NTNU – Trondheim
Norwegian University of
Science and Technology

Predicting Outcomes of Association Football Matches Based on Individual Players' Performance

Johanne Birgitte Linde
Marius Løkketangen

Master of Science in Computer Science
Submission date: June 2014
Supervisor: Helge Langseth, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

This master's thesis concludes our five years study in Computer Science, at the Norwegian University of Science and Technology.

Predicting the outcome of football matches is a research area where it is possible to earn a lot of money, if the generated predictions are accurate enough. In this thesis we develop three prediction models, based on a model proposed by Rue and Salvesen. Our models are scaled versions of the original model, where the scaling factors are determined by the strength of the players participating in a match. They are modelled as Bayesian networks, where the predictions are found by the Markov chain Monte Carlo method Gibbs sampling.

The models are applied to the betting market for three seasons, using three different betting strategies, along with the unscaled Rue and Salvesen model. Over these three seasons, our best model, the GoalScaled model, is able to outperform the baseline Rue and Salvesen model and earn money in all seasons.

Sammendrag

Denne masteroppgaven er avslutningen på vårt 5-årige studium i Datateknologi ved Norges teknisk-naturvitenskapelige universitet.

Å forutsi utfall av fotballkamper er et forskningsområde der det er mulig å tjene store penger på tipping dersom prediksjonene er nøyaktige. I denne oppgaven utvikler vi tre prediksjonsmodeller med utgangspunkt i en modell utviklet av Rue og Salvesen. Modellene våre er skalerte versjoner av den originale modellen, der skaleringsfaktorene er basert på hvilke spillere som deltar i kampen. De tre modellene er Bayesianske nett, der prediksjonene er funnet ved hjelp av Gibbs sampling, en Markov chain Monte Carlo metode. Modellene blir brukt på tippemarkedet i tre sesonger og med tre ulike tippestrategier, sammen med den opprinnelige Rue og Salvesen modellen. I løpet av disse tre sesongene er en av våre modeller, GoalScaled, den beste modellen. Denne modellen er også bedre enn Rue og Salvesen sin modell, og i stand til å tjene penger over de tre sesongene.

Acknowledgements

We would like to express our deep gratitude to our supervisor, Professor Helge Langseth, for all his help with this thesis. Thank you for your inspiring feedback, your dedication and your invaluable assistance.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter describes our motivation for this project. It also contains the goals we wish to achieve, along with the research questions we wish to answer. The outline of all remaining chapters is given in section 1.3.

1.1 Motivation

Association football (hereby referred to as ”football”) is one of the most popular sports in the world. People of all ages around the globe enjoy playing the sport, and even more love to watch it being played by professionals. Each weekend arrives with a new set of matches, and discussions arise on how well the different teams, players, and managers will perform. This has led to the rise of a large betting market, with bookmakers allowing for bets being placed on almost all aspects of the game.

Predicting outcomes of future football matches is a difficult task, because there are important variables that cannot be observed before a match is played. Injuries, suspensions, psychological effects and luck are examples of such variables. If a key player is injured or sent off during a match this could have a huge impact on the match outcome. If one team attacks for an entire match creating a lot of chances without scoring, while the opponent scores on their only attempt, the outcome might be considered as deeply unfair. Balls hitting the woodwork, players making individual errors, and faulty decisions by the referee, may often be enough to tip the game in one direction or the other. Even though such uncertainties could affect a single match, they may be evened out in the long run. Therefore, we believe it is possible to predict the probabilities of outcomes and earn money in the betting market in the long run.

1.2 Research Questions and Goals

This section contains the research questions to guide our work, and the goals that should be fulfilled in order to answer the research questions.

1.2.1 Research questions

The aim of this master thesis is to answer the following research questions:

> **Research question 1:** Is it possible to extend one of the existing football outcome prediction models to incorporate data about individual players?

If we should have developed a new model from scratch, this would have required more than just five months of research. Therefore, we wanted to extend one of the existing models. As described in our specialization project (Linde and Løkketangen, 2013), adding more detailed data to a prediction model could result in better predictions.

> **Research question 2:** Can we develop a purely data-driven system based on this model?

We needed a data driven system because this would make it possible to run the model by everyone, not just expert users. A data-driven system does also mean that the results are equal each time the system is used, which makes it possible to compare results found by different simulations on different machines, with different users.

> **Research question 3:** Is our model able to outperform the original model on the betting market?

The ultimate test for our developed model is to see if it is better than the original model on the betting market. Hopefully, it will also gain income, not just ”lose less” than the existing model.

1.2.2 Goals

We have stated three goals that should be achieved in this thesis. The first goal is

> **Goal 1:** An existing prediction model should be extended with information based on individual players.

This goal answers Research question 1. The extension of an existing model is the first step on our way to creating a prediction system.

Our second goal is

> **Goal 2:** Both the original prediction model and its extension(s) should be implemented as a part of a system that is able to process match input and generate predictions.

This goal is stated in order to achieve Research question 2. By implementing both the extensions and the original model, this makes it possible to compare them on several levels. The predictions created by the different models can be evaluated against each other, and the results can be applied on the betting market, or compared directly.

Our third goal follows

> **Goal 3:** Our developed predicion model(s) should be compared to the existing model on the betting market.

To be able to compare the models on the betting market, odds should be gathered and different betting strategies should be investigated and implemented. The implemented system should also have the ability to place bets and display statistics.

1.3 Outline of the Report

The outline of this thesis is as follows:

- **Chapter 1: Introduction** contains the motivation for conducting this research, as well as the research questions and goals.
- **Chapter 2: Background Theory** provides an overview of the knowledge required for the rest of the thesis.
- **Chapter 3: Prediction Models** describes the prediction models developed in this thesis.
- **Chapter 4: Implementation** is an overview of how we created and implemented our system for predicting outcomes.
- **Chapter 5: Experimental Set-Up** shows the parameters used by the prediction models and the set-up of the algorithms used for predictions.
- **Chapter 6: Results** presents the results and a discussion around them.

- **Chapter 7: Conclusion and Further Work** summarises the results and draws conclusions based on them. Also, our thoughts on how to extend the project are presented.

Chapter 2

Background Theory

In this chapter we will present the background knowledge required for the rest of this thesis.

2.1 Bayesian Network

A Bayesian network is a structure used to represent dependencies among variables, (Pearl, 1988). The network consists of nodes representing random variables, and arrows between the nodes, representing the connections between them. Each arrow is a direct link, and no cycles are allowed. Therefore, the network is a directed, acyclic graph, a DAG. If there is an arrow from node A to node B, A is a parent of B. The nodes can have either discrete or continuous domains.

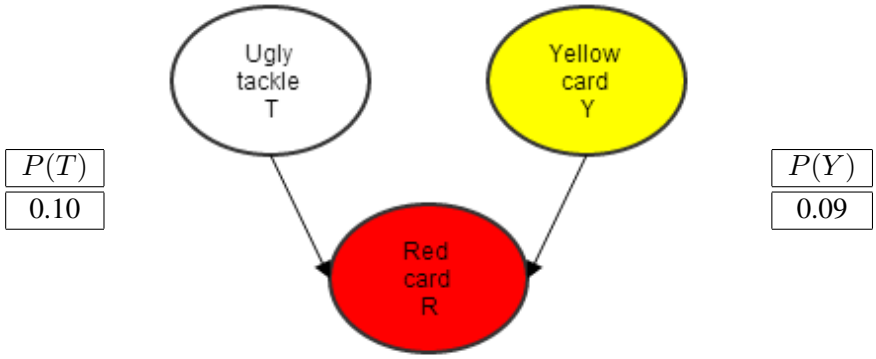
Each node in a Bayesian network must have a probability distribution table, where the probability distribution of the current node is given with respect to its parents, as shown by Equation 2.1.

$$P(X_i|Parents(X_i)) \tag{2.1}$$

Here, $Parents(X_i)$ are all the nodes that have an arrow to node X_i. By multiplying all the nodes given their parents, the joint distribution of the network is found, shown in Equation 2.2.

$$P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i|Parents(x_i)) \tag{2.2}$$

An example of a Bayesian network is shown in Figure 2.1. This network represents the dependencies between conducting an ugly tackle, already having a yellow



$P(T)$
0.10

$P(Y)$
0.09

T	Y	$P(R)$
true	true	0.95
true	false	0.50
false	true	0.15
false	false	0.02

Figure 2.1: A Bayesian network with its probability distribution tables.

card, and receiving a red card at a given time in a match. The symbols used in the figure are T for tackle, Y for yellow card and R for red card. The tables are interpreted such that the probability $P(T) = 0.10$ means that it is a 10% chance of the tackle being ugly (and hence a 90% chance of the tackle being legal). The node representing a red card is conditioned on the nodes for ugly tackle and yellow card, and its probabilities are as given with respect to its parents' nodes. For instance it is a 95% chance of receiving a red card if a player conducts an ugly tackle, and is already booked once. Therefore, it is a 5% chance that he is not shown the red card. This might be a result of the referee looking at something else, the referee making an error, and all the other events that might lead to the referee misinterpreting the situation.

2.2 Markov Chains

A Markov chain is a special class of the Bayesian networks that also capture the dynamic properties of a domain, so called dynamic Bayesian networks. A dynamic Bayesian network is a network where the states of the nodes changes over time, and the current value of the node is dependent on both the previous value of the same node, as well as the current values of its parent.

Formally, a Markov chain is a way to represent random variables that changes

over time, where each iteration is only dependent on the previous state of the variable, (Norris, 1997). This makes it a "chain" of variable values, with the Markov assumption. The Markov assumption is that the future is conditionally independent given the past. The future value is only dependent on the present state, which incorporate all the earlier states.

The changes between subsequent states are called transitions. These transitions are random. This makes the Markov chain non-deterministic, and it is therefore difficult to calculate a certain value the chain will have at a given time. However, it is possible to calculate the distribution over the outcomes.

As an example, the attack and defence strength of a team can be modelled as a Markov chain. Let us assume that the attack and defence strength of a team changes over time. It is reasonable to assume that there is some correlation between the strengths in subsequent matches, and that the previous match has more impact on the current match than a match way back in time. Like this, the attack strength in the current match, α_t, will be based on the attack strength in the previous match, α_{t-1}, which again is based on the attack strength in the match before, α_{t-2}. This shows that all the known information is incorporated into the current attack strength, which again is conditional independent given the earlier attack strengths, shown formally in Equation 2.3. Therefore, we can model attack and defence strengths as a Markov chain, where the strengths in the next match are only dependent on the strengths in the match before.

$$\alpha_{t+1} \amalg \{\alpha_0, \alpha_1, ..., \alpha_{t-1}\} | \alpha_t \tag{2.3}$$

2.3 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a way to simulate random samples from a distribution. The Monte Carlo method uses several repeated samples based on random numbers that can bee seen as representing the distribution. The Metropolis-Hastings algorithm and Gibbs sampling are such methods, and further described in section 2.4 and section 2.5, respectively.

Markov chain Monte Carlo methods are used to do inference in Bayesian networks. Inference is to calculate the posterior probability of some query variable given observations of some other variables. The way the Markov chain Monte Carlo methods draw inference are by first initializing all the nodes with random samples. Then, they look at a single node at a time, calculating its value with regard to the values of the other nodes.

The worst case with respect to computational complexity is when all the nodes in a network are connected to all the other nodes. This network has all the nodes in the same clique. In such a network, to draw exact inference would require a large

amount of computing power, and would be very time consuming. In fact, exact inference in Bayesian networks are NP-hard, and it is unknown how to solve them in polynomial time (Russel and Norvig, 2010). As a result, Markov chain Monte Carlo methods are used to draw inference in such networks. These approximation methods are also NP-hard, but since it it possible to end the simulation at any time, the simulation is conducted without the extensive use of computing power required for exact inference.

2.4 The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a Markov chain Monte Carlo method used for selecting a sequence of random samples from a probability distribution. It was first described by Metropolis et al. (1953) and Hastings (1970). A thorough description can be found in Chib and Greenberg (1995). The algorithm is applied when direct sampling is difficult.

The main idea is to generate a Markov chain with stationary distribution identical to the distribution we want to sample from, where $\pi(x)$ is known as the stationary distribution. This distribution is the one that is reached by sampling infinite number of times, and is unique given some regularity assumptions that are fulfilled in our domain. The randomly selected samples by the algorithm have the property that their distribution will approach the stationary distribution $\pi(x)$ as more and more samples are selected. An approximation to $\pi(x)$ can be found by choosing a large enough number of samples. During the selection of samples, the next sample is only dependent on the current samples, as described by the Markov property in section 2.2.

The pseudo code for the Metropolis-Hastings algorithm is shown in Algorithm 1.

Algorithm 1 Pseudo code for the Metropolis-Hastings algorithm.

1: **for** $t = 1...N$ **do**
2: $\theta^* \leftarrow q(\theta_{t-1}, \cdot)$
3: $u \leftarrow uniform(0, 1)$
4: **if** $u \leq acc(\theta_{t-1}, \theta^*)$ **then**
5: $\theta_t \leftarrow \theta^*$
6: **else**
7: $\theta_t \leftarrow \theta_{t-1}$
8: **end if**
9: **end for**
10: **return** $\theta_1...\theta_N$

Note that

$$acc(\theta_{t-1}, \theta^*) = \min\left(\frac{p(\theta^*)q(\theta^*, \theta_{t-1})}{p(\theta_{t-1})q(\theta_{t-1}, \theta^*)}, 1\right). \tag{2.4}$$

The function $q(\theta_1, \theta_2)$ is called the proposal distribution. This is the probability of the value θ_2 given the previous value θ_1, and can be expressed as $P(\theta_2|\theta_1)$. The $\cdot$ in the function q means that the argument can take any value. u is a sample for the uniform distribution.

Equation 2.4 is reduced to

$$acc(\theta_{t-1}, \theta^*) = \min\left(\frac{r(\theta^*)q(\theta^*, \theta_{t-1})}{r(\theta_{t-1})q(\theta_{t-1}, \theta^*)}, 1\right) \tag{2.5}$$

by setting $r(\theta) = K \times p(\theta)$. Here, $p(\theta)$ is the given distribution of the variables θ, and K is a normalizing constant, that might not be known. Note that the K can be cancelled out in Equation 2.4, since it is used in both the numerator and the denominator of the fraction. However, if the distribution is of a known form, the normalization constant K is also known.

In the pseudo code and in Equation 2.4 and Equation 2.5, the acc is the probability that the current candidate point, represented by θ^* is accepted. The candidate point θ^* is an arbitrary point selected from the distribution $q(\theta_1, \theta_2)$. When a candidate point is accepted, its value is added to the selection of samples. If it is not accepted, the acc value of a new candidate point is calculated, and the process is conducted again, until a large enough number of samples are found (as described above).

2.5 Gibbs Sampling

Gibbs sampling is a way to do inference in Bayesian networks, and a special case of the Metropolis-Hastings algorithm. The difference between the Gibbs sampler and the Metropolis-Hastings algorithm is that the proposal function is designed such that the candidate point θ^* is always accepted, meaning that acc in Equation 2.5 is equal to 1. Gibbs sampling is explained by Casella and George (1992), and this section is based on their paper.

One of the Bayesian networks used in this thesis is shown in Figure 2.2. It contains two nodes for each team in each match, the team's attack and defence strengths, shown as the white nodes. In addition, there are nodes representing the number of goals scored by each of the teams in each of the matches, the green nodes. These nodes do naturally also represent the number of goals conceded by the opponent. The red node is a result that is not yet observed, and should be predicted. The strengths are linked with the result nodes of the given match, which

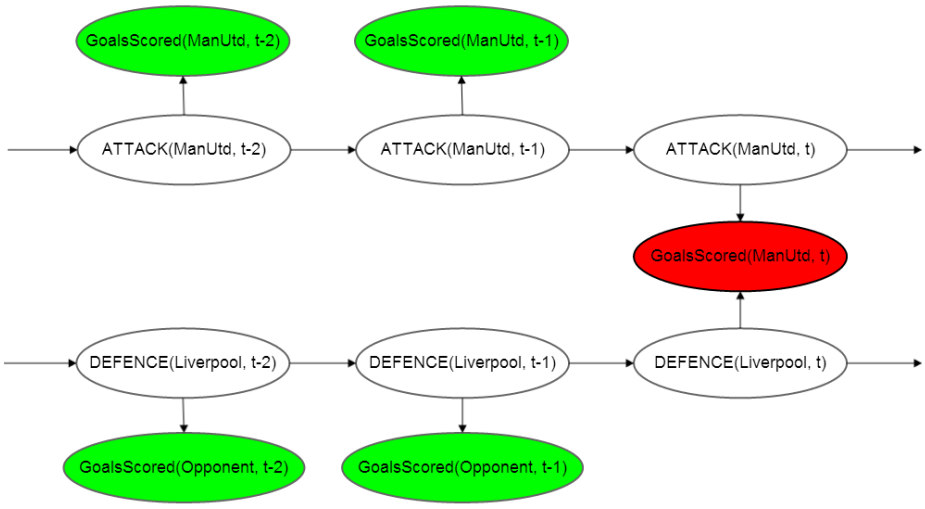


Figure 2.2: A Bayesian network showing the development of the attack strength of Manchester United and the defence strength of Liverpool, and how these strengths are used to find the number of goals scored by Manchester United in the match between them.

means that halfway through a season, all nodes in the network will be correlated given the evidence. The reason is that each of the teams play against each other two times during a season, one time at home and one time away. Normally, each team has met all the other once halfway through the season. Here, all the latent variables are correlated, and a Markov chain Monte Carlo method should be used to do inference, as described in section 2.3. Therefore, we decided to use Gibbs sampling on our network.

Figure 2.3 shows a small example of a part of a Markov chain. The chain contains a Bayesian network where the number of goals scored is dependent of the attack strength of Manchester United and the defence strength of Liverpool. For simplicity, we only look at the attack strength of Manchester United here. At the first time step of the Markov chain, t-2, the attack strength is set to A. During the next time step, t-1, the value is changed to B, and during time step t, this value is C. The selection of such a variable is described below.

In our network, there are two different types of nodes, the observed ones and the ones that should be predicted. The values of the observed nodes are kept constant, while the ones that should be predicted have a Gaussian distribution for the attack and defence strengths, and a Poisson distribution for the goals scored and conceded.

The value that should be used to predict the next attack or defence node is drawn at random from the Gaussian distribution, or from the Poisson distribution

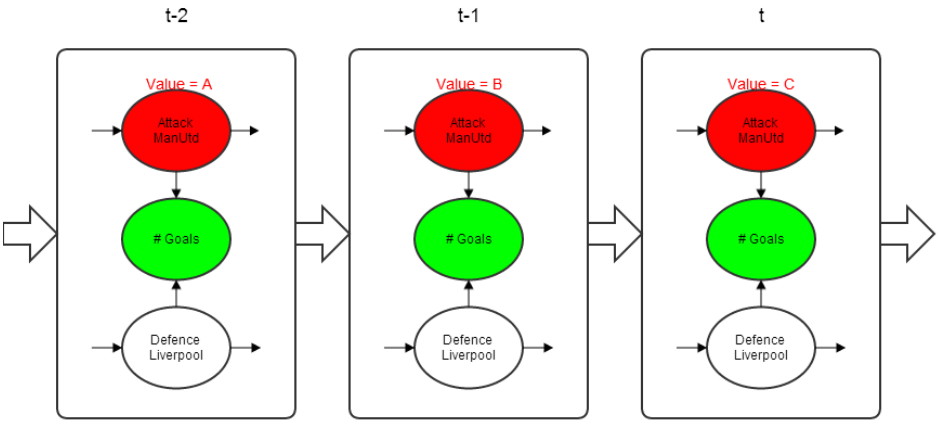


Figure 2.3: The development of a Bayesian network as a Markov chain.

if the node is a goal node. Then, the new value of the node is calculated.

In the next iteration, this calculated value is the new expected value for the Gaussian distribution for the attack or defence node, or Poisson distributed with λ based on the attack and defence strength. The new distribution is used when the value of the next node is calculated. In this manner, all the nodes update their values, until the change in the calculated values are below some threshold, or the sampling has been conducted a specified number of times.

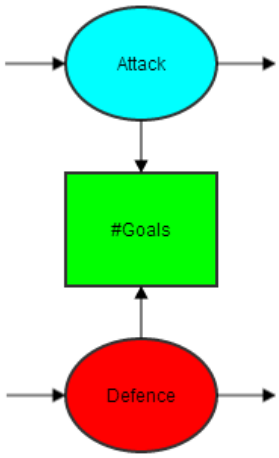
An example of Gibbs sampling is shown in Figure 2.4. Figure 2.4a shows a small part of a Bayesian network. In Figure 2.4b, the Gaussian distribution of the attack strength is presented. The random aspect of the Gibbs sampling is used to draw a value from the distribution at random, here, 0.7. The Gaussian distribution of the defence strength is the graph in Figure 2.4c, and the random value selected from this is 0.3. These two values are used to model the Poisson distribution of the number of goals scored in this match, with the Poisson parameter λ. The attack and defence strengths are represented by the values α and δ, respectively. The definition of λ is shown in Equation 2.6.

$$\log(\lambda) = \alpha - \delta \tag{2.6}$$

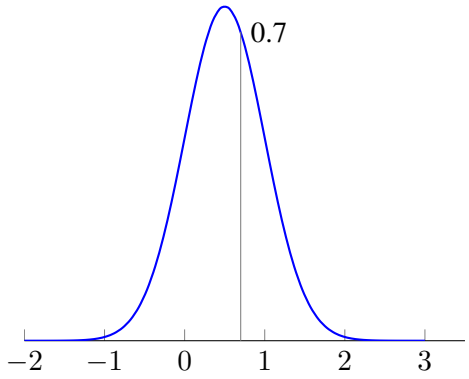
Equation 2.7 shows how the λ found in Equation 2.6 is used in the Poisson distribution for number of goals.

$$\#Goals = Poisson(\lambda) = Poisson(\mathrm{e}^{\alpha-\delta}) = Poisson(\mathrm{e}^{0.7-0.3}) \tag{2.7}$$

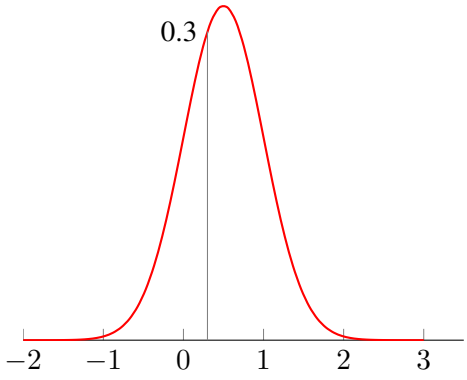
Next, a value is drawn at random from the Poisson distribution found in Equation 2.7. The calculated distribution is shown in Figure 2.4d, and the value drawn



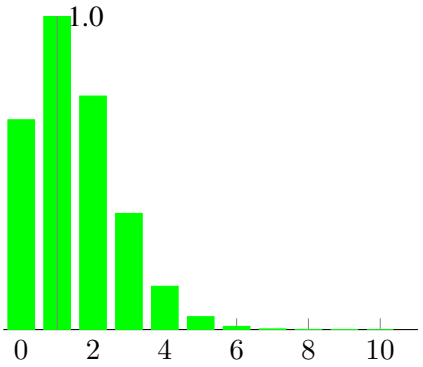
(a) A part of the Bayesian network.



(b) Gaussian distribution of attack.



(c) Gaussian distribution of defence.



(d) Poisson distribution of goals with an attack strength of 0.7 and a defence strength of 0.3.

Figure 2.4: A part of a Bayesian network with the distributions representing its nodes.

at random is 1.0. This value is added to the samples. Then, the simulation is conducted again, with new Gaussian and Poisson distributions.

A basic explanation of Gibbs sampling can be found in (Russel and Norvig, 2010, pp.536-538) and (Mitchell, 1997, p.176).

2.6 JAGS - Just Another Gibbs Sampler

JAGS is a program licensed under the GNU General Public License and provides the ability to draw inference in Bayesian networks using Markov Chain Monte Carlo (MCMC). The program is described in the user manual by Plummer (2011). To use the program, a model must first be defined in a .jags file, using R code. A small example model is shown below.

```
1  # Defining the constants as data
2  data {
3      noTeams <- 20
4      noRounds <- 38
5  }
6  # Definition of the model
7  model {
8      precision ~ dgamma(10, 1)
9      for (i in 1:noTeams)
10     {
11         attack[i, 1] ~ dnorm (0, 1/precision)
12         for (j in 2:noRounds)
13         {
14             attack[i, j] ~ dnorm(attack[i, (j-1)], 1/precision)
15         }
16     }
17 }
```

In line 2 to 5, the constants are assigned their values, using <-. All the variables that are not assigned at runtime with ~ must be sent to the model through the data object. The model is declared in lines 7 to 17. `dgamma` creates a gamma distribution, precision, with shape parameter 10 and scale 1, while `dnorm` creates a Gaussian distribution for attack with the expected value 0 and variance as $\frac{1}{precision}$. The attack node is dependent on previous attack nodes, as shown in line 11. This is in fact the implementation of a Bayesian network.

After the model is defined, it is compiled and initialized. The next stage is adaptation and burn-in. During the adaptation, monitors are defined to make JAGS keep track of the values produced for the different nodes. In the code above, it would be natural to add a monitor for the attack node. Burn-in is used to discard the n first samples, which are not recorded by the monitor. This is done because the Gibbs sampling will converge after some time (after n samples), and the first n samples is therefore regarded as noise, and discarded. This is further described in section 5.1.

The last step of the program is monitoring. During this phase, the simulated values (with a monitor) are written to a file. The output of the program are two files, one index file and one coda (data) file. The index file contains the name of the monitor, and the line numbers for where to find the values in the data file. By adding a monitor to the attack variable above, the index file can look like this (with 20 simulations of each match):

```
1 attack[1,1] 1 20
2 attack[1,2] 21 40
3 ...
```

The coda file will then contain the values for the attack strength of team 1 in round 1 at line 1 to 20, the values for team 1 in round 2 at line 21 to 40 etc. After these files are written, the JAGS program is finished, and the files can be post processed and incorporated into a larger program.

2.7 Odds Explained

This section is taken from our specialization project, Linde and Løkketangen (2013).

In football, there exists a wide range of odds for almost every aspect of a game. Multiple bookmakers provide odds for the end results, the number of goals scored during a match etc. We have focused on the odds presented for the different outcomes of a match, either home victory, draw or away victory.

Odds can be represented either as a decimal number, as a fractional number or several other specific representations for different parts of the world. The common way to represent odds in Europe is the decimal way, which is the one we use in this paper. British bookmakers favour fractional odds, but the conversion between decimal and fractional is straight forward. It is done by subtracting 1 from the decimal number and convert to fraction, and conduct the division (of the fraction) and add 1 to reverse the process.

The bookmakers operate with a built in margin, assuring them to earn money over time. This margin is implemented such that the sum of their predicted outcomes are greater than 1. The built in margin is often between 5% and 7%. If several bets are placed on a single outcome, the odds for that outcome is adjusted, and the bookmakers will still earn money.

If p is the predicted probability of an outcome and ω is the odds set by the bookmakers on that outcome, the sum $p \times \omega > 1$ would result in a bet being placed on that outcome, because it has a positive expected return. For a single game, the odds for home victory, draw and away victory would be ω_H, ω_D and ω_A, respectively. The predicted probabilities for the same outcomes are p_H, p_D and p_A.

$$p_H \times \omega_H > 1 \tag{2.8}$$

$$p_D \times \omega_D > 1 \tag{2.9}$$

$$p_A \times \omega_A > 1 \tag{2.10}$$

If either Equation 2.8, Equation 2.9 or Equation 2.10 is true, a bet is placed on that match outcome. By placing bets in such a way, an accurate prediction system will over time be able to earn money. However, if the prediction system is inaccurate, it is difficult to earn money in this way.

Given fair odds, one should have expected that by placing a bet with the value of $\frac{1}{\omega_H}$ on home victory, a bet of $\frac{1}{\omega_D}$ on draw and a bet of $\frac{1}{\omega_A}$ on away, the expected payback should be 1 credit, as shown in Equation 2.11.

$$\frac{1}{\omega_H} + \frac{1}{\omega_D} + \frac{1}{\omega_A} = 1 \tag{2.11}$$

The return on the investment, r, is found by rearranging Equation 2.11, and is shown in Equation 2.12.

$$\frac{1}{\frac{1}{\omega_H} + \frac{1}{\omega_D} + \frac{1}{\omega_A}} = r \tag{2.12}$$

By inserting some real values into Equation 2.12, we are able to find the real return on investment. Norsk Tipping is the only betting company with offices in Norway. For the Premier League match between Liverpool and Norwich City at the 4th of December 2013, the provided odds were 1.25, 5.25 and 8.25 for home, draw and away, respectively (NorskTipping.no, 2013). The return on the investment is then:

$$\frac{1}{\frac{1}{1.25} + \frac{1}{5.25} + \frac{1}{8.25}} = 0.90 \tag{2.13}$$

The return of 0.90, found in Equation 2.13, shows that we can only expect to get back 90% of our money on average by betting on this match. This means that Norsk Tipping will earn 10% of all the money that were spent on this match. In fact, Norsk Tipping will adjust their odds continuously in such a way that they always receive the same percentage of the spendings. Therefore, to be able to earn money, the developed prediction system must be better than the bookmaker's prediction system.

2.8 Betting Strategies

Several different betting strategies exist. A betting strategy determines how bets should be placed, and the amount to stake on each bet. Whether the system is able to gain income or not is highly dependent on the betting strategy used, as described by Langseth (2013). In this section we explain some common betting strategies. Here, odds are represented as ω and the outcome probability calculated by our prediction system as p. The betting strategies we have implemented only allows for one bet per match (either home, draw or away). The size of each bet is defined in units, meaning that if one unit is placed, you either lose that unit or win $(\textit{(Size of unit)} \times \omega - \textit{(Size of unit)})$.

2.8.1 Fixed bet

Place a bet of one unit if the expected return of the bet is positive. That is, if $\omega \times p > 1$. If there are more than one possible bets with positive expected return, place a bet on the outcome with the highest expected return.

2.8.2 Modified fixed bet

During our experiments, we have seen a tendency that the fixed bet strategy leads to a large amount of bets being placed on away outcomes with very high odds. Using the modified fixed bet strategy we want to avoid this by introducing a safety margin variable, ϵ, that prevents our system from betting too often on these unlikely away outcomes. The betting rule becomes: Place a bet of one unit if $\omega \times p > 1 + \epsilon$. The value for ϵ can be different for home, draw and away outcomes.

2.8.3 Fixed return

If the expected return of a bet is positive, place a bet of size $\frac{1}{\omega}$. Using this strategy, a successful bet will always result in the same return, and less risk will be taken on low-probability bets. The same safety margin that was introduced in the Modified fixed bet strategy is also used in our version of Fixed return.

2.8.4 Wallpapering Fog

This betting strategy is inspired by a strategy described in a blog called Wallpapering Fog by Charles (2014). The thought behind this strategy is that if a prediction system is accurate enough, it will earn money by simply placing a bet on the most likely outcome of each match. It works as follows: If the probability of a draw is

predicted to be more than 27%, place a bet on a draw. If not, place a bet on the most likely outcome.

The threshold for draw bets is used because a draw is almost never predicted to be the most likely outcome of a match. The threshold value of 27% is used because matches end in a draw about 27% of the time, according to the author. This is also supported by Clarke and Norman (1995), who calculated that 26.7% of all matches in the EPL seasons between 1981-1991 ended in a draw.

This strategy is interesting, because it does not care if the expected return of a bet is positive. The traditional gambler would never place money on bets that he believes will not pay off in the long run, because this simply does not make sense. However, as we will see in chapter 6, this style of betting sometimes outperforms the others mentioned in this section.

2.8.5 Probability bet

This betting strategy makes use of the values found by Clarke and Norman (1995) on how often a match ends in a home victory, away victory or draw. If our betting model predicts an outcome to be more likely than 48.7%, 26.7%, or 24.6% for home, draw or away outcomes, respectively, place a bet of one unit on that outcome.

2.8.6 Kelly

This betting strategy was proposed by Kelly (1956). Langseth (2013) used this strategy for football betting, and the description given here is based on his paper. The goal of using this strategy is to find the amount to stake on a bet b_i that maximises the utility of that bet. The utility here is defined as $\log B$, where B is the amount in your bankroll. The utility of going broke is therefore minus infinity. The expected utility of a bet b_i is

$$E\left(U_{b_i}\right) = p_i \log\left(B + \omega_i b_i\right) + \left(q - p_i\right) \ln\left(B - b_i\right), \tag{2.14}$$

which is maximized for

$$b_i = B\frac{p_i\omega_i - 1}{\omega_i - 1}. \tag{2.15}$$

2.9 Performance Measures

The performance of a football prediction system can be measured in several ways. This section investigates three common measures, and is taken from our specialization project, Linde and Løkketangen (2013).

2.9.1 Statistical measures

The easiest way to evaluate a prediction system, is to report the percentage of correctly classified outcomes of matches. However, a system with a lower classification percentage might still earn more money on the betting market than a system with a higher classification percentage. The reason is that the odds might be different for the different correctly classified matches, such that one might lose a lot on the few wrongly classified matches.

On average, around 50% of the matches end with a home victory. Koning (2000) reported that 46% ended with a home victory in the Dutch Eeresdivision for a couple of seasons. It is therefore possible to achieve a precision of about 50% by classifying all the matches as home win.

The main reason for predicting outcomes of matches is often to earn money. A high percentage of correctly classified matches is important, but it is not the only factor to consider. How to place the money is just as important.

2.9.2 Income from betting

Another way to estimate the performance of the model, is to apply it to the betting market. The performance is then based on if the system is able to earn money or not. This should be tested during several seasons, as some of the prediction models (e.g. Goddard and Asimakopoulos (2004)) were able to earn money in one season, but not in the next.

How to place the bets is a research area in itself. Langseth (2013) investigates the effect of five different betting models used with three different prediction algorithms. The main idea behind the placements of bets is that the expected return should be larger than 1 while placing a bet of 1, as explained in section 2.7. The different approaches tested are fixed bet, fixed return, Kelly-ratio, variance-adjusted, and Markowitz portfolio management with the strategies conservative, intermediate and aggressive. The different models are tested on two Premier League seasons, 2011/2012 and 2012/2013. For the first season, all the betting approaches are able to earn money (except 1 of the 21 combinations), but for the second season, only one of the 21 combinations earns money. As a result, how to place bets should be thoroughly considered before testing a prediction system. It also shows that it is important with an algorithm that can produce correct predictions.

2.9.3 Rank Probability Score - RPS

The Rank Probability Score (RPS) was investigated for use at football prediction systems by Constantinou and Fenton (2012). The evaluation of the algorithms is based on the relations between the three different outcomes home, draw and away.

It states that the difference between the two results home and away are greater than the difference between home and draw. If the result is 1-0, it would only require 1 goal from the away team to make the result a draw, but 2 goals to end up with away. During the evaluation of the predictions, the penalty of predicting away when the result is home should therefore be greater than predicting the same match as a draw. The formula for the RPS for football prediction evaluation is shown in Equation 2.16.

$$RPS = \frac{1}{2}\sum_{i=1}^{r}\left(\sum_{j=1}^{i} p_j - \sum_{j=1}^{i} e_j\right)^2 \tag{2.16}$$

Here, p_j is the predicted outcomes and e_j is the observed outcomes. Since the formula looks at the distances between the predicted outcomes and the observed outcomes, the closer the result is to 0, the better is the prediction.

Even though a system receives a good RPS score, it might not earn money from betting. As described in subsection 2.9.2, this depends on the applied betting strategy. The advantage with such an RPS evaluation is that the evaluation of the system is only based on the predicted probabilities, one does not need to take into account the behaviour of the bookmaker. This can be a good measure of accuracy, and another way to evaluate the system than through betting.

2.10 Summary of our Specialisation Project

During the autumn 2013, we conducted a specialisation project as a pre-study and preparation for this masters thesis (Linde and Løkketangen, 2013). The main part of this project was a structured literature review, which is presented in its entirety in Appendix A.

2.10.1 The models investigated

In the literature review, we investigated the existing models for prediction of outcomes of football games. Here we will present a brief summary of these models.

Poisson based models

Several of the models we investigated were Poisson models, where the main thought is that the number of goals scored by each team are Poisson distributed. These models include Maher (1982), Dixon and Coles (1997), and Rue and Salvesen (2000), where the newer models slightly outperform the older ones. Rue and

Salvesen (2000) use a Bayesian network and Markov Chain Monte Carlo to draw inference.

The Bradley-Terry model

Another approach to the prediction problem is the Bradley-Terry model. This model has been used with good results in high scoring sports like basketball, American football and Australian football. Cattelan et al. (2013) extend the model to predict results of Serie A matches, where the outcome of the model is either home, draw or away. They claim that the predictions would be better if more detailed data were included in the model, and say that it should be easy to extend it to incorporate this data.

ELO rating

ELO rating was applied to football prediction by Hvattum and Arntzen (2010). This is a rating system where each team has a rating that is updated after each match. The higher the rating, the better the team. They developed two kinds of ELO rating models, one basic model and one goal based extension of this model. Unfortunately, we were not able to find a comparison between the Poisson based models, the Bradley Terry model and these ELO models, and could not draw a conclusion about which of these models that is the best one. The only conclusion was that none of the models are able to gain certain income over time.

Pi-rating

A model that was able to gain profit was the Pi-rating developed by Constantinou and Fenton (2013). They created a model based on the three factors home advantage, that recent results have the most impact on the next result, and that it is more important for a team to win than to increase goal difference.

Ordered probit regression model

The last investigated model was an ordered probit regression model, created by Goddard and Asimakopoulos (2004). This model uses information about the significance of a match, previous results, the travel distance, and information about the teams' participation in other matches outside the league (like the UEFA cup, Champions League, FA cup etc.). When applied to the betting market, the model is able to outperform the bookmakers in one of the two seasons it was tested on. It would have been interesting to see it applied to several more seasons, in order to draw a conclusion if it is better than the bookmakers or not.

2.10.2 Improvements of the models

Several of the papers above mentioned that the inclusion of more detailed data in the prediction models should lead to better predictions. More detailed data refers to factors such as number of shots in a game, number of tackles, or which players that participated in the game, among others. The prediction models proposed by Maher (1982), Dixon and Coles (1997),Rue and Salvesen (2000), Goddard and Asimakopoulos (2004), Hvattum and Arntzen (2010), and Constantinou and Fenton (2013) all use the goals scored in previous matches as their main factor. However, at the time some of these models were presented, more detailed information was not as easily available as it is today. Dixon and Coles (1997) assumed that prediction models based on a richer amount of data would give better predictions, and Rue and Salvesen (2000) claimed that "It is of major importance to include more data than just the final match result in the model, but this depends on what kind of data is (easily) available and useful." (Rue and Salvesen, 2000).

Today, the Internet provides us with a huge amount of easily available data. Sites like WhoScored.com (2014) contains extensible data on almost all aspects of football matches in several leagues. This makes it possible for us to test if inclusion of more detailed data will make an existing prediction model perform better.

At the end of our specialisation project, we decided to conduct a feasibility study by creating a simple model based on the players that started a match. Therefore, we created a small regression model where the results of a match were sent in as parameters together with the average of the ratings for each player in that match. This was compared to a model where the average ratings were based on a team as a whole. Both models performed almost equally well (or bad), without any significant differences. However, in a few cases, the model based on individual players was slightly better than the other. As a result, we decided to focus on individual players in this master's thesis.

2.11 The Rue and Salvesen Prediction Model

As described in the previous section, the model presented by Rue and Salvesen (2000) was an extension of the model presented by Maher (1982). The model was also developed after the model by Dixon and Coles (1997). Andresen and Dubicki (2013) implemented a simplified version of this model for their masters thesis, which is the model we have used and extended for our experiments. The reason for this choice is that most prediction systems we have investigated have used this approach, and the model by Rue and Salvesen (2000) seemed to perform better than the older Poisson based models. It only makes use of the results of previous matches as its observed variables, and seemed easy to extend with information

concerning individual players.

2.11.1 The goal model

In this model, the two most important properties of a team are considered to be its attack and defence strength, α and δ respectively. The number of goals scored by each team in a match are assumed to be Poisson distributed, and are conditioned on the attacking and defensive abilities of the two teams involved. A simple goal model using this assumption is shown in Equation 2.17.

$$\begin{aligned} \log\left(\lambda_{A,B}^{(x)}\right) &= \alpha_A - \delta_B \\ \log\left(\lambda_{A,B}^{(y)}\right) &= \alpha_B - \delta_A \end{aligned} \tag{2.17}$$

Here, α_A represents the attack strength of team A, and δ_B represents the defence strength of team B. x and y are the numbers of goals scored by team A and B, respectively. The two equations find the λ-parameters for two Poisson distributions. The distribution with the $\lambda_{AB}^{(x)}$ -parameter represents the number of goals scored by team A, and the distribution with the $\lambda_{AB}^{(y)}$ -parameter represents the number of goals scored by team B.

Psychological effect of underestimating

Rue and Salvesen (2000) included a variable in their model that takes into account the psychological effect of one team underestimating the abilities of a weaker team. This variable is defined as

$$\Delta_{AB} = \frac{\alpha_A + \delta_A - \alpha_A - \delta_B}{2}. \tag{2.18}$$

The psychological effect variable is included in Equation 2.17 as follows:

$$\begin{aligned} \log\left(\lambda_{A,B}^{(x)}\right) &= \alpha_A - \delta_B - \gamma\Delta_{AB}, \\ \log\left(\lambda_{A,B}^{(y)}\right) &= \alpha_B - \delta_A + \gamma\Delta_{AB}. \end{aligned} \tag{2.19}$$

Here, γ determines the impact of the psychological effect.

Home ground advantage

Home ground advantage is represented as a global constant in this model, and is included in the goal model as defined in Equation 2.20.

$$\begin{aligned} \log\left(\lambda_{A,B}^{(x)}\right) &= c^{(x)} + \alpha_A - \delta_B - \gamma\Delta_{AB} \\ \log\left(\lambda_{A,B}^{(y)}\right) &= c^{(y)} + \alpha_B - \delta_A + \gamma\Delta_{AB} \end{aligned} \tag{2.20}$$

Team A is here the home team, and team B is the away team. $c^{(x)}$ and $c^{(y)}$ describe the empirical mean of home and away goals.

Simplified version of the goal model

The original model by Rue and Salvesen (2000) included a correction factor for increasing the probability of 0-0 and 1-1 results at the cost of 1-0 and 0-1 results. Also, if a team scored more than five goals in a match, the sixth goal did not add any information to the scoring abilities of that team. These two properties were not included in the implementation by Andresen and Dubicki (2013), so the goal model we have used as a starting point in this project is a simplified version of the goal model by Rue and Salvesen (2000).

2.11.2 The time model

It is desirable for the attack and defence strengths of a team to vary over time. Rue and Salvesen (2000) use Brownian motion to tie together attack strengths at the two time points $t^{'}$ and $t^{''}$ (where $t^{'} \leq t^{''}$):

$$\alpha_A^{t''} = \alpha_A^{t'} + \left(B_{\alpha,A}\left(\frac{t''}{\tau}\right) - B_{\alpha,A}\left(\frac{t'}{\tau}\right)\right) \frac{\sigma_{\alpha,A}}{\sqrt{1-\gamma\,(1-\gamma/2)}}. \tag{2.21}$$

Here, $\sigma^2_{a,A}$ is the prior variance for α_A. The time parameter τ is common to all teams and gives the inverse loss of memory rate for α_A^t, $var\left(\alpha_A^{t''}|\alpha_A^{t'}\right) \propto \frac{\sigma^2_{\alpha,A}}{\tau}$.

Chapter 3

Prediction Models

This chapter presents our developed prediction models. All these models are extensions of the model proposed by Rue and Salvesen described in Rue and Salvesen (2000). The three created models are named GoalScaled, AttackScaled and AttackAndDefenceScaled.

3.1 Data Set

The data were collected by crawling `http://www.whoscored.com` and sub sites for the selected seasons. We gathered data about three English Premier League Seasons, the 11/12, 12/13 and 13/14 seasons. Details about the gathered data are explained in section 4.1 and section 4.2.

3.2 Rue and Salvesen

The model we have used as our baseline model, is a simplified version of the model presented by Rue and Salvesen (2000). This simplification was created by Andresen and Dubicki (2013) during their masters thesis, and does not include freak results (matches where at least one of the teams scored at least 5 goals) and higher probabilities for the results 0-0 and 1-1. The full implementation is shown in section B.1, and further described in section 2.11.

3.3 Scaling Based on Goals

Our simplest model uses information about how many of a team's earlier goals the players in the predicted starting line-up have scored, and is called Goal Scaled. A

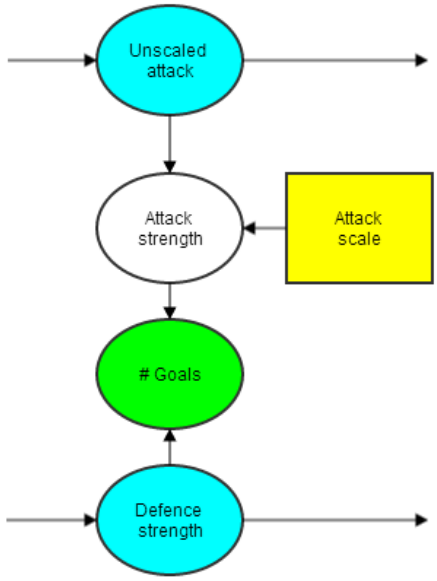


Figure 3.1: A Bayesian network where the attack strength is scaled with a scaling factor.

scaling factor is included in the baseline model, and is meant to take into account the loss of attacking strength if one of the most scoring players is absent from the starting line-up. This scaling factor is found by adding together a player's goals, and dividing them with the total number of minutes played by the player. Then, the average of all the players in the starting line-up is found, divided by the number of goals per minute for the team. The goal scale for a single team in match j, with the starting line-up consisting of the players in i is shown in Equation 3.1. m is the previous matches for the team.

$$GoalScale_j = \frac{1}{j-1} \times \sum_{m=1}^{j-1} \frac{\frac{1}{11} \times \sum_{i=1}^{11} \frac{playerGoals_{i,m}}{\max(minsPlayed_{i,m},1)}}{\frac{teamGoals_m}{90}} \tag{3.1}$$

The logarithm of this scaling value is then added to the attack strength found by the Gibbs sampler (in JAGS), and the now scaled attack is used to predict the number of goals scored by the team.

The reason for such a scaling is that the number of goals has the greatest impact on the outcome of a match. Therefore, scaling the strength using information about the starting line-up's contribution to the number of goals, was a natural place to start when developing our model. The full .jags implementation of this prediction model can be found in section B.2.

A Bayesian network of the implementation is shown in Figure 3.1. Here, the blue nodes are the predicted attack and defence strengths, and are the ones that are used in subsequent predictions. The yellow node is the observed scaling factor.

3.4 Data Intensive

After developing the GoalScaled model described in the previous section, we wanted to extend it with more of the available information. As a result, we developed a scaling factor for the attack strength and the defence strength using several attributes.

3.4.1 Choosing attributes to use in our models

The reason for scaling both the attack and defence strength was that we then could include variables relevant for both attack and defence in our model. The thought here is that if, for instance, a player that normally contributes with a lot of key passes and interceptions is missing from a team's starting line-up, this should be incorporated in the team's attack and defence strengths. To decide which attributes to include in these scaling factors, we used the results presented by Ellefsrød (2013). He conducted a correlation R^2 test, showing which attributes that had the greatest impact on the result of a match, and which that had none. Based on this, we chose to include goals scored, scoring attempts, attempts on target, through balls, key passes (passes that lead to a scoring opportunity), and successful passes in the attack scale. Attributes included in the defence scale were total tackles, total clearances, interceptions, and duels won. Some of these attributes were not a part of the R^2 test by Ellefsrød, but we chose to include them because we believe they might have an impact on the outcomes of football matches.

The scaling values were found by taking the average of all the values registered in the database. An example query is shown below:

```
1 SELECT   AVG(CAST(DuelWon AS Float)) AS Average
2 FROM     TeamMatch
```

Then, the scaling factor was calculated by finding the attack and defence contribution for each of the players in the probable starting line-up, adding them together and dividing by the average attack and defence contribution for the team.

3.4.2 Scaling the attack value

The implementation of the attack scaling factor used for all individual players is shown below.

```
private double CalculatePlayerAttackContribution(
                    PlayerIndividual player)
    {
        var minsPlayed = player.MinutesPlayedSoFar < 90 ? 90
                         : player.MinutesPlayedSoFar;
        return (double)player.GoalsScoredSoFar
                    / (minsPlayed * 1.4)     * 0.4513
             + (double)player.ScoringAttSoFar
                    / (minsPlayed * 14.11)   * 0.045
             + (double)player.OnTargetScoringAttSoFar
                    / (minsPlayed * 4.63)    * 0.135
             + (double)player.TotalThroughBallSoFar
                    / (minsPlayed * 1.7)     * 0.283
             + (double)player.TotalAttAssist
                    / (minsPlayed * 10.52)   * 0.084
             + (double)player.SuccessfulPassesSoFar
                    / (minsPlayed * 367)     * 0.0017;
    }
```

This method gives each of the players in the expected starting line-up a scaling factor for his attacking strength. Then, the total scaling factor for the team is the sum of all these 11 players, divided by the scaling factor found by sending in a TeamIndividual to an identical method.

The different numbers used in the calculations above are found by investigating the matches in the 11/12 and 12/13 Premier League seasons. By investigating the number of goals scored, we found that, on average, 1.4 goals are scored in a match by a team. Goals are weighted by 1, since this is the most important measure for the winner of a match. Next, on average, there are 14.11 scoring attempts by a team during a match. The weights for the scoring attempts are found by dividing 1.4 by the total number of scoring attempts, generally shown in Equation 3.2.

$$Weight = \frac{1.4}{Event} \tag{3.2}$$

When all the weights are found, the numbers are normalised, and multiplied with the scaling factor. Table 3.1 shows the number of occurrences, their weights and the normalised values.

An implementation of the simplified Rue and Salvesen prediction model (described in section 3.2) with the attack scaled in this manner was one of the prediction models we tried. This model is called AttackScaled. In this model, the predicted value of an attack strength is multiplied with the logarithm of the scaling factor, and this scaled attack value is the one that is used to predict the number of goals scored by a team in the following match. The .jags implementation is presented in section B.2, and is the same implementation as the one used for the GoalScaled model. The Bayesian network of this model is shown in Figure 3.1,

Table 3.1: The number of occurrences of different attack events in a game, and their scaling factor.

Event	Occurrences per match	Weight	Scaling factor (normalized weight)
Goals	1.4	1.0	0.4513
Scoring attempts	14.11	0.1	0.045
Attempts on target	4.63	0.3	0.135
Through balls	1.7	0.629	0.283
Assist attempts (passes that leads to a scoring attempt)	10.52	0.1864	0.084
Successful passes	367	0.0038	0.0017
Sum		2.2192	1.0

which is further described in section 3.3.

3.4.3 Scaling the defence value

The defence scale is calculated in the same manner as the attack scale, but based on different events. The implementation on player level is shown below. This scale is calculated for each of the players in the probable starting lineup, summed together, and divided by the calculated defence scale for the team as a whole. The defence scale for the team as a whole is calculated by a method similar to the one below, where the input is a TeamIndividual instead of a PlayerIndividual.

```
1  private double CalculatePlayerDefenceContribution(PlayerIndividual
       player)
2  {
3      var minsPlayed = player.MinutesPlayedSoFar < 90 ? 90
4                     : player.MinutesPlayedSoFar;
5      return (double)player.TotalTackle
6                  / (minsPlayed * 18.39)  * 0.319
7           + (double)player.TotalClearance
8                  / (minsPlayed * 30.99)  * 0.19
9           + (double)player.Interception
10                 / (minsPlayed * 15.77)  * 0.373
11          + (double)player.DuelWon
12                 / (minsPlayed * 49.69)  * 0.118;
13 }
```

Here, we decided to disregard the number of goals conceded as an own event. The average number of goals conceded per match are 1.4. This number is still included when finding the weights, as shown in Equation 3.2. The reason for this

Table 3.2: The number of occurrences of different defence events in a game, and their scaling factor.

Event	Occurrences per match	Weight	Scaling factor (normalized weight)
Total tackles	18.39	0.076	0.319
Clearances	30.99	0.04352	0.19
Interceptions	15.77	0.0888	0.373
Duels Won	46.69	0.0282	0.118
Sum		0.2382	1.0

decision is that the predictions produced by this approach were better than the predictions produced when including the number of goals conceded as an own event. The events and their weights are shown in Table 3.2.

We conducted a few runs with an implementation of the simplified Rue and Salvesen algorithm with a scaled defence value, but the results were no better than the AttackScaled model. Therefore, we decided to not include the model that only contains defence scale any further in our report.

3.4.4 Combining the attack and defence scale

The most comprehensive model includes a scaling factor on both the attack and defence strengths in the simplified Rue and Salvesen model, described in section 3.2. The scaling factors for attack and defence are the ones described in subsection 3.4.2 and subsection 3.4.3, respectively. These factors are included in the prediction model by adding the logarithm of the attack scaling factor to the predicted value of the attack strength, and using this new value to predict the number of goals scored. The defence strength and defence scaling factor is used in the same manner to predict the number of goals conceded.

Figure 3.2 shows the Bayesian network with scaling values on the attack and defence strengths. The yellow nodes are the observed scaling factors, while the blue nodes are the predicted attack and defence strengths, before they are scaled. The scaled values (the white nodes) are used to predict the number of goals scored or conceded. For the next prediction, the **blue** nodes are used further. The .jags implementation of this network shown in section B.3.

This model is called ScaledAttackAndDefence in the rest of our thesis.

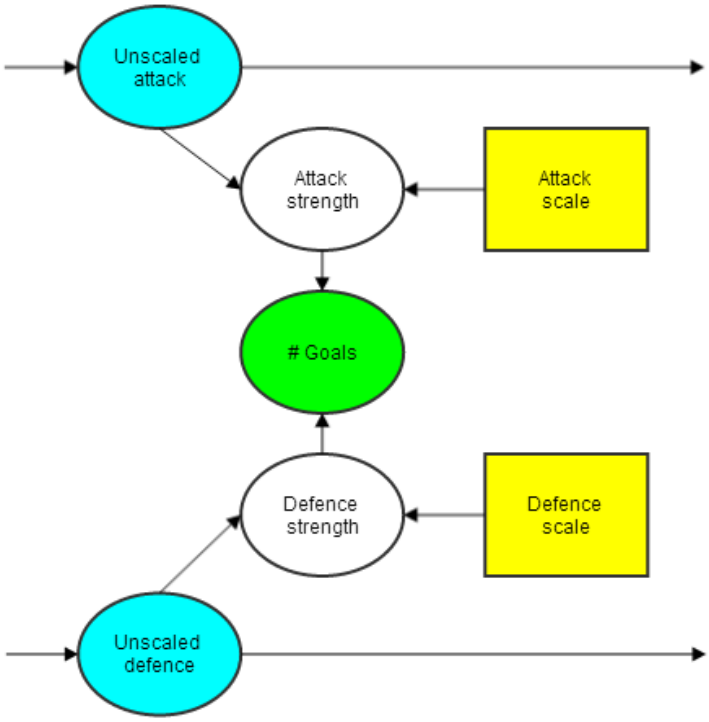


Figure 3.2: A Bayesian network with scaling values on the attack and defence strengths.

3.5 Fulfilment of Goals

In this chapter, we have presented three extensions of the exiting Rue and Salvesen model. We decided to scale the attack and defence predictions, and use the scaled values to predict the number of goals. The creation of these three scaled models fulfils Goal 1 from section 1.2, which again relates to Research question 1 from the same section.

Chapter 4

Implementation

This chapter describes how we created and implemented a system for generating predictions using the prediction models described in chapter 3. The content in this chapter consist of a description of the crawler, set-up of the database, the SharpJags extension to Visual Studio and a section about the C# implementation. How to run our systems are described in Appendix C.

4.1 Crawler

This section is an extended and modified version of the crawler description in our specialization project, (Linde and Løkketangen, 2013, pp.28-29).

An Internet crawler developed by Martin Belgau Ellefsrød for his master thesis was used for collecting team-specific match information (Ellefsrød, 2013). This crawler was written in PHP, using WampServer to access a specified site containing the desired information about a match (WampServer.com, 2013). We have modified this crawler, enabling it to fetch data regarding individual players.

This data was collected from `http://www.whoscored.com/`. The crawler accesses the site and relevant sub sites, and downloads web pages in raw text. This text is then processed by a PHP script using regular expression matching to extract the wanted information. An overview of the regular expressions used can be found in Appendix D.

The collected data are very detailed. A team's performance in a single match is described by approximately 150 attributes, and a player's performance in a single match can be described by over 100 attributes. Although we collect all these attributes, we will use only some of them for our experiments. Currently, 56 of them are stored in our database, but it is easy to include more information if that

is a wish. General match info is also gathered, such as goals scored, which teams are involved, match ID and the date of the match. In addition, we fetched the predicted starting lineups and the actual players involved for each of the matches. All of the matches in Premier League 2011/2012, 2012/2013 and 2013/2014 have predictions for the lineup, except three of the matches in 2012/2013.

Odds for each of the three seasons were downloaded from `http://www.football-data.co.uk/englandm.php` (Football-data.co.uk, 2014). This site contains files with odds provided by several betting agencies, and average and maximum odds. The odds are presented in .csv files, which makes them easy to parse and store in the database.

4.2 Database

To be able to access and process all the data in an efficient manner, we created a database. The ER-diagram for this database is presented in Figure 4.1. The database contains tables for players, teams, matches, odds, predictions, and the relations between them. A description of the different tables follows.

Stage

Each season is a different stage. This makes it possible to store several seasons from different leagues and countries in the database. The fact that each match is linked to a stage makes it possible to gather all the matches in a given season with a single query.

Team

All teams are stored with only their unique id and their name. Note that there is no direct link between a team and a stage. The reason for this decision is that this makes it possible to relegate or promote teams between different years (stages) without having to add a new replacement team or make changes to the existing entry (which could make the database useless for more than one year at a time).

TeamMatch contains all the information about how a specific team played in a specific match. All the attributes in this table are presented in Appendix E.

Player

The players are stored with only name and id. The motivation for not having an entry for the player's team, is the same as the one described above for team and stage. A player may play in different teams during a season, and also change teams between different seasons.

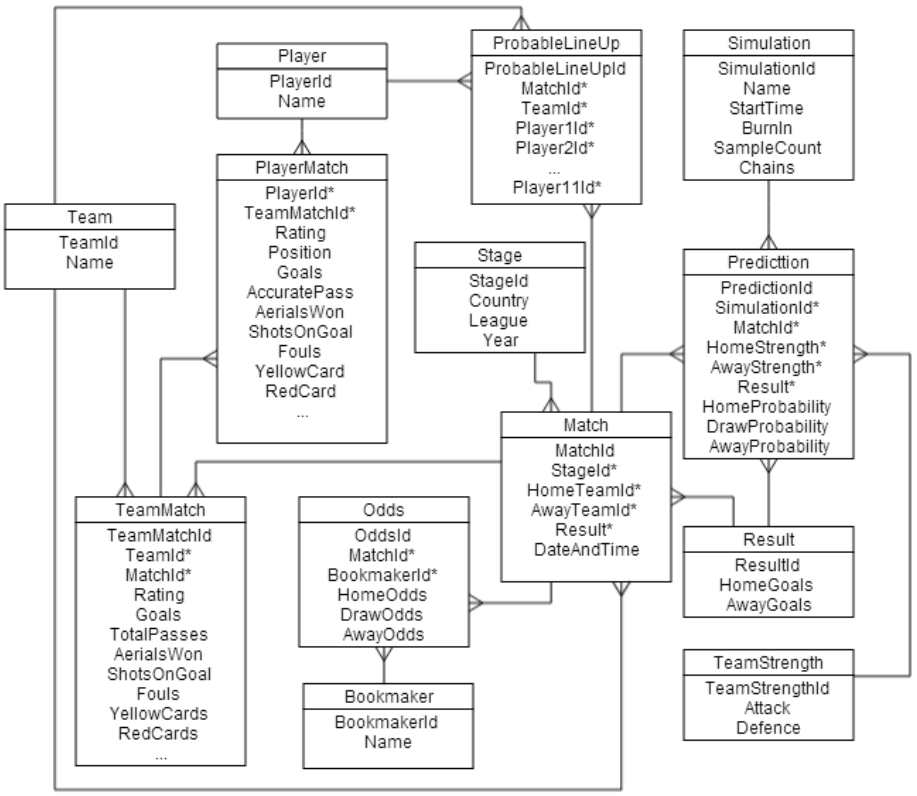


Figure 4.1: The ER-diagram for the database

PlayerMatch stores data about a single player's performance in a given match, for a given team (represented by TeamMatch). All the attributes in this table are presented in Appendix E.

Simulation

A simulation is stored each time the program is started with one of the implemented simulation models. Each simulation has multiple predictions, one prediction for each match. The prediction contains the predicted result, the predicted probability for home, draw and away, and the predicted attack and defence strengths for each of the two teams involved.

Bookmaker

The bookmaker has an id and a name. The bookmaker's odds are stored in the Odds table, with its probabilities for home, draw and away, and which match the predictions concerns.

4.2.1 NHibernate

The database was created by the use of NHibernate (NHibernate, 2014). NHibernate is an extension to .NET, which makes it possible to handle all entities as objects directly in the code. The result is that the database is abstracted away, and all the CRUD-operations (Create, Read, Update, Delete) take place by handling object instances. Inserting a new object is done by creating an object in code, and then save it to the database through a new database session. The mapping between an object and an entity in the database is shown below.

```
1  public class Player
2  {
3      public virtual int PlayerId { get; set; }
4      public virtual string Name { get; set; }
5  }
6
7  // The mapping to the database is done in this class
8  public class PlayerMap : ClassMapping<Player>
9  {
10     public PlayerMap()
11     {
12         Id(x => x.PlayerId);
13         Property(x => x.Name);
14     }
15 }
```

To initiate the communication with the database, a new session must be created. The session makes it possible to read from the database. If something is going to be written to the database, a transaction should be initialised as well. By calling $Save()$ or $SaveOrUpdate()$, the new or updated object is saved to the current session. If the object also should be saved to the database, the transaction must be committed. The following code snippet describes this process.

```
using (var session = NHibernateConfig.OpenSession())
{
    // Reads the player with PlayerId = 1
    var playerId = 1;
    var player = session.Get<Player>(playerId);

    using (var tx = session.BeginTransaction())
    {
        // All the writing to the database is done here.
        // The player is saved to the database with his new name.
        player.Name = "Steven Gerrard";
        session.SaveOrUpdate(player);
        tx.Commit();
    }
}
```

The entire database was created and is accessed and updated in this manner. During initialization of a new simulation, all the needed information is found in the database. After a simulation, the simulation results are saved to the database as well.

More than 50 variables concerning the contribution of each of the players in each of the matches are stored in the database, resulting in a large database. This large amount of data results in a noticeable search time. The search time is not more than some milliseconds for each query, but when we extract information about 22 players in each game, and their contributions, with 380 games in a season, this adds up to some minutes. Also, the simulation of a single match takes some time (around 10 minutes for 10000 simulations of one match, with 3 chains).

4.3 SharpJags

SharpJags is an extension to Visual Studio and C# that makes it possible to interpret and execute .jags files. SharpJags is written by Thomas Andresen, and can be downloaded from github at `https://github.com/thrandre/SharpJags`, (SharpJags, 2014). We experienced a problem having to little RAM when running a lot of samples (more than around 2000 simulations per match). Therefore, we wrote our own output handler for the results of the jags simulations.

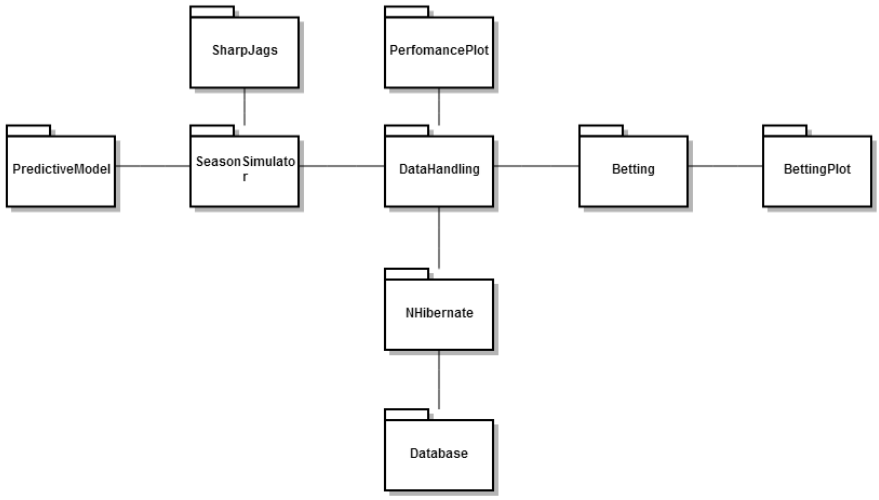


Figure 4.2: Package diagram of the C#-program

4.4 The C# Program

All the pre and post processing of the predictions are done in a C# program that we have developed. A package diagram is shown in Figure 4.2. Settings are controlled from a class that consists of static variables, the Constants class. Here, variables for if the database should be updated are set, which prediction model, which betting strategy and which bookmaker that should be used for the next simulation. The results of the simulations are stored in the database, which makes it possible to try other betting strategies or generate new diagrams without having to rerun the simulation.

4.4.1 Inheritance

During the implementation, our goal was to create a program that was easy to extend and maintain. This has been achieved by the use of inheritance. Adding a new JAGS simulation model is done by adding two new classes, a subclass of SeasonSimulation and a subclass of PredictiveModel, and updating the switch-case that selects models. Adding a new betting model is done in the same manner, by adding a new subclass to the superclass BettingModel.

4.4.2 Reasons for C#

We chose to do the implementation in C# because we had some earlier experience with the language. Also, the program developed for the masters thesis by Andresen and Dubicki (2013) was written in C#, and in the beginning we investigated the opportunity for extending their existing code. The last reason was the SharpJags plugin to Visual Studio, which could be used almost out of the box for our project.

4.5 Fulfilment of Goals

In this chapter, we have described the implementation of the system. Our system is purely data-driven, and contains implementations of all the prediction models from chapter 3. The implementation fulfils Goal 2 stated in section 1.2, and relates to Research question 2 from the same section.

Chapter 5

Experimental Set-Up

This chapter describes the set-up of our experiment. The parameters used for the Markov chain Monte Carlo are described, as well as the data-set, the prediction models used and the betting strategies applied for testing.

5.1 Parameters for Markov Chain Monte Carlo

Before conducting an experiment it is important to assure that the results received are reliable. In order to get as reliable results as possible for our experiments, the parameters for the Markov Chain Monte Carlo (MCMC) algorithm needed to be set correctly. In this section we will explain what the MCMC parameters do, what their values were in our experiments, and why we chose those values.

5.1.1 Burn-in

The plots in Figure 5.1 illustrate how sample values for a single variable evolve during the execution of an MCMC algorithm. Observe that the initial values in these two plots are different. Looking at the leftmost plot, sample values for the three chains are quite different initially, although they do converge to values within a certain range after some rounds of sampling. This is the range of the stationary distribution of the variable (see section 2.4). The time it takes for sample values to get within this range is called the burn-in phase. Since the samples generated during the burn-in phase are dependant of the initial value of the variable, it is desirable to discard them. The burn-in parameter serves this purpose, and decides how many samples we should discard before we start monitoring sample values.

In the rightmost plot of Figure 5.1, the burn-in parameter is set to 500. Here, all the samples are within the same value range. This is because we have discarded

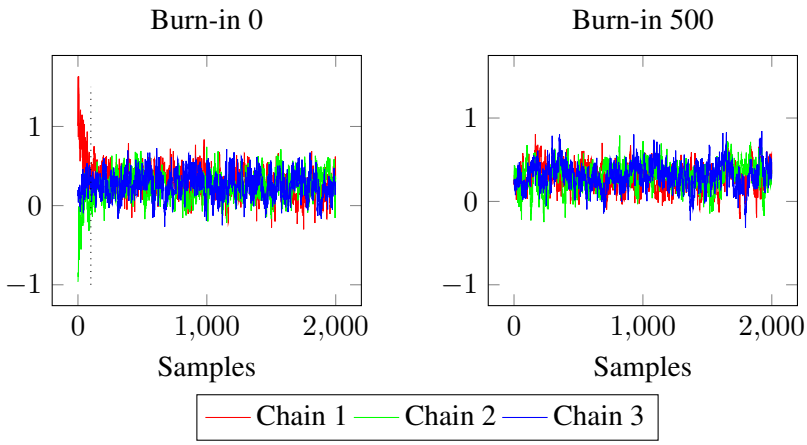


Figure 5.1: Sample values for a single variable with the burn-in parameter set to 0 and 500. The burn-in parameter is used to make sure that we discard all samples left of the dotted line.

the samples that were generated during the burn-in phase.

The distributions of the sample values in Figure 5.1 are illustrated in Figure 5.2. We can easily observe that the variable in this example is a normal distributed variable. We also see that two of the distributions in the leftmost plot have a long ”tail”. This is because the samples generated during the burn-in phase are kept. In the rightmost plot of Figure 5.2, all samples from the burn-in phase are discarded, and we see that the ”tails” are gone.

Initial values of variables can be set manually, or JAGS can generate them randomly. In our experiments, initial values were generated randomly by JAGS. We have found that setting the burn-in parameter to 500 samples (as in the example above) was sufficient for our experiments.

5.1.2 Thinning

During an MCMC simulation, each sample is correlated with samples generated before and after it. If the simulation generates a chain of several samples that are strongly correlated, this might affect our predictions by putting too large weight on the values of those samples. Therefore, we needed to assume that our samples were independent and identically distributed. In order to assure this we only kept every n’th sample and discarded the rest. The thinning parameter decides how many samples that should be generated for each one we keep. Figure 5.3 shows autocorrelation plots of attack strength samples when the thinning parameter is set to 1, 10, and 20, respectively. These plots show how correlated a sample is

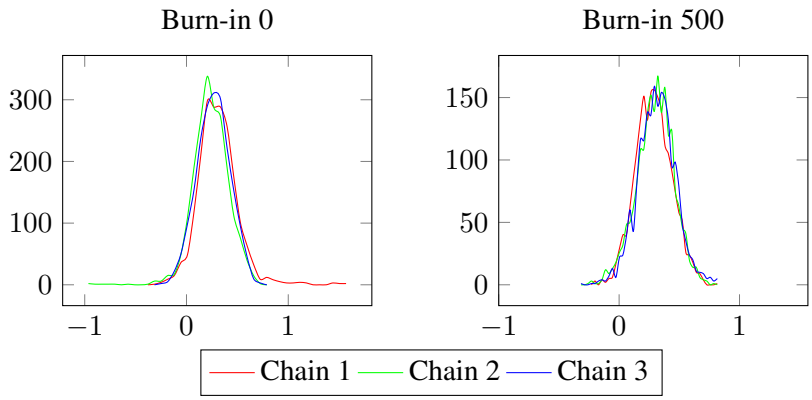


Figure 5.2: Distribution plots of sample values for an attack variable, created from simulations where the burn-in parameter is set to 0 and 500. As can be observed, two of the distributions in the leftmost plot has ”tails”. The samples that cause this ”tail” are samples generated during the burn-in phase.

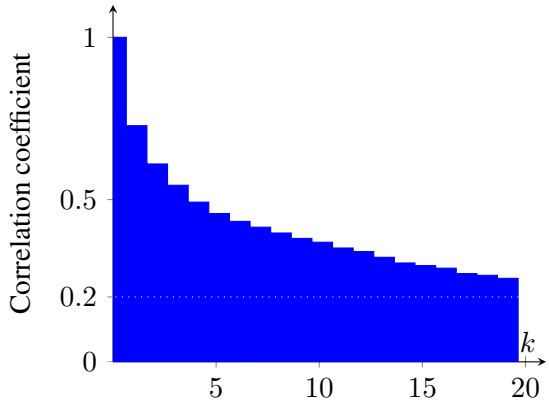
to the samples that are generated after it (with distance k). As we see from these figures, setting the thinning parameter to 20 is sufficient to avoid keeping too many strongly correlated samples.

5.1.3 Sample count

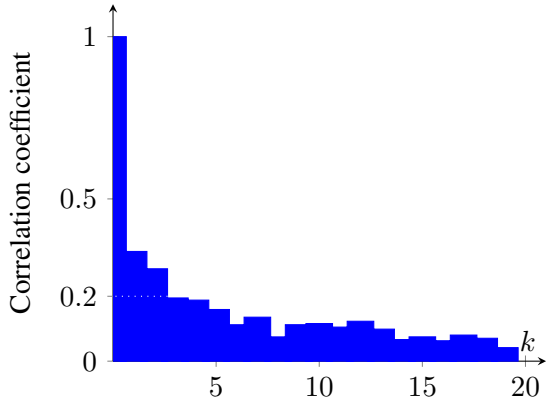
The number of samples that should be generated for each variable during a simulation (after the burn-in phase) is decided by the sample count parameter. More samples mean more accurate predictions, but it will also increase the run-time of the system. We needed to generate enough samples to represent an approximation of the stable distribution for all variables. We have performed simulations with a various number of samples and found that generating 40000 samples was sufficient for obtaining reliable results (meaning that the predictions are quite similar if the same simulation is executed several times). Statistical methods exist for finding how many samples are needed for representing an approximation of the stable distribution, but we chose to find the value of this parameter manually through hands-on experience.

5.1.4 Chains

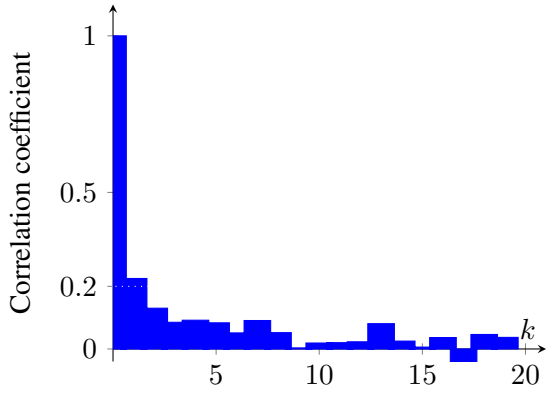
The chains parameter determines how many times the MCMC algorithm should be run (in parallel). Running the simulation several times is necessary because the Markov chain could ”get stuck” in a local neighbourhood outside the convergence range of a variable and generate samples of undesired values. Although separate



(a) Thinning 1



(b) Thinning 10



(c) Thinning 20

Figure 5.3: Autocorrelation plots with different values for the thinning parameter. The dotted white line defines an upper threshold value for which the correlation coefficient is acceptable.

chains are independent of each other, you would expect all chains to converge in the same neighbourhood. However, the initial values in one chain might be far outside the desired value range and spend a lot of time (in worst case, infinite) converging to this range.

For our experiments we have used three chains. Figure 5.1 and Figure 5.2 displays the sample values that are kept for an "attack strength" variable in one of our experiments. By investigating such plots, you can easily see if one of the chains are "stuck" because the sample values of this chain would be in another range than sample values in the other chains. Although this is not the case in these plots, it would have been evident if one chain generated sample values that were in a different range than the other chains. If this had been the case, we would have to consider re-parametrising our model.

5.2 Data

The data we have used for our experiments are match data for three seasons of the English Premier League (EPL), the 11/12, 12/13, and 13/14 season, collected from WhoScored.com (2014). The first half of each season is used for training our models, and the second half is used for predicting outcomes and betting.

5.3 Models

The models that are tested in our experiments are the simplified Rue & Salvesen model (Rue and Salvesen, 2000), along with the three extensions we have created (explanations of these models are given in chapter 3). These models will be referred to as RueSalvesen, GoalScaled (section 3.3), AttackScaled (subsection 3.4.2) and AttackAndDefenceScaled (subsection 3.4.4).

5.4 Betting strategies

The betting strategies used in our experiments are Fixed bet, our modified version of Fixed bet, Fixed return, Wallpapering, and Kelly (see section 2.8 for an explanation of all betting strategies used). However, we will only present the results from experiments using the Fixed bet, Fixed return and Wallpapering betting strategies. This is done to keep the presentation of results clean, and because we feel these represent three strategies with differences that are easily comparable.

The Fixed bet strategy is the most aggressive betting strategy, and tries to maximize its income by finding the most profitable bets. The Fixed return strategy is more careful when placing bets, placing only $1/\omega$ units per bet (ω represents

the odds). It also has a safety margin that prevents it from placing bets where the expected return is below the safety margin threshold, set to 5% here. The Wallpapering strategy does not use the provided odds at all, and places bets only based on the predictions.

The bookmaker odds used are the average odds given by eight different bookmakers, collected from Football-data.co.uk (2014). The unit size is set to 1 for all strategies. We start out with nothing in our bankroll, and are given a maximum of 1 unit to bet on each match.

Chapter 6

Results

The results from all our experiments are presented in this chapter. We have conducted experiments using the four prediction models RueSalvesen, GoalScaled, AttackScaled and AttackAndDefenceScaled, as explained in chapter 3. These models have been tested using the three betting strategies Fixed bet, Fixed return, and Wallpapering. An overview of the set-up of the experiment can be found in chapter 5.

6.1 English Premier League 2011/2012

Table 6.1 contains a summary of how well the different prediction models performed when paired with three different betting strategies in the 2011/2012 season of the English Premier League (EPL). We notice that all combinations, except the AttackAndDefenceScaled model paired with the Wallpapering strategy, made money this season. We also notice that the Wallpapering strategy was the least

Table 6.1: The final income after combining prediction models with betting strategies for the EPL 11/12 season.

Model	Betting Strategy		
	Fixed Bet	Fixed Return	Wallpapering
RueSalvesen	30.61%	23.76%	5.42%
GoalScaled	24.41%	24.83%	13.89%
AttackScaled	25.98%	17.29%	2.76%
AttackAnd-DefenceScaled	33.34%	22.32%	-1.38%

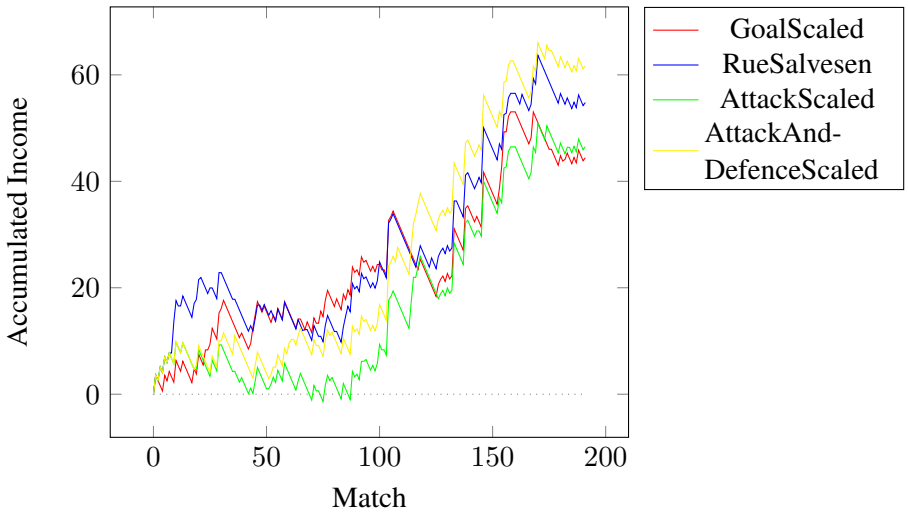


Figure 6.1: Accumulated income (in units) from betting with the Fixed bet betting strategy for the EPL 11/12.

profitable.

The RueSalvesen model was outperformed by a scaled model for all the betting strategies. The GoalScaled model yielded the best results for both the Fixed return and Wallpapering betting strategies, while the AttackAndDefenceScaled model was best when using Fixed bet. The model that only scales the attack strength was worse than the RueSalvesen model in all cases.

6.1.1 Fixed bet

Figure 6.1 shows the accumulated incomes from when the Fixed bet strategy was used with the four prediction models. As seen by the plot, all the prediction models had a positive accumulated income, with the largest income from the AttackAnd-DefenceScaled model, which gained a total income of 33.34% (61.68 units).

The Fixed bet strategy can decide not to place a bet on a match if none of the expected returns are greater than 1. Therefore, the number of bets placed by the prediction models are different. Table 6.2 shows the number of bets placed by each of the prediction models. As the table shows, the prediction model that placed the most bets, the AttackAndDefenceScaled model, was also able to earn the most money, even if the percentage of successful bets was 0.11% lower than the percentage for the RueSalvesen model.

By investigating Figure 6.1 as well, we see that both the RueSalvesen and the

Table 6.2: Overview of bets with the Fixed bet strategy on the EPL 11/12 season.

Prediction model	Number of bets placed	Successful bets	Success percentage	Return
RueSalvesen	179	66	36.87%	30.61%
GoalScaled	182	64	35.16%	24.41%
AttackScaled	179	61	34.08%	25.48%
AttackAnd-DefenceScaled	185	68	36.76%	33.34%

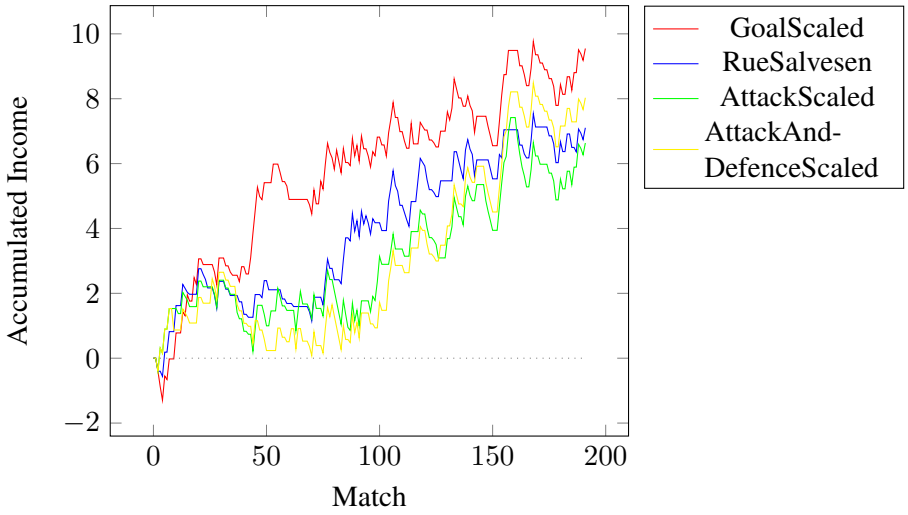


Figure 6.2: Accumulated income (in units) from betting with the Fixed return betting strategy for the EPL 11/12.

AttackAndDefenceScaled model had large jumps in their accumulated income, meaning that a bet on a high odds match was correct. Unfortunately, it seems that after a bet on a high odds match has been won, the system seems to have a longer period of losses than if it won on a low odds match. The reason might be that the result could be a ”lucky win”, and during the next rounds of simulation, the system therefore thinks that the underdog was the better team, and adjusts their attack and defence strength accordingly in the next match.

Table 6.3: Placement of bets in match 43 to 46 in the EPL 11/12 seasons, using the Fixed return betting strategy and the GoalScaled prediction model.

Teams	Date	Result	Our prediction ($P(Result)$)	Bookmaker's prediction ($1/\omega$)
Norw vs Bolt	04.02.14	2-0(H)	0.56	0.50
QueP vs WolW	04.02.14	1-2(A)	0.36	0.25
Stok vs Sund	04.02.14	0-1(A)	0.47	0.30
WesB vs Swan	04.02.14	1-2(A)	0.37	0.28

6.1.2 Fixed return

In Figure 6.2, the betting strategy Fixed return is used with the four prediction models. Here, we see that all the four models performed almost equally well. The biggest difference, and the one that made the GoalScaled model receive a higher return than the other ones, was the placement of bets in match 43 to 46. In these matches, the GoalScaled model gained an advantage that the other models were not able to catch up with later. The matches, our model's predictions, the bookmaker's predictions and the results are shown in Table 6.3. As the table shows, the predictions produced by GoalScaled were between 0.06 and 0.17 higher than the bookmaker's, meaning that our model had more faith in those results than the bookmaker.

6.1.3 Wallpapering

The Wallpapering strategy was the betting strategy with the lowest overall winnings this season, and the only one that resulted in a negative return for a prediction model, (the AttackAndDefenceScaled with -1.38%). How the Wallpapering betting model performed on the betting market is shown in Figure 6.3. As we see in the figure, the Wallpapering strategy is obliged to place a bet on each of the matches, and its success is based on winning many wagers with low odds, instead of winning a few with higher odds. This strategy was the one that showed the largest differences in income among the prediction models. As Table 6.1 shows, the returns were between -1.38% for AttackAndDefenceScaled, and up to 13.89% for GoalScaled. RueSalvesen was the second best model with a return of 5.42%.

Table 6.4 shows how the predictions made by the different prediction models affected the decisions made by the Wallpapering strategy. We notice that the GoalScaled model had a larger success rate for both home, draw and away outcomes. It was especially on draw outcomes that the GoalScaled model outperformed the others, and it placed a bet on a draw outcome 28.8% of the time. This seems to be reasonable, as about 26.7% of all matches end in a draw (Clarke and

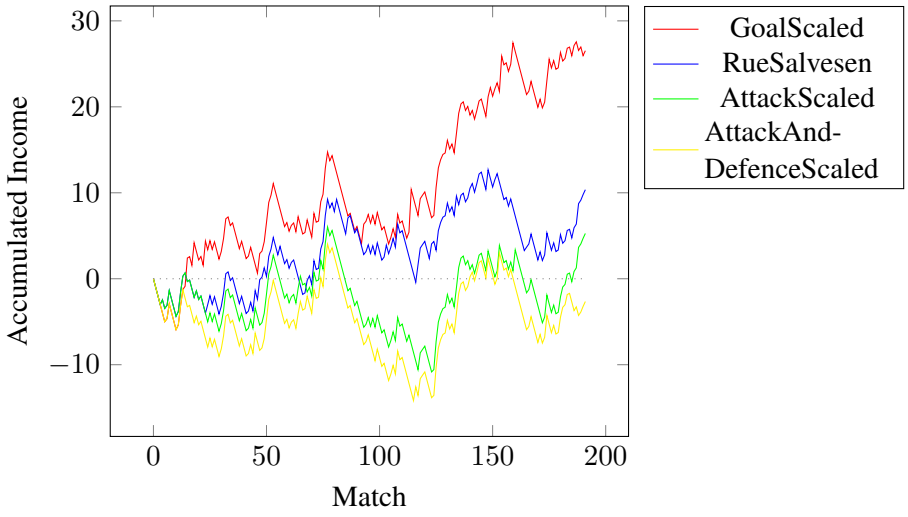


Figure 6.3: Accumulated income (in units) from betting with the Wallpapering betting strategy for the EPL 11/12.

Table 6.4: Betting statistics for the Wallpapering strategy in the EPL 11/12 season.

	Prediction model			
	Rue-Salvesen	Goal-Scaled	Attack-Scaled	AttackAnd-DefenceScaled
Bets placed	191	191	191	191
Successful bets	100	99	95	91
Success %	52.36%	51.83%	49.74%	47.64%
Home bets	116	102	110	108
Draw bets	32	55	36	39
Away bets	43	34	45	44
Won home bets	67	60	61	59
Won draw bets	10	20	13	12
Won away bets	23	19	21	20
Success % home	57.76%	58.82%	55.46%	54.63%
Success % draw	31.25%	36.36%	36.11%	30.76%
Success % away	53.49%	55.88%	46.67%	45.45%
Avg return home	3.5%	1.5%	-1%	-2.3%
Avg return draw	4.4%	33.7%	23.3%	5%
Avg return away	11.5%	19.0%	-4.4%	-4.9%

Table 6.5: The final income after combining prediction models with betting strategies for the EPL 12/13 season.

Model	Betting Strategy		
	Fixed Bet	Fixed Return	Wallpapering
RueSalvesen	-21.29%	-16.65%	0.56%
GoalScaled	-26.24%	-23.89%	11.76%
AttackScaled	-39.24%	-25.81%	0.9%
AttackAnd-DefenceScaled	-37.57%	-31.8%	-2.07%

Norman, 1995). The big difference in return in Figure 6.3 is explained by the higher success percentage on draw and away bets for the GoalScaled model. The reason is that the odds on draw and away matches in most cases are the highest odds, and a good success percentage here is vital for receiving a good overall profit.

6.1.4 Comparison

Figure 6.4 illustrates the difference between the Fixed bet and Wallpapering betting strategies. The Fixed bet strategy will only place a bet if the expected return is positive, while the Wallpapering strategy will place a bet on the most likely outcome (or a draw if it is more probable than 27%).

28.57% of the bets placed using the Fixed bet strategy this season were on outcomes with odds greater than 5.0. These bets generated an average profit of 38.27%, and 17.31% were successful. In comparison, 47.12% of the bets placed using the Wallpapering strategy had a negative expected return. These returned a profit of 7.67%, and 58.89% were successful. This indicates that bets with a negative expected return have a greater chance to be successful than high odds bets, but the expected return is much lower.

6.2 English Premier League 2012/2013

Table 6.5 contains a summary of how well the prediction models performed when paired with different betting strategies for predicting outcomes in the 12/13 season of the EPL. We observe that, although the RueSalvesen and GoalScaled models performed better than the other two this season, all models had more problems with predicting outcomes than in the 11/12 season.

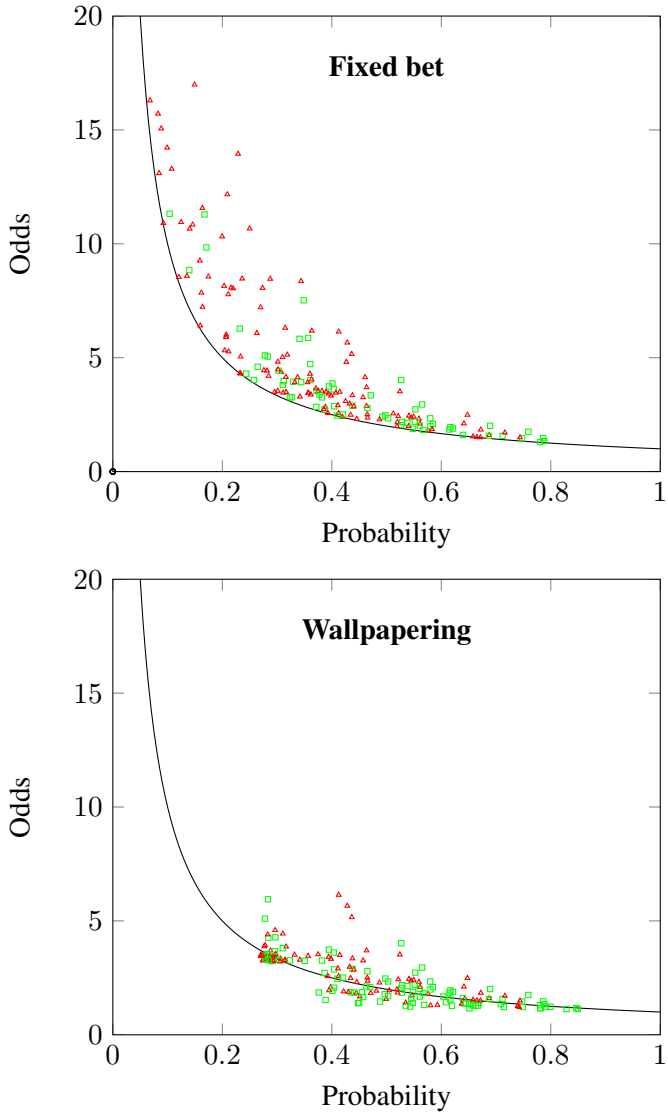


Figure 6.4: Overview of bets placed using the GoalScaled prediction model combined with the Fixed bet (upper plot) and Wallpapering (lower plot) betting strategies in the EPL 11/12 season. Green squares and red triangles indicate a winning or losing bet, respectively. Bets above the solid black line have a positive expected return.

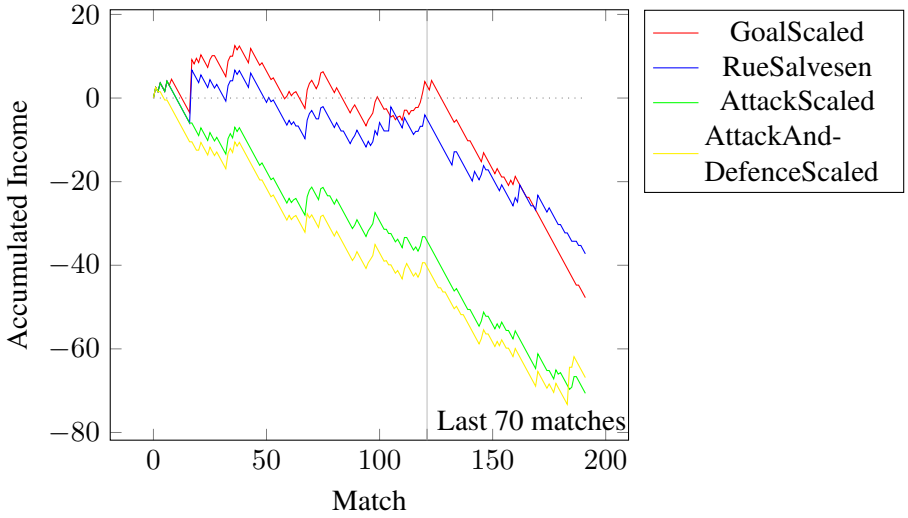


Figure 6.5: Accumulated income (in units) from betting with the Fixed bet betting strategy in the EPL 12/13 season.

6.2.1 Fixed bet

The success of the Fixed bet strategy in the previous season was followed by loss of about equal proportions this season. Figure 6.5 illustrates the performance of using the Fixed bet strategy with the different prediction models. We notice that the RueSalvesen and the GoalScaled models performed about equally bad, and that the same applied to the AttackScaled and AttackAndDefenceScaled models. We also notice that it was especially in the last 70 matches that all models performed poorly, and it was the GoalScaled model that took the hardest hit. An explanation to this might be that out of the last 70 matches this season, 35.7% ended in a home victory, 32.9% in a draw, and 31.4% in an away victory. This is quite far off the usual distribution of outcomes reported by Clarke and Norman (1995) (48.7% home, 26.7% draw and 24.6% away).

Out of all bets placed using the Fixed bet strategy this season, 26.01% were on matches with odds above 5.0. 4.98% of these bets were successful, and resulted in a total loss of -47.72%. This was the opposite of last season, where we experienced a profit of 38.27% from the same type of bets. If we were lucky in the 11/12 season or unlucky this season, we do not know. However, we do know that the RueSalvesen model had the best result concerning bets with odds above 5.0 this season. It had a success rate of 7.84% and had an average return of -39.39% for each such bet made. An overview of how the different prediction models per-

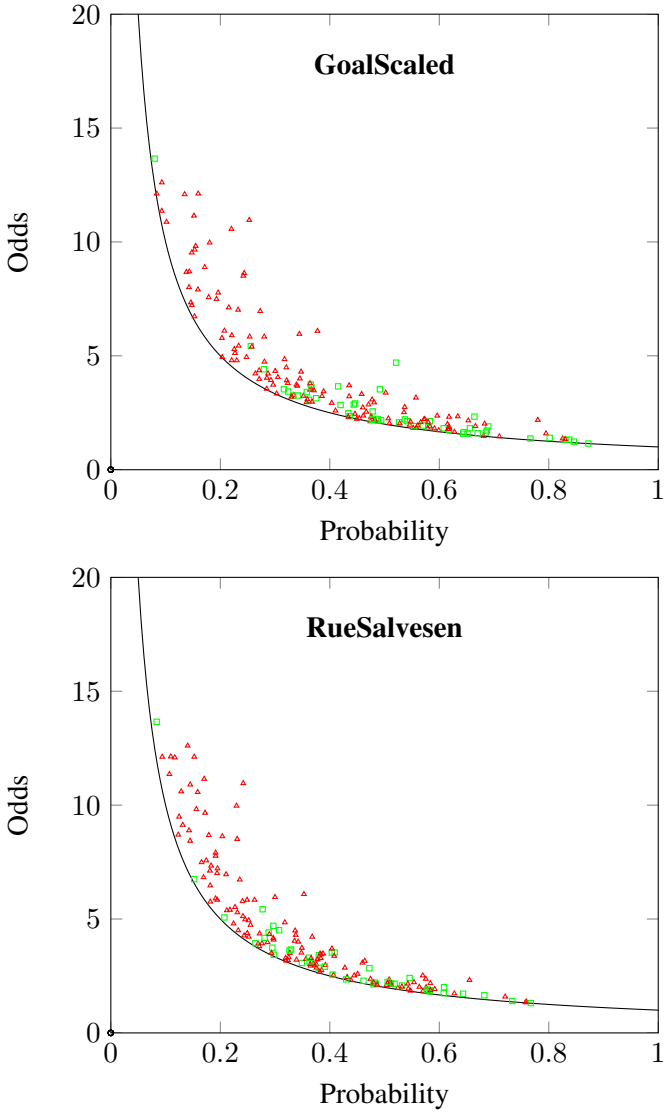


Figure 6.6: Overview of bets placed using the GoalScaled and RueSalvesen prediction models combined with the Fixed bet betting strategy in the EPL 12/13 season. Green squares and red triangles indicate a winning or losing bet, respectively. Bets above the solid black line have a positive expected return. We see that there is not much difference between the two models, but the RueSalvesen strategy finds some more winning bets with high odds.

Table 6.6: Performance of the different models where the bets are placed on outcomes with odds greater than 5.0 in the EPL 12/13.

Model	Portion of bets	Success rate	Return
RueSalvesen	29.14%	7.84%	-39.39%
Goal Scaled	23.08%	4.76%	-54.55%
AttackScaled	25.56%	2.17%	-88.20%
AttackAnd-DefenceScaled	26.40%	4.35%	-67.47%

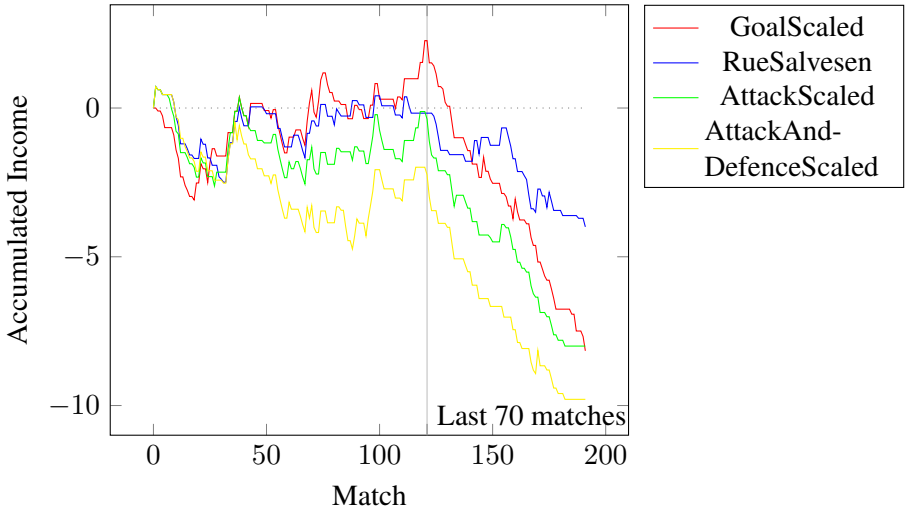


Figure 6.7: Accumulated income (in units) from betting with the Fixed return betting strategy in the EPL 12/13 season.

formed regarding bets with odds above 5.0 is displayed in Table 6.6. A comparison of how bets were placed by the RueSalvesen and GoalScaled models is illustrated in Figure 6.6.

6.2.2 Fixed return

The Fixed return strategy suffered the same fate as the Fixed bet strategy this season. Figure 6.7 shows how the different prediction models performed using this strategy. We notice that the RueSalvesen model performed better than the other models in the last 70 matches.

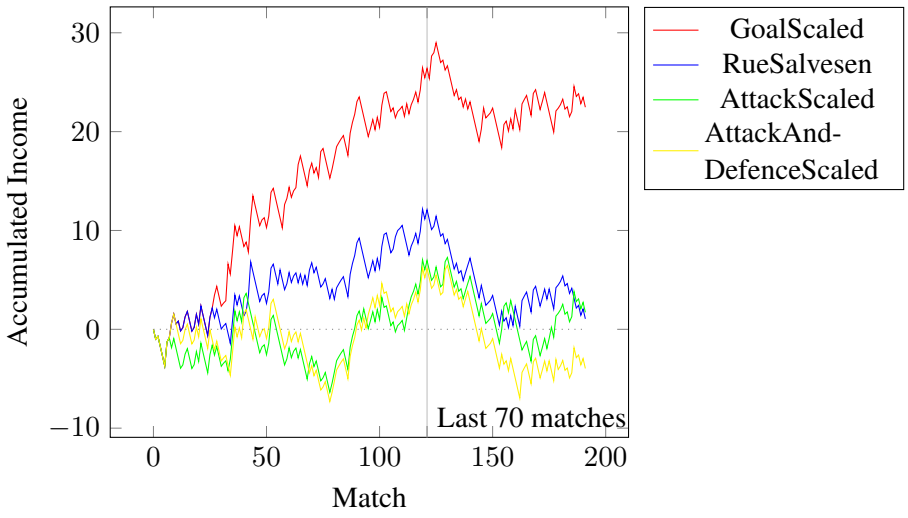


Figure 6.8: Accumulated income (in units) from betting with the Wallpapering betting strategy in the EPL 12/13 season.

Table 6.7: Performance of the different models concerning bets placed on outcomes with negative expected return in the EPL 12/13.

Model	Portion of bets	Success rate	Return
RueSalvesen	61.78%	61.86%	8.92%
Goal Scaled	44.50%	68.24%	22.42%
AttackScaled	53.93%	63.11%	11.90%
AttackAnd-DefenceScaled	53.40%	60.78%	8.29%

6.2.3 Wallpapering

The Wallpapering strategy performed about equally well both seasons. In fact, this was the only betting strategy that made money this season. Figure 6.8 shows the Wallpapering betting strategy used with the four prediction models. Combined with the GoalScaled prediction model, it generated a profit of 11.76%, which was a lot better than all other combinations.

As mentioned in subsection 2.8.4, the Wallpapering strategy places bets where a traditional gambler would not, namely where the expected return is negative. Table 6.7 shows the success of allowing for this kind of bets to be placed. The GoalScaled prediction model was the one placing the fewest amounts of bets with negative expected return, and still was the one with the greatest profit. In the

Table 6.8: The final income after combining prediction models with betting strategies for the EPL 13/14 season.

Model	Betting Strategy		
	Fixed Bet	Fixed Return	Wallpapering
RueSalvesen	-14.55%	-0.35%	-2.06%
Goal Scaled	15.93%	6.45%	5.82%
AttackScaled	-11.37%	-2.71%	-2.37%
AttackAnd-DefenceScaled	-18.01%	1.75%	-11.08%

final 70 matches it earned a profit of 25.6% on bets with negative expected return, which probably is the reason why the Wallpapering strategy was such a success this season, especially for the GoalScaled model.

6.3 English Premier League 2013/2014

Results from betting on outcomes in the EPL 13/14 season are presented in Table 6.8. We notice that this season the GoalScaled model outperformed the others using any betting strategy. The difference was especially large using the Fixed bet strategy, where the GoalScaled model earned a profit of 15.93%, while the others lost money.

6.3.1 Fixed bet

How the different models performed using the Fixed bet strategy is visualised in Figure 6.9.

The GoalScaled model seemed to be superior this season, especially when using the Fixed bet strategy. One reason why this was the case might be the advantage the GoalScaled model has, being able to detect if any top goal scorers are out of a starting line-up. For instance, Newcastle United's top goal scorer Loïc Remy (14 goals) was out of the starting line-up 8 matches in the second half of the season due to injury and a suspension. Newcastle United also sold Yohan Cabaye (7 goals) during the winter transfer window. These two players scored 21 of Newcastle United's 43 goals this season, so it is reasonable to assume that the team was weakened in their absence. Table 6.9 shows the predictions made by the GoalScaled and RueSalvesen models for matches where Loïc Remy was suspended or injured. It also includes the final three matches of the season, where Loïc Remy was back in the starting line-up. We see that when Loïc Remy disappears from the starting line-up, and when he comes back in, the GoalsScaled

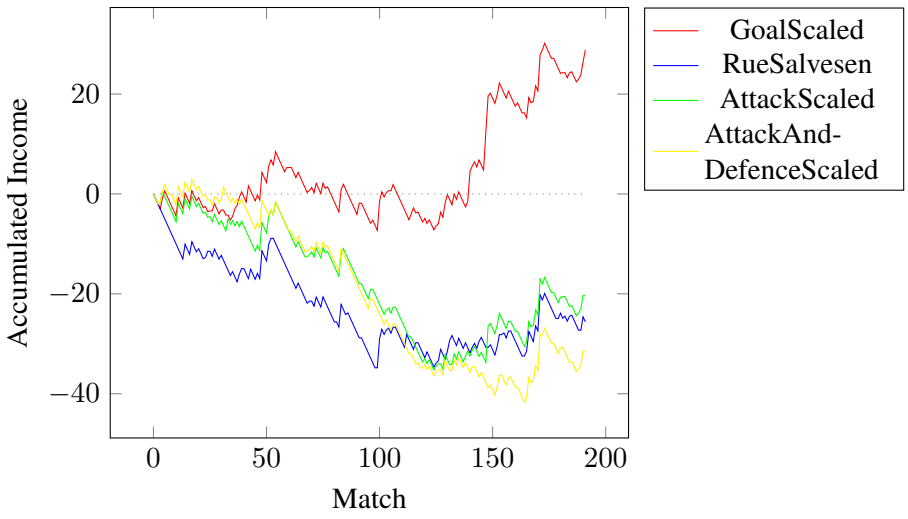


Figure 6.9: Accumulated income (in units) from betting with the Fixed bet strategy in the EPL 13/14 season.

Table 6.9: Probabilities for actual outcomes of matches where Newcastle United's top goal scorer Loïc Remy was missing from the starting line-up, generated by the GoalScaled and RueSalvesen models. Green numbers indicate that a bet was placed on the correct outcome. Red numbers indicate that a bet was placed, but on the wrong outcome. The betting strategy used is Fixed bet in the EPL 13/14 season.

Match	Date	Result	Probability(actual outcome) GoalScaled	RueSalvesen	Bookmaker
Remy suspended					
NewU vs Sund	1/2	0-3(A)	0.39	0.20	0.24
Remy injured					
Fulh vs NewU	15/3	1-0(H)	0.29	0.31	0.35
NewU vs CryP	22/3	1-0(H)	0.41	0.43	0.54
NewU vs Ever	25/3	0-3(A)	0.50	0.36	0.45
Sout vs NewU	29/3	4-0(H)	0.63	0.60	0.58
NewU vs ManU	5/4	0-4(A)	0.53	0.45	0.51
Stok vs NewU	12/4	1-0(H)	0.72	0.71	0.47
NewU vs Swan	19/4	1-2(A)	0.59	0.49	0.38
Remy back in starting line-up					
Arse vs NewU	28/4	3-0(H)	0.79	0.75	0,79
NewU vs Card	3/5	3-0(H)	0,52	0,44	0.50
Live vs NewU	11/5	2-1(H)	0,87	0,89	0.83

Table 6.10: Probabilities for the actual outcomes of matches where Aston Villa's top goal scorer Christian Benteke was injured, generated by the GoalScaled and RueSalvesen models. Green numbers indicate that a bet was made on the correct outcome. Red numbers indicate that a bet was made, but on the wrong outcome. The betting strategy used is Fixed bet in the EPL 13/14 season.

Match	Date	Result	Probability(actual outcome)		
			GoalScaled	RueSalvesen	Bookmaker
Benteke injured					
AstV vs Fulh	5/4	1-2(A)	0.33	0.19	0.26
CryP vs AstV	12/4	1-0(H)	0,47	0.51	0,48
AstV vs Sout	19/4	0-0(D)	0.27	0.23	0.30
Swan vs AstV	26/4	4-1(H)	0,62	0.19	0.60
AstV vs Hull	3/5	3-1(H)	0.27	0.36	0.41
ManC vs AstV	7/5	4-0(H)	0,91	0.85	0.88
Tott vs AstV	11/5	3-0(H)	0.83	0.65	0.67

model responds to this quickly. The RueSalvesen model has no idea that the top goal scorer is missing, so it needs to observe Newcastle's form without him for some matches before it responds. The same accounts for when he comes back in. Over these 12 matches, the GoalScaled model won 9 out of 11 bets, generating a profit of 60.88%, while the RueSalvesen model lost 8 out of 11 bets, resulting in a profit of -50.29%.

Another example is Aston Villa's Christian Benteke, who scored 10 out of their 38 goals this season. He was out of their starting line-up the last 7 matches because of an injury. Table 6.10 compares how the RueSalvesen and GoalScaled prediction models performed over these 7 matches. The GoalScaled model gained an income of 16.3%, while the RueSalvesen model lost -70.2%.

From these two examples, it is clear that the GoalScaled model has an advantage by knowing if the top goal scorer is missing from a starting lineup, and this is reflected in the predictions made. One would expect that the AttackScaled and AttackAndDefenceScaled models also should perform better in these two cases, as they also penalise the attack strength of teams when players who scores many goals are absent. They do perform better than the RueSalvesen model in the two examples above, but not as good as the GoalScaled model. This is probably because they do not penalise the absence of a goal scorer as much as the GoalScaled model does. In matches where the top goal scorer is absent, as in our two examples above, it is therefore quite logical that the GoalScaled model shows the best performance. Maybe, if a key midfield player was absent, the AttackScaled and AttackAndDefenceScaled models would catch this and outperform the other two

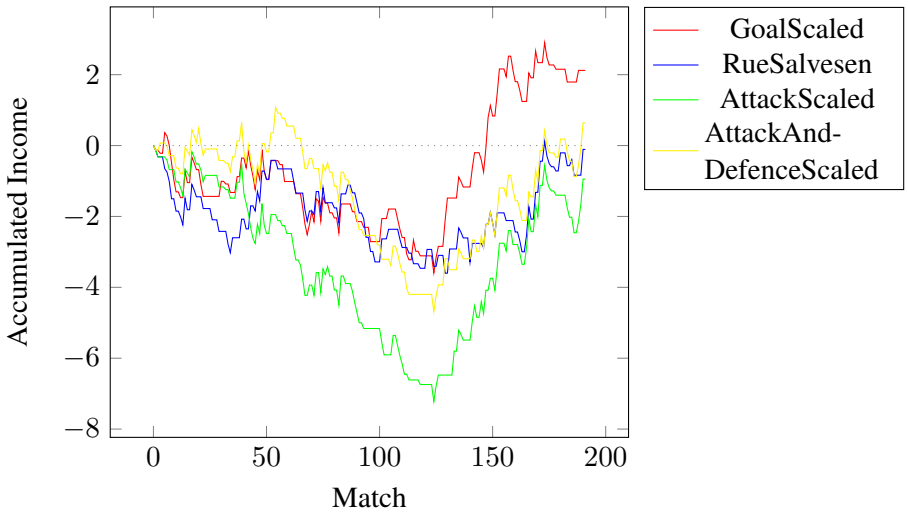


Figure 6.10: Accumulated income (in units) from betting with the Fixed return strategy in the EPL 13/14.

models, but we have not found any good examples of this happening.

6.3.2 Fixed return

Figure 6.10 shows the results from betting with the Fixed return strategy. We notice that the difference between the GoalScaled model and the others is not as big as when using the Fixed bet strategy, but that the GoalScaled model still is better.

6.3.3 Wallpapering

The results of betting using the Wallpapering betting strategy is shown in Figure 6.11. As in the two previous seasons, the GoalScaled model generated the highest income this season. The AttackAndDefenceScaled model generated a negative income, and the RueSalvesen and AttackScaled models were in between generating about the same profit.

6.4 Overall Performance

In this section we will present the results of betting through all the three seasons from the previous sections. Since the AttackScaled and AttackAndDefenceScaled models did not match the results of the RueSalvesen and GoalScaled models, we

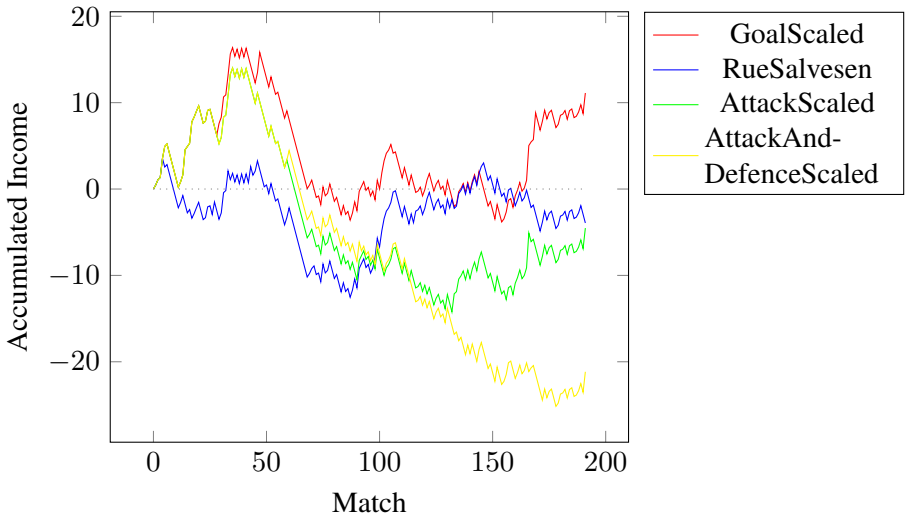


Figure 6.11: Accumulated income (in units) from betting with the Wallpapering strategy in the EPL 13/14.

Table 6.11: The final income after combining prediction models with betting strategies for the EPL 11/12, 12/13, and 13/14 seasons.

Model	Betting Strategy		
	Fixed Bet	Fixed Return	Wallpapering
RueSalvesen	-1.52%	3.53%	1.31%
GoalScaled	4.68%	3.33%	10.49%
AttackScaled	-8.31%	-2.22%	0.43%
AttackAnd-DefenceScaled	-6.82%	-1.09%	-4.85%

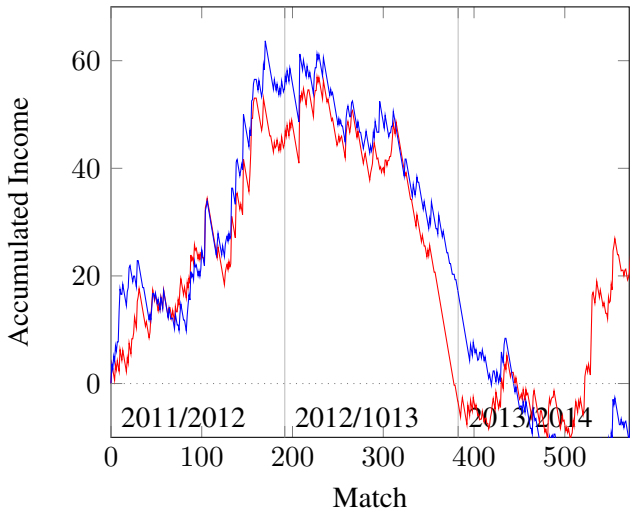


Figure 6.12: Result of using the GoalScaled (red) and RueSalvesen (blue) models with the Fixed bet strategy over all three seasons. Accumulated income is given in units.

will focus our discussion in this section on the two latter models. Table 6.11 shows the total income from betting using different combinations of prediction models and betting strategies.

6.4.1 Fixed bet

The three seasons discussed in the previous sections proved to be quite different in terms of the difficulty of finding the most profitable bets (as the Fixed bet strategy tries to do). The results from betting with the Fixed bet strategy through all these seasons are displayed in Figure 6.12.

Overall, the GoalScaled model outperformed the RueSalvesen model using the Fixed bet strategy, even though it only performed better in one out of three seasons. We believe this was the case because the GoalScaled model knows if a top goal scorer is unavailable for a match, as explained in subsection 6.3.1, and this knowledge really seemed to pay off in the 13/14 season of the EPL.

Table 6.12 compares the GoalScaled and RueSalvesen models in terms of high-odds (or low-probability) bets. The Fixed bet strategy would be successful if the models were good at finding outcomes where the odds are set too high by the bookmaker. We see from Table 6.12 that both the RueSalvesen and the GoalScaled model places bets on high-odds outcomes quite often, and the RueSalvesen model more often than the GoalScaled model. We also see that the success of these types

Table 6.12: Performance of the GoalScaled and RueSalvesen models regarding bets with odds greater than 5.0.

Model	Portion of bets	Success rate	Return
EPL 2011/2012			
RueSalvesen	34.08%	18.08%	39.43%
GoalScaled	30.77%	17.86%	37.41%
EPL 2012/2013			
RueSalvesen	29.14%	7.48%	-39.39%
GoalScaled	23.08%	4.76%	-54.55%
EPL 2013/2014			
RueSalvesen	28.41%	10%	-25.82%
GoalScaled	26.52%	16.67%	16.44%
All seasons combined			
RueSalvesen	30.57%	12.35%	-8.64%
GoalScaled	26.79%	13.7%	4.06%

of bets vary a lot from season to season, and seems to be closely related to the total success of the Fixed bet strategy.

As explained in section 2.7, bookmakers adjust their odds if many bets are made on a specific outcome in order to maximize their profit. The odds we have used in our experiments are the final odds that were set right before matches started. These odds were probably adjusted such that the outcome least attractive to pundits would become more attractive to bet on. This might be the reason why the Fixed bet strategy places bets on high-odds outcomes so often.

In the final rounds of a season, teams might find themselves in one out of three situations: Some teams have a chance of winning something, for instance winning the league or qualifying for a European cup. Other teams might be threatened by relegation and have to win their final matches in order to stay in the league. The final situation is the one of teams that do not have a chance of winning anything nor be relegated. Teams that find themselves in one of the two first situations will have more at stake in the final rounds of the season than the teams that find themselves in the third situation. Matches between teams in different situations might be affected by this. The bookmakers know which situation each team is in, and this might be reflected in their odds. However, the prediction models we have used do not consider the situation each team is in, and this could also be an explanation to the poor results we experienced at the end of the 12/13 season.

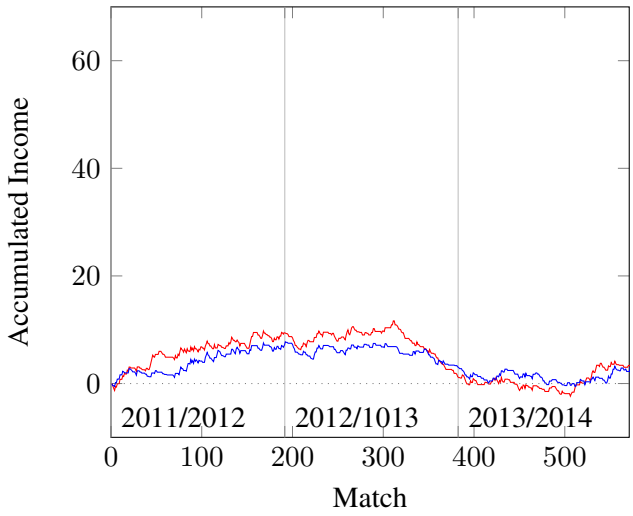


Figure 6.13: Result of using the GoalScaled (red) and RueSalvesen (blue) models with the Fixed return strategy over all three seasons. Accumulated income is given in units.

6.4.2 Fixed return

The Fixed return strategy is less risky than the other two strategies. The amount spent on bets is smaller, and the return in units is therefore also smaller. The provided safety margin leads to fewer bets being made, and the amount spent on each bet is on average smaller than with the other models. As we can see in Figure 6.13, there are no large ”jumps” in the plot. This is because the Fixed return strategy has small stakes on bets, and the income of a successful bet is always 1 unit. The risk of using this strategy is therefore low, and so is the potential gain.

6.4.3 Wallpapering

Figure 6.14 illustrates how well the GoalScaled and RueSalvesen models performed when paired with the Wallpapering strategy over all three seasons. We notice that the GoalScaled model generated a quite stable, positive, growth of income. The RueSalvesen model also made money over the three seasons, but without the same steady increase in income.

The idea of placing bets on outcomes with negative expected return did not sound like a good one at first. Table 6.13 displays the statistics of placing such bets using the Wallpapering strategy. We notice that such bets were placed more often when using the RueSalvesen model than when using the GoalScaled model. This means that the RueSalvesen model will more often consider a bet on the favourite

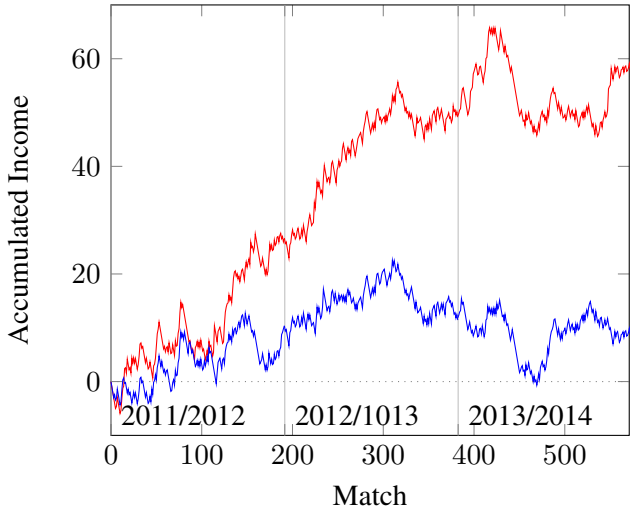


Figure 6.14: Result of using the GoalScaled (red) and RueSalvesen (blue) models with the Wallpapering strategy over all three seasons. Accumulated income is given in units.

Table 6.13: Performance of the GoalScaled and RueSalvesen models regarding bets with negative expected return.

Model	Portion of bets	Success rate	Return
EPL 2011/2012			
RueSalvesen	55.5%	53.77%	-3.56%
GoalScaled	47.12%	58.89%	7.67%
EPL 2012/2013			
RueSalvesen	61.78%	61.86%	8.92%
GoalScaled	44.5%	68.24%	22.42%
EPL 2013/2014			
RueSalvesen	54.97%	59.05%	-0.73%
GoalScaled	45.03%	60.47%	6.92%
All seasons combined			
RueSalvesen	57,42%	58.36%	1.82%
GoalScaled	45.55%	62.45%	12.23%

Table 6.14: Betting decisions made by the Fixed bet strategy when given probabilities from three different chains. Even though all three chains provide almost the same prediction, we observe that the betting decision can be easily affected. All values in this table are from a match between QPR and Norwich, played February 2nd, 2013, which ended 0-0 (D).

	Prediction			$1/\omega$			
Chain	H	D	A	H	D	A	Bet
Chain1	0.42	0.30	0.28	0.51	0.29	0.26	A
Chain2	0.43	0.30	0.27	0.51	0.29	0.26	A
Chain3	0.43	0.31	0.26	0.51	0.29	0.26	D
All chains	0.43	0.30	0.27	0.51	0.29	0.26	A

to win to have a negative expected return. This is also an explanation of why the RueSalvesen model places bets more often on outcomes with odds greater than 5.0 when using the Fixed bet strategy. It is interesting to see that both models made money over three seasons by making bets with negative expected return. This could be explained by the uncertainty in our predictions, which we will consider in subsection 6.4.4.

6.4.4 Uncertainty

There is some uncertainty involved when using the MCMC algorithm. Figure 6.15 displays the results of using single chains for betting, and shows how this uncertainty impacts our betting results. We see that the Fixed bet strategy seems to be more affected by this uncertainty than the Wallpapering strategy. This is to be expected, since the Fixed bet strategy can decide whether to bet or not based on minor differences between our and the bookmaker's predictions. The same accounts for the Fixed return strategy. If a prediction is about the same as the bookmaker's prediction, a change as small as +-1% could make the expected return positive or negative, and we then decide how to bet based on this small difference.

Figure 6.16 visualises the expected return of placing bets on home victories in all matches of the second half of the EPL 13/14 season. The predictions used for creating this plot were generated by the GoalScaled model. If the expected return is above 1 in this plot, the Fixed bet strategy would place a bet on that outcome. There are quite a few outcomes where the expected return is just above or below 1. If we had used predictions from a different run of the GoalScaled model, the expected return of these outcomes might be slightly altered, leading to different betting decisions using the Fixed bet strategy.

Table 6.14 illustrates how a small change in our predictions can affect the decisions made when using the Fixed bet strategy. In "Chain3", the probability for

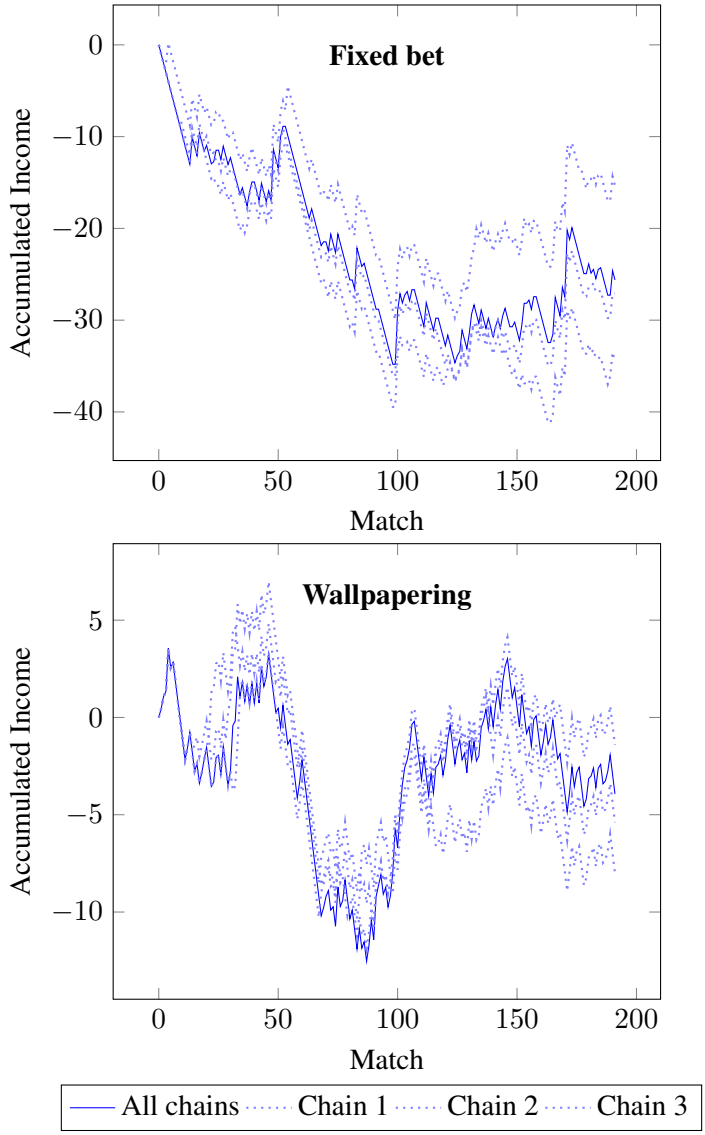


Figure 6.15: Results of using predictions from single chains for betting in the EPL 13/14 season using the RueSalvesen prediction model.

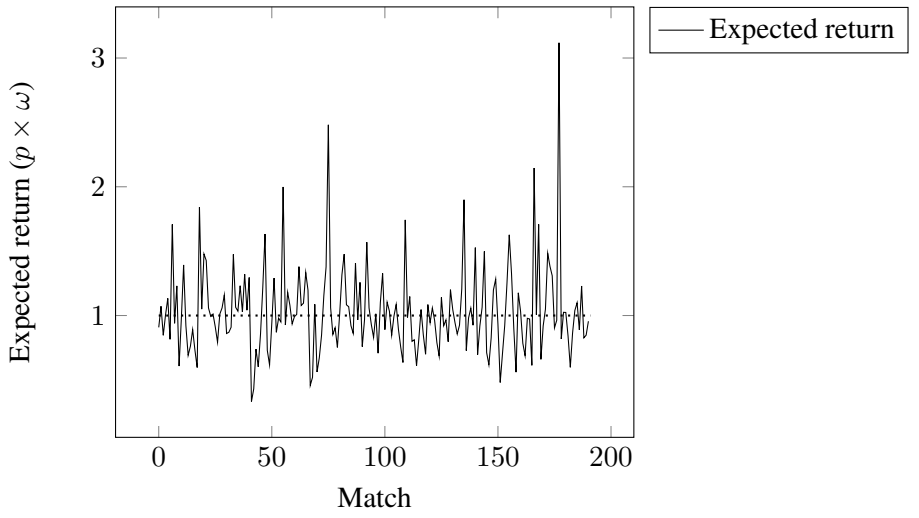


Figure 6.16: Expected return of betting on home victories in the second half of the EPL 13/14 season. The predictions used here are generated by the GoalScaled prediction model.

a draw outcome has the highest expected return, while in all other cases it is the away outcome. If the Wallpapering strategy had been used in this case, it would have bet on a draw in all chains, because the probability of a draw outcome is above 27% in all chains. This shows that the decisions made by the Wallpapering strategy will not be affected as much by small differences. This will only happen if the probability of a draw is about 27%, or if the probabilities for home and away victory are about the same.

We know that the odds provided by different bookmakers also differ. This means that the decisions made by the Fixed bet strategy could change by simply using another bookmaker. This would work if we could rely on our predictions being very accurate, but it seems that we can not rely on this at this stage. The fact that the Fixed bet strategy would perform differently if tested using different bookmakers makes it difficult to evaluate how good our prediction models really are. The Wallpapering strategy will always make the same betting decisions given a set of predictions.

6.4.5 Varying the odds

We have also conducted some tests using the maximum odds among the same eight bookmakers that provided the average odds in earlier experiments. The results

Table 6.15: The returns for the three seasons 11/12, 12/13 and 13/14 with the maximum odds.

Model	Fixed Bet	Fixed Return	Wallpapering
EPL 2011/2012			
RueSalvesen	44.07%	32.67%	10.12%
GoalScaled	33.81%	27.92%	19.64%
EPL 2012/2013			
RueSalvesen	-12.90%	-14.06%	4.83%
GoalScaled	-15.17%	-19.17%	16.79%
EPL 2013/2014			
RueSalvesen	-9.82%	0.63%	1.80%
GoalScaled	25.47%	22.80%	10.71%
All seasons combined			
RueSalvesen	7.12%	6.37%	5.58%
GoalScaled	18.08%	11.54%	15.71%

for the different seasons are shown in Table 6.15. We have only presented the results from the runs of the GoalScaled model and the RueSalvesen model, since this is enough to illustrate the point (the behaviour was similar among the two other models). By comparing this table with the tables 6.1, 6.5, 6.8 and 6.11, it is obvious that higher odds will yield higher returns with all the betting strategies and all the prediction models. This is also an indication of that when developing a new prediction model, it should be tested with average odds or odds from a given bookmaker, not the maximum odds. Also, while using the model in the future, using it with maximum odds means that a lot of time must be spent in order to search for the highest odds among several bookmakers.

Another point that can be illustrated by comparing the tables, is that the Wallpapering strategy still seems to be a better factor for evaluating how good a model is than the other betting strategies. The Wallpapering strategy puts money on the matches where the prediction system finds it reasonable, without regarding the odds. This means that higher odds only will result in a larger income, not a change in the placement of the bets. Fixed bet, on the other hand, will often change its placements of bets to outcomes with higher odds. And a match with high odds will often be an outcome that is more uncertain. This can lead to the loss of a larger percentage of the wagers, but since the odds on average are higher, the total outcome is larger. If the Fixed return strategy is to bet on the same match as earlier, it will place a smaller amount on the wager. The result is the same income as earlier, but the income in percent is larger than before.

Table 6.16: The returns for the three seasons 11/12, 12/13 and 13/14 with the odds provided by Norsk Tipping.

Model	Fixed Bet	Fixed Return	Wallpapering
EPL 2011/2012			
RueSalvesen	16.60%	-8.81%	-1.37%
GoalScaled	14.27%	14.75%	5.90%
EPL 2012/2013			
RueSalvesen	-39.65%	-34,41%	-5.32%
GoalScaled	-32.79%	-36.12%	4.56%
EPL 2013/2014			
RueSalvesen	-20.01%	-14.66%	-6.44%
GoalScaled	8.19%	5.77%	1.12%
All seasons combined			
RueSalvesen	-13.99%	-17.51%	-1.08%
GoalScaled	-2.53%	-4.29%	3.86%

Norsk Tipping is the only legal bookmaker in Norway. They usually have a margin between 10% to 13% (found by investigating the odds of a number of random matches in the three seasons). They provided us with their odds from the three seasons we have investigated. The performance of the GoalScaled and RueSalvesen models using their odds is shown in Table 6.16. By combining the three seasons, the GoalScaled model performed better than the RueSalvesen model, and was able to earn money using the Wallpapering betting strategy in all seasons. This shows that the GoalScaled model combined with the Wallpapering betting strategy is able to earn money even when the bookmaker uses a margin as high as 10%-13%.

6.4.6 Rank probability score

Table 6.17 contains the rank probability score (RPS) for all prediction models from our experiments. How to calculate the RPS is described in subsection 2.9.3. We notice that even though the RueSalvesen model seems to be outperformed by the GoalScaled model when it comes to betting, it performs slightly better than the other models when evaluated using RPS. By comparing the tables 6.1, 6.5 and 6.8 to the RPS, we did not find any correlations. We can therefore conclude that the RPS and the betting market has little to do with each other, and are two different measures for performance. Having a good result on one of them does not directly imply a good result at the other measure.

Table 6.17: Rank probability score for the different prediction models for the EPL 11/12, 12/13 and 13/14 seasons.

Model	Avg RPS			
	2011/2012	2012/2013	2013/2014	Overall
RueSalvesen	0.207	0.198	0.201	0.202
GoalScaled	0.210	0.203	0.196	0.203
AttackScaled	0.211	0.206	0.203	0.207
AttackAnd-DefenceScaled	0.210	0.209	0.204	0.208

6.5 Fulfilment of Goals

In this chapter, we have applied our prediction models to the betting market, together with the RueSalvesen model. This fulfils Goal 3 stated in section 1.2. We also showed that we were able to beat the original model on the betting market with the GoalScaled model developed by us, over three seasons. By beating the benchmark model and fulfilling Goal 3, we answer Research question 3 from the same section.

Chapter 7

Conclusion and Further Work

In this chapter we present the conclusions drawn based on the experiments conducted in this thesis. The Further Work section contains ideas and suggestions for improvements of the prediction system in the future.

7.1 Conclusion

In this thesis we have created a purely data-driven system for predicting outcomes of football matches. We have also created an extension of a simplified version of the prediction model proposed by Rue and Salvesen (2000). This extension includes information on the probable starting line-up of each team in a match. We have shown that, even though football is a sport involving a lot of chance events and luck, it is possible to earn money in the betting market over three seasons. Furthermore, our GoalScaled model outperforms the RueSalvesen model over these three seasons, generating a profit of 10.49% when combined with the Wallpapering betting strategy. This indicates that including information on individual player level can lead to more accurate predictions.

7.1.1 Comparing the models

The GoalScaled model only aims to catch if an attacker that normally contributes with many goals is missing from the starting line-up. We have also created two models including more detailed information on each player, in an attempt to catch if players that contribute in other ways also are missing from the starting line-up. The AttackScaled and AttackAndDefenceScaled models showed no improvement over the RueSalvesen model. This might be because the level of detail becomes too high, or our scaling values become too dominant. However, there are a lot of ways

to include this type of information in a prediction model, and we can not state that doing this is a bad idea. We can only conclude that our approach did not pay off. We think the reason is that since the original RueSalvesen model is based only on goals, it is unnatural to create scaling factors that include information about other attributes. Instead, the two models should have been altered, such that the Bayesian network contained nodes for all the attributes that were used for scaling. Another factor to consider here, is that we did not conduct any tests for finding the different attributes to use in our scales. We used the results presented by Ellefsrød (2013), but we should have conducted correlation tests with more than two attributes at a time, not only comparing everything to the number of goals scored.

We believe the main reason for the GoalScaled model outperforming the RueSalvesen model, is that the GoalScaled model reacts to changes much faster than the original model. When one of the most scoring players in a team is absent, this is reflected in the predictions immediately in the GoalScaled model. The RueSalvesen model must observe some matches before the change shows in the predictions. When the player then returns again, the GoalScaled model's predictions takes this into account, while the RueSalvesen must again wait some matches before this is reflected in the predictions.

7.1.2 Applying the models to the betting market

Even though it is satisfying to generate an income of 40% in one season, this does not mean that the system generating this income will perform well in another season. We have seen this in chapter 6. Even though all models performed excellently in the EPL 2011/2012 season, the results in the following season were poor, which shows the importance of testing a system over several seasons. If we were able to identify what makes a season profitable, this could be very promising, and could be used in our betting decisions.

In subsection 6.4.5, we tested our system with the highest available odds. This increased the income for all the models in all the seasons. Actually, this shows that it is possible to gain a total income of 18.08% using the GoalScaled model combined with Fixed bet over the three seasons. It also shows that a system should be evaluated using average odds or odds from a given bookmaker, because the results will be more equal to the actual results the system would have achieved in real life. It is possible to gather high odds for each of the matches during a weekend, but this is time consuming and difficult. We have found no other papers or models that were able to gain certain income over time using average odds. Constantinou and Fenton (2013) presented a model which earned money over ten seasons, but they used maximum odds.

7.1.3 Betting strategies

The betting strategies explored in chapter 6 can be compared with stock trading strategies. Some involve high risk and high potential gain, while others have less risk and less potential gain.

For the Fixed bet and Fixed return strategies to work, we need to beat the bookmakers' predictions in order to find the most profitable outcome to bet on. The difference between our predictions and the bookmaker's is often small, and our predictions have some uncertainty attached to them, as shown in subsection 6.4.4. Deciding on how to place bets based on small differences between our and the bookmaker's predictions can therefore be considered as high risk. However, if it works, the potential gain is high. The Fixed return strategy has a safety margin that prevents it from placing a bet if this difference is very small, which means that less risk is involved when using this strategy. The amount placed on each bet is smaller on bets with high odds than on bets with low odds, making the risk even lower.

Using the Wallpapering strategy for betting only relies on our system being able to find the most probable outcome, and not necessarily the most profitable one. This means that the uncertainty attached to our predictions does not affect our decisions as much as when using the other betting strategies. As we saw in Table 6.11, trying to find the most probable outcome seems to be a better strategy than trying to find the most profitable outcome in our case. If one wants to find the most profitable outcome, we feel that more of the uncertainty in the generated predictions needs to be eliminated.

7.1.4 Fulfilment of goals and answering research questions

In section 1.2, we stated three research questions that should be answered in this thesis. In addition, we stated three goals that should be achieved in order to answer the research questions.

Our first goal was to extend an existing prediction model. We decided to extend a simplification of the model presented by Rue and Salvesen (2000). In total, we created three different versions of this model, where the model was extended by scaling the attack and defence strengths with different factors. All the models incorporated information about which players that were present in a probable starting line-up. The simplest model, GoalScaled, scaled the attack strengths of teams based on how many of a team's goals the players in the starting line-up had scored. The two other models had more comprehensive scaling factors, where one scaled the attack value, and the other one scaled both the attack and defence value of the teams. The achievement of this goal is described in chapter 3.

The second goal was to implement the models, and create a prediction system based on these models. We created a purely data-driven system, implemented mostly in C#. The program consisted of a crawler, a database, the prediction models, and classes to process the predictions, place bets, evaluate returns, create plots and display statistics. This goal is fulfilled in chapter 4.

The third and final goal regarded the evaluation of our system. It should be tested on the betting market together with the original model. In chapter 6, this goal is fulfilled, and the results from applying the models to the betting market using three different betting strategies are presented. The betting strategies used were Fixed bet, Fixed return and Wallpapering. We saw that the Wallpapering strategy was the best to evaluate the prediction models, since this strategy was less affected by noise in the predictions than the other two.

The results from the tests on the betting market show that our GoalScaled model is able to gain profit over three seasons, and beats the RueSalvesen prediction model with two of the three betting strategies. It does also have a total positive return for all the betting strategies when looking at all the three seasons together, as the only prediction model where this is a fact. In total, GoalScaled has the highest return, with 10.49% over the three seasons combined, using average odds and the Wallpapering betting strategy. By using maximum odds, this number is increased to 15.71%.

By fulfilling the three goals, we were able to answer our three research questions. We showed that it is possible to extend an existing prediction model with information about individual players (Research question 1), we created a purely data-driven system based on the model (Research question 2) and we were able to outperform the existing model on the betting market (Research question 3).

7.2 Further Work

This section contains our ideas and plans for further extensions of the prediction system.

7.2.1 Initial values of team strengths at the start of the season

Our system does not make use of any information on how good the different teams are at the start of a new season. However, if information on the previous season exists it could be used to estimate the initial strength of each team. The problem here is that teams are active in the transfer market in the summer. Players and managers are sold, bought and sacked between seasons, which makes it difficult to determine how well a team will perform in one season based on the previous. In addition to this, clubs are promoted from a lower division each year, and it is

difficult to determine how well these will perform in a higher division. However, basing initial team strength on a team's earlier performance might be better than initialising them randomly. And for the newly promoted clubs, their team strength could be set based on historical data on which league position such teams usually end up in. Expert knowledge could also be applied at this stage to ensure that clubs who are considered to have done poorly or good on the transfer market have their initial team strengths adjusted accordingly.

7.2.2 Detecting injured and banned players

In this paper the focus has been on using probable starting line-ups for scaling team strengths. An alternative to this approach may be to detect and rule out injured and banned players from the squad, players we know are unavailable for an upcoming match. Team strength could then be scaled based on all available players in a squad, instead of basing it on probable line-ups. Information on unavailable players is available on the WhoScored preview site for each match (WhoScored.com, 2014).

7.2.3 Create an even more data intensive prediction model

As stated in subsection 2.10.2, many authors of previous research papers suggest that prediction models could perform better if more detailed data is included in their model. Our inclusion of information on which players that start a match is just one way of doing this. One could also include information on passing statistics, key passes, tackles, and so on, as variables directly in the Bayesian network. This way of adding more detailed data was investigated by Andresen and Dubicki (2013) and Ellefsrød (2013). It could be interesting to investigate if such models would perform better if they were combined with the scaling technique introduced in this paper.

7.2.4 Season bets

In this paper we have only considered betting on the outcome of single football matches. Bookmakers offer many other ways of betting. For instance you could bet on the exact result of a football match, how many bookings a team receives, or you could combine such bets. One type of betting that could be interesting for a system such as ours is seasonal bets: Which team will win the league and which teams will be relegated?

Our system supports simulating and creating predictions for single matches where all results from matches played before that match is known. This way the training set is extended for every match, and the system is updated with all known

information for each new match. This is the type of simulation that is run in all our experiments. However, the system also supports running simulations where only the results from matches in the training set is known. This type of simulation could be used for predicting the outcome of all the remaining matches of a season, and predict the final standings in the league table. We have not conducted any experiments for predicting final league positions, but this is something that could be interesting to try in the future.

7.2.5 Enhancing the Wallpapering betting strategy

The Wallpapering betting strategy had a stable, positive performance when used with our GoalScaled prediction model. We believe this stability comes from the fact that it makes rule based decisions, and that it does not care about what the potential gain of a bet is. The amount to bet on each match, however, could be experimented with in order to maximize the profit of the bets we make. This means that the Wallpapering strategy is first used to select which outcome to place a bet on, and then a new strategy is applied to decide which amount to bet on the match. This could be as simple as investigating the odds, and betting more on matches with odds below some threshold than matches above the threshold. Or the amount to bet could for instance be dependent on the difference between our predictions and the bookmaker's, or it could be dependent on what the odds are, such as in the Fixed bet strategy.

7.2.6 Betting for real

Currently, our football prediction system does not have support for generating predictions for next weekend's matches, but this can quite easily be implemented. It would be interesting to see if generated predictions for next weekend's matches could be used by an expert in order to place bets. We saw in chapter 6 that our prediction system can have trouble predicting outcomes at the end of a season. But, when the system is used by someone who has expert knowledge on football (and knows how the system works), he could decide where to place bets based on the probabilities generated by the system.

Bibliography

Ali, A., 2011. Measuring soccer skill performance: a review. Scandinavian Journal of Medicine & Science in Sports 21 (2), 170–183.
URL http://dx.doi.org/10.1111/j.1600-0838.2010.01256.x

Andresen, T. R., Dubicki, D., 2013. The betting machine. Master's thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway.

Boulier, B. L., Stekler, H., 2003. Predicting the outcomes of national football league games. International Journal of Forecasting 19 (2), 257 – 270.
URL http://www.sciencedirect.com/science/article/pii/S0169207001001443

Carmichael, F., Thomas, D., Ward, R., 2000. Team performance: the case of english premiership football. Managerial and Decision Economics 21 (1), 31–45.
URL http://dx.doi.org/10.1002/1099-1468(200001/02)21:1<31::AID-MDE963>3.0.CO;2-Q

Carron, A. V., Bray, S. R., Eys, M. A., 2002. Team cohesion and team success in sport. Journal of Sports Sciences 20 (2), 119–126, pMID: 11811568.
URL http://www.tandfonline.com/doi/abs/10.1080/026404102317200828

Casella, G., George, E. I., 1992. Explaining the gibbs sampler. The American Statistician 46 (3), 167–174.

Cattelan, M., Varin, C., Firth, D., 2013. Dynamic bradley–terry modelling of sports tournaments. Journal of the Royal Statistical Society: Series C (Applied Statistics) 62 (1), 135–150.

URL http://dx.doi.org/10.1111/j.1467-9876.2012.01046.x

Charles, N., 2014. http://www.wallpaperingfog.co.uk/2014/01/betting-using-my-model-of-premier.html, accessed: March 31, 2014.

Chib, S., Greenberg, E., 1995. Understanding the metropolis-hastings algorithm. The American Statistician 49 (4), 327–335.

Clarke, S., Dyte, D., 2000. Using official ratings to simulate major tennis tournaments. International Transactions in Operational Research 7 (6), 585–594.
URL http://dx.doi.org/10.1111/j.1475-3995.2000.tb00218.x

Clarke, S. R., Norman, J. M., 1995. Home ground advantage of individual clubs in english soccer. Journal of the Royal Statistical Society. Series D (The Statistician) 44 (4), pp. 509–521.
URL http://www.jstor.org/stable/2348899

Constantinou, A. C., Fenton, N. E., 2012. Solving the problem of inadequate scoring rules for assessing probabilistic football forecast models. Journal of Quantitative Analysis in Sports 8 (1).
URL http://www.degruyter.com/view/j/jqas.2012.8.issue-1/1559-0410.1418/1559-0410.1418.xml

Constantinou, A. C., Fenton, N. E., 2013. Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries. Journal of Quantitative Analysis in Sports 9 (1), 37–50.
URL http://www.degruyter.com/view/j/jqas.2013.9.issue-1/jqas-2012-0036/jqas-2012-0036.xml

Dixon, M. J., Coles, S. G., 1997. Modelling association football scores and inefficiencies in the football betting market. Journal of the Royal Statistical Society: Series C (Applied Statistics) 46 (2), 265–280.
URL http://dx.doi.org/10.1111/1467-9876.00065

Duch, J., Waitzman, J. S., Amaral, L. A. N., 06 2010. Quantifying the performance of individual players in a team activity. PLoS ONE 5 (6), e10937.
URL http://dx.doi.org/10.1371%2Fjournal.pone.0010937

Dyte, D., Clarke, S. R., 2000. A ratings based poisson model for world cup soccer simulation. The Journal of the Operational Research Society 51 (8), pp. 993–

998.
URL http://www.jstor.org/stable/254054

Ellefsrød, M. B., 2013. The betting machine. Master's thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway.

Football-data.co.uk, 2014. http://football-data.co.uk/, accessed: December 12, 2013 and May 16, 2014.

Forrest, D., Goddard, J., Simmons, R., 2005. Odds-setters as forecasters: The case of english football. International Journal of Forecasting 21 (3), 551 – 564.
URL http://www.sciencedirect.com/science/article/pii/S0169207005000300

Goddard, J., 2005. Regression models for forecasting goals and match results in association football. International Journal of Forecasting 21 (2), 331 – 340.
URL http://www.sciencedirect.com/science/article/pii/S0169207004000676

Goddard, J., Asimakopoulos, I., 2004. Forecasting football results and the efficiency of fixed-odds betting. Journal of Forecasting 23 (1), 51–66.
URL http://dx.doi.org/10.1002/for.877

Grehaigne, J.-F., Bouthier, D., David, B., 1997. Dynamic-system analysis of opponent relationships in collective actions in soccer. Journal of Sports Sciences 15 (2), 137–149, pMID: 9258844.
URL http://www.tandfonline.com/doi/abs/10.1080/026404197367416

Grund, T. U., 2012. Network structure and team performance: The case of english premier league soccer teams. Social Networks 34 (4), 682 – 690.
URL http://www.sciencedirect.com/science/article/pii/S0378873312000500

Hastings, W. K., 1970. Monte carlo sampling methods using markov chains and their applications. Biometrika 57 (1), 97–109.

Hughes, M., Franks, I., 2005. Analysis of passing sequences, shots and goals in soccer. Journal of Sports Sciences 23 (5), 509–514, pMID: 16194998.
URL http://www.tandfonline.com/doi/abs/10.1080/02640410410001716779

Hvattum, L. M., Arntzen, H., 2010. Using elo ratings for match result prediction in association football. International Journal of Forecasting 26 (3), 460 – 470, sports Forecasting.
URL http://www.sciencedirect.com/science/article/pii/S0169207009001708

Jamieson, J. P., 2010. The home field advantage in athletics: A meta-analysis. Journal of Applied Social Psychology 40 (7), 1819–1848.
URL http://dx.doi.org/10.1111/j.1559-1816.2010.00641.x

Joseph, A., Fenton, N., Neil, M., 2006. Predicting football results using bayesian nets and other machine learning techniques. Knowledge-Based Systems 19 (7), 544 – 553, creative Systems.
URL http://www.sciencedirect.com/science/article/pii/S0950705106000724

Kelly, J., September 1956. A new interpretation of information rate. Information Theory, IRE Transactions on 2 (3), 185–189.

King, N., Owen, P. D., Audas, R., 2012. Playoff uncertainty, match uncertainty and attendance at australian national rugby league matches. Economic Record 88 (281), 262–277.
URL http://dx.doi.org/10.1111/j.1475-4932.2011.00778.x

Koning, R. H., 2000. Balance in competition in dutch soccer. Journal of the Royal Statistical Society: Series D (The Statistician) 49 (3), 419–431.
URL http://dx.doi.org/10.1111/1467-9884.00244

Koning, R. H., Koolhaas, M., Renes, G., Ridder, G., 2003. A simulation model for football championships. European Journal of Operational Research 148 (2), 268 – 276, sport and Computers.
URL http://www.sciencedirect.com/science/article/pii/S0377221702006835

Lago, C., Martín, R., 2007. Determinants of possession of the ball in soccer. Journal of Sports Sciences 25 (9), 969–974, pMID: 17497397.
URL http://www.tandfonline.com/doi/abs/10.1080/02640410600944626

Langseth, H., 2013. Beating the bookie: A look at statistical models for prediction of football matches. In: Twelfth Scandinavian Conference on Artificial Intelligence. IOS Press, pp. 165–174.

Linde, J. B., Løkketangen, M., December 2013. Predicting outcomes of association football matches based on individual players' performance, specialization project in Intelligent Systems at the Department of Computer and Information Science, Norwegian University of Science and Technology.

Maher, M. J., 1982. Modelling association football scores. Statistica Neerlandica 36 (3), 109–118.
URL http://dx.doi.org/10.1111/j.1467-9574.1982.tb00782.x

McHale, I., Morton, A., 2011. A bradley-terry type model for forecasting tennis match results. International Journal of Forecasting 27 (2), 619 – 630.
URL http://www.sciencedirect.com/science/article/pii/S0169207010001019

McHale, I., Scarf, P., 2005. Ranking football players. Significance 2 (2), 54–57.
URL http://dx.doi.org/10.1111/j.1740-9713.2005.00091.x

McHale, I. G., Scarf, P. A., Folker, D. E., jul 2012. On the development of a soccer player performance rating system for the english premier league. Interfaces 42 (4), 339–351.
URL http://dx.doi.org/10.1287/inte.1110.0589

McHale, I. G., Szczepański, u., 2013. A mixed effects model for identifying goal scoring ability of footballers. Journal of the Royal Statistical Society: Series A (Statistics in Society), n/a–n/a.
URL http://dx.doi.org/10.1111/rssa.12015

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., 1953. Equation of state calculations by fast computing machines. The journal of chemical physics 21 (6), 1087–1092.

Mitchell, T. M., 1997. Machine Learning, international Edition. The McGraw-Hill Companies, Inc.

NHibernate, 2014. http://nhforge.org/, downloaded: January 13, 2014.

Norris, J. R., 1997. Markov Chains, 1st Edition. Cambridge University Press.

NorskTipping.no, 2013. https://www.norsk-tipping.no/spill/langoddsen?NT.mc_id=nt-langoddsen, accessed: 19:36, December 4, 2013.

Pearl, J., 1988. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann.

Plummer, M., 2011. Jags version 3.1.0 user manual, accessed April 23, 2014.
URL http://faculty.washington.edu/jmiyamot/p548/plummerm%20jags_user_manual.pdf

Pollard, R., Reep, C., 1997. Measuring the effectiveness of playing strategies at soccer. Journal of the Royal Statistical Society: Series D (The Statistician) 46 (4), 541–550.
URL http://dx.doi.org/10.1111/1467-9884.00108

Reed, D. D., Critchfield, T. S., Martens, B. K., 2006. The generalized matching law in elite sport competition: Football play calling as operant choice. Journal of Applied Behavior Analysis 39 (3), 281–297.
URL http://dx.doi.org/10.1901/jaba.2006.146-05

Rue, H., Salvesen, O., 2000. Prediction and retrospective analysis of soccer matches in a league. Journal of the Royal Statistical Society: Series D (The Statistician) 49 (3), 399–418.
URL http://dx.doi.org/10.1111/1467-9884.00243

Russel, S. J., Norvig, P., 2010. Artificial Intelligence: A Modern Approach, 3rd Edition. Pearson Education, Inc.

Ryall, R., Bedford, A., 2010. An optimized ratings-based model for forecasting australian rules football. International Journal of Forecasting 26 (3), 511 – 517, sports Forecasting.
URL http://www.sciencedirect.com/science/article/pii/S016920701000004X

Santos, R. M., 2013. Optimal soccer strategies. Economic Inquiry, no–no.
URL http://dx.doi.org/10.1111/ecin.12020

Sáez Castillo, A., Rodríguez Avi, J., Pérez Sánchez, J. M., 2013. Expected number of goals depending on intrinsic and extrinsic factors of a football player. an application to professional spanish football league. European Journal of Sport Science 13 (2), 127–138.
URL http://www.tandfonline.com/doi/abs/10.1080/17461391.2011.589473

SharpJags, 2014. https://github.com/thrandre/SharpJags, downloaded: January 13, 2014.

Spann, M., Skiera, B., 2009. Sports forecasting: a comparison of the forecast accuracy of prediction markets, betting odds and tipsters. Journal of Forecasting 28 (1), 55–72.
URL http://dx.doi.org/10.1002/for.1091

Stilling, S. T., Critchfield, T. S., 2010. The matching relation and situation-specific bias modulation in professional football play selection. Journal of the Experimental Analysis of Behavior 93 (3), 435–454.
URL http://dx.doi.org/10.1901/jeab.2010.93-435

WampServer.com, 2013. http://www.wampserver.com/en/, accessed: September 25, 2013.

WhoScored.com, 2014. http://www.whoscored.com/, accessed: November 5, 2013 and May 16, 2014.

Appendix A

Structured Literature Review

This chapter contains the structured literature review conducted for this project. First, we have a rationale for such a literature review. Second, the research questions for the review is presented. Third, we describe how we choose which papers to study, and then conclude the chapter with a review of the papers and a summary.

A.1 Rationale

We have conducted a structured literature review in order to get an overview of the existing literature in this field. A good review can be used as a basis when developing our own model. Such a review results in both knowledge of the domain, an overview of previous conducted research, and would give us an understanding of what we should include (and not include) in our own model.

A.2 Research Questions

Before conducting a structured literature review, it was important to have a goal in mind. This goal could be stated by the use of research questions.

> **Research question 1a:** Which existing solutions are there for predicting outcomes of matches in sport based on team performance?
>
> **Research question 1b:** Which existing solutions are there for predicting outcomes of matches in sport based on individual players' performance?

Table A.1: Search terms

	Group 1	Group 2	Group 3	Group 4
Term 1	Football	Individual	Analysis	Contribution
Term 2	Soccer	Player	Assess	League
Term 3			Determine	Match
Term 4			Model	Performance
Term 5			Predict	Rating
Term 6				Result
Term 7				Scoring

It was important to get a thorough understanding of the previous work conducted in the field. Answering these research questions would give us the required overview of the area.

> **Research question 2:** How are the existing solutions compared to each other regarding the model used, the data analysed and the sport observed?

To be able to decide which method we should use, or to create a model based on one of the existing ones, we needed to compare all the existing models, and find the ones best suited for our task. By comparing the models, we could also investigate strengths and weaknesses with the different models, and then decide which of the elements in the model we wanted to include in our model.

Almost all of the models use different datasets. Comparing the datasets was essential to see if this was something we could reuse, and also to see if the model was tested on a large enough amount of data.

The observed sport was important in case we wanted to reuse the model. If the sport was something else than football, we most likely needed to rewrite the model.

A.3 Literature Search

The sources used for finding state-of-the-art literature related to prediction of football matches were The Wiley Online Library and Google Scholar. Searching these sources covers a lot of relevant publishing journals. The sources were searched using combinations of the strings in table Table A.1. One search was conducted for each combination containing one term from each of the groups.

Table A.2: List of inclusion/exclusion criteria

ID	Criteria Description
IC 1	The study is mainly concerned with how individual players contribute to a team
IC 2	The study is mainly concerned with prediction of the results of football matches
IC 3	If the study is concerned with another sport than football, it is possible to extend the model to include football
EC 1	The study is mainly concerned about medical issues
EC 2	The study is mainly concerned about live betting
EC 3	The study is mainly concerned about economical issues

A.4 Articles Chosen for Further Investigation

By searching as described in section A.3, we received several hundreds of results. Therefore, we needed some restrictions on which papers to include in the study. The inclusion and exclusion criteria are shown in Table A.2. Only papers that matched at least one of the inclusion criteria and none of the exclusion criteria were included.

A.4.1 Abstract screening

First, we conducted a screening of the abstracts of the articles returned by the searches. After the abstract screening, we had a list with the following articles:

- Ali (2011): Measuring soccer skill performance: a review
- Boulier and Stekler (2003): Predicting the outcomes of national football league games
- Carmichael et al. (2000): Team performance: the case of English premiership football
- Carron et al. (2002): Team cohesion and team success in sport
- Cattelan et al. (2013): Dynamic Bradley-Terry modelling of sports tournaments
- Clarke and Dyte (2000): Using official ratings to simulate major tennis tournaments

- Clarke and Norman (1995): Home ground advantage of individual clubs in English soccer
- Constantinou and Fenton (2013): Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries
- Dixon and Coles (1997): Modelling association football scores and inefficiencies in the football betting market
- Duch et al. (2010): Quantifying the performance of individual players in a team activity
- Dyte and Clarke (2000): A ratings based Poisson model for world cup soccer simulation
- Forrest et al. (2005): Odds-setters as forecasters: The case of English football
- Goddard (2005): Regression models for forecasting goals and match results in association football
- Goddard and Asimakopoulos (2004): Forecasting football results and the efficiency of fixed-odds betting
- Grehaigne et al. (1997): Dynamic-system analysis of opponent relationships in collective actions in soccer
- Grund (2012): Network structure and team performance: The case of English Premier League soccer teams
- Hughes and Franks (2005): Analysis of passing sequences, shots and goals in soccer
- Hvattum and Arntzen (2010): Using ELO ratings for match result prediction in association football
- Jamieson (2010): The home field advantage in athletics: A meta-analysis
- Joseph et al. (2006): Predicting football results using Bayesian nets and other machine learning techniques
- King et al. (2012): Play-off uncertainty, match uncertainty and attendance at Australian national rugby league matches
- Koning (2000): Balance in competition in Dutch soccer

- Koning et al. (2003): A simulation model for football championships
- Lago and Martín (2007): Determinants of possession of the ball in soccer
- Langseth (2013): Beating the bookie: A look at statistical models for prediction of football matches
- Maher (1982): Modelling association football scores
- McHale and Morton (2011): A Bradley-Terry type model for forecasting tennis match results
- McHale and Scarf (2005): Ranking football players
- McHale et al. (2012): On the development of a soccer player performance rating system for the English Premier League
- McHale and Szczepański (2013): A mixed effects model for identifying goal scoring ability of footballers
- Pollard and Reep (1997): Measuring the effectiveness of playing strategies at soccer
- Reed et al. (2006): The generalized matching law in elite sport competition: Football play calling as operant choice
- Rue and Salvesen (2000): Prediction and retrospective analysis of soccer matches in a league
- Ryall and Bedford (2010): An optimized ratings-based model for forecasting Australian rules football
- Santos (2013): Optimal soccer strategies
- Sáez Castillo et al. (2013): Expected number of goals depending on intrinsic and extrinsic factors of a football player. An application to professional Spanish football league
- Spann and Skiera (2009): Sports forecasting: a comparison of the forecast accuracy of prediction markets, betting odds and tipsters
- Stilling and Critchfield (2010): The matching relation and situation-specific bias modulation in professional football play selection

A.4.2 Full text screening

Full text screening was conducted on all the papers mentioned in the previous section. During the full text screening, 21 of the articles were removed from the selection. A small summary of why they were excluded, follows:

- **Ali (2011) Measuring soccer skill performance, a review:** The article looks at measuring soccer skill performance based on physical attributes, and violates EQ1.
- **Boulier and Stekler (2003) Predicting the outcomes of National Football League games:** The model presented is specific to NFL. Also, it is not able to beat the bookmakers. Since it does not work, and since we have to rewrite it in order to use it, it is not relevant for our study.
- **Carron et al. (2002) Team cohesion and team success in sport:** No model is provided in the article. The results from the study is based on a questionnaire among the team members, which is something that we can not do in our project.
- **Clarke and Dyte (2000) Using official ratings to simulate major tennis tournaments:** The model is based on the ratings of each tennis player. No similar rating is available for football players. In addition, the model looks at points scored. A tennis match consists of several games in multiple sets and the number of points are much higher than the few goals that is scored in a football match. It is therefore difficult to used the presented model on football.
- **Forrest et al. (2005) Odds-setters as forecasters: The case of English football:** The model used in this paper is the model presented in Goddard (2005). Since no new model is presented in this article, this is not relevant.
- **Grehaigne et al. (1997) Dynamic-system analysis of opponent relationships in collective actions in soccer:** An analysis is conducted based on the movement of the players, and the conclusion is that the most common behaviour is passing the ball. This analysis is based on data that is not available to us, and no model is presented either. As a result, we can not make use of the results obtained in this paper.
- **Grund (2012) Network structure and team performance: The case of English Premier League soccer teams:** This article creates a network structure between the players at the teams, describing how they interact with each other. By analysing the data, they conclude that a high passing rate

within a team leads to better team performance. No prediction models are presented, and the results are found by analysing previous games, where the score is given. To be able to do such an analysis, we need access to whom where passing to whom in each match, and we do not have that data avilable.

- **Hughes and Franks (2005) Analysis of passing sequences, shots and goals in soccer:** They analyse how passes leads to goals, and conclude that longer passing sequences leads to more shots, and that direct play is more efficient for goal scoring than possession play. This could have been relevant if we knew how long passing sequences a team had on average, but we do only know the number of passes conducted by each team (and each player) in the game. Also, no model is presented, so this article is not relevant for us.

- **Jamieson (2010) The Home Field Advantage in Athletics: A Meta-Analysis:** The author looks at a lot of different sports to see if there is a home ground advantage in those sports. He states that "the home-field advantage for soccer was significantly stronger than that of any other sport". This article shows that there exists a home-field advantage in soccer, so we should use this in our model. This article could be referenced, but it is not that relevant because it examines a lot of sports, not only soccer. If we could find an article that only looked at soccer, maybe that would be better for our project.

- **King et al. (2012) Playoff Uncertainty, Match Uncertainty and Attendance at Australian National Rugby League Matches:** The article investigates what the impact uncertainties of match outcomes has to do with the attendance on matches in the Australian Rugby League. However, we are not interested in how many spectators are present at each game, and therefore, this article is not relevant for us. Also, the prediction model is specific for Australian rugby, and must be changed to be used with football.

- **Koning (2000) Balance in Competition in Dutch Soccer:** This article looks at economic balance, and how this balance has changed since the 70s. This violates EC3 and is not relevant for us.

- **Koning et al. (2003) A simulation model for football championships:** A Poisson model is used in order to predict who wins a tournament, where input to the model is a weighted average of goals scored. For us, this is not a relevant model, since it is specialized for national teams.

- **McHale and Morton (2011) A Bradley-Terry type model for forecasting tennis match results:** The article is concerned with predicting the outcome

of tennis matches. The ranking model presented takes into account court surface, date and game outcome, and does not reward players for frequent participation in tournaments (as is the case with the official ranking model). Since the model presented in this article aims to weigh the strengths of two individuals in tennis, an individual sport, we find the article not very relevant to our project.

- **McHale and Szczepański (2013) A mixed effects model for identifying goal scoring ability of footballers:** This article presents a model for trying to predict how many shots on goal a player has per match, and how many of these shots that produce goals. The aim of this article is to determine how much of a player's ability that is reflected in goal-scoring, to be used for estimating values of players on the transfer market. Although it is interesting to see that the ability to create shots can be modelled based on previous observations, we do not think we will be using the model presented in this article for modelling player performances as it will make our project work load too large.

- **Pollard and Reep (1997) Measuring the effectiveness of playing strategies at soccer:** In this article, team possession is analysed. If a team gets possession of the ball, what is the probability of that play ending with a shot/goal? This probability depends on where on the pitch the possession starts, if it is a set play or not, and so on. This is interesting for finding facts like "long attacking throws are more efficient than short throws". This article is concerned with analysing the efficiency of different strategies and we do not think it will come in handy for our project.

- **Reed et al. (2006) The generalized matching law in elite sport competition: football play calling as operant choice:** This article is about American football specific details, and is not relevant for us.

- **Ryall and Bedford (2010) An optimized ratings-based model for forecasting Australian rules football:** This article is about Australian football, but the authors state in the conclusion that the presented models should be applicable to other high-scoring sports as American football and basketball. Since football is no high-scoring sport, we find this article not relevant for our project.

- **Sáez Castillo et al. (2013) Expected number of goals depending on intrinsic and extrinsic factors of a football player. An application to professional Spanish football league:** This article describes a Bayesian regression model for the number of goals scored by each player on a team. The

Table A.3: List of quality criteria

ID	Criteria Description
QC 1	There is a clear statement of the aim of the research
QC 2	The study is put into context of other studies and research
QC 3	Algorithmic design decisions are justified
QC 4	The test data set is reproducible
QC 5	The study algorithm is reproducible
QC 6	The experimental procedure is thoroughly explained and reproducible
QC 7	It is clearly stated which other algorithms the study's algorithm(s) has been compared with
QC 8	The performance metrics used in the study is explained and justified
QC 9	Test results are thoroughly analysed
QC 10	The test evidence supports the findings presented

model can then be used to get an estimation of a player's performance given the number of goals he has scored. It focuses on number of goals per season, not per game. However, it can not be used to predict the expected number of goals for a player in the next match. Therefore, we do not characterize this article as relevant for our project.

- **Santos (2013) Optimal soccer strategies:** It seems that the article only looks at how the teams are playing, if they are attacking or defending. By conducting a retrospective analysis, they see if the strategy was optimal or not. It seems to be no prediction involved, so this article is not relevant for us.

- **Spann and Skiera (2009) Sports forecasting: a comparison of the forecast accuracy of prediction markets, betting odds and tipsters:** This article compares the prediction accuracy of prediction markets, betting odds and tipsters. The paper is only concerned with betting, and not relevant at the moment.

- **Stilling and Critchfield (2010) The matching relation and situation-specific bias modulation in professional football play selection:** This article is about American football, not soccer. It contains a model specific to that domain, and it is not relevant for us.

A.4.3 Quality assessment

The remaining papers were evaluated with regard to the quality criteria presented in Table A.3. The paper got the score 1 if it fulfilled the criteria, 0.5 if it partial fulfilled it, and 0 if it were not fulfilled.

The scores are presented as Author (Year) Title : $\mathbf{Score} = (\text{Points from QC1} + \text{Points from QC2} + ... + \text{Points from QC10})$.

Papers with a score below 8 will be removed from the study.

- Carmichael et al. (2000) Team performance: the case of English Premiership football: $\mathbf{8} = (1 + 1 + 1 + 1 + 1 + 1 + 0 + 0.5 + 0.5 + 1)$
- Cattelan et al. (2013) Dynamic Bradley-terry modelling of sports tournaments: $\mathbf{9} = (1 + 1 + 1 + 1 + 1 + 1 + 0.5 + 0.5 + 1 + 1)$
- Clarke and Norman (1995) Home ground advantage of individual clubs in English soccer: $\mathbf{8.5} = (1 + 1 + 1 + 1 + 1 + 1 + 0 + 1 + 1 + 0.5)$
- Constantinou and Fenton (2013) Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries: $\mathbf{9.5} = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 0.5 + 1)$
- Dixon and Coles (1997) Modelling association football scores and inefficiencies in the football betting market: $\mathbf{10} = (1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1)$
- Duch et al. (2010) Quantifying the performance of individual players in team activity: $\mathbf{8} = (1 + 0.5 + 1 + 1 + 1 + 1 + 0 + 1 + 0.5 + 1)$
- Dyte and Clarke (2000) A rating based Poisson model for world cup soccer simulation: $\mathbf{7.5} = (1 + 0.5 + 1 + 1 + 1 + 1 + 0 + 1 + 0.5 + 0.5)$
- Goddard (2005) Regression models for forecasting goals and match results in association football: $\mathbf{8.5} = (1 + 1 + 1 + 1 + 1 + 1 + 0.5 + 0.5 + 0.5 + 1)$
- Goddard and Asimakopoulos (2004) Forecasting football results and the efficiency of fixed odds betting: $\mathbf{9} = (1+1+1+1+1+1+0.5+1+0.5+1)$
- Hvattum and Arntzen (2010) Using ELO ratings for match result prediction in association football: $\mathbf{9.5} = (1 + 1 + 1 + 1 + 1 + 1 + 0.5 + 1 + 1 + 1)$
- Joseph et al. (2006) Predicting football results using Bayesian nets and other machine learning techniques: $\mathbf{7} = (1+0+0.5+1+0.5+0.5+1+1+0.5+1)$

- Lago and Martín (2007) Determinants of possession of the ball in soccer: $\mathbf{8.5} = (1+1+1+1+1+0+0.5+1+1+1)$
- Langseth (2013) Beating the bookie: A look at statistical models for prediction of football matches: $\mathbf{10} = (1+1+1+1+1+1+1+1+1+1)$
- Maher (1982) Modelling association football scores: $\mathbf{9.5} = (1+1+1+1+1+1+0.5+1+1+1)$
- McHale and Scarf (2005) Ranking football players: $\mathbf{2} = (1+0+1+0+0+0+0+0+0+0)$
- McHale et al. (2012) On the development of a soccer player performance rating system for the English Premier League: $\mathbf{8.5} = (1+1+1+1+0.5+0.5+0.5+1+1+1)$
- Rue and Salvesen (2000) Prediction and retrospective analysis of soccer matches in a league: $\mathbf{9.5} = (1+1+1+1+1+1+0.5+1+1+1)$

The articles removed were Dyte and Clarke (2000), Joseph et al. (2006) and McHale and Scarf (2005).

A.5 State of the Art Assessment

A.5.1 Different models for predicting outcomes of football games

The Poisson model

One of the pioneers in the field of predicting outcomes of football matches was Maher. He laid the foundation by creating a Poisson model based on the goals scored by the teams in a match (Maher, 1982). The model proposed consists of an attack and a defence strength for each of the teams. Also, home ground advantage was incorporated into the model as a static variable. Each of the teams will receive a team strength dependent on earlier played matches. At first, he states that the number of goals scored by each team are independent of each other. This constraint is too simple to be realistic, which he addresses later in the paper. He then creates a bivariate Poisson model, looking at the differences in goals scored by each of the teams, with the result that the teams are no longer independent of the opponents. The model is still static, meaning that a team's attack and defence strengths are constant.

Maher's model is extended by Dixon and Coles (1997). They want to create a more dynamic model, stating that recent results are more suitable to describe a team's current form than results way back in time. This is incorporated into

the model by adding a time variable, exponentially down weighting results further back in time. In addition, home advantage and attack and defence strengths are parts of the model.

In contrast to Maher (1982), Dixon and Coles do not use a bivariate Poisson distribution, where the goals scored by one team in a match are dependent on the goals scored by the opponent in the same match. To address this problem, they extend their basic model, keeping the marginally distribution of goals scored by each of the teams Poisson distributed.

A Poisson distribution of the goals scored by each team is also the basis for Rue and Salvesen (2000). In their paper, Rue and Salvesen extend the Poisson distribution to include a time factor, a psychological effect between the teams and the average goal intensity. The time factor is used to give higher weight to more recent results, as Dixon and Coles (1997) also did. The psychological effect constant is used when a high rated team plays against a low rated team. This constant tries to represent the fact that the strong team might underestimate the weaker team. The average goals scored is included because they found that the results from previous matches did not bear enough meaning to yield accurate predictions of next results. The whole equation is modelled in a Bayesian network, and Markov chain Monte Carlo is used to draw inference (be able to predict results) from the network.

As opposed to Maher (1982) and Dixon and Coles (1997), Rue and Salvesen (2000) do not look at the home advantage. Instead, they use the mentioned psychological effect. Clarke and Norman (1995) investigated home advantage, and found this to be important in football. Rue and Salvesen (2000) propose the inclusion of home advantage as an extension of the model. It is therefore a bit strange that they do not include this factor as well as the psychological effect.

The Bradley-Terry model

Football results can also be predicted using a Bradley-Terry model, as done by Cattelan et al. (2013). The authors develop such a model, and use it to predict the outcomes of NBA (basket) and Serie A (football) matches. The outcome from the model is either win, draw or loss for football, meaning that no numbers of scored goals are predicted. This paper presents both a static and a dynamic model. In the static model, each match is simulated and a table with the predicted final standings is presented. The included variables are home advantage, and the strength for each team, which is found by an exponentially weighted moving average.

The dynamic model presented assumes that a team has two separate abilities, its home ability and its away ability. In this model, time is included, weighting recent home matches more than home matches long time ago for the home ability, and similar for the away ability. This makes it possible to give each team a team-

specific home advantage.

As the previous presented models, this model uses only information about results of earlier played games. The authors suggest that a model using more of the post-match information available, would produce better predictions. They state that it is easy to extend their model to use vectors of relevant parameters instead of single-valued ones.

ELO rating

Hvattum and Arntzen (2010) investigate two types of ELO rating, and compare these to six specified prediction models. ELO rating is a measure of a team's current strength, and is updated after each match. After a match, each of the two participating teams get a score, 1 if they won, 0.5 if the match ended with a draw, and 0 if they lost. The sum of the scores from a match will then always be 1. The received score for each of the teams is then used to update their ELO rating. This is the basic ELO model, named ELO_b.

The formula for the ELO model contains a variable k. This k is a constant in the basic model. The extended ELO model is a model based on the goal difference between the two teams. The intention of this model is to incorporate the fact that a win with more goals often is a result of greater difference between the two teams than a win with fewer goals (e.g a 3-0 win indicates a greater dominance in the game than a 2-1 win). This goal difference is expressed by making the variable k be a function of the goal difference. This goal model is called ELO_g.

The two ELO ratings were compared to these six measures:

- UNI: Ignores all available information and predicts a uniform distribution over the results ($\frac{1}{3}$ home win, $\frac{1}{3}$ draw and $\frac{1}{3}$ away win).
- FRQ: Uses the observed frequencies of the different outcomes for the given data.
- GOD_b: This is a model based on model 4 in Goddard (2005).
- GOD_g: This is a model based on model 3 in Goddard (2005).
- AVG: Calculated from the normalized inverse of the average of the odds provided for the match.
- MAX: As AVG, but this model uses the normalized inverse of the best odds provided.

Hvattum and Arntzen (2010) concludes that their model were not able to gain profit, but it performed better than the six benchmark models it was compared to.

We found it difficult to see how this model performed compared to the previous mentioned models in this sections, since no comparison is available. The only conclusion we can see is that none of the models are able to gain certain income over time.

Pi-rating

The pi-rating system is presented in the paper by Constantinou and Fenton (2013). The model is based on three assumptions. These assumptions are that the model should include home advantage, that recent results are more important than old results while predicting, and that it is more important for a team to win than to increase the goal difference in a match.

Each match is handled in a cycle of six steps, where the two first steps are conducted before the match, and the four last are conducted after the match:

1. An expected goal difference is calculated for each team.
2. The expected goal difference is calculated for the match, by subtracting the home team's expected goal difference from the away team's expected goal difference.
3. The observed goal difference is calculated.
4. The prediction error is found.
5. The error is weighted.
6. A new home and away rating is calculated for each of the teams.

The paper does not describe how well the model performed on match predictions directly, but the predictions are used for betting. By accounting for a built in margin in the bookmaker's odds of 5 %, the system is able to gain profit over five Premier League seasons. This is the only one of the presented models that are able to earn money on betting.

An ordered probit regression model

Goddard and Asimakopoulos (2004) create an ordered probit regression model trying to predict outcomes of Premier League football matches. The model is based on parameters like results from previous matches, the significance of the match for the end season standings, the travel distance and the involvement of the team in games outside the league (e.g., in the FA cup).

The team quality is found by looking at the team's performance for the last 12 months. The home and away qualities of a team are based on its recent achievements at home and away, respectively. This gives each team a home and away quality specific for them, which sounds reasonable, since there are individual differences amongst the teams for how well they perform at home and away.

The significance of a match is calculated by assuming that each of the other teams receive 1 point in their matches, and then seeing if the current team will win the league, or be promoted or avoid relegation if they win the match. This variable seems wise to include, especially at the end of the season, while teams play to win the league, avoid relegation, or come in fourth place or better, to be able to qualify for the Champions League. One can however question if this variable should be included at the start of the season, since these measures are not so clear at an early stage.

Team confidence is incorporated into the variable representing the team's involvement in other tournaments. The authors saw that early elimination from the FA cup would result in a decrease in the confidence amongst the players of a team.

The model is tested on the betting market for games in Premier League for the 98/99 and 99/00 season. The model is able to outperform the bookmakers in 98/99, but not in 99/00. In both of the seasons, the model performs better than the bookmakers in the spring. This can be a result of the variable representing the significance of the match. It would have been interesting to see the model tested on several seasons, to see if it always performs good at the end of a season. Also, since the result was loss of money in one season, and gain in the other, it would have been nice to see if this was just luck (or bad luck), to indeed be able to say if this model is able to beet the bookmakers or not.

Using the models for betting

Some of the authors tested their models on the betting market, with various luck. If a model should be able to gain income, it must be better than the bookmakers to predict outcomes. A bookmaker will add a margin to his prediction, resulting in the percentages representing the prediction of the outcome being higher than 1 (as described in section 2.7). This makes it difficult to earn money, even if the prediction system is pretty good.

Langseth (2013) evaluates the models proposed by Maher (1982), Dixon and Coles (1997), and Rue and Salvesen (2000). He compares Maher's model to a dynamic version of the same model inspired by Rue and Salvesen (2000), and a data intensive model with regard to how they perform trying to beat the bookmakers in the 11/12 and 12/13 season of the Premier League. The three models are called STATIC, DYNAMIC and DATA INTENSIVE, respectively. The performance of

the prediction models are evaluated by the use of five betting strategies (one of the strategies used three different sub-strategies, conservative, intermediate and aggressive, resulting in a total of 7 betting strategies investigated).

For the 11/12 season, almost all the combinations of models and betting approaches are able to gain profit. In the following discussion, this is mostly due to two unlikely results, Liverpool loosing at home to both Wigan and West Bromwich. These results had high odds, yielding a large return on the placed bets. Without these matches, a loss would have been suffered from the betting. During the next season, only one of the combinations of models and bets is able to gain a positive result, and then only an income of 0.4%.

The results show that it is difficult to beat the bookmakers. Several models for betting were investigated, without anyone being able to beat the bookmakers in a consistent manner. Regarding the different prediction models, they all performed reasonably well in the 11/12 season without any large differences. In the 12/13 season, however, the STATIC and DYNAMIC lost more than the DATA INTENSIVE. This could be a clue that more data should be included in order to get better predictions for the outcomes of the matches, which supports the thoughts we have about creating a model based on the involved players.

A.5.2 Important variables to consider while developing a prediction system

Home ground advantage

Clarke and Norman (1995) examined the home ground advantage of all teams in the English Football League between the 1981/82 and 1990/91 seasons. The home ground advantage was calculated based on how well a team played at home opposed to playing away. For the ten seasons studied, home teams won 48.7% of the time and 26.7% resulted in a draw. Also, 59.9% of all goals scored were made by home teams. The home advantage was found to be the same for all divisions in the English Football League. However, differences between seasons were shown to be highly significant. There were also some indications of club-effect on home ground advantage, although not strong enough to make any conclusions.

Lago and Martín (2007) quantified the advantage of playing at home in terms of possession. Four variables were included in this analysis; match status (whether a team is winning or losing), venue (home or away) and the identities of the two teams involved. It was found that a team has 4-6% more possession of the ball when playing at home than when playing away, which supports that home advantage is a fact.

Maher (1982) used a constant factor for home ground advantage and assumed that this was the same for all teams.

After reading the articles mentioned in this section we believe we have evidence that supports incorporating home ground advantage in our model.

Rating individual players

McHale et al. (2012) describe how individual players are rated by the EA Sports Player Index. Some of the difficulties faced when rating football players are discussed. For instance, it is hard to compare a defender with an attacker because their performances should be measured by completely different factors (scoring goals versus preventing goals). Also, a player may start a match as a defender and end as an attacker, which makes rating the player even more complex. The EA Sports Player Index is composed of six sub-indices, each awarding different kinds of match contributions.

Sub-index 1 considers match outcome and player contributions. Each player is awarded points for each contribution he makes for his team, like passes, tackles, dribbles, and crosses. Each contribution has an impact on how many shots a team has in a game (and shots produce goals). The value of each contribution within a team is based on how many points the team benefits from each contribution (if a team scores 0 goals, no players will be awarded any points). Sub-index 2 takes into account the amount of time each player has spent on the pitch combined with the points earned for his team. Sub-index 3 considers the time played by a player compared to all players on his team, multiplied by the average amount of points earned by all teams in the league (1.34). This means players are rewarded for being on the pitch regardless of the contributions they make. Sub-index 4 awards points to scoring players, while sub index 5 rewards assists. Sub-index 6 rewards clean sheets. Defensive players are rewarded more than attacking players by this sub-index. Finally, the total points a player receives is a weighted sum of all sub-indices.

A table of the 20 top rated players of the Premier League 2008-2009 season is presented, showing a mix of players from all positions and different teams.

This article serves as an example of determining how well players perform, and it is interesting for us to see what kind of variables we should consider for rewarding a player or team.

Comparing goal based and result based models

Goddard (2005) investigates the differences in predictions when using goal based and result based models. This is done by creating four models, where model 1 and 2 are bivariate Poisson models. Model 1 is based on number of goals scored and model 2 is based on the three outcomes win, draw or loose. Model 3 and 4

are ordered probit regression models, where model 3 is goal based and model 4 is result based. The models are all tested on ten Premier League seasons.

As a result, he concludes that none of the models outperforms the others in a consistent way, and the difference is rather small. He can therefore not state which of the models that are best for predicting outcomes. Since both the approaches yield equal results, we can choose the approach that is best fitted for our model.

Accessibility of data

The team strengths calculated by the models proposed by Maher (1982), Dixon and Coles (1997), Rue and Salvesen (2000), Goddard and Asimakopoulos (2004), Hvattum and Arntzen (2010), and Constantinou and Fenton (2013) do all use the goals scored in previous matches as their main and most dominant factor.

At the time some of these models were presented, it was difficult to gather other information from the matches than the results. Now, the Internet makes it possible to gather enormous amounts of data in a relative easy way. Sites like WhoScored.com (2014) presents extensible data on almost every aspect of a game. It contains numbers of passes, shots, saves, tackles, probable line-ups (before the match is played) and many other facts from each match. It also provides statistics and ratings for each of the players. Dixon and Coles (1997) assume that it would have been possible to perform better predictions with models based on a richer amount of data. Also Rue and Salvesen (2000) mention this as a point to improve, claiming that ”It is of major importance to include more data than just the final match result in the model, but this depends on what kind of data is (easily) available and useful.” (Rue and Salvesen, 2000).

Given the huge amount of available data at the moment, we should strive to create a model based on more than just results of previous matches. Including the presence of the different individual players in a team, such a model could take advantage of more of the provided data.

A.5.3 Models based on individual players

The Opta Index

Carmichael et al. (2000) used the Opta Index of player performance statistics in order to create a function based on match performance. The function should take into account match specific data, like passes and shots. The created model is composed of a weighted variable for the observed team playing at home, differences between N ”play” and ”team characteristic” variables, who the opponent is and an error term.

A discussion is made on which variables that have positive or negative impact on a team's performance. Interesting elements from this discussion are that some variables, like number of blocked shots, can both be positive for the attacking team (a goal opportunity), and for the defending team (a good defence). Also, a lot of defensive statistics might mean that the opponent is attacking all the time. With this discussion, all the observed variables are assigned a +, a - or a +- to indicate if they are positive, negative or both for the observed team. The model is tested on the 97-98 season of the English Premier League, to see if the predicted signs (+, -, +-) were correct.

Shooting, passing, tackles, clearances and blocks are found to be important variables for calculating a team's strength. Red cards contribute negatively to the match outcome. It would have been interesting to see if using these variables with data from individual players in the starting line-up, would make this model perform even better.

Flow network

Duch et al. (2010) created a model based on individual players' impact on the game. They set up a "flow network", where each player is a node, and the edges between the players have weights according to how many completed passes that were conducted from one player to the other. In addition to nodes for each player, two nodes were added representing shots on goal and shots wide. The weights of the edges between player nodes and shot nodes represent how many shots a player makes during a game. Ratings for individual players are calculated by looking at the probability of a player being involved in "a flow of passes" that ends with a shot. Defensive players are also rewarded for intercepting passes made by the opposition.

The overall team ratings are represented by the average rating of all individual players. This approach was shown to yield pretty good results in identifying good players and teams. However, it requires detailed statistical data on passes; who made the pass, who received the pass, and where on the pitch the pass was made. The data used in this article is from the 2008 European Cup, which was special in terms of the amount of statistics data that were made officially available. Unfortunately we do not have access to such data on the English Premier League, and this approach is therefore not reproducible for our project.

Bradley-Terry model

In subsection A.5.1, the dynamic Bradley-Terry model by Cattelan et al. (2013) was discussed. The authors suggest that this model possibly would perform better

if more detailed data is included. They also state that the model is easily extendible for this purpose. To us, this seems like a promising approach for our project.

A.6 Summary

We have conducted a literature review in a structured and systematic way. The review answers the research questions stated in section A.2. The existing models for predicting outcome of football games have been investigated and compared to each other. We have seen that there exists some promising models based on individual players. Also, we found that more data should be incorporated into the models in order to get better results. Money management is an important field when betting on outcomes, and should be further investigated as a prediction model is created.

Appendix B

Implementation of the Prediction Models

This chapter contains the .jags implementations of the different prediction models used in this thesis.

B.1 Rue and Salvesen

This section contains the .jags implementation of the simplification of the model presented by Rue and Salvesen (2000). The simplification was created by Andresen and Dubicki (2013). We have conducted some small modifications, like variable renaming and changing of some of the indexes, without changing the logic of the program.

```
### RueSalvesen by Andresen and Dubicki (2013)
################################################################################

data {
  # Set C^(x) and C^(y) as constants
  # Values reported by Rue & Salvesen
    homeGoalAvg <- 0.395
    awayGoalAvg <- 0.098
}

model {
  ### Time model
  ################################################################################

  tau ~ dgamma(10, 1)
  precision ~ dgamma(10, 0.1)

  # Loop through all teams and all rounds
```

```
  for(t in 1:noTeams) {
    # Initial distribution of attack/defence strength
    attack[t, 1]  ~ dnorm(0, precision)
    defence[t, 1]   ~ dnorm(0, precision)

    # Evolve the attack/defence strength over time with
    # Brownian motion (Wiener process).
    ## The parameters are normally distributed with mean equal to
    ## the parameter value of the previous round. Loss-of-memory
    ## effect provided by variance parameter.
    for(s in 2:noRounds) {

      attack[t, s]  ~ dnorm(
                    attack[t, (s-1)],
                    (
                      ((abs( days[s,t] - days[s-1,t] )) / tau) * precision
                    )
                  )

      defence[t, s]   ~ dnorm(
                    defence[t, (s-1)],
                    (
                      ((abs( days[s,t] - days[s-1,t] )) / tau) * precision
                    )
                  )
    }
  }

  ### Goal model
  ##############################################################################

  # Parameters:
  ## attack[t, s] = attack strength for team t at given round s
  ## defence[t, s] = defence strength for team t at given round s
  ## team[1, i] = home team in game i
  ## team[2, i] = away team in game i
  ## round[1, i] = games played so far of home team in game i
  ## round[2, i] = games played so far of away team in game i
  ## goalsScored[1, i] = goals scored by home team in game i
  ## goalsScored[2, i] = goals scored by away team in game i

  # Give the delta-parameter some room to move
  gamma ~ dunif(0, 0.1)

  # Loop through all the games in correct order
  for(i in 1:noMatches) {

    # delta param for psychological effect of underestimating
    delta[i]      <-  (
                  attack[team[1, i], round[1, i]] +
                  defence[team[1, i], round[1, i]] -
                  attack[team[2, i], round[2, i]] -
                  defence[team[2, i], round[2, i]]
              ) / 2

    log(homeLambda[i])  <-  (
                  homeGoalAvg +
```

```
                    (
                      attack[team[1, i], round[1, i]] -
                      defence[team[2, i], round[2, i]] -
                      gamma * delta[i]
                    )
                  )

    log(awayLambda[i])  <-  (
                    awayGoalAvg +
                    (
                      attack[team[2, i], round[2, i]] -
                      defence[team[1, i], round[1, i]] +
                      gamma * delta[i]
                    )
                  )

    goalsScored[1, i] ~ dpois( homeLambda[i] )
    goalsScored[2, i] ~ dpois( awayLambda[i] )

  }
}
```

B.2 Goal Scaled and Attack Scaled

The Goal Scaled and Attack Scaled model are based on the simplification of the Rue and Salvesen model presented in section B.1. The two simulations use the same .jags script because both of the models scale the attack value of a team. The only difference is the scaling factor sent in to the script.

```

###Goal Scaled RueSalvesen / Attack Scaled RueSalvesen
#############################################################################

data {
  # Set C^(x) and C^(y) as constants
  # Values reported by Rue & Salvesen
    homeGoalAvg <- 0.395
    awayGoalAvg <- 0.098
}

model {
  ### Time model
  ###########################################################################

  tau ~ dgamma(10, 1)
  precision ~ dgamma(10, 0.1)

  # Loop through all teams and all rounds
  for(t in 1:noTeams) {
    # Initial distribution of attack/defence strength
    unscaledattack[t, 1]  ~ dnorm(0, precision)
    defence[t, 1]   ~ dnorm(0, precision)
    attack[t,1] <- unscaledattack[t, 1]
```

```
25
26      # Evolve the attack/defence strength over time with
27      # Brownian motion (Wiener process).
28      ## The parameters are normally distributed with mean equal to
29      ## the parameter value of the previous round. Loss-of-memory
30      ## effect provided by variance parameter.
31      for(s in 2:noRounds) {
32
33        unscaledattack[t, s]  ~ dnorm(
34                      unscaledattack[t, (s-1)],
35                      (
36                        ((abs( days[s,t] - days[s-1,t] )) / tau) * precision
37                      )
38                    )
39
40        defence[t, s]   ~ dnorm(
41                      defence[t, (s-1)],
42                      (
43                        ((abs( days[s,t] - days[s-1,t] )) / tau) * precision
44                      )
45                    )
46              attack[t, s] <- unscaledattack[t, s] + log( scale[s, t] )
47      }
48    }
49
50    ### Goal model
51    ############################################################################
52
53    # Parameters:
54    ## attack[t, s] = attack strength for team t at given round s
55    ## defence[t, s] = defence strength for team t at given round s
56    ## team[1, i] = home team in game i
57    ## team[2, i] = away team in game i
58    ## round[1, i] = games played so far of home team in game i
59    ## round[2, i] = games played so far of away team in game i
60    ## goalsScored[1, i] = goals scored by home team in game i
61    ## goalsScored[2, i] = goals scored by away team in game i
62
63    # Give the delta-parameter some room to move
64    gamma ~ dunif(0, 0.1)
65
66    # Loop through all games in correct order
67    for(i in 1:noMatches) {
68
69      # delta param for psychological effect of underestimating
70      delta[i]      <-  (
71                    attack[team[1, i], round[1, i]] +
72                    defence[team[1, i], round[1, i]] -
73                    attack[team[2, i], round[2, i]] -
74                    defence[team[2, i], round[2, i]]
75                ) / 2
76
77      log(homeLambda[i])  <-  (
78                    homeGoalAvg +
79                    (
80                      attack[team[1, i], round[1, i]] -
81                      defence[team[2, i], round[2, i]] -
```

```
                    gamma * delta[i]
                  )
                )

    log(awayLambda[i])  <-  (
                  awayGoalAvg +
                  (
                    attack[team[2, i], round[2, i]] -
                    defence[team[1, i], round[1, i]] +
                    gamma * delta[i]
                  )
                )

    goalsScored[1, i] ~ dpois( homeLambda[i] )
    goalsScored[2, i] ~ dpois( awayLambda[i] )

  }
}
```

B.3 Scaled Attack and Defence

The Scaled Attack and Defence model is an extension of the Goal Scaled / Scaled Attack model presented in section B.2.

```
### RueSalvesenAttDefScale
################################################################################

data {
  # Set C^(x) and C^(y) as constants
  # Values reported by Rue & Salvesen
    homeGoalAvg <- 0.395
    awayGoalAvg <- 0.098
}

model {
  ### Time model
  ##############################################################################

  tau ~ dgamma(10, 1)
  precision ~ dgamma(10, 0.1)

  # Loop through all teams and all rounds
  for(t in 1:noTeams) {
    # Initial distribution of attack/defence strength
    unscaledattack[t, 1]  ~ dnorm(0, precision)
    unscaleddefence[t, 1]   ~ dnorm(0, precision)
    attack[t, 1] <- unscaledattack[t, 1]
    defence[t, 1] <- unscaleddefence[t, 1]

    # Evolve the attack/defence strength over time with
    # Brownian motion (Wiener process).
    ## The parameters are normally distributed with mean equal to
    ## the parameter value of the previous round. Loss-of-memory
    ## effect provided by variance parameter.
```

```
    for(s in 2:noRounds) {

      unscaledattack[t, s]  ~ dnorm(
                    unscaledattack[t, (s-1)],
                    (
                      ((abs( days[s,t] - days[s-1,t] )) / tau) * precision
                    )
                  )

      unscaleddefence[t, s]   ~ dnorm(
                    unscaleddefence[t, (s-1)],
                    (
                      ((abs( days[s,t] - days[s-1,t] )) / tau) * precision
                    )
                  )
            attack[t, s] <- unscaledattack[t, s] + log( ascale[s, t] )
            defence[t, s] <- unscaleddefence[t, s] + log( dscale[s, t] )
    }
  }

  ### Goal model
  ######################################################################

  # Params:
  # Parameters:
  ## attack[t, s] = attack strength for team t at given round s
  ## defence[t, s] = defence strength for team t at given round s
  ## team[1, i] = home team in game i
  ## team[2, i] = away team in game i
  ## round[1, i] = games played so far of home team in game i
  ## round[2, i] = games played so far of away team in game i
  ## goalsScored[1, i] = goals scored by home team in game i
  ## goalsScored[2, i] = goals scored by away team in game i

  # Give the delta-parameter some room to move
  gamma ~ dunif(0, 0.1)

  # Loop through all games in correct order
  for(i in 1:noMatches) {

    # delta param for psychological effect of underestimating
    delta[i]       <-  (
                  attack[team[1, i], round[1, i]] +
                  defence[team[1, i], round[1, i]] -
                  attack[team[2, i], round[2, i]] -
                  defence[team[2, i], round[2, i]]
              ) / 2

    log(homeLambda[i])  <-  (
                  homeGoalAvg +
                  (
                    attack[team[1, i], round[1, i]] -
                    defence[team[2, i], round[2, i]] -
                    gamma * delta[i]
                  )
                )

```

```
88      log(awayLambda[i])  <-  (
89                      awayGoalAvg +
90                      (
91                        attack[team[2, i], round[2, i]] -
92                        defence[team[1, i], round[1, i]] +
93                        gamma * delta[i]
94                      )
95                    )
96
97      goalsScored[1, i] ~ dpois( homeLambda[i] )
98      goalsScored[2, i] ~ dpois( awayLambda[i] )
99    }
100 }
```

Appendix C

Running of the Code

In addition to this report, we have delivered a .zip file containing the implementations done this spring. We have assured that the code is well documented with intuitive method and variable names, and added describing comments where necessary. This chapter describes how to run the delivered crawler and prediction system.

C.1 Internet Crawler

The delivered crawler is an extension of the crawler developed by Martin Belgau Ellefsrød for his thesis, (Ellefsrød, 2013).

The approach to use the crawler is as follows:

- Install WampServer from `http://www.wampserver.com/en/`.
- Place the delivered crawler folder in the folder `C:/wamp/www/`.
- Start the installed WampServer. The server should be offline at all times, preventing others from accessing files on your computer.
- Go to `http://localhost80/internetcrawler/hp/premier-league-table.php`. If this is not possible, open the file `C:/wamp/bin/apache/apache2.4.4/conf/httpd.conf` and exchange `Listen 80` with a random 4-digit number. Then, change the 80 in the url with the number selected.
- In case you get a message that the server has timed out, click on the WampServer icon, select the folder PHP and open the file php.ini. Locate the variable `max_execution_time` and set its value to 120.

- The retrieved files should be stored in `/FootballPredictions/FootballPredictions/TextFiles/RawFiles/'season number'/RawFiles/`.

C.2 The Prediction System

The prediction system is developed in C#, and can be opened in Visual Studio by clicking the .project file. Here, the whole program is controlled from the file Constants.cs, and this file should be self explanatory. Before the program is built, the `CURRENT_USER` variable should be set to the path of the folder where the FootballPredictions folder is stored.

C.2.1 Databases

In total, four different databases are placed in the App_Data folder. The name of the one that should be used at the moment must be changed to Database.sdf. The contents of the different databases are (with simulation number in parenthesis):

- 5476 is the EPL 11/12 season.
  - GoalScaled(18)
  - AttackAndDefenceScaled(22)
  - RueSalvesen(26)
  - AttackScaled(29)
- 6531 is the EPL 12/13 season.
  - RueSalvesen(12)
  - GoalScaled(13)
  - AttackAndDefenceScaled(17)
  - AttackScaled(21)
- 7794 is the EPL 13/14 season.
  - RueSalvesen(4, in the database 7794_1)
  - AttackScaled(8, in the database 7794_1)
  - GoalScaled(6, in the database 7794_2)
  - AttackAndDefenceScaled(8, in the database 7794_2)

The results from the simulations are stored in the database, while the results from applying different betting models are stored in the folder TextFiles as .tex files.

Appendix D

Regular Expressions

This appendix provides an overview of regular expressions and a description of how we have used regular expressions in the crawler.

D.1 Background Theory

A regular expression is used to find pattern(s) in a text, and consists of a set of rules. Some different patterns are shown in Table D.1.

Table D.1: An overview of the most common matching patterns in regular expressions

Symbol	Meaning
a	Matches the letter 'a'
.	Matches all characters, except line break
\n	Matches line break
*	Matches the previous character 0 or more times
?	Matches the previous character 0 or 1 time
()	All the letters inside the brackets are grouped together to a single element
\d	Matches a digit

D.2 Pattern Matching in the Crawler

A small part of the downloaded text is shown below. This text is a result of crawling `http://www.whoscored.com/Matches/614052/Live`, and displays line 1294 to 1357. The matching of regular expressions is done on this part of the text. 614052 is the unique id of this perticular game between Everton and Manchester United at 20th of August, 2012. Each line consists of an id, a name of a team or a player, the rating received and a number of statistics about the team's or player's involvement in the match.

```
1294 , [[31,'Everton',7.25,[['blocked_scoring_att',[2]],['
        att_miss_right',[1]],['post_scoring_att',[2]],['accurate_pass
        ',[196]],['att_miss_left',[1]],['total_tackle',[19]],[['
        att_post_high',[1]],['att_sv_high_right',[2]],['
        att_sv_low_centre',[2]],['won_contest',[7]],['
        att_goal_low_right',[1]],['att_sv_high_centre',[1]],['
        shot_off_target',[9]],['ontarget_scoring_att',[7]],['
        total_scoring_att',[18]],['aerial_lost',[18]],['
        att_sv_low_right',[1]],['fk_foul_lost',[13]],['total_throws
        ',[18]],['won_corners',[6]],['possession_percentage
        ',[30.8]],['aerial_won',[28]],['total_pass',[275]],['
        att_post_left',[1]],['att_miss_high',[1]],['att_miss_high_left
        ',[4]],['goals',[1]]]],[[5582,'Tim Howard',7.28,[[['
        accurate_pass',[6]],['touches',[42]],['saves',[3]],['
        total_pass',[16]],['good_high_claim',[2]],['formation_place
        ',[1]]]],1,'GK',24,0,0,'GK',34,187,88]
1295 ,[6105,'Phil Jagielka',8.53,[[['accurate_pass',[15]],['touches
        ',[41]],['clearance_off_line',[1]],['total_scoring_att
        ',[1]],['aerial_won',[5]],['total_pass',[19]],['total_tackle
        ',[2]],['aerial_lost',[1]],['formation_place',[5]]]],2,'DC
        ',6,0,0,'D(CR)',31,183,83]
1296 ,[8222,'Leighton Baines',8.19,[[['accurate_pass',[20]],['touches
        ',[57]],['won_contest',[2]],['total_scoring_att',[1]],['
        aerial_won',[3]],['total_pass',[29]],['total_tackle',[3]],['
        formation_place',[3]]]],2,'DL',3,0,0,'D(L)',29,170,74]
1297 ,[528,'Sylvain Distin',7.42,[[['accurate_pass',[15]],['touches
        ',[37]],['aerial_won',[4]],['total_pass',[20]],['total_tackle
        ',[1]],['aerial_lost',[2]],['formation_place',[6]]]],2,'DC
        ',15,0,0,'D(C)',36,193,87]
1298 ,[3030,'Tony Hibbert',7.43,[[['accurate_pass',[11]],['touches
        ',[43]],['total_pass',[17]],['total_tackle',[2]],['aerial_lost
        ',[1]],['fouls',[1]],['formation_place',[2]]]],2,'DR',2,0,0,'D
        (R)',32,175,71]
1299 ,[682,'Philip Neville',6.62,[[['accurate_pass',[21]],['touches
        ',[34]],['total_pass',[26]],['total_tackle',[2]],['aerial_lost
        ',[2]],['fouls',[1]],['formation_place',[4]]]],3,'MC',18,0,0,'
        D(R),DM(C)',36,180,69]
1300 ,[5047,'Leon Osman',7.19,[[['accurate_pass',[20]],['touches
```

```
    ',[40]],['won_contest',[1]],['total_scoring_att',[3]],['
    aerial_won',[1]],['total_pass',[23]],['post_scoring_att
    ',[1]],['total_tackle',[3]],['aerial_lost',[1]],['
    formation_place',[7]]]],3,'MR',21,1,81,'M(CLR)',32,173,67]
,[22738,'Marouane Fellaini',8.87,[[['blocked_scoring_att',[1]],['
    six_yard_block',[1]],['accurate_pass',[35]],['touches
    ',[66]],['won_contest',[2]],['total_scoring_att',[6]],['
    aerial_won',[9]],['total_pass',[48]],['goals',[1]],['
    post_scoring_att',[1]],['total_tackle',[1]],['aerial_lost
    ',[5]],['fouls',[6]],['formation_place',[10]],['
    man_of_the_match',[1]]]],3,'AMC',25,1,90,'M(C)',26,194,85]
,[3553,'Steven Pienaar',7.04,[[['accurate_pass',[20]],['touches
    ',[43]],['total_scoring_att',[4]],['aerial_won',[2]],['
    total_pass',[25]],['aerial_lost',[1]],['fouls',[1]],['
    formation_place',[11]]]],3,'ML',22,0,0,'AM(CLR)',31,173,69]
,[30395,'Darron Gibson',7.82,[[['blocked_scoring_att',[1]],['
    accurate_pass',[25]],['touches',[48]],['yellow_card',[1]],['
    won_contest',[2]],['total_scoring_att',[2]],['aerial_won
    ',[1]],['goal_assist',[1]],['total_pass',[33]],['total_tackle
    ',[2]],['aerial_lost',[1]],['fouls',[1]],['formation_place
    ',[8]]]],3,'MC',4,0,0,'M(C)',26,183,90]
,[19969,'Nikica Jelavic',6.81,[[['accurate_pass',[8]],['touches
    ',[26]],['total_scoring_att',[1]],['aerial_won',[3]],['
    total_pass',[18]],['total_tackle',[2]],['aerial_lost',[3]],['
    fouls',[3]],['formation_place',[9]]]],4,'FW',7,1,90,'FW
    ',28,187,84]
,[9314,'Steven Naismith',5.97,[[['touches',[1]],['total_pass
    ',[1]],['aerial_lost',[1]],['formation_place',[0]]]],5,'Sub
    ',14,2,90,'AM(LR)',27,178,72]
,[92547,'Ross Barkley',0,[[['formation_place',[0]]]],5,'Sub
    ',20,0,0,'AM(C)',20,189,76]
,[36491,'Jan Mucha',0,[[['formation_place',[0]]]],5,'Sub',1,0,0,'
    GK',31,189,87]
,[14114,'Victor Anichebe',0,[[['formation_place',[0]]]],5,'Sub
    ',28,0,0,'AM(R),FW',25,185,80]
,[6023,'Johnny Heitinga',6,[[['formation_place',[0]]]],5,'Sub
    ',5,2,90,'D(C),DM(C)',30,182,69]
,[31826,'Seamus Coleman',6.26,[[['touches',[7]],['total_tackle
    ',[1]],['formation_place',[0]]]],5,'Sub',23,2,81,'D(R),M(R)
    ',25,178,67]
,[69627,'Magaye Gueye',0,[[['formation_place',[0]]]],5,'Sub
    ',19,0,0,'AM(L)',23,179,73]
],['4411',[[5,1]
,[8,3]
,[2,3]
,[6,5]
,[6,3]
,[4,3]
,[8,5]
```

```
,[4,5]
,[5,9]
,[5,7]
,[2,5]
]]
]
,[32,'Manchester United',6.63,[['blocked_scoring_att',[4]],['
    att_miss_right',[1]],['accurate_pass',[571]],['att_miss_left
    ',[3]],['total_tackle',[15]],['total_offside',[1]],['
    att_sv_low_left',[1]],['att_sv_low_centre',[3]],['won_contest
    ',[7]],['shot_off_target',[6]],['ontarget_scoring_att',[4]],['
    total_scoring_att',[14]],['aerial_lost',[28]],['fk_foul_lost
    ',[11]],['total_throws',[23]],['won_corners',[8]],['
    possession_percentage',[69.2]],['aerial_won',[18]],['
    total_pass',[646]],['att_miss_high_left',[2]]]],[[79554,'David
     de Gea',7.55,[['accurate_pass',[8]],['touches',[37]],['saves
    ',[6]],['aerial_won',[1]],['total_pass',[15]],['
    formation_place',[1]]]],1,'GK',1,0,0,'GK',23,192,76]
,[9685,'Nemanja Vidic',7.06,[['accurate_pass',[48]],['touches
    ',[67]],['total_scoring_att',[1]],['aerial_won',[6]],['
    total_pass',[52]],['aerial_lost',[5]],['fouls',[3]],['
    formation_place',[6]]]],2,'DC',15,0,0,'D(C)',32,189,84]
,[6410,'Patrice Evra',6.66,[['blocked_scoring_att',[1]],['
    accurate_pass',[49]],['touches',[87]],['total_scoring_att
    ',[1]],['aerial_won',[1]],['total_pass',[58]],['total_tackle
    ',[1]],['aerial_lost',[1]],['formation_place',[3]]]],2,'DL
    ',3,0,0,'D(L)',32,175,76]
,[11,'Paul Scholes',7.04,[['accurate_pass',[92]],['touches
    ',[107]],['yellow_card',[1]],['won_contest',[1]],['
    total_scoring_att',[1]],['aerial_won',[1]],['total_pass
    ',[95]],['total_tackle',[3]],['fouls',[2]],['formation_place
    ',[8]]]],3,'DMC',22,0,0,'DM(C)',39,168,70]
,[18296,'Luis Antonio Valencia',6.79,[['accurate_pass',[50]],['
    touches',[95]],['aerial_won',[2]],['total_pass',[60]],['
    total_tackle',[3]],['aerial_lost',[1]],['fouls',[2]],['
    formation_place',[2]]]],3,'DR',7,0,0,'D(R),M(R)',28,181,78]
,[69956,'Tom Cleverley',7.28,[['blocked_scoring_att',[1]],['
    accurate_pass',[78]],['touches',[100]],['won_contest',[1]],['
    total_scoring_att',[2]],['aerial_won',[1]],['total_pass
    ',[85]],['total_tackle',[5]],['aerial_lost',[3]],['fouls
    ',[1]],['formation_place',[4]]]],3,'DMC',23,1,85,'M(CLR)
    ',24,175,67]
,[42862,'Shinji Kagawa',7.3,[['accurate_pass',[68]],['touches
    ',[85]],['won_contest',[2]],['total_pass',[75]],['total_tackle
    ',[1]],['aerial_lost',[2]],['formation_place',[10]]]],3,'AMC
    ',26,0,0,'AM(CL)',24,172,64]
,[2115,'Michael Carrick',6.68,[['accurate_pass',[61]],['touches
    ',[80]],['aerial_won',[2]],['total_pass',[67]],['total_tackle
    ',[1]],['aerial_lost',[5]],['formation_place',[5]]]],3,'DC
```

```
    ',16,0,0,'D(C),DM(C)',32,183,74]
,[19104,'Nani',5.8,[['accurate_pass',[32]],['touches',[54]],['
    yellow_card',[1]],['total_scoring_att',[1]],['total_pass
    ',[36]],['total_tackle',[1]],['aerial_lost',[3]],['fouls
    ',[3]],['formation_place',[7]]],3,'AMR',17,1,78,'AM(LR)
    ',27,175,66]
,[3859,'Wayne Rooney',6.54,[['accurate_pass',[40]],['touches
    ',[64]],['won_contest',[2]],['total_scoring_att',[4]],['
    aerial_won',[3]],['total_pass',[49]],['aerial_lost',[5]],['
    formation_place',[9]]],4,'FW',10,0,0,'AM(CL),FW',28,178,78]
,[39308,'Danny Welbeck',6.07,[['blocked_scoring_att',[2]],['
    accurate_pass',[20]],['touches',[31]],['won_contest',[1]],['
    total_scoring_att',[3]],['total_pass',[23]],['aerial_lost
    ',[2]],['formation_place',[11]]],4,'AML',19,1,68,'AM(CLR),FW
    ',23,185,73]
,[21723,'Anderson',6.06,[['accurate_pass',[9]],['touches
    ',[12]],['total_scoring_att',[1]],['total_pass',[10]],['
    formation_place',[0]]],5,'Sub',8,2,85,'DM(C)',25,176,69]
,[92914,'Scott Wootton',0,[['formation_place',[0]]],5,'Sub
    ',31,0,0,'D(C)',22,188,78]
,[4564,'Robin van Persie',5.98,[['accurate_pass',[4]],['touches
    ',[13]],['aerial_won',[1]],['total_pass',[7]],['aerial_lost
    ',[1]],['formation_place',[0]]],5,'Sub',20,2,68,'FW
    ',30,183,71]
,[8166,'Ashley Young',5.99,[['accurate_pass',[12]],['touches
    ',[18]],['total_pass',[14]],['formation_place',[0]]],5,'Sub
    ',18,2,78,'AM(CLR)',28,180,65]
,[2213,'Dimitar Berbatov',0,[['formation_place',[0]]],5,'Sub
    ',9,0,0,'AM(C),FW',32,188,79]
,[10620,'Anders Lindegaard',0,[['formation_place',[0]]],5,'Sub
    ',13,0,0,'GK',29,193,80]
,[4092,'Rafael',0,[['formation_place',[0]]],5,'Sub',21,0,0,'D(R)
    ',23,173,80]
],['4231',[[5,1]
,[2,3]
,[8,3]
,[6,5]
,[4,3]
,[6,3]
,[3,7]
,[4,5]
,[5,9]
,[5,7]
,[7,7]
]]
]
]
], 0];
```

First, we needed to extract the parts of the text that described the different teams, one part for the home team and one part for the away team. The notation is described in Table D.1. The text regarding the home team is extracted by this expression, where homeTeamId, homeTeamName, awayTeamId and awayTeamName are variables:

```
\n,  \[\[homeTeamId,homeTeamName,(.*\n)*,\[awayTeamId,awayTeamName,
```

The expression for the away team is:

```
\n,\[awayTeamId,awayTeamName,(.*\n)*.*\];
```

Second, the id, name and rating for each player should be extracted, as well as the id, name and rating for the team. The following regular expression runs in a loop, and is executed on both of the two teams:

```
\[\d*,'.*?',\d*(\.\d*)?,\[\[\[
```

The last step is to perform trimming on the strings to remove brackets, spaces, etc., and update the .csv files with the new ratings.

Appendix E

Attributes in the Database

E.1 Match Data

The tables E.1, E.2, E.3, E.4, E.5, E.6 and E.7 contain an overview of all the attributes we have chosen to store about players and teams in our database. The attributes for the teams are the sum of the attributes for all the players that participated in the match for the given team. These attributes are collected for each of the matches.

Table E.1: General information about the match.

Attribute	Description
Team	Team ID
Match	Match ID
Rating	Average rating
PossessionPercentage	Possession percentage

Table E.2: Information about the goals scored and conceded.

Attribute	Description
Goals	Goals scored
OwnGoals	Own goals scored by this team
OwnGoalAccrued	Own goals scored by opposing team
AttPenGoal	Goals scored on penalties

Table E.3: Information about the performance with the ball.

Attribute	Description
Touches	Total touches
UnsuccessfulTouch	Total unsuccessful touches
GoalAssist	Number of assists
TotalPass	Total passes
AccuratePass	Accurate passes
TotalThroughBall	Total through balls
AccurateThroughBall	Accurate through balls
TotalFinalThirdPasses	Total final third passes
SuccessfulFinalThirdPasses	Successful final third passes
TotalAttAssist	Total passes that lead to a shot
OffTargetAttAssist	Passes that lead to a shot off target
BigChanceCreated	Big chances created
TotalOffside	Total offsides
Dispossessed	Number of times team is dispossessed

Table E.4: Information about the shots.

Attribute	Description
TotalScoringAtt	Total scoring attempts
BlockedScoringAtt	Total blocked scoring attempts
OnTargetScoringAtt	On target scoring attempts
ShotOffTarget	Off target scoring attempts
PostScoringAtt	Shots hitting the post
BigChanceMissed	Big chances missed

Table E.5: Information about defending contribution.

Attribute	Description
TotalTackle	Total tackles
LastManTackle	Last man tackles
TotalClearance	Total clearances
EffectiveClearance	Effective clearances
SixYardBlock	Six yard blocks
ClearanceOffLine	Clearances off the line
ErrorLeadToGoal	Errors that lead to a goal
ErrorLeadToShot	Errors that lead to a shot
Interception	Total interceptions
InterceptionWon	Total interceptions won
BallRecovery	Total ball recoveries

Table E.6: Information about set pieces.

Attribute	Description
CornerTaken	Total corners
TotalThrows	Total throws
PenaltyConceded	Total penalties given to the opposition
FkFoulWon	Free kicks in favour of the team
FkFoulLost	Free kick in favour of the opposition
YellowCard	Total yellow cards
RedCard	Total red cards

Table E.7: Information about duelling.

Attribute	Description
AerialLost	Number of aerial duels lost
AerialWon	Number of aerial duels won
DuelLost	Total number of duels lost
DuelWon	Total number of duels won
WonContest	Number of successful dribbles
TotalContest	Number of dribble attempts
PenaltySave	Number of penalties saved