Mii Erik Samyo Haugård & Kim Long Vu

Computational models for live-prediction of football matches

Specialization Project, Autumn 2018

Artificial Intelligence Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

Abstract

Football betting has increased in popularity over the past years. With the increased computation power we have today and the emergence of machine learning, analyzing sports-event data is more available and common than it has previously been. The analyzed data can be used by the athletes and teams to gain advantages over their opponents, but also by ordinary persons that try to beat the bookies. In order for the bookies to stay in front, their models need to be better than the bettors.

In this project, we explore what has been done in the field of predicting the outcome of football matches, in order to attempt building a better model. We consider what has been done with regards to methods that have been used, but also what kind of inputs been fed into the models and their importance. Previous work has shown good results with the use of neural networks when trying to predict the outcome of a match, prior to its start. We also explore the effect different match events has shown to have on the outcome of a football match.

We will try to build a model that is able to predict the outcome of an ongoing football match, instead of trying to predict pre-game. This includes finding out who will win the rest of the match (e.g. the last 30 minutes). The main focus is to predict matches in the English Premier League, where historic data is provided by Sportradar.

Preface

This report was written by Mii Erik Samyo Haugård and Kim Long Vu as the Specialization Project (TDT4501) during the autumn of 2018. We are both students at the Norwegian University of Science and Technology (NTNU), in the Artificial Intelligence group at the Department of Computer science (IDI). This project will be a starting point for our master's thesis next semester, which will also be in the same domain. We would like to thank Helge Langseth for excellent guidance throughout the semester, and also Sportradar for making this project possible by providing data.

Mii Erik Samyo Haugård and Kim Long Vu
Trondheim, December 12, 2018

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this project we will have a look at what techniques and methods has been used when trying to predict the outcome of football matches. We will look at both approaches for predicting matches, pregame and while they play out. In addition we will try to find what kind of events has an effect on the outcome of a match, which can be used as features in our own model in the future.

1.1 Background and motivation

Analyzing sports-event data has over the last years been an important aspect in professional sports. NBA is one of the sports where analyzing data have made a great impact on the game. After the league started using a system that tracked every movement each player did on the court, the team coaches got a new arsenal of information that could be used to further improve their game. The most significant change was the increased amount of three-point shots taken. The analytics revealed that three-pointers on average led to more points than a two-point shot, even when the two-point shot had a higher accuracy. This lead to a near 50% increase of three-point shots taken per game [Kopf, 2017].

One of the world's most popular sports football, not to be mistaken with American football which is also called football by some, is also embracing data analytic. As one of the leading sports when it comes to popularity, huge amount of money is involved in the game. Mistakes could cost clubs lots of money, and it is not rare that team managers are fired due to poor results. New technology are used by the teams to monitor their own players, while opponents' match data are available from other parties. With the increased level of statistical data from the biggest football leagues available, some of these mistakes could be avoided with the right use of data [Evans, 2018].

The match data is not only used by the teams to analyze each other. Bookmakers is also a huge benefactor of the increased data availability in football. With more information on the teams, and good models for predicting the outcome of matches, the bookmakers can set more correct odds. Football tops the list over the most betted on sports worldwide [van Lier, 2018]. The possibility to place bets on a various number of leagues and all

the different types of in-game events, makes football unique in the betting industry. E.g. some bookmakers even provides the possibility to bet on Norwegian 8th division games [NordicBet, 2017]. With the betting industry being this huge in football, having more accurate odds could lead to an advantage for the bookmakers.

The increased data availability in combination with the artificial intelligence blooming in the recent years, the potential for even better models for predicting the outcome of matches are possible. AI is now playing a bigger role in helping bookmakers determining the odds, but there are also many that uses this technology to try to beat the bookies.

1.2 Sportradar

Sportradar, founded in 2000, is an international organization that provides analyzed sports data to bookmakers and other instances. As part of a thesis at Norwegian University of Science and Technology, Sportradar started of using crawler technology to monitor the online betting market. Among the services they provide today are odds comparison, both pregame and live, detailed live coverage of matches, and a system for detecting betting related match fixing [UEFA, 2018].

Sportradar is the provider of the data that is going to be used for this project. The data includes a huge amount of game statistics from a great number of leagues, including the biggest ones, where detailed information of game events are provided. They expressed a wish to find better models to predict how the rest of a match will play out in real time, as the live betting market represents a big portion of the total betting industry. Sportradar specified that it was especially which team was going to score the next goal, and who was going to win the rest of the match that were among the most popular types of bets.

1.3 Research goals and questions

The main goal of this project is to explore models that have been used to predict football matches. We will focus on models for pregame prediction that could be adapted to live predictions, and models that predict ongoing matches. As this is a project prior to a master thesis where we will try to predict the outcome of ongoing matches, we have the following research questions:

1. What models can be used to predict match outcome?
2. Which features are important for the outcome of a match?
3. How can event data be used to update match predictions in real-time?

1.4 Project structure

This project is divided into the following chapters:

Chapter 2: Background theory introduces the main theoretical aspects behind the models discussed in Chapter 3.

Chapter 3: State of the art gives an overview of what has been done in previous work when it comes to predicting football matches.

Chapter 4: Data presents the main data sources available.

Chapter 5: Future work tackles how the previous work, discussed in Chapter 3, along with the data available, undergone in Chapter 4, can be combined to make our own model to predict the outcome of ongoing matches.

Chapter 6: Summary gives a summary of what has discussed in this paper, especially in regards to the research questions.

Chapter 2

Background theory

This chapter covers the main theory that are relevant for this project. This includes some of the different techniques that are used by the models discussed in Chapter 3, which are both probability based approaches and machine learning techniques.

2.1 Poisson

A Poisson distribution is often applied when modelling the number of times an event occurs within a time interval. The Poisson distribution is applicable when the number of events can be counted in whole numbers, the occurrences of the events are independent of each other, the average frequency is known, and it is possible to count the number of times the event has occurred [Brooks, 2007]. For instance, a football team scores three goals in each match, on average. This becomes their expectation, but there will be some variations on how many goals are scored in each match. Sometimes there will be none, while there sometimes might be as many as five goals. Assuming that goals scored by the team are independent of each other, the Poisson distribution can be used to tell the how probable it is that the team will score 1, 3, or 6 goals during a match. The Poisson distribution is shown in Figure 2.1.

Given the average number of an event occurring, the probability for a number x is given by the equation

$$f(x) = \frac{\lambda^x e^\lambda}{x!}, \tag{2.1}$$

where λ is the average number of occurrences in the time interval and e is the base of the natural logarithm. With this, one could calculate the probabilities for the different number of events occurring, and generate a probability table for better readability. Table 2.1 shows the probability tables for the football example.

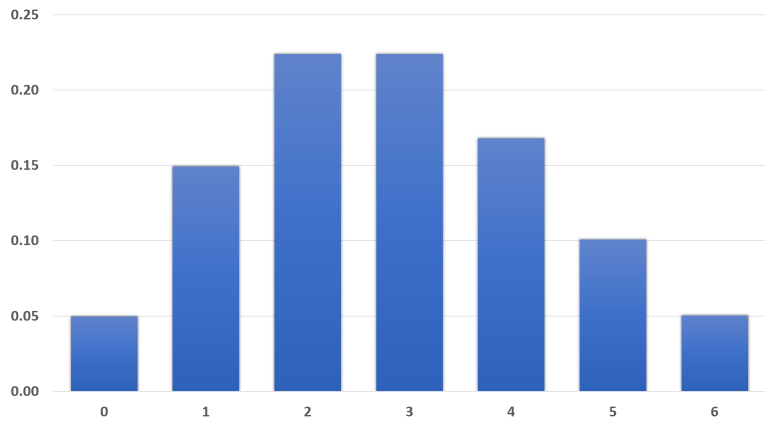


Figure 2.1: The chart shows the Poisson distribution for the football example, where the average number of goals by a team is three each match (meaning $\lambda = 3$).

Table 2.1: The Poisson distribution table for the football example, where the average number of goals is three each match (meaning $\lambda = 3$).

x	0	1	2	3	4	5	6
f(x)	0.04979	0.14936	0.22404	0.22404	0.16803	0.10082	0.05041

2.2 Bayesian networks

A Bayesian network (BN) is a probabilistic representation of a set of variables and their relationships. The variables in a BN are represented by nodes, where directed links connects them. The links displays the dependencies between nodes. If there is an arc from a node X to node Y, X is a parent of Y. As node X is the parent of node Y, it has an direct effect on its child. Given X is Y's only parent, Y's has a probability distribution $\mathbf{P}(Y|X)$. In general, each node X_i has a probability distribution $\mathbf{P}(X_i|Parents(X_i))$, where $Parents(X_i)$ is all the parents of X_i.

A simple example of a BN, taken from Russell and Norvig [2016], is shown in Figure 2.2. The figure shows the topology and the belonging conditional probability tables. The example illustrates an alarm which detects burglars. However, the alarm also responds to minor earthquakes. If the alarm sounds, there is a probability that John and Mary will call. The network shows that both burglary and earthquake are parents to alarm, and therefor directly affects the probability for the alarm to sound. An earthquake and burglary however, does not effect each other at all. The figure also shows that there are no dependencies between burglary and earthquake, despite the fact that they are both parents to alarm. This means that they do not influence each other.

The tables next to each node in the Figure 2.2 shows the conditional distributions, and are called a conditional probability table. These tables show the probability for an event, given the occurrence (or absence) of the parent event(s). Each row in the tables represent a conditioning case, representing a possible case of the parent nodes. E.g. the top row in

the alarm's table indicates that there is a 95% probability for the alarm to sound if there is a burglary and an earthquake.

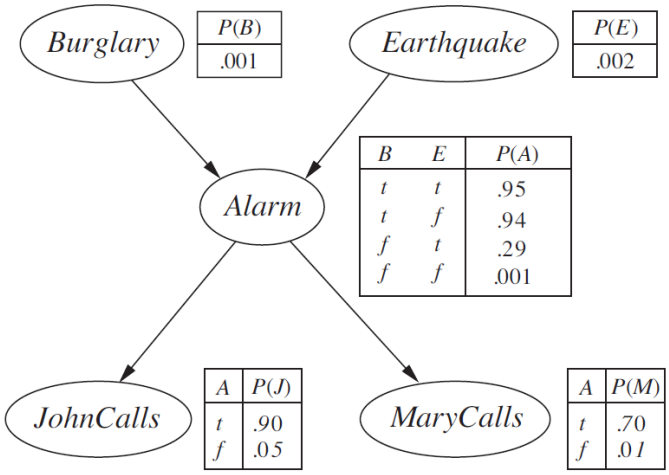


Figure 2.2: An example of a simple Bayesian network, showing both the topology and the conditional probability tables. [Russell and Norvig, 2016]

In general, the probability for a combination of variables in a network is given by

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

where *parents(X_i)* is the parents of x_i. As stated by Russell and Norvig, this equation can be used to guide knowledge engineers in constructing the topology for a BN. The equation is then rewritten using the product rule:

$$P(x_1, ..., x_n) = P(x_n|x_{n-1}, ..., x_1)P(x_{n-1}, ..., x_1). \tag{2.3}$$

Repeating this process, we end up with following product:

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i|x_{i-1}, ..., x_1). \tag{2.4}$$

Comparing this with Equation 2.2, we get that

$$P(X_i|X_{i-1}, ..., X_1) = P(X_i|parents(X_i)), \tag{2.5}$$

which says that the BN correctly represents a domain if each node, given its parents, is conditionally independent of each predecessors. Each node is conditionally independent of its other predecessor in the node ordering. To satisfy this the set of variables required needs to be determined, with an ordering of them. To have a more compact representation of the domain, it is preferred to have variables ordered in a way that effects follows causes.

Further, a minimal set of parents for the variables are chosen, while Equation 2.5 is still satisfied. Additionally, the parents of a variable X_i should contain all variables that directly influence X_i. With the domain from Figure 2.2, John deciding to call is influenced by both the occurrence of an earthquake and a burglary, but is not directly influenced by them. These events affects whether or not the alarm sound, which is the only reason for John to call. This gives:

$$P(JohnCalls|MaryCalls, Alarm, Earthquake, Burglary) = P(JohnCalls|Alarm), \tag{2.6}$$

which indicates that *Alarm* is the only parent node for *JohnCalls*.

2.3 Logistic regression

Logistic regression is often used with linear classification where the goal is to model the probability of a variable being true or false. The difference from linear regression is that the logistic regression predicts binary values rather than a continuous value, using a soft threshold function. Instead of fitting a line through the data (hard threshold function), the logistic regression fits an "S" shaped logistic function which is a soft threshold function, also called a sigmoid function. The process of fitting the weights of this model to minimize loss on a data set is called logistic regression [Russell and Norvig, 2016]. The different functions can be seen in Figure 2.3.

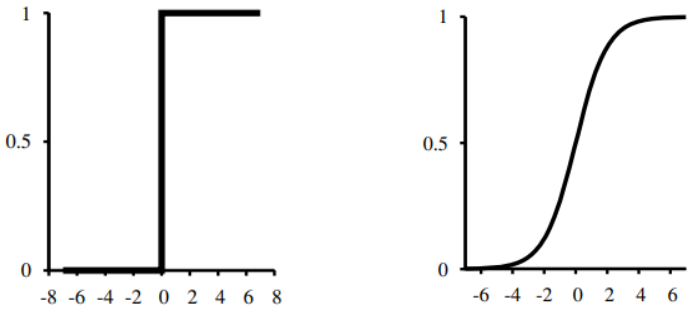


Figure 2.3: (a) is the hard threshold function Threshold(z) and has the possible output of 1 and 0. The function also has the feature that the function is nondifferentiable at z = 0. (b) is the Logistic(z) function, also called a sigmoid function. [Russell and Norvig, 2016]

Koslowsky et al. [2018] states that Logistic regression also seeks to predict the effects of the prediction variables on the prediction result. This can be done by looking at the value of the regression coefficients which represents the relationship between the variable and prediction response. The coefficient are often called weights and is either positive or negative which tells us what kind of impact the variable has on the prediction result.

In Russell and Norvig [2016], the complexity for a linear function can be specified as

a function of weights. We can consider a family of regularization functions:

$$Complexity(h_w) = L_q(w) = \sum_i |w_i|^q, \tag{2.7}$$

where $q = 1$ and $q = 2$ is L1 and L2 regularization, respectively. L1 and L2 regularization techniques is used with the loss function in regression, where the regularization minimizes the sum of absolute values and the latter minimizes the sum of squares (see Equation 2.7 for reference). L1 regularization tends to produce a sparse model, which means that it often sets weights to zero, effectively declaring the corresponding attributes to be irrelevant. This can make it easier for humans to understand, and decreases the probability of overfitting. Both of these function has a regularization factor often called λ, which is a hyperparameter to control the regularization strength.

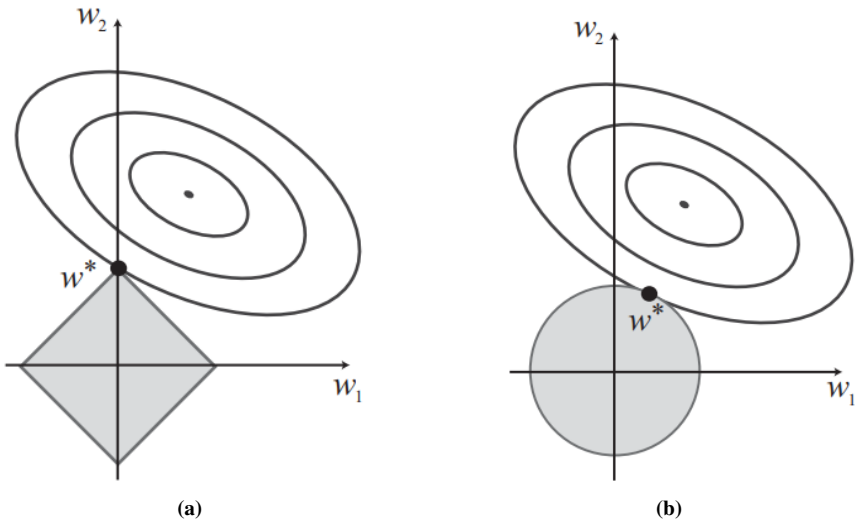


Figure 2.4: (a) L1 regularization having a diamond shape and (b) L2 regularization having a circle shape that represent the set of point **w** in two-dimensional weight space that have a less complexity than c; the solution will have to be inside this box. The concentric ovals represent contours of the loss function, with the minimum loss at the center. [Russell and Norvig, 2016]

Figure 2.4 shows why L1 regularization tends to create a sparse model. We want to find the point on the box that is closest to the minimum loss and in Figure 2.4a we can see that it will be common for the corner of the diamond to find the minimum rather than the sides, just because the corners are pointy. And of course the corners are the points that have a value of zero in some dimension. In the other hand L2 regularization has a circle rather than a diamond. We can see that, in general there is no reason to find the minimum close to one of the axes, making the L2 regularization less likely to produce zero weights [Russell and Norvig, 2016].

In a more practical sense, the L1 regularization technique shrinks the less important feature's coefficient to zero, removing some features altogether. This is preferable when

dealing with feature selection where we have an enormous amount of features [Nagpal, 2017]. L2 regularization in the other hands, penalizes values by limiting weights, but usually does not set them directly to zero.

Ordinary regression only predict two categories (true or false), but when dealing more than two categories where each category has a sequential order where one category is higher than the previous category, ordered regression is the best fitted model. The model predicts the probability for each category.

Another form of regression is probit regression, which differs from logistic regression in how the *f(*)* is defined in the predictor function:

$$\hat{Y} = f(\alpha + \beta x) \tag{2.8}$$

Logistic regression uses the cumulative distribution function of the logistic distribution, while the probit uses cumulative distribution function of the standard normal distribution which is defined by Russell and Norvig as:

$$F(x) = \int_{-\infty}^{x} P(z)dz = \frac{1}{2}(1 + erf(\frac{z-\mu}{\sigma\sqrt{2}})) \tag{2.9}$$

2.4 Neural networks

Neural networks (NN) also called artificial neural networks are computer systems inspired by the use of neurons from the human brain. Each neuron produces an output when a linear combination of its inputs exceeds a threshold, which can be of a soft threshold or a hard threshold. A collection of several neurons that is connected together is called a neural network. The properties of the neurons and the topology makes the property of the network. The mathematical representation of the neuron can be seen in Figure 2.5. The

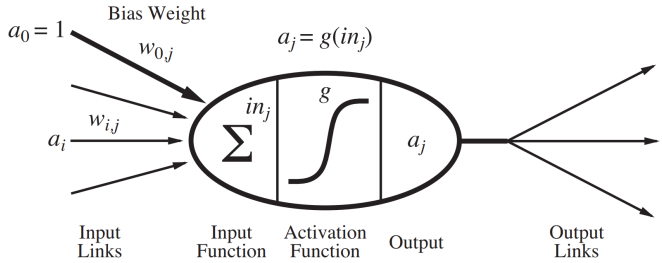


Figure 2.5: A mathematical representation of a neuron. The neurons output activation $a_j = g(\sum_{i=0}^{n} w_{i,j}\, a_i)$, where a_i is the output activation of unit *i* and $w_{i,j}$ is the weight on the link from unit *i* to this limit. [Russell and Norvig, 2016]

neurons are called units or nodes in the networks and is connected via links between each other. A link from node *i* to node *j* serves as a way to propagate the activation a_i from *i* to *j*. Every link has a numeric weight $w_{i,j}$ to it, which determines the power and sign of the connection [Russell and Norvig, 2016].

To produce an output from the network, each unit *j* has to compute the weighted sum of its inputs:

$$in_j = \sum_{i=0}^{n} w_{i,j} a_i \tag{2.10}$$

And then use an activation function *g* to derive the output, where the activation function *g* is typically a hard threshold function or a soft threshold function, creating a perceptron or a sigmoid perceptron:

$$a_J = g(in_j) = g(\sum_{i=0}^{n} w_{i,j} a_i) \tag{2.11}$$

After choosing the mathematical model for the neurons, a way of connecting them is required. There are two ways to connect the neurons. The first one is creating a feed-forward network that only has a connection one direction, meaning that it creates a acyclic graph. Every node gets its input from an upstream node(s) and produces output to a down-stream node(s). It has no internal state and no connections forming any cycles in the network. The feed-forward network itself represents a function of its current input because of its lack of internal states [Russell and Norvig, 2016].

The other way is by creating a recurrent neural network (RNN), which on the contrary uses the output from a node and feeds it into its own input, thus creating a cycle. This creates the possibility for the activation levels of the network to create a dynamic system that might reach a stable state or chaotic state. Since these types of network has nodes where the output may be dependent on previous inputs, we can say that RNN supports short-term memory.

Layers is a central concept in NNs, which forces each node to receive its inputs from other nodes in the preceding layer. A single-layer network will have its nodes connected directly from the input to the output, while a multilayer network will have one or more hidden layers that are connected to each other rather than directly to the output. Figure 2.6 illustrates a NN with a single hidden layer.

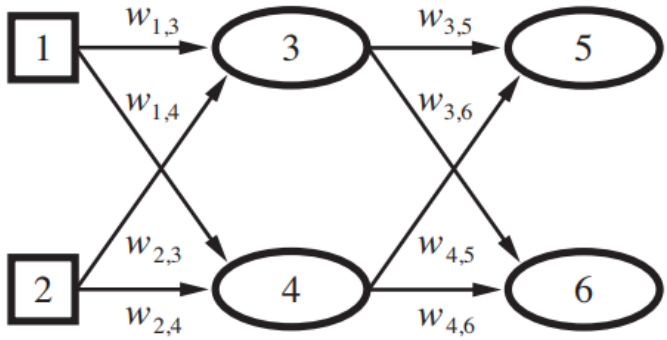


Figure 2.6: A neural network with two inputs, one hidden layer of two units, and one output unit. $W_{i,j}$ represents the associated weights from node *i* to *j*. [Russell and Norvig, 2016]

Increasing the dimensionality in the network by connecting large numbers of units into networks of arbitrary depth, will make it possible to solve more complex problems, such

as the *XOR* problem which is not linearly separable. As the network expands, introducing more parameters to the model, the problem with overfitting may occur. Overfitting can occur to all kinds of learners and causes the learner to produce an output that corresponds too similar to a particular set of data, thus failing to fit unseen data or produce future predictions inaccurately.

There are many techniques to prevent overfitting in NNs. One of them is by using previously mentioned L1 and L2 regularizations and its hyperparameter to control regularization strength. Early stopping is a technique where the data is divided into training, validation and test sets. For every iteration, the validation set's error is checked, and the learning process stops when the validation set's error rises. Dropout is another regularization method commonly used where every unit of the NN, expect the units in the output layer, is given a probability of being ignored temporarily in the [Skalski, 2018]. Then for each iteration, neurons gets randomly selected to drop according to the probability that is defined. The results of this that for each time we use the network, we get a smaller NN, which tends to overfit less. Figure 2.7 illustrates a network where four neurons has been dropped.

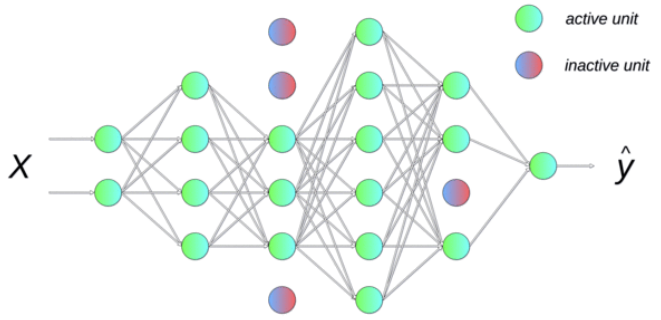


Figure 2.7: A neural network where four neurons has been ignored. The green dots represents active neurons, while the blue/red represents inactive neurons. [Skalski, 2018]

Chapter 3

State of the art

Trying to predict football matches has been a topic of research since long before the computational tools and power we have today. There are publications dating back to over 60 years ago, that tries to find the magic formula to predict football matches [Moroney, 1951].

3.1 Probability based prediction

A common approach to predict match outcome, without the use of machine learning, is to use Poisson distribution to measure the probability of the number of goals scored in a match. Maher is one of the first to investigate the use of the Poisson model with football scores. Earlier attempts has been made to fit a Poisson distribution to the number of goals scored in a match, but as Maher states in his paper, these earlier attempts rejected the use of the the Poisson model in favour of another model [Moroney, 1951, Reep et al., 1971].

Maher works further with the Poisson model in his paper. To calculate the Poisson distribution for football results, the two opposing teams' attacking and defencing strength, α and β respectively, are used. Only the number of goals, the opposing teams, and the venue of the match are parameters for calculating the strengths values. When playing at home, team *i*'s number of goals against team *j* is modelled as a Poisson variable, X_{ij}. The number of goals scored by team *j* is also modeled as a Poisson variable, Y_{ij}. Maher assumes these two, X_{ij} and Y_{ij}, to be independent of each other, meaning they can be evaluated as two separate games at each end of the pitch. Based on this, the outcome of a game between team *i* and team *j* is given by the distribution

$$P(X_{ij} = x, Y_{ij} = y|\alpha, \beta, k) = Poisson(x|k \cdot \alpha_i\beta_j) \cdot Poisson(y|\alpha_j\beta_i), \qquad (3.1)$$

where *k* is the home field advantage assumed to be equal for all teams. An important note is that α and β do not vary over time.

The model was used on four English leagues over three season, giving a total of 12 data sets. Comparing the expected (calculated) with the observed (actual) goal frequencies, some systematic differences could be seen. The model underestimated the probability for

one and two goals to be scored, and overestimated the probability that none and more than four goals would be scored. The difference between the expected and the observed frequencies are quite small, but added together would lead to a rejection of the model.

Further extensions of this model has been made to Maher's model. Dixon and Coles [1997] used results from 6629 league and cup matches from the top four English divisions from 1992 to 1995, and made a model that was able to generate score probabilities. They added a function $\tau(x, y)$ to Equation 3.1, that adjusted the probability for the low-scoring match results compared to Maher's model. Dixon and Coles also brings up the limitation with the model being static and that the attack and defence strength of a team is considered as constant through time by this model. This is not the case in reality, as a team's performance, considering both attack and defence, could vary from one time period to another. Dixon and Coles handles this by taking into account that recent matches reflects a team current form better than matches earlier in history.

Even though Maher's paper was published in 1982, the model is still relevant today. With the growth of emergence of football betting, many hobbyist have used similar kind of models to predict the outcome of matches [Ammon, 2016, Cronin, 2017a]. These model calculates the goal probabilities for each team, which could be a good aid for different kinds of bets. Multiplying these probabilities and plotting the data into a table, as shown in Table 3.1, makes it easier to see what the probability for different outcomes is. E.g. is the probability that the home team does not score is below 4%.

Table 3.1: Results from prediction for a match using Poisson. The blue field indicates the Poisson distribution for each of the teams while the green fields indicates home win, red field indicates away win and the yellow field shows the probabilities for the different draw scores.

	Goals: Home team	0	1	2	3	4
Goals: Away Team	Probability for number of goals	3.65%	12.07%	19.99%	22.07%	18.27%
0	36.11%	1.32%	4.36%	7.22%	7.97%	6.60%
1	36.78%	1.34%	4.44%	7.35%	8.12%	6.72%
2	18.73%	0.68%	2.26%	3.74%	4.13%	3.42%
3	6.36%	0.23%	0.77%	1.27%	1.40%	1.16%
4	1.62%	0.06%	0.20%	0.32%	0.36%	0.30%

The two articles, by Ammon and Cronin, do not mention how their model's predictions did compared to the actual results. Even though they do not record any results, it seems like this model uses features that are necessary to predict a football match. The attacking and defencing strength for each team, that are the basis of this model, are calculated using other factors that could be useful in other models as well. These are the goal statistics during the season for each team, the number of goals scored and conceded in both home and away matches. However, only using these features leaves out other information that could be vital. Important factors like player suspensions and injuries, how important the game is (motivation, researched in Constantinou et al. [2012]), and how many games a team has played the last week are not taken into account.

The usage of BNs as a model has also been explored. Joseph et al. [2006] constructed

a BN to predict the outcome of matches played by Tottenham Hotspurs Football Club, in the period 1995-1997. This expert BN was based almost entirely on subjective judgment, but performed rather well in its prediction. This BN involves specific players and therefor only relevant for the two seasons.

The expert model built by Joseph et al. included only the few features shown in Figure 3.1. The *Attack* measure is based on the presence of three key players: Sherringham, Anderton and Armstrong. Combined with the position played by a player named Wilson, whether or not he played in the midfield, defined the *Spurs_Quality*. How their performance would be in each match, is naturally also dependent on the quality of the opposing team. The evaluation of the opposing teams was done with a simple 3-point scale: High, medium and low. Lastly, the venue where the matches were played also had an impact on the match results.

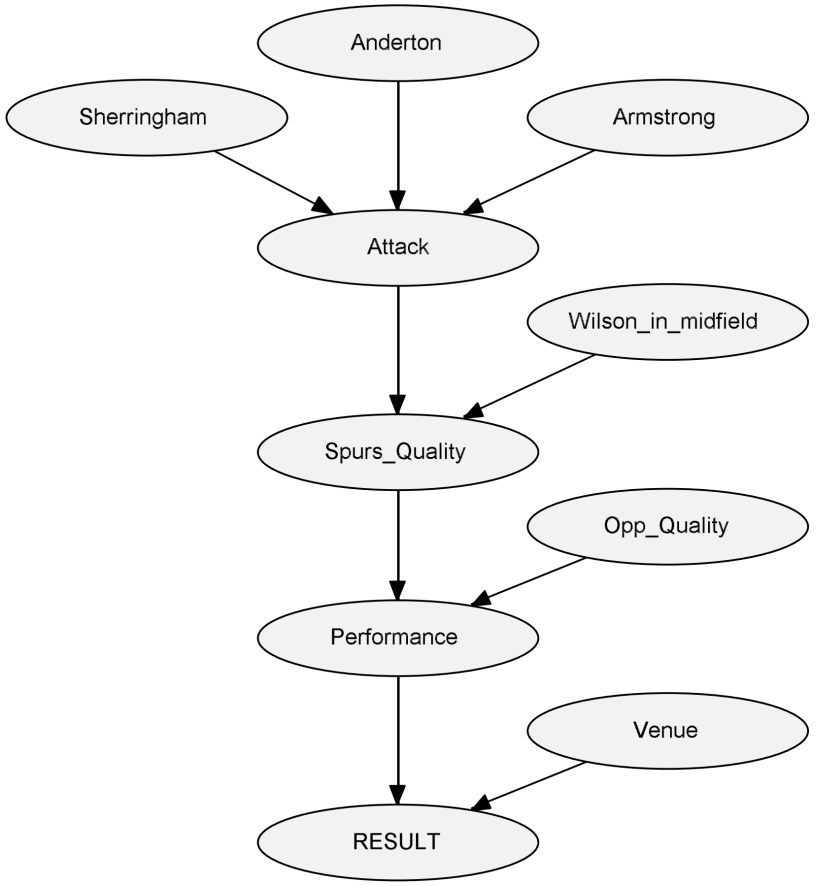


Figure 3.1: The expert constructed Bayesian Network for Tottenham's performance. [Joseph et al., 2006]

The paper was written almost ten years after the two seasons in consideration. This

was done to compare how the expert BN performed when set up against other machine learning techniques. Among these were MC4 Decision trees, a BN learned from statistical data, a naive BN, and the K-nearest neighbour technique. The results, after testing all the different learners, showed that the expert BN performed best. The expert BN achieved an overall accuracy of 59.21%, which was significantly higher than the others. Even though this model is outdated due to its player specific features, this study still shows the potential of expert generated BNs.

In Constantinou et al. [2012], a model called pi-football was created to predict the outcomes of the matches during the Premier League (PL) season 2010 - 2011. The model pi-football is a BN based on both objective and subjective information, which together gives a subjective forecast. Historical data from 1993 to 2010 are used to calculate the level of team-strength, which are the basis of an objective forecast. The strength of a team for each season is based on on the total number of points they achieved that season. The subjective information is based on three components: Form, psychology and fatigue. These three components, along with the objective forecast are all linked together, as shown in Figure 3.2, and gives a subjective forecasts.

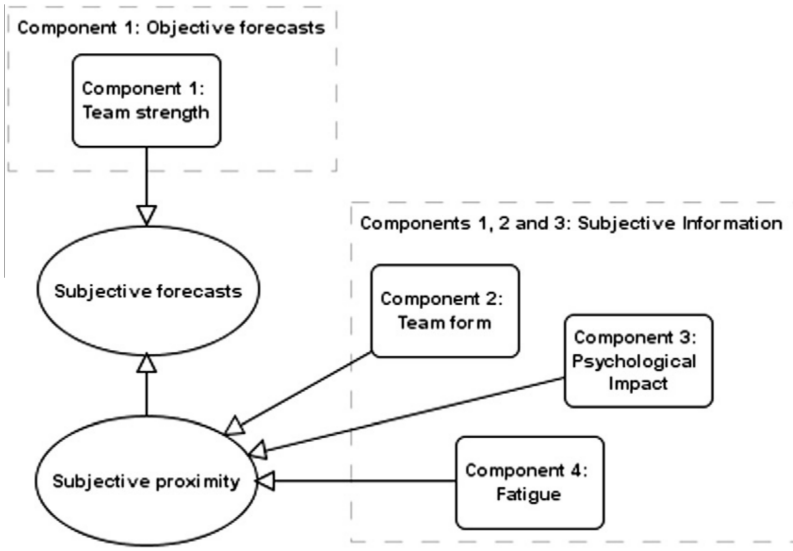


Figure 3.2: The Bayesian network called pi-football and how its components are linked together. [Constantinou et al., 2012]

Looking at Figure 3.2, pi-football seems like a rather simple model, but the subjective components are based on several factors. Figure 3.3 shows the BN for estimating a team's form. The form is represented with a value on a scale from 0 to 1, where 0.5 indicates that the team is performing as expected. Higher than 0.5 means that the team performs better than expected. As seen in the figure, the availability of key players, recent results and first team players are considered when estimating a teams form. Some of the aspects that are taken into consideration regarding the psychological impact are the teams confidence, their spirit and motivation, and managerial impact. The last subjective components, fatigue, is

determined by the toughness and how many first team players that rested the previous match, how many days it has been since the last match, and whether or not the first team players have participated for the national team.

Figure 3.3: The Bayesian network used to estimate a team's form in pi-football. [Constantinou et al., 2012]

In total, all these components make up a complex network involving a great number of features requiring good team insight. In order for such a model to perform well, the knowledge of the expert is important. For this study, the subjective input is not provided by a top expert in the field, or as Constantinou et al. puts it; "the subjective inputs were provided by a member of the research team who is certainly not an expert on the English Premier League" [Constantinou et al., 2012]. Nevertheless, Constantinou et al. refers to good results when he assess the quality of the system. He tests pi-football's accuracy by using a scoring rule called RPS [Epstein, 1969]. The RPS is computed for three forecasts: An objective forecast, a forecast by pi-football, and a bookmakers' forecast. The test proved that the accuracy improved when using subjective information rather than just objective, however it also appeared that the bookmakers had higher overall accuracy. This might suggest that bookmakers also use subjective information in addition to standard football data available to public.

To determine the profitability of the forecasts, a betting simulation was performed. Three sets of bookmakers' odds where considered: the best available, the mean, and the most common. Overall, 35% of the bets were won. The mean odds won was around 3 for all the three sets, with a max of 9 for set with the best available odds. Constantinou et al. achieved a profit of 8.40%, 2.86% and 9.48% for the three set, respectively. According to Constantinou et al., no other published work has been successful at beating all available bookmakers over a period of time. Based on this, he concludes with pi-football being a great success.

Constantinou has also done more recent studies in this field. In Constantinou [2018], he created a model with dynamic rating called Dolores. The rating system is called pi-rating and originates from pi-football [Constantinou et al., 2012]. The system provides a relative measure of superiority between adversaries for each league and is used as an input to a hybrid BN. This rating is used when predicting the outcome of a match by finding other random matches that has the same rating, ignoring the date, the place, or the teams of the match. This eliminates the problem of finding historical data that is sparse and temporally dependent, where recent data is more important than old data. Hybrid BN are simply BN models that incorporate both discrete and continuous variables. The data set consist of 216,743 matches from leagues throughout the world with each sample providing information about name of the home and away team, the football league, the date of the match, and the final score in terms of goals scored. From this study Constantinou has described Dolores, which provides empirical proof that a model can make a good prediction for a match outcome between teams x and y even when the prediction is derived from historical match data (only goal scoring data) that neither x nor y participated in. The model is also used for protability-based validation, based on bookmakers odds from 21 different leagues and over a period of approximately seven football seasons. The results indicate marginal profits of 1.09% return on investment (ROI) over all top divisions, and marginal losses of 1.57% ROI over all lower divisions. These numbers are not so impressive and worse than he has previously achieved [Constantinou et al., 2012], but it gives empirical proof that the model was able to generalize over all leagues and division based on goal data.

3.2 Prediction with machine learning

The emergence of machine learning has allowed us to discover patterns that previously were unknown, and have shown good results in classification tasks. The outcome of a football match can be seen as a classification task, where one of the three outcomes win, draw, lose is to be predicted. It can also be viewed as an prediction task, where we want the outcome of the model to be the match score. Naturally, there has been attempts in classifying football matches with the use of machine learning techniques.

An L2-regularized logistic regression model was used in Kerr [2015], where the model used 5-fold cross validation to select the optimum penalty factor. The model was chosen because it did not only give the results of the classification task, but also what level of impact each feature had on the result. The goal of the experiment was to show that they could learn what non-obvious features, constructed using the available ball-event data, are associated with a games outcome. The data used for the model was recorded ball-event data from previous games, where an event is a pass, shot, etc. The most interesting aspect

of this approach was the fact that the authors were focusing on using features that are not obviously correlated with the outcome of the game. The methodology of the experiment was executed in five steps. The first step being the transformation of the data and the construction of features to create a data set suitable for supervised learning. In this step each match from the data set gets randomly chosen a team A and B, where the outcome being 1 if team A won and 0 otherwise. The feature sets was created by using expert knowledge and by looking at traditional statistics for football games. The features was separated into two models, Obvious Model which contained obvious features and Non-Obvious Model which contained less obvious features. The next two steps, 2 and 3 was splitting the set into a 80-20 ratio for the training and test set and performing a z-score normalization, with the formula

$$z = \frac{x - \mu}{\sigma}, \tag{3.2}$$

where μ is the mean and σ is the standard deviation of the training values given a particular feature. Step 4 was to train the logistic regression model with the feature sets from the earlier steps. If the model predicted a value greater than 0.5, they assumed that team A won the game. By using a logistic regression model, they were able to use the coefficient value for a feature as a proxy for its importance in determining a team's success. To maximize accuracy they also performed 5-fold cross validation in order to select the regularization factor, which had a search space of 0.001, 0.01, 0.1, 0.5, 1, 1.5, 10. The last step is presenting the accuracy attained on the test sets for the different models, which was 74% for the Obvious Model and 75.6% for the non-obvious. A third and final model was also created with features from both of the models, achieving 84% accuracy. The feature results associated with a game outcome is also presented, which was the goal of experiment. The features are ranked after their absolute weight in order to gain insight into their importance. The top features for the Non-Obivious model can be seen in Table 3.2, while the top features for the mixed model can be seen in Table 3.3, where the 5$^{\text{th}}$ feature is from the Non-Obvious model.

Table 3.2: Results from Kerr [2015]'s experiment, showing which non-obvious feature had the most impact on the match outcome, ranked by the absolute value of their weights.

Rank	Feature	Weight	Relative Weight
1.	Difference in the number of crosses	-0.522	0.0618
2.	Number of crosses Team A	-0.445	0.0526
3.	Difference in the average shot distance	0.428	0.0507
4.	Number of crosses team B	0.414	0.0489
5.	Number of well positioned shots	0.401	0.0474

A majority of the explored models have focused on the attack and defence strength of a team, but there are a lot of other features that come into play resulting in a goal. Kerr explored some of these. One of the most surprising revelations is that having more crosses can decrease the chance of the team scoring a goal. According to Kerr, this is because crossing the ball more frequently results in either fewer opportunities to score, or worse opportunities to score. From a feature set of several non-obvious features, the difference in number of crosses had the most impact on the match outcome (see Table 3.2). This feature

Table 3.3: Results from the mixed model in Kerr [2015]'s experiment, showing which feature had the most impact on the match outcome. The features is ranked by the absolute value of their weights and consist of a combination of obvious and non-obvious features.

Rank	Feature	Weight	Relative Weight
1.	Difference in the number of shots on target	0.672	0.0929
2.	Number of shots on target team A	0.555	0.0766
3.	Number of shots on target team B	-0.499	0.0690
4.	Home	0.433	0.0598
5.	Difference in the number of crosses	-0.394	0.0544

compared to every other feature used in Kerr's experiment could be seen from Table 3.3.

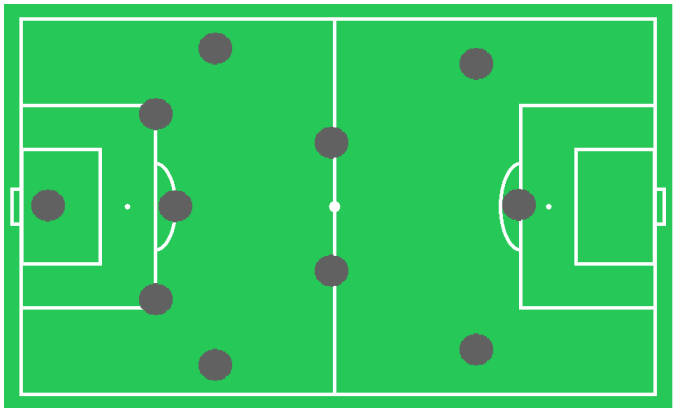
Bradley [2018] attempts to predict football matches using player ratings from the video game FIFA, with the use of a NN. EA releases a new version of FIFA every year, naturally with updated player ratings. The idea to use data from FIFA is that we will not have to consider potential missing key players and players that transfer to a new team, assuming these player ratings are accurate and independent of leagues. Data from matches in the PL seasons from 2013 to 2018 were used, including 22 player ratings for each match. The teams formation was also taken into consideration. So, instead of having a vector of length 22 as input for each match, an extended vector of length 32 were used. The 16 first fields represent the home team's line up, while the last 16 are the away team. Having it consequently in this particularly order, might make the NN discover a home team advantage. Each section of the vector represents a position on the football field. The first component in the vector is for the goalkeeper. The next section, including the next six fields, represents the defenders. If a team played a formation with four defenders, the last two fields in the sector representing the defenders would be left as 0. Figure 3.4b shows the input vector for a team playing the formation 5-2-3 (shown in Figure 3.4a).

Bradley used a NN with dropout and early-stopping. The optimal network structure for this data was with two hidden layers, one with 16 nodes and the last with eight. The dimension of the NN is therefor 32-16-8-3, as there are three options for the output (home win, away win, draw). In order to test his system, Bradley used a betting strategy to try to profit from his model. Something the betting strategy included were restrictions on what kind of bets were allowed, e.g. not placing bets on odds greater than 3.2. Betting on the 2017-2018 season's matches, Bradley achieved an ROI of 11%, where he won 50% the bets.

3.3 Live predictions

Bookmakers offer their clients the opportunity to place bets during the match. Different kinds of bets are available, and as the match plays out the odds naturally changes. Thus, techniques to predict matches live, as they play out, are naturally being explored.

An approach related to live prediction, is Boice's attempt to predict the outcome of the 2018 World Cup. In Boice [2018], a model is made based on creating Poisson distributions for each team and a matrix showing all the possible match scores with their probability.



(a)

89	81	82	82	82	86	0	85	88	0	0	0	0	0	91	84	84	0

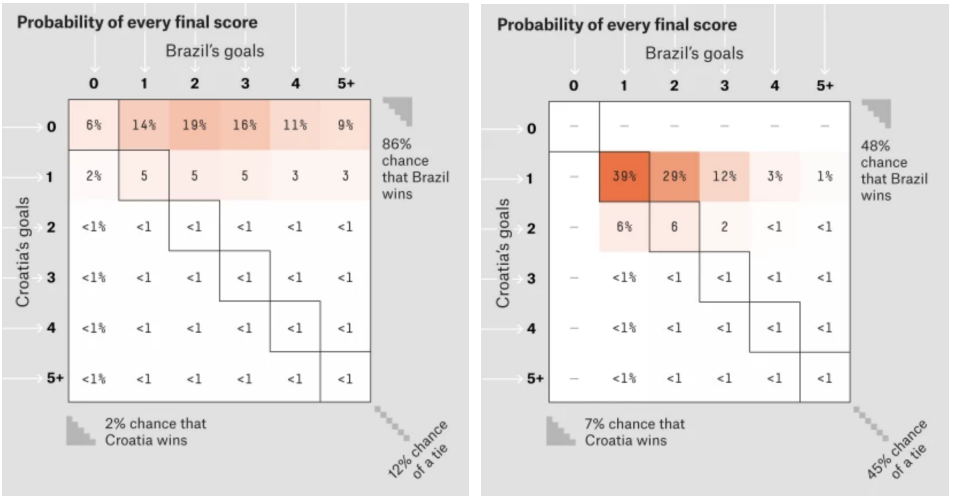
(b) Bradley [2018]

Figure 3.4: How the input vector (b) will look like for a team playing the formation 5-2-3 (a). The first field represent the goalkeeper. The next six are reserved for up to six defenders. The following seven are for the players playing in the midfield, while the last four represents the attackers.

This makes it possible to find the pre-game probability of winning for each team. The difference with this model and the models based on Maher, is that the attack and defence strengths used in this model are FiveThirtyEight's own variable called SPI ratings. SPI ratings are their own estimates of overall team strength, which are made up from match-based and roster-based ratings. These ratings are generated from data stored in their own database that contains matches that dates back to 1905. Based on the teams ratings, a win/loss/draw probability matrix for a match are generated with the use of the Poisson distributions of expected goals for each team. Further, in order to forecast which team will win the World Cup, they used all the matrices with Monte Carlo simulations, which will simulate thousands of tournaments resulting in a winner based on how many times the team occurs in a winning simulation.

Live match predictions were also implemented in their work, which calculated each teams chances of winning, losing or drawing a match in real time. The live model works almost the same way as the pre-game predictions. First the number of goals expected in the remaining time was calculated for each team. Then the Poisson distribution is created based on these numbers and fused together to create a matrix. In the end the current score of the match is combined with the matrix, which gives the score probabilities in real time. Figure 3.5 shows how the probability matrix differences for the Brazil - Croatia match in the 2014 World Cup. Brazil was initially expected to be a clear winner, with a probability of 86%. After 65 minutes, with the score 1-1, this probability has now declined to 48%.

There are factors that needs to be taken into consideration as the match plays out. Some important aspect Boice have considered, are that the scoring intensity at the end of



(a) Pre-game **(b)** After 65 minutes

Figure 3.5: Pre-game vs. live probabilities for the outcome in Brazil - Croatia, 2014 World Cup. How the probabilities of the final score changed after the game had played 65 minutes. [Larsson, 2018]

a match is higher than at the beginning. Added time is also an important, which extends the last period of the match. This is calculated by number of bookings so far in the match, and whether or not it is a close match. Red cards also give a significant advantage and are taken into account. According to Boice, having one more player than the opposing team is worth three times more than a home-field advantage. In addition, after exploring the data, Boice discovered that a team that is down by a goals tends to have a higher scoring rate than whats indicated prior to the match.

Boice provided both pre-game and live predictions based on their model on their website throughout the World Cup, with the probabilities adjusted as the matches played out. As the article was written prior to the World Cup start, they have not concluded with how well their system performed. They do however have the predictions on which team will advance through each stage available on their website [FiveThirtyEight, 2018]. Comparing the predictions for the group stage and the actual results indicates that their system performed rather well. Of the 16 teams that advanced from the group stage, Boice [2018] predicted 14 of them. Considering each team are in a group with three others, and in total two of them advances, 0.875% correct is at least better than random guessing (given the probabilities for different outcomes [Smith, 2017]). The two teams that were predicted to advance , but got knocked out out the group stage, are Germany and Poland. At least Germany were considered by many to have an easy process in advancing to the next stage. Of the 14 correctly predicted teams to advance, 11 of them were also predicted correctly whether they would finish first or second in their respective groups.

Pettersson and Nyquist [2017] use RNNs to predict the outcome of football matches. The data set consisted of matches from multiple seasons of many leagues from differ-

ent countries (63 in total), in addition to tournaments that included teams from several countries (e.g Champions League). Information and events from these matches was used as input into the network. This includes lineups, the starting players' starting positions, goals, cards, substitutions and penalties. Using an RNN ensures that the input data can have different size, which is important as the number of events in a match varies. Still, the input vector for each event has to have the same shape when fed into the network, meaning information for different events need to be merged into a input form that the network can handle. One of the methods Pettersson and Nyquist used to solve this problem is by using a one-hot vector containing all attributes for all events. Pettersson and Nyquist's model uses LSTM, and also a softmax classifier to represent a probability distribution over the three possible classes. The architecture of the model is shown in Figure 3.6.

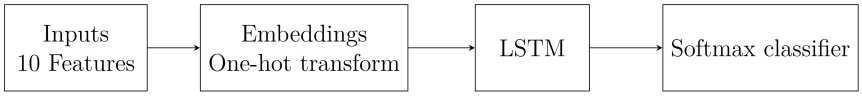


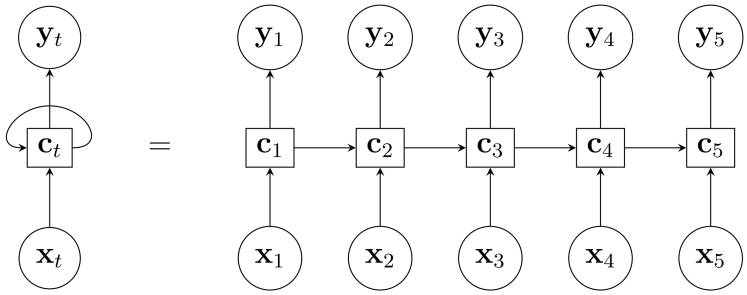
Figure 3.6: The RNN's high level architecture. [Pettersson and Nyquist, 2017]

Two different RNNs are compared, a many-to-one RNN and a many-to-many RNN. A many-to-many has an output at each timestep, while a many-to-one only has an output after the sequence of input (see Figure 3.7). Predictions were done using these RNNs, with 15 minutes intervals for each match. The results from the prediction showed that the many-to-one performed better from the 60th minute and onwards, while a configuration of the many-to-many performed better up to this point. For each time step, the predictions are made by using events that have occurred up to this point. Table 3.4 shows the prediction accuracy at each time step for one of the many-to-many models and the many-to-one model. Initially, with only teams and lineups being known, the systems do not perform much better than random guessing. Both gets higher accuracy as the match nears the final whistle, as to be expected with more events feed into the system and less time left to score for the teams.

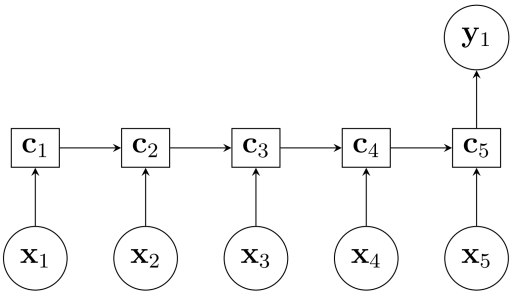
Table 3.4: Prediction comparison between the best configuration of many-to-many model and the many-to-one model. Derived from Table 4.6 in Pettersson and Nyquist [2017].

	Prediction accuracy during match							
Model	0	15	30	45	60	76	90	Full Time
Many to many	**0.4396**	**0.4479**	**0.4705**	**0.5151**	0.5831	0.6797	0.8048	0.8868
Many to one	0.3335	0.3539	0.4151	0.5048	**0.6280**	**0.7409**	**0.8825**	**0.9863**

The test consisted of both classification and prediction, where the many-to-one model showed an overall higher accuracy than the many-to-many model. The many-to-many



(a) Many-to-many RNN model, with an output at each time step.



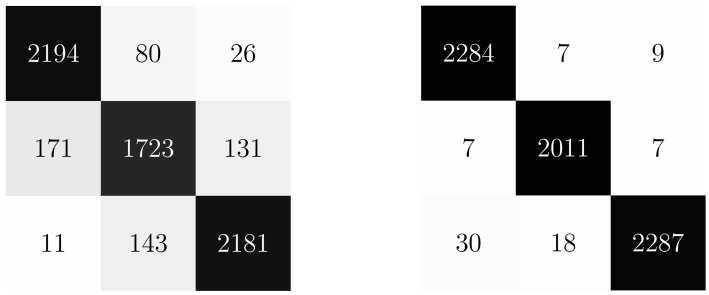
(b) Many-to-one RNN model, with an output only after the sequence of input.

Figure 3.7: The two different RNN models that were used in Pettersson and Nyquist [2017].

model was tested with different parameters, none performing better than the many-to-one model. The many-to-one approach calculates the accuracy at the end of the sequence, while the many-to-many averages over all the events. The classification results had a training accuracy of 100% and a test accuracy of 98% using the many-to-one model. The best test accuracy from a many-to-many model was 88%. However, the many-to-many model was ”closer” to the correct answer when classifying wrong in the cases of home win and away win. Meaning in the cases of home win, only a few were classified as away win compared to draw. The same goes for the cases with away win; when classified wrong the system classified mostly as draw and not home win. With the many-to-one model, the model often classified the other team winning, rather than draw, when classifying wrong. This is illustrated by the confusion matrices in Figure 3.8.

3.4 Additional findings

The ELO rating is a rating model which originated from chess [Elo, 1978]. Unlike Maher [1982]'s model, the ELO rating is time dependent and updates the teams' ratings after each match. ELO rating has been used to predict matches. Hvattum and Arntzen [2010] used an implementation of the ELO ratings for football teams to predict the match results. The implementation consisted of a method which used the ELO differences as covariate



(a) Confusion matrix from the many-to-many model.

(b) Confusion matrix from the many-to-one model.

Figure 3.8: The two matrices shows the classification results from the many-to-many model (a) and the many-to-one model (b). Each row corresponds to the actual results home win, draw, and away win respectively. The column represents, with the same order, what was predicted. A perfect classification would only have complete black cells along the diagonal, indicating 100% accuracy in the predictions. [Pettersson and Nyquist, 2017]

in an ordered logit regression model. Through several comparison between other benchmarks predictions, the results were worse than the two benchmark models based on the market odds, but better than the remaining four models that were based on team performance and past matches. Two of the beaten benchmark models were quite simple, where one used an uniform distribution on the possible outcomes (giving probability of $\frac{1}{3}$ for each) and the other model used the frequency of each outcome in previous matches to give a probability. The two other models are derived from the work of Forrest et al. [2005], that used ordered probit regressions to predict matches. From these comparisons Hvattum and Arntzen found out that statistical loss functions were more efficient than economic measures in differentiating between the different predictions methods. In conclusion ELO ratings appear to be useful when handling information from past results. Regarding football prediction, the rating had a lot of importance during the prediction, a small rating difference is a highly significant predictor of the match outcomes.

Using only bookmakers' odds to predict matches has also proven to be quite effective. Odachowski and Grekow [2012] tries to predict the outcome of football matches based on changes in bookmakers odds, as odds change due to the bets placed. Odachowski and Grekow assumes that a gambler has reasons to place a bet, and that it is not done simply on a whim. Assuming this implies that the changes in the odds is based on knowledge on the competing teams. Changes of the odds the last 10 hours preceding a match, updated every 10 minutes, along with the match outcome were used as input. Matches over a six month period were used, which included 2615 matches. Different features were derived from the odds changes and used in the models. Some of these are the maximum value, the initial value, and the difference in the initial and final value.

Table 3.5 displays some of Odachowski and Grekow findings. They use different classification algorithms, where a Decision Table gave the highest accuracy when trying to predict one of the three outcomes: Home win, draw or away win. With this algorithm, an accuracy of 46.51% was achieved. He got higher accuracy when he used a binary classifier

for each of the three match results. With a classifier for a home win, the binary classifier would simply classify if the home team wins or not, meaning a draw and an away win would be classified the same. Different classification algorithms were tested here as well, with varying results. The highest accuracy were obtained by the home win classifier with a 70.56% accuracy, using a Bagging algorithm. The highest accuracy for the away win classifier was 65.46% and 56.99% for the draw classifier, using a Naïve Bayes and an Ensemble Selection classifier respectively.

Table 3.5: Results from predictions based on odds change [Odachowski and Grekow, 2012]. Highest accuracy was achieved when using a binary classifier to classify whether the home team won or not.

Type of classifier	Best Algorithm	Accuracy
Home win, draw or away win	Decision Table	46.51%
Home win or not	Bagging	70.56%
Away win or not	Naïve Bayes	65.46%
Draw or not	Ensemble Selection	56.99%

An important fact that needs to be taken into consideration is that each feature can have a different level of impact, dependant on which team is playing. Hirotsu and Wright used a statistical model to obtain insights into the characteristics of teams. From the tests using data from the 1999-2000 PL, they found out that scoring goals at home had a massive impact for some teams while others less significant. Hirotsu and Wright used a interaction parameter called $\lambda_{scorehome(A)}$ from his model, which is the propensity for team A scoring a goal when playing at home. This parameter gives an indication of a team's home preference with regard to scoring goals. Table 3.9 presents the top five and the bottom five teams ranked after the value of the parameter.

Team	$\lambda_{\text{scorehome}}$	*Home goals − away goals*
Coventry City	1.13 (0.44)	29
Tottenham Hotspur	0.44 (0.37)	23
Watford	0.34 (0.43)	13
Bradford City	0.31 (0.42)	14
Chelsea	0.28 (0.37)	17
⋮	⋮	⋮
Derby County	−0.44 (0.38)	0
Leeds United	−0.44 (0.35)	0
Sunderland	−0.44 (0.36)	−1
Aston Villa	−0.46 (0.38)	0
Middlesbrough	−0.46 (0.38)	0

Figure 3.9: Hirotsu and Wright [2003]'s results from his experiment on advantage of scoring goals at home. Top five and bottom five teams are presented where the standard errors is in parentheses.

Figure 3.9 shows that Converty City is the highest ranked team based on this parameter. This means that the team has the strongest tendency to score goals when playing at home compared to other teams. This is not so surprising as Converty City scored 38 goals at home and only nine goals away. On the other hand Middlesbrough had the lowest value

indicating a tendency to score less goal at home rather than away compared to other teams. This is also an reflection of their goals scored, where they scored 23 goals both at home and away. Other interesting observations is that no team had a particular overall tendency of gaining or losing possession at home and that some teams had a better strength at gaining possession against another team. These observations is interesting when it comes to using events as features, because it means that the events happening during a live game could affect each team differently.

In Lucey et al. [2014] they analyzed the 10 second interval before a goal was scored and found out that strategic features like defender proximity, interaction of surrounding players, speed of play, coupled with the shot location are important features which has an impact on determining the likelihood of a team scoring a goal. The information derived from this can be a useful tool for teams to analyze which series events often lead to goals, and further find defensive measure against these. In the case of real-time betting, this information is rather useless, as there is a delay when placing bets [Matchbook, 2018]. Events within a 10 second interval prior of a goal would therefor probably not be possible to use to predict a goal for betting purposes.

3.5 Machine learning in other sports

Machine learning techniques have also been applied to other sports. Detailed game statistics for sports like American football and basketball has been available a long time compared to many other sports, and attempts to predict game results using machine learning techniques can be found from even before this millennium.

Purucker [1996] tried to predict American football (AF) results in the National Football League (NFL) with the use of a NN model. He tried both unsupervised and supervised methods, where the latter clearly gave the best results. Only four features, that were considered to be the most important for winning a AF game, were used in the methods. These were both AF specific features in the form of yards gained and rushing yards gained, and the more general features turnover margin and time of possession. Data from the first rounds were used as training data, while he tried to predict the outcome of the games in the 15^{th} and 16^{th} round.

Purucker found that using unsupervised methods were not very successful. Unsupervised methods tries to divide the data into clusters based on similarities. The best results Purucker obtained using unsupervised methods were with the use of a Hamming network [Gupta and Singh, 2011]. The input vectors are clustered based on their Hamming distances [Burch, 2018] between them. This method was not able to distinguish the teams based on the input, and predicted successfully in only 50% of the cases. According to Purucker, The Hamming network are designed to classify binary patterns and not vectors that have large Hamming distances, so the poor results were not unexpected.

The use of supervised methods gave slightly better results. 64,3% of the games in the 15^{th} round were predicted correctly at best, which is nine out of 14 games. Purucker also tried using discrete input, where the values where scaled. Positive values where represented as 1, values near zero as 0, and negative as -1. Using the discrete input had a lower accuracy, with only 57.1%. This seems only natural as two dissimilar input vectors converted into discrete form may appear as similar. Purucker added another feature before

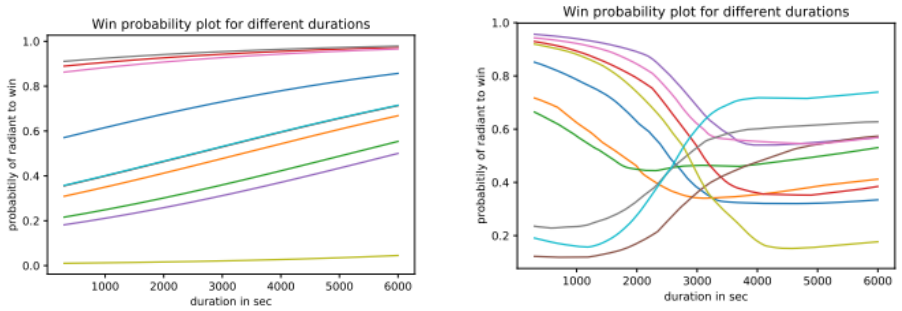
round 16, the Las Vegas line. The line addressed every NFL game and favoured teams over each other. Introducing this feature before the 16^{th} round increased the number of correct predictions to 12 games.

Kahn [2003] achieved higher prediction accuracy, with an extension of Purucker's work. Kahn used data from the 13 first rounds in the 2003 NFL season, which was in total 208 games. In addition to Purucker [1996]'s features, Kahn uses differential statistics as input, which he defines as the difference between offensive and defensive statistics. This includes features from Purucker's models, but also if a team plays at home or away. In addition to the data set containing games from the full season, Khan also used a set consisting of games from the three most recent weeks too see what would give a better prediction.

As with Purucker, Khan also used a NN with back-propagation. The games featured in week 14 and 15 were used for testing, and Khan obtained better results than Purucker. An accuracy of 75% were achieved by Khan, which is significantly higher than the accuracy achieved by Purucker. In addition, the training set including data from the whole season gave better predicting results than the set only featuring the last three weeks. This system also did well compared to experts, as Khan reports that eight sportscasters from ESPN.com predicted on average only 63% of the games correctly.

In more recent times, the competitive electronic sports (eSports) community has started to merge with the betting industry and with increasing tournament prizes and views, it took no time before the bettors came along. The eSports industry is valued at 900 million dollars, where a winning professional team can get up to several millions of dollars [Meola, 2018]. Dota 2 is a popular multiplayer online battle arena game which had a prize pool of 25.5 million dollars for the international 2018 tournament. The game has over 440.000 active players [Steam, 2018] where two teams are facing each other each match, playing as a character of their choice. The game has serious market for competitive gaming with over 15 million people watching the world tournament online. Dota 2 is created by Valve Corperation, which also created the digital distribution software platform, Steam. Larsson [2018] developed an AI called Znipe Sense and was used to predict which team would win in a Dota 2 match. The system used a feed-forward NN and a logistic regression model to compare results. They trained the network using data sets from previous studies and mined data from the STEAM application programming interface (API). The first data set contained over a million professional matches while the second only had around six thousand. Both data sets had the same features where each match had 24 features like skill level of the match, duration, character picks and team won to mention a few. The accuracy achieved from training on these features was 68% with the NN and 64% with the logistic regression model. In addition to these 24 features, Larsson tried to add specific features like synergy and counters, which is the win rate for a pair of characters and the win rate for one character when playing against another character. This was done for every character in the game. One feature unique to Znipe Sense, was each players skill level which was more relevant for the current match instead of just an overall skill level. The final feature set used to train Znipe Sense ended up containing 647 features. The logistic regression model was trained the same way, but did not have greater weighting on recent games which the NN took into consideration. With the new features added the models gained a significantly increase in accuracy. Accomplishing around 75% accuracy for both of the models. While

testing the AI with live and unseen data the model got 73% accuracy while the expert panel, which job is to commentate and predict the games, only managed to get 43%.



(a) **(b)**

Figure 3.10: Prediction graphs where we can see the win probability over time for the regression model (a) and the neural networks model (b). Each color represents individual match predictions. [Larsson, 2018]

Predictions over time was also examined by using a duration as a feature in the models, which made it possible to predict outcome over time. Bear in mind that this is not live prediction during the game, but rather the win probability for a team based on the game duration. When exploring predictions over time Larsson found that the graph for the logistic regression model almost had a linear relationship between the duration and probability of winning (see Figure 3.10a). This relationship is not expected to be linear, because the game has characters designed to perform better earlier and worse later compared to other characters. The same graph was created with the NN (Figure 3.10b), and even though the phenomenon of fluctuating performance on characters over time was not provided to the models, the network was still able to learn the phenomenon from the data. This is one of the main reasons as to why NN were used as the final model for Znipe Sense.

Chapter 4

Data

Using a good selection of features is crucial for creating a good model to predict football matches. Finding features that to some degree affects the outcome of a match, and using these features in a model, is a central part of creating accurate match predictions. These can be gathered from different sources.

4.1 Sportradar

As mentioned initially, Sportradar provides data that is going to be used in this project. With access to their API, detailed information of both historic matches and matches happening real time can be gathered. How detailed a match is covered depends on the tournament's level of popularity. The PL, the league of interest for this project, is naturally well covered. How all the information flows and can be accessed is presented in the Sportradar's Soccer API Map, shown in Figure 4.1. E.g. this is how to find a team's number of goals conceded in the last 15 minutes of a match:

- Call the *Tournament List* by its id and find the team's id
- Call the *Team Statistics* using the previously found id
- Locate the *Goals by Time 76-90* field in the response...

Sportradar has a lot of detailed information about leagues, teams and players that can be analyzed prior to a match. They also provide updated match data during a live game in real-time, which can be useful for making predictions as a match is playing out.

4.1.1 Pre-match data

The rest of a team's season statistics available from Sportradar can be seen in Table 4.1. The data retrieved from this endpoint is for a specified tournament, and is divided into

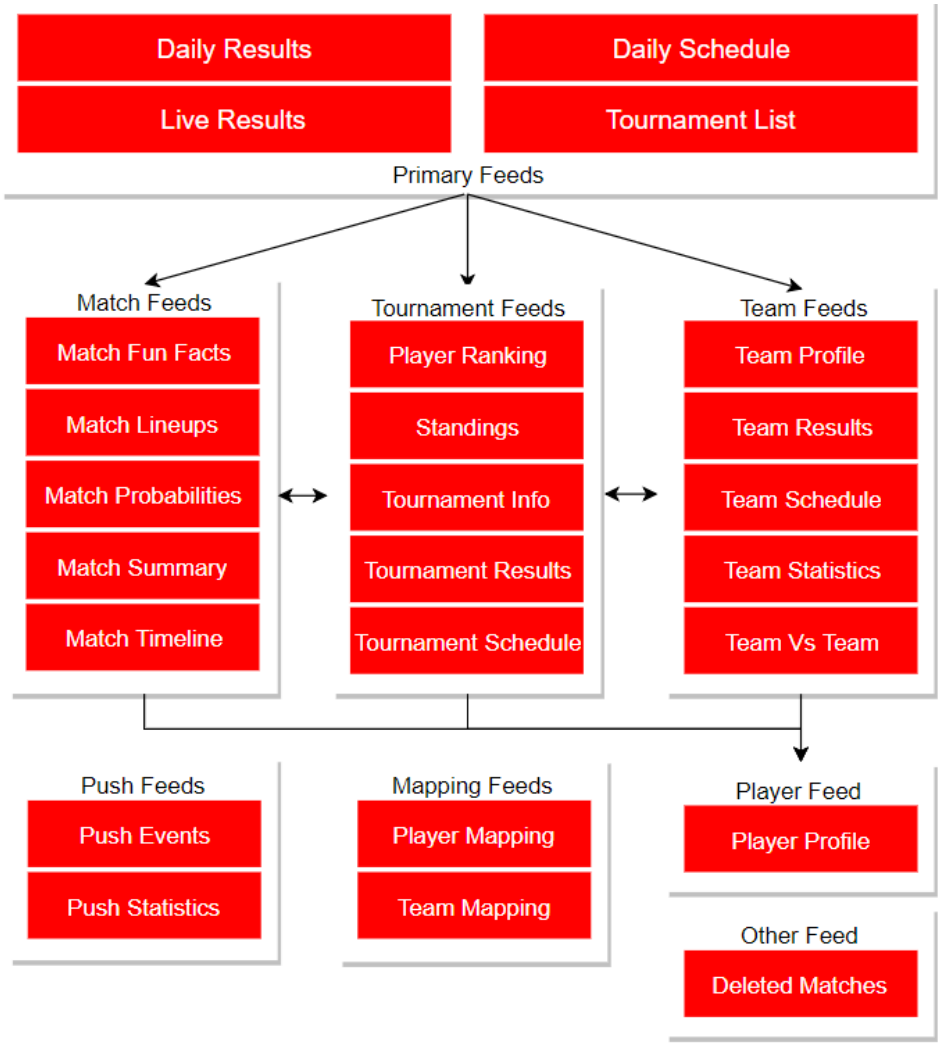


Figure 4.1: Sportradar's API Map. [Sportradar, 2018b]

categories, where the data in *Team Season Statistics Data Points* and *Player Season Statistics Data Points* contains a lot of valuable information. The others categories contains basic team data, such as how many matches they have played in the specified tournament. In Section 3.1, we saw that number of goals scored and conceded were used to calculate a team's attacking and defencing strength, which again was used to predict the outcome of a match. This data can be found in the team statistics, in the *Team Season Statistics Data Points*. The goals scored and conceded for each of the six 15 minutes intervals in a match, which are also included in the teams' statistics, could also be useful data. Using approaches discussed in Section 3.1, we can calculate teams' attacking and defencing

Table 4.1: Sportradar API - Team statistics endpoint. [Sportradar, 2018b]

Sport & Category Info Data Points:	Category Country Code Category Id Category Name	Sport Id Sport Name
Team Info Data Points:	Abbreviation Coverage Matches Covered (Bronze) Coverage Matches Covered (Gold) Coverage Matches Covered (Platinum) Coverage Matches Covered (Silver) Coverage Matches Played	Coverage Scheduled Matches Country Country Code Id Name
Player Info Data Points:	Id Matches Played	Name
Tournament Info Data Points:	Current Season End Date Current Season Id Current Season Name Current Season Start Date	Current Season Year Tournament Id Tournament Name
Team Season Statistics Data Points:	Ball Possession Cards Given Corner Kicks Form Free Kicks Goal Attempts Goals Goals by Foot Goals by Headers Goals by Time 0-15 Goals by Time 16-30 Goals by Time 31-45 Goals by Time 46-60 Goals by Time 61-75 Goals by Time 76-90	Goals Conceded Goals Conceded by Time 0-15 Goals Conceded by Time 16-30 Goals Conceded by Time 31-45 Goals Conceded by Time 46-60 Goals Conceded by Time 61-75 Goals Conceded by Time 76-90 Goals Scored Matches Played Matches Won Offsides Shots Blocked Shots off Goal Shots on Goal - Total
Player Season Statistics Data Points:	Assists Cards Given Chances Created Clean Sheets Corner Kicks Crosses Duels Header Duels Sprint Duels Tackle Goal Attempts Goal Line Clearances Goals by Headers Goals Conceded Goals Scored	Offsides Own Goals Passes Long Passes Medium Passes Short Penalties Faced Penalties Saved Performance Score Shots Blocked Shots Faced Shots off Goal Shots on Goal Substituted In Substituted Out

strength for the 15 minutes intervals. Boice [2018] also stated that the scoring intensity

at the end of a match is higher than at the beginning. Using this data could give better predictions for which team is going to win the rest of the match or the next 15 minutes period. Bookmakers often provides different bets for the 15 minutes intervals in a match. This could be whether or not there will a goal, or which team will win that period.

Other possibly valuable data included in this category are number of cards received and number of shots on target. From Section 3.3 we remember Boice [2018] saying that a team receiving a red card was worth three times more than a home-field advantage for the opposing team. Using this information in a model, a match prediction might differ if one of the teams has a high frequency of cards. From Kerr [2015]'s study we saw that the feature that had the most impact on the match outcome is the difference in number of shots on target (see Table 3.2). Using this data from the current season should therefor be crucial when predicting the outcome of a game. In addition, the other shot data, shots blocked and shots off target, could also prove to be have predicting abilities. The team's form is also available in *Team Season Statistics Data Points*, which Constantinou et al. used as an feature in his pi-football (see Section 3.1). This is gives the results from the last four matches, where a returned string "WWWW" indicates wins in the four most recent matches.

From Table 4.1, we can see that the number of crosses, both the number successful crosses and the total number, are available in *Player Season Statistics Data Points*. The results from Kerr [2015]'s study also revealed that the teams' difference in the number of crosses was the feature that had the most impact on the match outcome (see Table 3.2 in Section 3.2), when a model with non-obvious features was used. Using the mixed model, this feature was still in the top 5 list over the features that had the most impact on the match outcome (see Table 3.3). Kerr's mixed model also showed that the number of shots on target was one of the most important features, which is also available from this API call.

The match lineups are also available from Sportradar. With the call *Match Lineups*, the both teams' lineups for the queried match are returned. This can be called for historic matches, but also for coming matches as the lineups usually are available an hour prior to the match. The lineups naturally includes the starting players and the substitutes, but also the teams' formations and the players' positions. In Section 3.2 we saw that this information had predictive abilities, with Bradley using lineups (with a rating for each player) to predict matches. As mentioned, he achieved an ROI of 11% by placing bets based on these predictions, which is impressive.

4.1.2 Live-match data

The team statistics does not divide the data based on where the team played, home or away, which is often important information. However, the team statistics are naturally based on each of the matches played in the current season. Detailed information on how these matches have played out are available. Table 4.2 shows some of the data points received when calling *Match Timeline*. The API call divides the data into different categories that contains match info such as which team played home and away, detailed event data, player and team information. The category *Team Match Statistics Data Points* includes high-level match summary data like ball possession, corner kicks, cards, shots off and on goal, and offsides. The category *Play by Play Info Data Points* provides detailed event data,

Table 4.2: Sportradar API - Match timeline endpoint (some parts wrere left out). [Sportradar, 2018b]

Boxscore Data Points:	Clock Stoppage Time Match Score Period Score Period Number Period Type Aggregate Score	Aggregate Winner Id Period Match Status Status Winner Id
Team Info Data Points:	Team Abbreviation Team Country Team Country Code	Team Id Team Name Team Qualifier (home/away)
Player Info Data Points:	Player Id	Player Name
Play by Play Info Data Points:	Match Score Ball Location - Coordinates Event Coordinates Location Event Id	Event Period Event Time Event Team Event Type
Play Details Data Points:	Assisting Player Goal Scorer Goal Scorer Method	Substitution Player In Substitution Player Out
Player Match Statistics Data Points:	Assists Chances Created Crosses Duels Header Duels Sprint Duels Tackle Fouls Committed Goal Line Clearances Goals by Head Goals by Penalty Goals Conceded Goals Scored Interceptions Minutes Played Offsides Own Goals Passes Long	Passes Medium Passes Short Penalties Faced Penalties Missed Penalties Saved Performance Score Red Cards Shots Blocked Shots Faced Saved Shots Faced Total Shots Off Goal Shots On Goal Substituted In Substituted Out Was Fouled Yellow Cards Yellow Red Cards
Team Match Statistics Data Points:	Ball Possession Corner Kicks Fouls Free Kicks Goal Kicks Offsides	Penalties Missed Red Cards Shots Off Target Shots On Target Shots Saved Throw Ins

including the event type, the field coordinates of the event, match time and player involved the event. This API call can be called for historic matches, but also for live matches. For live matches, each event gets added as they happen. In Section 3.3, we saw that Pettersson and Nyquist fed event data into an RNN to predict the outcome of a ongoing match. These

event data were limited to goals, cards, substitutions and penalties. Including other event data available from this Sportradar API call, such as free kicks, corners, and shots on and off target will hopefully improve the accuracy of the predictions.

The category *Player Match Statistics Data Points* contains data on the individual players from the match, which could be useful. Their number of successful header duels, sprint duels won, successful tackles and successful passes are recorded. Comparing these data for a current game with their historic data could give an indication that a player is performing unusually well, or that he might perform below par.

As the match timeline is updated as events happens, the information from a current match can be used when trying to predict the outcome on the rest of it. One could have believed that there would be a considerable delay from the occurrence of an event to when it is available from an API, but this is not the case with the Sportradar API. Sportradar also provides match visualization tools, and claims their delays is of less than a second from when an event happens to when it appears on the visualization tools [Sportradar, 2018a]. The visualizations are based on data fed from the API, meaning the data retrieved from the API (e.g. the *Match Timeline*) is as up to date as possible. Figure 4.2 shows how the first goal in the 2018/2019 PL (penalty set by Paul Pogba in Manchester United - Leicester City) is presented by Sportradar with the *Match Timeline* API call.


```
{
    "id": 447785902,
    "type": "score_change",
    "time": "2018-08-10T19:02:38+00:00",
    "match_time": 3,
    "match_clock": "2:28",
    "team": "home",
    "x": 88,
    "y": 48,
    "period": 1,
    "period_type": "regular_period",
    "home_score": 1,
    "away_score": 0,
    "goal_scorer": {
        "id": "sr:player:111802",
        "name": "Pogba, Paul",
        "method": "penalty"
    }
},
```


Figure 4.2: Sportradar API - Match timeline response. [Sportradar, 2018b]

From the two tables (Table 4.1 and 4.2), we can see that there are a lot of different possible features available. In addition to the features discussed above, that has been used by the models in Chapter 3, some of the other features might prove to be useful in a model as well. This is just from two of Sportradar's API calls. Even though the data that has been used by the the models discussed in Chapter 3 are mainly found in these two calls, there are still other API calls that could contain valuable information. For instance, the *Player*

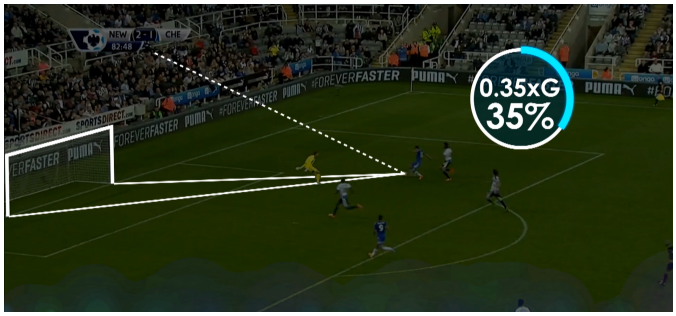
Profile contains information about the players position. If this position does not match with the position he plays a given match, could that have an effect on the outcome of a match? With all the information available at Sportradar, many combination of features can and should be tested in order to see which ones has predictive abilities.

4.2 Understat - Expected goals

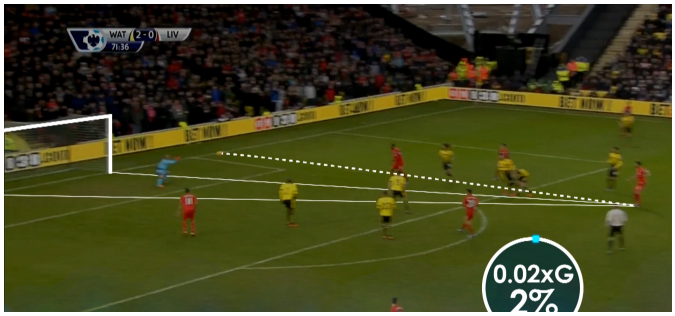
In addition to the features that are available from Sportsradar, there are also other features that could give a good indication on how a game will play out. One of these is the expected goals metric.

Expected goals (xG) is a football metric that gives an indication of what the chance for a shot to hit the back of the net is. Statistics for xG are available from many sources online, but they are not always the same as there are some differences in the models used to calculate them [Cronin, 2017b]. E.g. Opta, a big sports analytics company, uses a variable themselves defined as "big chance", which is not easy to quantify, but defined as a situation where a player should be expected to score [Opta, 2018]. Calculating the xG is naturally based on several other, more common variables as well. Both the angle and distance the shot was taken from, in addition to how many players there are between the position the shot was taken from and the goal, will have an impact on the xG score for that shot. What kind of assist led to the shot will also affect the xG score, whether it was a cross or a short pass right inside the penalty area. If there was a header or a volley following a cross, that would result in different xG scores. Figure 4.3 shows two different shot situations and their corresponding xG score. In the situation shown Figure 4.3b, the shot is taken from a greater distance, and with a greater number of players between the shot position and the goal, compared to the shot in 4.3a.

How the xG scores are calculated based on the variables mentioned above is not thoroughly explained by Opta. Understat is also a provider of an xG metric. They do not specify which parameters they are using, but simply tells that they have used a trained NN with over 100.000 shots and over 10 parameters for each shot [Understat, 2018d]. Nevertheless, Understat has xG scores, for both individual players and each team, available at their websites, making it possible to use this metric in a model without necessarily knowing how the metric is calculated. The xG value gives a more reliable picture on how a match actually played out and how strong a team is. Opta [2017] highlights the Italian team Juventus' 2015-2016 season as a prime example. Juventus started this season rather poorly, compared to their usual standard, with only 3 victories in the first ten matches. Looking at the xG values for Juventus and their opponent for each of these matches, showed that Juventus had a higher xG value than all of their opponents in each of the first ten matches. They scored 11 goals, when the xG value showed that they should have scored 19. In addition, they conceded nine goals, when this number should be five. Looking at Figure 4.4a, Juventus was at this point placed 12^{th} in the Italian league, a league they would usually be a contender for the top position, even though they had the highest xG value and the lowest xGA value(combined xG for the opposing teams they have faced). The xG values showed that Juventus' performance was rather good this season as well, compared to the actual results. After a couple more matches, Juventus started to both score and concede goals that better reflected their performance according to the xG values, and after ten more



(a)



(b)

Figure 4.3: Two different shot situations. One (a) from a good position with 0.35xG, and one (b) from a worse position with only 0.02xG. The latter indicates that this shot will go in only once for every 50 shots attempted. [Opta, 2017]

matches they had climbed to the second spot on the table. From Figure 4.4b, we can see that they actually scored more goals than one could expect according to their xG value. Juventus ultimately went on and won the Italian league this season.

Boiled down to a single match, xG can give a good indication on how well two teams plays against each other. Imagine a match where the half-time score is 1-0, but the xG score for the same match are actually 0.65-3.43. This indicates that the leading team currently has had a very good pay off on the few chances they had, while the loosing team has higher amount of chances without any of them resulting in a goal. If the teams keep up the same frequent of chances in the second half, one could expect that the loosing team would get an equalizer, at the very least.

The xG metric could also be used to assess a team's expected number of goals prior to match. Understat keeps track of the xG value for every player on each team during a season, along with their number of minutes on the field. With this, an xG for a player per 90 minutes (xG90) can easily calculated. Adding the xG90 value for each player on the line-up for two facing teams can be useful when trying to predict the outcome of a game.

Additionally, Understat provides data about the formations played by each team, along with the xG values for each formation. The xG value and xG against (xGA, xG for the opposing teams) from both teams are being kept track of. Figure 4.5 shows the xG values

№	Team	M	W	D	L	G	GA	PTS	xG	xGA	xPTS
1	Roma	10	7	2	1	25	12	23	17.29 -2.71	8.46 -3.54	20.21 -2.79
2	Napoli	10	6	3	1	21	8	21	14.95 -6.05	8.82 +0.82	18.19 -2.81
3	Fiorentina	10	7	0	3	18	8	21	15.83 -2.17	8.49 +0.49	18.83 -2.17
4	Inter	10	6	3	1	10	7	21	14.78 +4.78	11.35 +4.35	16.57 +4.43
5	Sassuolo	10	5	3	2	13	10	18	9.17 -3.83	9.40 +0.60	13.49 -4.51
6	Lazio	10	6	0	4	15	15	18	13.63 -1.37	10.02 -4.98	16.24 -1.76
7	Atalanta	10	5	2	3	13	11	17	11.74 -1.26	12.99 +1.99	13.78 -3.22
8	AC Milan	10	5	1	4	12	15	16	9.98 -2.02	10.32 -4.68	13.50 -2.50
9	Sampdoria	10	4	3	3	18	14	15	13.04 -4.96	15.23 +1.23	12.27 -2.73
10	Torino	10	4	3	3	16	15	15	10.25 -5.75	9.52 -5.48	13.21 -1.79
11	Chievo	10	3	3	4	13	10	12	9.90 -3.10	12.31 -2.31	11.87 -0.13
12	Juventus	10	3	3	4	11	9	12	18.39 +7.39	5.01 -3.99	21.94 +8.94

(a)

№	Team	M	W	D	L	G	GA	PTS	xG	xGA	xPTS
1	Napoli	20	13	5	2	41	16	44	33.39 -7.61	15.34 -0.66	39.72 -4.28
2	Juventus	20	13	3	4	37	15	42	33.70 -3.30	10.94 -4.06	42.87 +0.87
3	Inter	20	12	4	4	25	13	40	27.12 +2.12	23.22 +10.22	29.94 -10.06
4	Fiorentina	20	12	2	6	37	21	38	31.07 -6.93	16.99 -4.01	37.12 -0.88
5	Roma	20	9	8	3	37	23	35	33.17 -3.83	21.27 -1.73	35.91 +0.91
6	AC Milan	21	9	6	6	29	25	33	28.12 -3.88	21.07 -3.93	32.80 -0.20
7	Sassuolo	20	8	8	4	25	21	32	24.06 -5.94	20.51 -0.49	29.75 -2.25
8	Empoli	21	9	5	7	27	26	32	20.34 -8.66	24.71 -1.29	24.74 -7.26
9	Lazio	20	8	4	8	25	29	28	23.75 -1.25	18.62 -10.38	31.29 +3.29
10	Chievo	20	7	6	7	26	22	27	21.49 -4.51	23.95 +1.95	25.16 -1.84
11	Torino	20	7	5	8	27	26	26	23.95 -3.05	19.48 -6.52	29.85 +3.85
12	Atalanta	21	7	5	9	21	24	26	23.39 +2.38	28.22 +4.22	27.21 +1.21

(b)

Figure 4.4: The table for the Italian league after ten (a) and 20 (b) rounds for the 2015/2016 season. The expected results according the the xG values shows that Juventus' performance was better than the actual results. The results bettered gradually and better reflected their xG values. Juventus won the league this season. [Understat, 2018b]

for the different formation played by Fulham in the first 12 rounds of the current season of PL. This shows that the 4-3-3 formation is by far the most played formation by Fulham, and naturally this formation has a higher xG value than the second most played formation. Looking at the xG values per 90 minutes shows that while the xG is the same for both formations, the xGA is much lower for the second most played formation. This could be useful information for both Fulham, who is at the bottom of the PL table after the first 12 rounds, and for models trying to predict matches.

Unfortunately, the xG values are only available for finished matches at Understat, and not for an ongoing game. However, Understat provides xG values for each 15 minutes period of a match. Figure 4.6 shows data for Liverpool after the first 13 rounds in the 2018-2019 season. From this we can see that Liverpool, by far, has the highest xG value in the last 15 minutes of the match. This is also the time period they have their highest xGA value. Using data from each of the intervals could be useful when predicting an ongoing game.

Currently, a search in Google Scholar retrieves no documented work that have used the xG metrics to predict football matches. However, xG is a well known metric that is more

№	Formation	Min	Sh	G	ShA	GA	xG	xGA	xGD	xG90	xGA90
1	4-3-3	628	79	9	122	15	7.00 -2.00	16.82 +1.82	-9.82	1.00	2.41
2	4-2-3-1	268	37	1	43	8	2.99 +1.99	4.57 -3.43	-1.58	1.00	1.53
3	4-4-1-1	69	5	0	17	2	0.53 +0.53	2.09 +0.09	-1.56	0.69	2.73
4	3-4-3	56	15	1	6	2	0.88 -0.12	0.71 -1.29	0.17	1.42	1.14
5	4-2-2-2	46	5	0	3	0	0.19 +0.19	0.13 +0.13	0.06	0.37	0.26
6	4-1-4-1	20	0	0	2	1	0.00	0.23 -0.77	-0.23	0.00	1.02
7	4-4-2	20	3	0	1	1	0.06 +0.06	0.37 -0.63	-0.31	0.29	1.68
8	3-4-2-1	17	2	0	1	1	0.06 +0.06	0.52 -0.48	-0.47	0.30	2.78

Figure 4.5: Fulham's xG data for different formations played in the 2018/2019 season. The data is from the first 12 rounds. [Understat, 2018a]

№	Timing	Sh	G	ShA	GA	xG	xGA	xGD	xG/Sh	xGA/Sh
1	1-15	27	2	8	0	3.86 +1.86	1.29 +1.29	2.58	0.14	0.16
2	16-30	31	4	21	1	4.24 +0.24	1.51 +0.51	2.73	0.14	0.07
3	31-45	30	4	17	0	3.93 -0.07	1.12 +1.12	2.81	0.13	0.07
4	46-60	31	6	14	0	4.61 -1.39	1.14 +1.14	3.47	0.15	0.08
5	61-75	21	3	19	1	2.32 -0.68	1.69 +0.69	0.63	0.11	0.09
6	76+	48	6	26	3	6.49 +0.49	3.61 +0.61	2.88	0.14	0.14

Figure 4.6: Liverpool's xG data for different time intervals in the 2018/2019 season. The data is from the first 13 rounds. [Understat, 2018c]

and more used in football analysis and betting [Toby, 2017]. There are even companies that offers betting courses, where the xG metric is an important factor [Orio Sports, 2018]. Orio has posted some of their customers' feedbacks on their website, where one of them claimed he hit 500 from a bankroll of 50. He had an algorithm for making money in sports betting, where one of the things missing from his algorithm was the concept of xG.

4.3 WhoScored

A lot of data is available from either Sportradar or Understat, but there are some potentially useful information that is not provided by these two providers. This has to be gathered from other sources. WhoScored has detailed information on the big leagues, naturally including the PL. WhoScored provides some interesting player data that is not available from the two other sources. The most interesting one is their player ratings. After a match, players are evaluated based on their performance. This is something that is common to see in newspapers and other sources that writes about matches, where the players are evaluated by some experts. Finding articles for all the matches of interest in order to get the player ratings is a rather time consuming job. However, at WhoScored, each player that has appeared in a match this season has their own rating. Their ratings for every match they have played are available, but also the average for each tournament in the current season (e.g 2018/2019 PL) is possible to gather from WhoScored's website [WhoScored, 2018a]. Figure 4.7 displays the top 10 players in the PL this season (after the first 12 rounds), according to their average match ratings. Using this information can give an indication on how strong a starting line up is.

From Bradley [2018], we have already seen that using player ratings has resulted in

good accurate predictions. Using ratings based on how players have performed the current season as an input in a method, combined with the formation and the players' positions from Sportradar, might be a good alternative to Bradley's approach.

In addition to providing the ratings based on historic performance, player ratings for an ongoing match are also available at WhoScored. These ratings are naturally dynamically adjusted based on the player performances. If a player's rating has a big variance compared to their average rating, this should probably be taken into consideration when trying to predict the outcome of a match. Figure 4.8 shows the player ratings for the match between Manchester City and Bournemouth (played on December 1st, 2018).

Premier League Player Statistics

Summary | Defensive | Offensive | Passing | Detailed

View: Overall Home Away Filter: Minimum apps All players

R	Player	Apps	Mins	Goals	Assists	Yel	Red	SpG	PS%	AerialsWon	MotM	Rating
1	Eden Hazard Chelsea, 27, M(CLR),FW	8(3)	783	7	4	1	-	2.7	86.3	1	6	7.95
2	Raheem Sterling Manchester City, 23, M(CLR),FW	10	862	6	5	1	-	2.7	87	0.3	4	7.90
3	David Silva Manchester City, 32, M(CLR)	10	837	4	2	1	-	2.3	88.5	0.4	1	7.78
4	Sergio Agüero Manchester City, 30, AM(CL),FW	12	884	8	4	2	-	4.5	87.5	0.3	1	7.76
5	Mohamed Salah Liverpool, 26, AM(CLR),FW	12	1034	6	3	-	-	4	75.8	0.5	1	7.56
6	Fernandinho Manchester City, 33, D(R),DMC	12	1053	1	2	2	-	1.3	88.3	2.4	2	7.48
7	Sadio Mané Liverpool, 26, AM(CLR)	11	972	6	-	1	-	2.5	78.5	1	4	7.44
8	Bernardo Silva Manchester City, 24, AM(CLR)	11(1)	962	3	3	2	-	1.7	85.4	0.1	1	7.44
9	Hugo Lloris Tottenham, 31, GK	8	720	-	-	-	-	-	69.9	0.3	1	7.39
10	Gylfi Sigurdsson Everton, 29, M(CLR),FW	12	1005	5	2	1	-	2.4	78	0.3	2	7.38

Figure 4.7: WhoScored's player ratings. The table displays the top ten players in the Premier League this season after the first 12 rounds, according to their average player ratings. Only players more appearances than the average number of appearances is displayed, but information for all players is available. [WhoScored, 2018a]

Figure 4.8: WhoScored's player ratings. The figure displays the players' rating after 86 minutes in the match between Manchester City and Bournemouth. [WhoScored, 2018b]

Chapter 5

Future work

Based on previous work in the domain of predicting football matches, we have a good basis for creating our own predictive model. Our focus is to predict live matches, and unfortunately most of the previous work discussed in Chapter 3 is based on predicting matches prior the kick off. We have seen that both statistical and machine learning approaches has been made, and we will make suggestions to a couple of models for predicting live matches.

5.1 Models

After the data is gathered, the model that is going to use the data needs to be built. The use of BNs has given good results, where the pi-football from Constantinou et al. [2012] achieved an incredible profit of 9.48% based on the prediction made with his BN. Joseph et al. [2006] achieved an overall prediction accuracy of 59.21% with his expert constructed BN. Even though not all BNs are dependent on experts, both these networks are heavily dependent on domain knowledge. This require insight we must admit we do not possess. We will therefor focus on the use of NN.

5.1.1 Neural network

Bradley [2018] achieved an ROI of 11% with his NN, using team formation and player ratings as features. Instead of using the player ratings from the video game FIFA, player ratings from WhoScored can be used. These ratings are updated after each match, and should better reflect player's current level than the FIFA ratings which are released once a year. Bradley's NN using this type of information has shown good promise, but additional data could be included. The model can also be adapted into predicting each 15 minutes period in a match, in which bookmakers offers different bets.

We have seen that a popular approach when predicting matches is with the calculation of the teams' attacking and defending strengths. These values are basically based on the goals they have scored and conceded, and the strengths are also specified for home

and away games. As we saw in Section 4.1, Sportradar provides the goals scored and conceded data for each 15 minutes intervals in a match. With this data, we can calculate teams attacking and defencing strengths for each corresponding 15 minutes interval. Also Understat provides xG data for each 15 minutes period of a match. With these data we can see that some teams tends to score at different periods of the match.

Combining these data and feeding it into a NN to predict a 15 minutes period should give better results compared to using only player ratings and over strength values. With bookmakers offering bets on 15 minutes periods, this could be important. Naturally, different network architectures needs to be tested in order to find the optimal one. Additionally, match events up to the start of the period could also be used as features in the NN. The event data should give an indication on the trend in the match, and should therefor be taken into consideration when making a prediction.

Alternatively, the teams ELO ratings could also be used as a metric for the teams strength. However, this does not update during a match, but could still be used in a combination with event data.

5.1.2 Recurrent neural network

Instead of predicting each of the 15 minutes period of a match, one could feed events into a model as they happen to generate new predictions during a live game. Starting of with an initial probability distribution for the match outcome, that could be derived from bookmakers' odds, where match events could be used to update these probabilities. Pettersson and Nyquist [2017] used an RNN to predict live matches. They fed event data into their network, but these were limited to goals, cards, substitutions and penalties. From the data available from Sportradar (see Section 4.1), a lot of other events are also recorded.

This will be a good starting point when building another model. Building a similar RNN, but using other events as well. We have seen from Kerr's work how some features impact the match result. The top features were shots on target, which team played home and the number of crosses. Feeding this data into an RNN, in addition to the features already used by Pettersson and Nyquist, might improve the live predictions.

The goals scored and conceded for each 15 minutes interval from Sportradar, and the xG data from Understat for the same intervals, could also be included in this model. This data can be used to calculate the expected number of goals in the remaining minutes of a match, which can then be fed into the RNN. Hopefully, this will give good predictions on who will win the rest of the match. Additionally, this could be used for bets regarding how many goals are to be scored in the remaining minutes of a match.

5.2 Data selection

Selecting features from the data is probably the most important and difficult aspect of the future work. From Chapter 3 we have seen that different sets of features have been used in the different models. What works well with our model might not conform with what has worked well with other models. We will have to try different set of features to find the optimal one. With all the features available from Sportradar and the other sources, we should test different kinds of combinations, even features not mentioned in Chapter 3.

We will first run with features used by similar models that has showed good results, then extend the model with more features to see how the predictions are affected. We will leave no stone unturned.

Larsson discovered a phenomenon in the popular game Dota 2, where game characters were designed to perform better earlier and worse later compared to other characters. A similar phenomenon can be apparent in football. Football teams might have higher performance during the early stages in the match while other teams can have their strongest period closer to the end. It might be important to use NN for our future model if we want to take this phenomenon into consideration, as it was the model that was able to discover the phenomenon in Larsson [2018]. For instance, the effect of a red card stated by Boice [2018], where a red card is worth three times more than a home-field advantage, might be such an aspect where the effect is dependent on when the card occurs in the match. Hirotsu and Wright [2003] also had some interesting findings: events during the game has different effect based on the team. To implement this into the model, we have to we have to find how much a particular event impacts a team and and map each possible event to every team in the PL.

How many season of historic data that is going to be used when training a model also needs to be considered. Sportradar, Understat and WhoScored have data available from several seasons. Understat provides data for the fewest season, but still has data available for the five most recent seasons. The models discussed in Chapter 3 had a varying number of seasons that were used to train the models. The model pi-football [Constantinou et al., 2012] used historical data from 1993 to 2010 in order to predict matches in the PL season 2010-2011. On the other hand, Bradley's model were based on player ratings from the video game FIFA, which reflect the players' current level. As the data is available, we will test our future model with both data from various seasons and from the current season. A possibility is to weigh recent seasons with larger magnitude than earlier ones, as the teams tends to change players quite often. E.g. did Mohamed Salah's arrival at Liverpool prior to the 2017-2018 have huge positive effect for the team's performance [Howarth, 2018].

Even though the 2018 version of Liverpool differs from the Liverpool in 2013, does not mean that the data from 2013 is outdated and irrelevant. The data from the earlier seasons can have valuable information about how a match plays out. How a match between two teams with strengths X and Y will play out, can be derived from historic matches where the strength difference and other features is similar. For instance, one could see from earlier matches that if the home team after 70 minutes has had a ball possession of 70%, over 10 shots on goal, over 10 crosses and with the score 0-0 they are likely to score in the last 20 minutes.

Chapter 6

Summary

Throughout our study, we have tried to find answers to the research questions listed in Section 1.3:

1. What models can be used to predict match outcome?

   There has been created different models with the purpose of predicting the outcome of a football match. Maher was one of the first with his model based on calculation of team strength, while in more recent times there has been attempts with both Bayesian networks and neural networks. As the Bayesian networks studied in this project were heavily dependent on expert knowledge, we will set our focus on the use of neural networks, which has been showing good promise.

2. Which features are important for the outcome of a match?

   Different features have been used by the different models studied in this project. Team strengths, based on goals scored and conceded, have been used to make predictions with success. Home team advantage is considered an important factor. A study revealed that the difference in number of crosses also had a impact on the end results. Using players' ratings and team formation has also given good results when used in a neural network. We have also seen that different features might affect teams differently.

3. How can event data be used to update match predictions in real-time?

   In Section 3.3, we discussed a model that used event data with a recurrent neural network to predict live matches. However, these event data were limited to goals, cards, substitutions and penalties. As we have a lot of other event data available as well, which has shown to have an impact on the outcome of a match, we believe that feeding this additional data into a network would give better live predictions.

In this project have we studied some of the work done in the field of predicting the outcome of football matches. With football being one of the world's most popular sports and also topping the list over the most betted sports, a lot of attempts has been made

in order to create better match-predicting models, also with the use of machine learning techniques. We have looked at some of the models that has been used, and the features these models has been utilizing. With data we have available, provided Sportradar, we have gotten some ideas on how to further improve football match predictions. We have a great faith in that a neural network, combined with the correct features, can deliver accurate predictions of the outcome of a football match.

Bibliography

Christopher Ammon. How I use Poisson Distribution to successfully predict winning soccer outcomes, December 2016. URL http://fspinvest.co.za/articles/strategies/how-i-use-poisson-distribution-to-successfully-predict-winning-soccer-outcomes-7331.html. Accessed: 2018-09-12.

Jay Boice. How Our 2018 World Cup Predictions Work, June 2018. URL https://fivethirtyeight.com/features/how-our-2018-world-cup-predictions-work/. Accessed: 2018-10-25.

Bradley. Predicting Football Matches using EA Player Ratings and Tensorflow, July 2018. URL https://towardsdatascience.com/predicting-premier-league-odds-from-ea-player-bfdb52597392. Accessed: 2018-11-08.

E Bruce Brooks. Statistics - The Poisson Distribution, 2007. URL https://www.umass.edu/wsp/resources/poisson/. Accessed: 2018-12-01.

Kelly Burch. How to Calculate Hamming Distance, October 2018. URL https://sciencing.com/how-to-calculate-hamming-distance-12751770.html. Accessed: 2018-11-06.

Anthony C. Constantinou. Dolores: a model that predicts football match outcomes from all over the world. *Machine Learning*, pages 1–27, 5 2018. ISSN 0885-6125. doi: 10.1007/s10994-018-5703-7.

Anthony C Constantinou, Norman E Fenton, and Martin Neil. pi-football: A Bayesian network model for forecasting Association Football match outcomes. *Knowledge-Based Systems*, 36:322–339, 2012.

Benjamin Cronin. Poisson Distribution: Predict the score in soccer betting, April 2017a. URL https://www.pinnacle.com/en/betting-articles/Soccer/how-to-calculate-poisson-distribution/MD62MLXUMKMXZ6A8. Accessed: 2018-09-13.

Benjamin Cronin. An analysis of different expected goals models, November 2017b. URL https://www.pinnacle.com/en/betting-articles/Soccer/expected-goals-model-analysis/MEP2N9VMG5CTW99D. Accessed: 2018-09-24.

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

Arpad E Elo. *The rating of chessplayers, past and present*. Arco Pub., 1978.

Edward S Epstein. A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, 8(6):985–987, 1969.

Dean Evans. Engineering Football Excellence With Sensors, Stats And Data Analytics, 2018. URL https://www.intel.co.uk/content/www/uk/en/it-management/cloud-analytic-hub/data-powered-football.html. Accessed: 2018-10-30.

FiveThirtyEight. 2018 World Cup Predictions, 2018. URL https://projects.fivethirtyeight.com/2018-world-cup-predictions/?ex_cid=endlink. Accessed: 2018-11-06.

David Forrest, John Goddard, and Robert Simmons. Odds-setters as forecasters: The case of English football. *International journal of forecasting*, 21(3):551–564, 2005.

Amit Kumar Gupta and Yash Pal Singh. Analysis of Hamming network and MAXNET of neural network method in the string recognition. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pages 38–42. IEEE, 2011.

Nobuyoshi Hirotsu and Mike Wright. An evaluation of characteristics of teams in association football by using a Markov process model. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(4):591–602, 2003.

Matthew Howarth. What records has Salah set in 2017/18?, April 2018. URL https://www.uefa.com/uefachampionsleague/news/newsid=2554423.html. Accessed: 2018-12-01.

Lars Magnus Hvattum and Halvard Arntzen. Using ELO ratings for match result prediction in association football. *International Journal of forecasting*, 26(3):460–470, 2010.

Anito Joseph, Norman E Fenton, and Martin Neil. Predicting football results using Bayesian nets and other machine learning techniques. *Knowledge-Based Systems*, 19 (7):544–553, 2006.

Joshua Kahn. Neural network prediction of NFL football games. *World Wide Web electronic publication*, pages 9–15, 2003.

Matthew George Soeryadjaya Kerr. *Applying machine learning to event data in soccer*. PhD thesis, Massachusetts Institute of Technology, 2015.

Dan Kopf. Data analytics have made the NBA unrecognizable, October 2017. URL https://qz.com/1104922/data-analytics-have-revolutionized-the-nba/. Accessed: 2018-10-16.

Sam Koslowsky, Senior Analytic Consultant, and Harte Hanks. On Variable Importance in Logistic Regression, August 2018. URL https://www.predictiveanalyticsworld.com/patimes/on-variable-importance-in-logistic-regression/9649/. Accessed: 2018-12-11.

Marcus Larsson. Enhancing the value proposition of live esports consumption with AI technology, 2018.

Patrick Lucey, Alina Bialkowski, Mathew Monfort, Peter Carr, and Iain Matthews. quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data. In *Proc. 8th annual mit sloan sports analytics conference*, pages 1–9, 2014.

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

Matchbook. Live Betting Overview, 2018. URL https://www.matchbook.com/page/getting_started/live-bet-overview/. Accessed: 2018-11-10.

Andrew Meola. How eSports has given rise to competitive gaming betting and gambling with skins and real money, 2018. URL https://www.businessinsider.com/the-rise-of-esports-betting-and-gambling-2018-1?r=US&IR=T&IR=T. Accessed: 2018-10-31.

Michael Joseph Moroney. *Facts from figures*. Penguin books, 3 edition, 1951. Revised 1962.

Anuja Nagpal. L1 and L2 Regularization Methods, 2017. URL https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c. Accessed: 2018-12-01.

NordicBet. NORDICBET SETTER ODDS PÅALLE LANDETS FOTBALLSERIER, Mars 2017. URL https://www.nordicbet.com/no/blogg/fotball/nordicbet-setter-odds-p-alle-landets-fotballserier/. Accessed: 2018-10-10.

Karol Odachowski and Jacek Grekow. Using bookmaker odds to predict the final result of football matches. *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 196–205, 2012.

Opta. Opta Expected Goals, 2017. URL https://www.youtube.com/watch?time_continue=96&v=w7zPZsLGK18. Accessed: 2018-10-19.

Opta. Advanced Metrics - Expected goals (xG), September 2018. URL https://www.optasports.com/news/optas-event-definitions/. Accessed: 2018-10-19.

Orio Sports. The Football Betting Masterclass: How to win at football betting, 2018. URL https://courses.oriosports.com/optin-21985780. Accessed: 2018-11-12.

Daniel Pettersson and Robert Nyquist. Football Match Prediction Using Deep Learning, 2017. URL http://publications.lib.chalmers.se/records/fulltext/250411/250411.pdf.

Michael C Purucker. Neural network quarterbacking. *IEEE Potentials*, 15(3):9–15, 1996.

Charles Reep, Richard Pollard, and Bernard Benjamin. Skill and chance in ball games. *Journal of the Royal Statistical Society. Series A (General)*, pages 623–629, 1971.

Stuart Russell and Peter Norvig. Artificial intelligence-a modern approach 3rd ed, 2016.

Piotr Skalski. Preventing Deep Neural Network from Overfitting, 2018. URL https://towardsdatascience.com/preventing-deep-neural-network-from-overfitting-953458db800a. Accessed: 2018-12-01.

Adam Smith. Sky Sports bust common football myths: Home advantage?, 2017. URL https://www.skysports.com/football/news/11096/10955089/sky-sports-bust-common-football-myths-home-advantage. Accessed: 2018-11-08.

Sportradar. Sportradar - Data Services, 2018a. URL https://www.sportradar.com/rights-holder-solutions/data-services/. Accessed: 2018-11-10.

Sportradar. Soccer Extendend v3, 2018b. URL https://developer.sportradar.com/docs/read/football_soccer/Soccer_Extended_v3#soccer-extended-v3-api-map. Accessed: 2018-11-08.

Steam. Dota 2 player count, 2018. URL https://steamcharts.com/app/570. Accessed: 2018-10-31.

Toby. How Expected Goals (xG) Will Change The Way We Bet On Football, 2017. URL https://punter2pro.com/expected-goals-xg-football-betting-analysis/. Accessed: 2018-11-05.

UEFA. Protecting the game, October 2018. URL https://www.uefa.com/insideuefa/protecting-the-game/integrity/index.html?redirectFromOrg=true. Accessed: 2018-10-10.

Understat. Fulham xG stats for the 2018/2019 season, 2018a. URL https://understat.com/team/Fulham/2018. Accessed: 2018-11-14.

Understat. Serie A xG Table and Scores for the 2015/2016 season — Understat.com, 2018b. URL https://understat.com/league/Serie_A/2015. Accessed: 2018-10-14.

Understat. Liverpool xG stats for the 2018/2019 season, 2018c. URL https://understat.com/team/Liverpool/2018. Accessed: 2018-12-01.

Understat. Understat, 2018d. URL https://understat.com/. Accessed: 2018-11-10.

Jen van Lier. The Most Popular Sports to Bet on Today, May 2018. URL https://bitcoinchaser.com/bitcoin-sportsbook/most-popular-sports-to-bet-on-today. Accessed: 2018-10-12.

WhoScored. Premier League Player Statistics, 2018a. URL https://no.whoscored.com/Regions/252/Tournaments/2/Seasons/7361/Stages/16368/PlayerStatistics/England-Premier-League-2018-2019. Accessed: 2018-11-14.

WhoScored. WhoScored - Manchester City 3-1 Bournemouth, 2018b. URL https://no.whoscored.com/Matches/1285008/Live/England-Premier-League-2018-2019-Manchester-City-Bournemouth. Accessed: 2018-12-01.