Department of Computer Science

Submitted in part fulfilment for the degree of BEng.

Predicting the Outcome of Football Matches Using Machine Learning

Tom Loomes

28 April 2025

Supervisor: Tarique Anwar

Contents

List of Figures

List of Tables

Statement of Ethics

The Department of Computer Science at the University of York outlines three main ethical principles that any project should adhere to. This project is mindful of the following:

- Avoidance of harm - This project will not cause any direct physical harm due to no human participants and no environmental harm due the absence of physical components. This project could be construed as promoting gambling, but the mention of bookmakers and gambling is purely for academic comparison. A person's choice to gamble is their own and those who may have a gambling disorder should be advised on the contents of this report.
- Informed Consent - There are no human participants involved in this project so this is not relevant.
- Data Protection - The data used in this project is freely available, containing historical football match statistics so there is no requirement for any form of data protection.

Executive Summary

The aim of this project was to create a machine learning model capable of predicting football matches based on historical match data. The primary motivation was to enable more accurate predictions than those provided by bookmakers, potentially offering bettors a competitive edge. Additional objectives included identifying which match statistics have the most influence results and assessing the predictability of different football leagues.

Research into proposed solutions to similar problems showed the effectiveness of machine learning methods in predicting match outcomes. But this often came at the cost of an imperfect set of predictions, with many omitting the ability to predict draws from their research and using outdated datasets.

Historical match data was collected in the form of `.csv` files, before combining the data from these sources and cleaning the resulting dataset. The match statistics from this dataset were then analysed, for the strength of their correlation and influence in predicting the outcome of a football match, with poorly performing statistics removed. Further feature engineering was then used to enhance the predictive quality of these metrics, creating streak-based and net statistics for a game-by-game insight into a team's performance. A variation of the ELO rating system was used to help create an additional measure of a team's strength, incorporating the margin of victory or loss to provide a more accurate appraisal of a team's quality for the model to learn from.

The devised model features a LSTM to capture temporal dependencies on individual match data for a team, a MLP to handle the aforementioned team ratings and the use of the XGBoost algorithm to create predictions from streak-based features from each team. The outputs from each of these methods are then concatenated to provide a final prediction, which is then evaluated against both validation and test datasets.

The hyperparameters for the three involved machine learning methods were optimised through the use of a grid search algorithm, applying a brute-force approach to discover the best combination of hyperparameters. The application of these optimal hyperparameters resulted in a peak predictive accuracy of 52.68%, showing a capable level of predictive power. This

level of accuracy is comparable with major bookmakers Bet365 and William Hill who, based on their odds favourite, attained a predictive accuracy of 54.24% and 52.90% respectively. This shows that the model offers a competitive predictive capability and can provide meaningful insights for bettors.

1 Introduction

1.1 Background and Context

Football is the most popular sport in the world with an estimated 3.5 billion fans worldwide [1]. As a consequence of its widespread popularity, results are of great interest to many key stakeholders, some of which include: players, managers, fans, and gamblers. Each party does their best to predict the outcome of a game - from tactical planning, match preparation, fan engagement and the potential financial rewards of correct predictions.

In the UK alone, the betting industry was worth £15.1 billion between 2022 and 2023 [2] and is expected to grow by a compound annual growth rate of 5.36% for the next five years [3]. Of this, the football betting market accounts for approximately £1.1 billion [2], underscoring its importance to many punters who stand to benefit from accurate predictions.

As machine learning and data analytics continue to advance, their applications within the sport continue to grow. A leading example is the Premier League team Brighton and Hove Albion, who have built a reputation for signing "hidden gems", promising young players who may not have been identified with traditional scouting methods, but who are believed to have potential to be successful due to in-depth statistical analysis [4]. This is done through external data analysis firm Jamestown, who act as part of Brighton's recruitment team and perform statistical analysis to provide a list undervalued potential targets that they believe would fit the club's playing style and squad [4]. One example of this was Moises Caicedo. The now Ecuadorian international was signed from Independiente del Valle for around £4.5 million in 2021 at 19 years old [5] and was sold 2 years later, after only 45 league appearances, with his value soaring to a then British record transfer fee of £115 million with his move to Chelsea [5]. Beyond recruitment, Brighton utilised a data-driven approach under manager Graham Potter in the 2021/2022 season with regards to opposition analysis [6], discovering weaknesses that may not have been apparent with regular scouting. This approach allowed them to effectively compete against far better resourced teams [6], achieving a 9th placed finish in the Premier League in 2021/22. Their efficiency and success is highlighted with them having 5th lowest salary bill in the league and having the 3rd

lowest "cost-per-point" in the division [6] for the season.

Another well-documented example of a Premier League club using data to their benefit is Liverpool. Despite being a team with much loftier ambitions than Brighton, Liverpool still used data analysis techniques to find value for money in their recruitment. Recruitment analyst Dr Ian Graham was appointed as part of a recruitment team overhaul in April 2012 [7] under new owners Fenway Sports Group. His mandate was to identify players capable of performing to the level of Liverpool's expectations, without having to pay the inflated transfer fees often associated with elite-level players.

Dr Graham developed multiple models to gain insights and used statistics like Expected Goals (xG) and Expected Assists (xA) [8]. xG is the measure of the quality of a chance (between 0 and 1) based on factors such as the location of the shot, angle of the shot, and contextual elements like defensive pressure. Similarly, xA measures the likelihood that a given pass will result in a goal accounting for the quality of the pass and the quality of the following shot, providing a value between 0 and 1, showing the probability that the passing player receives an assist. Other statistics used in Dr Graham's models included pressures, where a player presses an opponent, distance covered, player availability, injury history, and positional data, such as the areas on the pitch occupied by a player during a match. Some of these metrics were prioritised in order to find the players that best fit the high-energy playing style under at-the-time manager Jurgen Klopp, with metrics like pressures and distance covered being of particular importance.

One example of the success of this strategy can be seen in the signing of Egyptian winger Mohammed Salah. He was signed from AS Roma in 2017 for a fee of £36.9 million [8], with "Graham's model highlighting Salah's exceptional off-the-ball movements and his expected goals, suggesting the fee was a savvy investment" [8]. In the six full seasons since then, Salah has enjoyed extraordinary individual success, breaking the Premier League goalscoring record to win the Golden Boot in his debut season and scoring a ridiculous 211 goals in 349 games [9] cementing himself as one of the best players in the Premier League.

As a collective, Liverpool have achieved a similar level of success, winning the English Top Flight for the first time in 30 years in 2019/20, their first in the Premier League era [10]. They also won their sixth Champions League the year before in 2018/19, the FA Cup in 2021/22, the EFL Cup in 2021/22 and 2023/24 as well as the Club World Cup in 2019 [11], representing what The Independent [10] calls a "glittering period for the club". Other key contributors to The Reds' achievements, identified by Dr Graham's models, included Sadio Mané, signed for £34 million in 2016, recognised by data analysis for his "ability to carve out goal-scoring opportunities and disrupt

defensive alignments" [8] and Roberto Firmino, a player who ranked highly for "his pressing metrics and link-up play" [8] both proved to be outstanding value for money during their time at the club.

Along with the emergence of xG and xA, an increasing number of values and models are being developed to quantify actions on the pitch. One stat which will become increasingly widespread is Expected Threat (xT). xT was invented by Karun Singh [12] in 2018 and measures how likely a player's actions, whether that be a pass, dribble, or shot, influence the chances of a goal being scored. Similarly to xG and xA, values are between 0 and 1 and are assigned to an individual action for a player. For example, a player who delivers a defence-splitting through ball that sets up a winger to square the ball for a tap-in would not be credited with an assist. However, xT recognises the significance of such actions by assigning them a high value, providing a more comprehensive measure of creativity and influence. This metric addresses the limitations of traditional statistics by capturing the meaningful contributions of players, enabling more accurate evaluations in data analysis.

Similar possession value models continue to be developed, such as On-Ball Value (OBV) by Statsbomb [13]. This metric not only measures the affect of an action on a players' team scoring, but also on not conceding, measuring the net affect of an action, and additionally removes some possession values to remove bias towards players that play on "stronger" teams, giving a more accurate representation of the influence of an action compared with xT. The further extensive adoption of these metrics and models by football clubs can allow them to identify the most effective players, both from their own and opposition teams, and could perhaps inform tactical decisions, in trying to get the best from their own, and limiting the oppositions' creative threats.

As more teams adopt similar strategies to Brighton and Liverpool across the sport, and more detailed metrics and models are able to quantify teams' and players' performances, the importance of data becomes further emphasised due to its increased ability to draw more accurate conclusions. Another application of these methods is within the betting industry. Historically, bookmakers set odds based on historical data, expert judgement, and market sentiment [14]. However, in the new age of AI and machine learning, betting companies are able to incorporate more dynamic factors like a team's line-up, weather conditions, player injuries, and fans' confidence [14] as these new data-led methods are able to compute vast quantities of data. This can give live, in-play updates to odds that more accurately reflect the probabilities of an outcome in a match.

1.2 Aims and Objectives

The goal of this report is to determine whether a machine learning model can outperform bookmakers in predicting football match outcomes. This will be done by creating a hybrid machine learning model that combines multiple machine learning techniques. The success of the model is primarily defined by analysing how often the model's forecasts align with actual results compared to how accurate the result considered the most probable by the bookmakers' odds is.

Other areas of interest for this project could include identifying the most influential factors in predicting match outcomes, and examining how predictive accuracy varies across different leagues. These insights would help determine which features are most impactful in forecasting results and whether the model's performance is consistent across diverse competitive environments and historical contexts.

Success in this project would mean creating a model capable of predicting football match outcomes with a level of accuracy similar to, or better than bookmakers. The applications for this could include: use for bettors as tips to help them "beat the bookies", another metric used by betting companies to set odds for games, for sports broadcasters to provide pre-match insight into the chances of a particular result, and potential for football clubs to adjust tactics and personnel selection based on the likelihood of a particular result in an upcoming match.

2 Literature Review

2.1 Machine Learning Methods

Machine learning has experienced rapid advances in recent years, driven by the increased availability of powerful computational resources. Defined by Murphy [15] as a set of methods that can automatically detect patterns in data, using the uncovered patterns to predict data or perform decision making, machine learning is now integral to everyday life, from being used to power search engines, social media, and use in diagnosing conditions in healthcare.

Machine learning methods are becoming increasingly widespread within the sport, with teams like Liverpool and Brighton using machine learning models to improve the quality of their player recruitment. This section focusses on the machine learning methods and their use in predicting football match outcomes and investigates the processes and results used in these pieces of literature.

2.1.1 Logistic Regression

Logistic Regression is a form of supervised machine learning which, in the words of Bing Shen Choi et al.[16], models the relationship between the input features and the probability of a piece of data belonging to a certain class. Binary Logistic Regression makes use of the Sigmoid function to turn linear combination of inputs and weights into a value between 0 and 1. A decision boundary is a hyperplane used in Binary Logistic Regression to separate data points into positive and negative classes based on a probability threshold (normally 0.5) .

Prasetio et al. [17] used Binary Logistic Regression to predict football match results from the 2015/16 Premier League season. They used the following Binary Logistic Regression equation:

$$\Pi(x) = \frac{1}{1 + e^{-y}} \tag{2.1}$$

where:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 + X_4 \tag{2.2}$$

The y value is made up of four input features along with the constant β_0: Home Offence (X_1), Home Defence (X_2), Away Offence (X_3), and Away Defence (X_4). The corresponding β coefficient measures the weight of each input feature in predicting the outcome. They calculated the probability of two outcomes, with the decision boundary set at 0.5, meaning that a value of Π greater than 0.5 indicates a Home Win and a value equal to or less than 0.5 indicates an Away Win.

This approach allowed them to attain a peak predictive accuracy of 69.5%, showing how Logistic Regression, with only four input features, is able to predict results to a high level of accuracy. One limitation with this method, is the inability to predict draws, as Binary Logistic regression can only have two outcomes, in this case Home Win or Away Win. With 28.2% of matches ending in a draw in the Premier League in the 2015/16 season [18], this model's overall ability to accurately predict results is flawed, despite its excellent record in predicting wins and losses. The solution to this could be a Multinomial Logistic Regression, allowing for multiple possible outcomes, in this context, win, loss, and draw. One such solution, proposed by Raju et al. [19] managed to predict results with a 70.27% accuracy, marginally improving on Prasetio's model's performance, but with being able to predict draws, gives a more complete set of predictions.

2.1.2 Random Forest

Random Forest is an ensemble learning method that groups a "forest" of decision trees and summarises their predictions to produce an aggregate result. This method uses bagging, also known as bootstrap aggregating, where each tree is trained on a randomly sampled subset of the original dataset, creating diversity across the trees in the forest, with the aim of reducing variance and improving generalisation to data. Breiman [20] describes Random Forests as a combination of tree predictors where the value of each tree depends on the values of a vector sampled randomly and independently, with the same distribution for all of the trees. This aims to reduce overfitting, where a model generalises poorly to unseen, or test data, whilst improving the model's overall predictive accuracy. Random forest is often used to establish an initial baseline level of predictive accuracy to compare more advanced models against. The reason for this is that it tends to have a consistently good performance across a variety of tasks, without the need for large amounts of fine-tuning due to the low number of parameters.

Hucaljuk and Rakipović [21] tested a variety of machine learning models to

find the most effective at predicting football matches and used a random forest as one of their methods. They utilised two datasets: a "basic" dataset with selected features and an "expert" dataset, which included an additional feature representing team quality, subjectively rated by an expert. Their Random Forest classifier had a strong performance, compared with other methods, on the "basic" dataset, but struggled slightly by comparison on the "expert" data. One reason for this change in performance between datasets could be due to overfitting, with the model potentially leaning too heavily on the extra value in the "expert" dataset when predicting results. Subjective features, like the one added in the "expert" data, often introduce bias within Random Forest models, skewing the results and reducing its overall performance.

2.1.3 XGBoost

Extreme Gradient Boosting (XGBoost) is a tree boosting algorithm, meaning that it creates an ensemble of learners (normally decision trees) to create a strong predictive model. This is done by building tress sequentially, with each created tree aiming to correct the errors of the previous tree, with the algorithm minimising the loss function through the use of gradient boosting. This algorithm is the default choice of ensemble methods [22] due to its scalability in a range of scenarios. Additionally, it performs exceptionally on benchmark datasets in terms of both accuracy and computational efficiency when compared with similar machine learning algorithms such as Random Forests and Gradient Boosting [22].

Berrar et al.[23] successfully applied XGBoost to predict the outcomes of football matches in the 2017 Soccer Prediction Challenge, with their model achieving a predictive accuracy of 50.49%. However, accuracy alone does not capture the probabilistic quality of predictions, which is why they evaluated their model using the Ranked Probability Score (RPS). The XGBoost model achieved an RPSavg of 0.2023, making it the most accurate model in their experiments, outperforming their k-nearest neighbour (k-NN) model that was initially the best performer during the competition.

2.1.4 Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning method used primarily for binary classification problems. Initially proposed by Cortes and Vapnik [24], SVMs map input data into a high-dimensional feature space using kernel functions, allowing non-linear relationships to be handled. In the feature space, there is a hyperplane that optimally divides

the mapped data into classes. The hyperplane is chosen to maximise the margin, the distance between the hyperplane and the closest data points from each class, called the support vectors. The aim of this is to improve the generalisation of the model on test data by creating a larger "buffer zone", to better separate training data without errors[24].

A study by Rodrigues and Pinto [25], explored the use of various machine learning algorithms to predict the results of football matches, including a SVM. They evaluated the performance of these models by simulating betting scenarios. They calculated the profit that a bettor would have made if they had placed €2 on each prediction, using the odds available at the time of the match. In their initial forecasts, their SVM algorithm achieved the best accuracy in predicting results with 61.32%, which would have earned them a leading profit of €95.06 out of all of the methods, a profit margin of 12.51%. However, the SVM's performance was actually the poorest out of all their models in terms of predicting draws accurately, with a success rate of only 3.57%. This shows, similarly to Prasieto's [17] logistic regression model, that Rodrigues and Pinto's SVM is ultimately not viable for predicting the results of football matches.

2.1.5 Artificial Neural Networks

An Artificial Neural Network (ANN), sometimes known just as a Neural Network, is a machine learning model that is designed to replicate the way in which the human brain performs [26], [27]. Neural networks are made up of interconnected nodes called neurons which are structured into layers. Every neuron receives multiple input values which are then processed, with each neuron applying an activation function to the values to create non-linearity in the system, allowing more complex relationships to be modelled from the data. The output of the neuron can then be passed as an input value into the next layer of neurons in the network.

ANNs are made up of three main types of layers, which can each perform different functions. Input layers are the first layer of the network, where raw data enters the network and each neuron corresponds to a feature of the input data [26]. Next are the hidden layers, which are the neurons described previously, handling the computation within the network, allowing the network to learn patterns and extract features. Finally, the data is passed to the output layer where the final results from the computations are provided, this could be probabilities in a classification task or predicted values in regression scenarios.

Neural networks can be challenging to design and implement successfully, mainly due to the vast number of hyperparameters used in the training

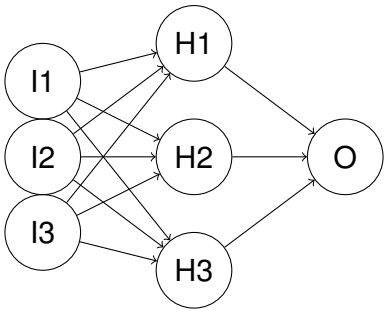

Figure 2.1: A simple ANN with three input nodes I, three hidden nodes H, and one output node O.

of a network. Key examples include: learning rate, batch size, optimiser algorithm, type of layer, number of epochs, choice of activation function, weight initialisation, and could include variations in network structure, such as the number of layers, and number of neurons per layer (sometimes known as the "width" of a layer)[28]. Adjusting these parameters to find an optimal solution is often time consuming and can be computationally expensive. Traditionally, methods such as grid search and random search have been used to fine-tune hyperparameters as these methods guarantee the optimal set of hyperparameters from the search pool. However, these methods are inefficient because of the sheer number of possible hyperparameter value combinations that could be used [28], many of which are likely to have a negative impact on the model's performance, leading to wasted resources and further time penalties.

Nevertheless, if optimised correctly, ANNs can yield outstanding results. Among the models tested by Hucaljuk and Rakipović [21] was an ANN with 5 hidden layers, which achieved the highest prediction accuracy out of all of their models at 68.8%. Tiwari et al. [29] used a Recurrent Neural Network (RNN) - a variation of an ANN that performs better on sequential data and incorporates information from previous inputs. They tested multiple models, using combinations of RNNs and LSTMs (Long Short Term Memory), a further specialisation of RNNs, with variations of hyperparameters. Their LSTM model managed to achieve a test accuracy of 80.75%, comfortably surpassing the performance of Hucaljuk and Rakipović's model [21], [29]. It does have to be noted, however, that this model, similarly to Prasieto [17], had only two output classes, meaning that draws were not predicted.

2.2 Team Ratings

Team ratings are a useful measure of a team's ability, and when accurately calculated, can be incredibly helpful in predicting the outcome of a game

before it is played. A team rating that can accurately portray the difference in quality between two teams and can be a highly influential data point for machine learning models to learn from.

The problem is, however, that calculating a value that accurately represents a team's ability compared with others can be challenging, due to various factors. Vaziri et al. [30] proposed three properties that an accurate rating system should consider:

1. **Opponent Strength**: Victories should weigh differently based on the opponent's calibre. A win against a higher-placed team should boost ratings more than one against a lesser side.

2. **Incentive to Win**: Teams with little at stake, particularly late in the season of league formats, may underperform. Ratings should account for this, reducing the weight of wins against unmotivated opponents.

3. **Sequence of Matches**: The rating of a team should consider the difficulty of their matches. A tough run of matches or an easy schedule should not disproportionately skew the ratings and should focus solely on results.

The use of these properties aims to ensure that a rating system is able to provide team ratings that are accurate and reflect the team's strength compared to others. Some of the most prominent methods for this will be outlined below.

2.2.1 Win-Loss

The Win-Loss method is the simplest method of rating a team. It involves simply awarding a team the most rating points for winning games, less for draws, and the least for losses. The highest ratings will reflect the team that has won the most games, but does not account for two of the properties mentioned above. Incentive to win and Sequence of matches. This means a team's high rating could be purely due to having played the lesser teams in the competition or teams with little to play for, meaning it does not give a true reflection of a team's strength.

2.2.2 Elo

The Elo ranking system, created by Arpad Elo [31], was originally designed to rank chess players. The outcome of a particular match was primarily

determined by the difference in existing ratings of the two parties (whether that be chess players or sports teams). If a lower-rated team defeats a higher-rated opponent, they gain more rating points compared to when a higher-rated team wins, rewarding unexpected results. This method takes into account Vaziri et al.'s [30] first and second properties of an accurate rating system, as there are more rating points on offer for beating a well-rated team and there is no rating-related incentive for losing matches. However, one fault of this method is that it does not satisfy the third property, which is the sequence of matches that a team plays. This means that if the order of the fixtures changed, with the results staying the same, the teams' ratings would change.

A variation of the Elo system is used to determine the Fifa World Rankings, which ranks all of the national football teams worldwide. The formula was adapted in 2018 to become the following:

$$P = P_{before} + I \times (W - W_e) \tag{2.3}$$

Where:

- P is the resulting points after the match.
- P_{before} is the points before the match is played.
- I is the importance of the match. With a higher value for more important games.
- W is the result of the match with a value of 1 for a win, 0.5 for a draw, and 0 for a loss.
- W_e is the expected result of the match, which itself is given by the formula:

$$W_e = \frac{1}{10^{(\frac{-dr}{600})} + 1} \tag{2.4}$$

Where:

- dr is the difference in ratings between the two playing teams.

Via FIFA[32].

This highlights the suitability of the use of the Elo ranking system as part of football match prediction as it is used by football's governing body, at the highest level of competition.

A further football-specific variation of ELO, known as "Goal-based ELO", was proposed by Hvattum and Arntzen [33] who theorised that a win by a larger margin should be rewarded with a greater boost in rating. The

aim was to reflect the winning team's superiority, and conversely, a heavily beaten losing team should suffer a greater fall in rating due to their poor result. Their method for this involved accounting for a team's goal difference in a match to calculate their new ELO, with the formula structured similarly to the FIFA rankings formula with the adjustment magnitude (I in the FIFA Formula) replaced by the following formula:

$$k = k_o(1 + \delta)^{\lambda} \tag{2.5}$$

Where:

- k_0 is the initial magnitude.
- k is the adjusted magnitude used to update the ELO.
- δ is the goal difference of a match
- λ is a constant parameter that controls how strongly goal difference affects the magnitude.

This updated system performed better than the original ELO and other methods they proposed when tasked with predicting the outcome of matches based on previous data, only marginally beaten in terms of a lower loss by their AVG and MAX systems, derived from bookmakers odds.

2.2.3 Glicko

The Glicko rating system was created by Mark Glickman [34] and aimed to improve upon some of the shortcomings of the Elo model. One key flaw that Glickman identified in the Elo system is that it does not account for inactivity. If a player or team has a long period without matches, their rating can become outdated, and the results of their subsequent matches may not accurately reflect their or their opponent's true abilities. This means that any rating adjustments as a result of these matches may be poorly weighted.

To attempt to fix this, he introduced a measure of rating reliability called Rating Deviation (RD). A higher RD value means that there is uncertainty in a player's rating, possibly due to the fact that the player may not be competing regularly [34]. The RD value changes dynamically over time, increasing during periods of inactivity and falling when players compete regularly, ensuring that ratings are more weighted towards those playing more recently.

Another key difference between Glicko and Elo is the way rating points are exchanged. In the Elo system, the winner's rating always increases by

the same amount as the loser's rating decreases. However, in the Glicko system, the adjustments depend on the RD values of both participants. This means that the rating change for the winner may differ from the rating change for the loser, since the RD determines the weight of a match's outcome on each player's rating. For instance, a match involving a player with a high RD will result in more significant rating adjustments compared to one with a low RD. By including the RD value, the Glicko system ensures that ratings are more reliable and reflective of current performance, making it suitable for competitions where participants compete irregularly or intermittently.

One key limitation of the Glicko system is that players who play regularly often have such low RD values that rating updates are only in much smaller amounts, which may not accurately reflect the changes in strength. This problem can be solved by setting a minimum threshold for the RD value [34], but this may require some fine-tuning in order to find a value that works effectively.

While the Glicko method is seen as an overall improvement over the Elo system in individual competitions like chess [34], it is known to struggle more in team sports. These difficulties are particularly pronounced in league competitions, where teams compete against the same opponents multiple times over the course of a season, encountering the problems associated with low RD values, as mentioned earlier.

3 Methodology

This section describes and justifies the methods to design the problem aims and objectives outlined in Section 1.2.

3.1 Dataset

To be able to achieve a high level of accuracy in predicting the outcome of matches, a large amount of accurate and consistent historical match data is needed. The dataset is included in the submission folder as `dataset.csv`.

3.1.1 Data Sources and Acquisition

One often used and easily accessible dataset in football machine learning tasks is the European Soccer Database [35]. This dataset was not suitable for this project as it contained outdated match data (between 2008 and 2016), as a result the output may not have been entirely accurate for current applications.

As a result, the decision was made to create a custom dataset from other data sources. This was mainly so there was complete control over the types of statistics included and their recency. The aim was to include up-to-date data, as mentioned previously. The data was taken from one source, Football Data [36] which provided information like (but not limited to) match results, goals, shots on and off target, and corners, as well as pre-match odds from a number of major bookmakers, including the likes of Bet365, Ladbrokes and William Hill. The data from Football Data could be downloaded from their website in the form of `.csv` files, with one file for each season for a particular league.

The data collected for each league was each full season between and including 2018/19 to the 2023/2024 season, giving six full seasons of complete data for each chosen league, a sufficient time period to draw conclusions from. There were seven leagues data was sourced from: the

English Premier League, the Spanish La Liga, the German Bundesliga, the French Ligue 1, the Italian Serie A, the Dutch Eredivisie, and the Belgian Pro League.

3.1.2 Combining Data Sources

Next, these `.csv` files obtained had to be combined for later use. Each of the files were combined into a dataframe within a Python script before moving onto the cleaning stage.

3.1.3 Data Cleaning

The data cleaning step, which is carried out in the same script, is crucial to ensure the datasets are accurate, complete, and consistent. If done correctly, a clean dataset should improve the performance of a machine learning model when compared with its previous dirty form. In commercial settings, it is estimated that dirty datasets that have incomplete customer and prospect data wastes 27% of revenue [37], so it is crucial that a dataset minimises missing and erroneous values.

In the context of my dataset, little cleaning is required due to the way the custom dataset has been created. The main issues faced were smoothing out small inconsistencies in data and column formats across the files. The first instance is converting the Date column across the files to the same format to allow for the matches to be ordered. This is done so that each match is processed chronologically by the machine learning model. Next, columns across the files are renamed to be consistent. The third and final example is the team name mapping. Some teams have different names, due to team rebranding or reforming. These names need standardising in the final dataset, allowing all the matches belonging to a team to be processed properly in sequence. This is done through a mixture of manual mapping and the use of the `get_close_matches` function from the difflib library.

3.2 Feature Engineering

Feature Engineering involves the transformation of a given feature space, typically using mathematical functions, with the aim of reducing the modelling error for a given target [38]. However, there is no single optimal

solution to successful feature engineering, often making the process often iterative and experimental. As a result, decisions made must be justified using numerical methods to get the best chance of a positive outcome.

3.2.1 Selecting Features

For this project, it is crucial that the features selected can accurately forecast future matches from past matches. The process for evaluating features began by applying the Pearson Correlation Coefficient (PCC, also sometimes referred to as just Correlation Coefficient) to each of the original match statistics within the dataframe, with the results shown as part of table 3.1 (Appendix A). This was used due to its simplicity and effectiveness in quantifying linear relationships [39], perfect for an initial analysis of the importance of each statistic in predicting results.

The analysis was performed against FTR (Full Time Result), where a value of 2 corresponds to a home win, 1 represents a draw, and 0 signifies an away win. A positive correlation close to 1 indicates that the feature is strongly associated with a home win, while a negative value close to -1 suggests a stronger association with an away win. Values close to 0 show that the feature has little to no influence on the match result.

This analysis showed that both half time and full time goals were unsurprisingly the most important features. Beyond that, shots and shots on target were strong indicators of a win either way, with other metrics having little to no impact on the result.

Another feature analysis used was Mutual Information (MI). MI measures the mutual dependence of two variables, both linear and non-linear, looking at the amount of uncertainty lost from the tested variable (match statistics) when the other variable is known [40](FTR). This method can be more informative than PCC as it is able to analyse non-linear relationships, crucial for understanding how match statistics together can influence a result. The results of this analysis can be seen in Table 3.2 (Appendix A). The high influence statistics this analysis identified are similar to PCC, with goals (half and full time), shots and shots on target performing well.

Next, it was important to check how much data was available for each of the features, as a feature with little coverage would not be helpful in creating accurate predictions. The results for the coverage of each feature can be seen in Table 3.3 (Appendix A), with every feature having a high level of coverage (>98%) meaning there is enough data from each to give detailed insights.

As a result of this analysis, the final raw features included were: Full Time Goals, Shots and Shots on Target. With each of these features recorded for the home and away teams in a match. The other raw features were dropped (corners, fouls committed, red cards, and yellow cards) due to their low PCC and MI values, indicating their lack of influence on the outcome of a game.

3.2.2 Team Rating System

The team rating system utilised on the dataset, was the "Goal-based ELO" system developed by Hvattum and Arntzen [33]. This was chosen due to its improved performance in predicting match outcomes compared with regular ELO, as mentioned in the literature review.

The Glicko rating system was not chosen due to its difficulties in accuracy with league competitions (which is where the contents of the dataset come from). This is because teams play regularly, often weekly, and as a result, low RD values mean that rating updates may be inaccurate [34].

The other team rating system identified in the literature review, Win-Loss, was decided against due to inaccuracies because of the sequence of matches a team plays. Especially in league formats, with how the data in this dataset is structured, teams that begin with playing against poorer or stronger teams can have their own team rating skewed as a result.

In this application of the goal-based ELO system, every team starts with a rating of 1000, with the base k-factor k set to 32 and the goal difference scaling exponent λ set to 0.8 (based on equation 2.5). The value of k was originally taken from Elo's initial implementation of the system [31], where 32 was the standard value in competitive rating systems, with Hvattum and Arntzen keeping the same value for their later iteration [33]. The λ value of 0.8 was chosen to allow for meaningful updates for the k value based on goal difference, without making the changes too volatile or reactive to a particular result.

3.2.3 Streaks and Team Form

One of the engineered features created was streaks. Due to the success of results-based streak features in similar machine learning tasks [16] [21] [25], they were chosen to be implemented for this model. The streaks created were Unbeaten Streak (how many games since a team lost), Win streak (number of games won in a row), Winless streak (how many games

since the last win), and Loss streak (number of games lost in a row). These streaks were applied for both the home and away teams involved in a fixture and are able to help give a deeper insight into a teams' recent form.

Similarly, features were implemented that recorded a rolling total of both the Home and Away Teams' points totals over the last 5, 10 and 20 games. This is able to give an insight into a team's form and give the model another meaningful statistic to help predict match outcomes.

The streaks created can be seen in Table 3.4 (Appendix A), along with their Pearson Correlation Coefficient calculated with the same process as described for the original features. The coefficients show that the points rolling totals were the most influential, with the points over the last 20 games having the highest value.

3.2.4 Additional Engineered Features

The next set of engineered features created was the "conceded" features for each team. For each of the raw statistics (full time goals, shots, shots on target) for each team, the values were applied to the opposing team as the "conceded" version. Whilst for an individual game this may not provide any insight, over a sequence of games, this highlights a team's defensive capabilities, by recording the opposition's attacking output against them over each game.

Applying similar logic to previously, "net" statistics were introduced. These features represent the difference between a team's performance and their opponent's across all the raw features. By capturing the balance of play, net statistics are able to provide a more comprehensive view of a team's overall strength. For example, a team with consistently high number of net shots on target is creating significantly more shooting (and in turn goalscoring) opportunities than they concede, indicating offensive dominance over their opponent. Conversely, a negative net value highlights defensive vulnerabilities.

Incorporating both these sets of features offers more informative metrics than just raw statistics alone, as it reflects both attacking and defensive capabilities for a particular team. These richer representations of team performance ultimately enhance the model's accuracy in predicting match outcomes.

All of the features created and included in the dataset are shown in Table 3.6, Appendix A, with the feature name, the name of the feature in the code, and which part of the model they are used in, with the model structure

explained in the next section.

3.3 The Model

This section outlines the structure of the machine learning model, as well as the optimiser and loss functions used. The code used to create and run the model can be found in the `Model.py` Python script.

3.3.1 Model Structure

The next stage was to create a Machine Learning structure capable of turning historical match data into predictions for future matches, before comparing these predictions against the actual outcomes to determine accuracy. Based on the success of Tiwari et al. [29] and Rahman [41], an LSTM-based model was used due to its ability to capture temporal dependencies. LSTMs learn from a sequence of past matches to aid in future predictions, allowing the model to capture temporal trends which can contribute to more accurate predictions. By analysing sequences of previous matches, the model can better capture a team's recent form and momentum, contributing to more accurate predictions. This makes LSTMs particularly effective for the task of predicting football matches, where recent form fluctuations can significantly impact match results.

To effectively model variations in team performances, two LSTMs are used to process match data, one for each teams' home games and one for each teams' away games. This decision was motivated by the impact of a team having home advantage, as within the dataset, 43.65% of the games resulted in a home win, with only 31.67% being an away win. By training separate LSTMs, the model can learn distinct patterns for teams playing at home versus away, better capturing the influence of venue on match outcomes.

Inspired by the success of Berrar et al. [23], an XGBoost algorithm was implemented as part of the model. In this instance, it was used to handle the streak-based data for each team. This algorithm was chosen for its ability to capture complex, non-linear patterns, which are common in the fluctuating performance trends of football teams. Additionally, XGBoost's regularisation and ensemble nature make it particularly effective in preventing overfitting, enhancing the model's robustness and accuracy when analysing streak-based data.

The final hidden states from both LSTMs, along with the XGBoost class probability predictions and Elo ratings, processed by a separate multilayer perceptron (MLP), are then concatenated into a unified feature representation. This combined feature representation was then passed through a fully connected feedforward network with a softmax activation function to produce the final match outcome predictions. The model's performance was then evaluated against the validation and testing data splits to assess its accuracy and ability to generalise. An abstraction of the model structure is shown below in Figure 3.1.

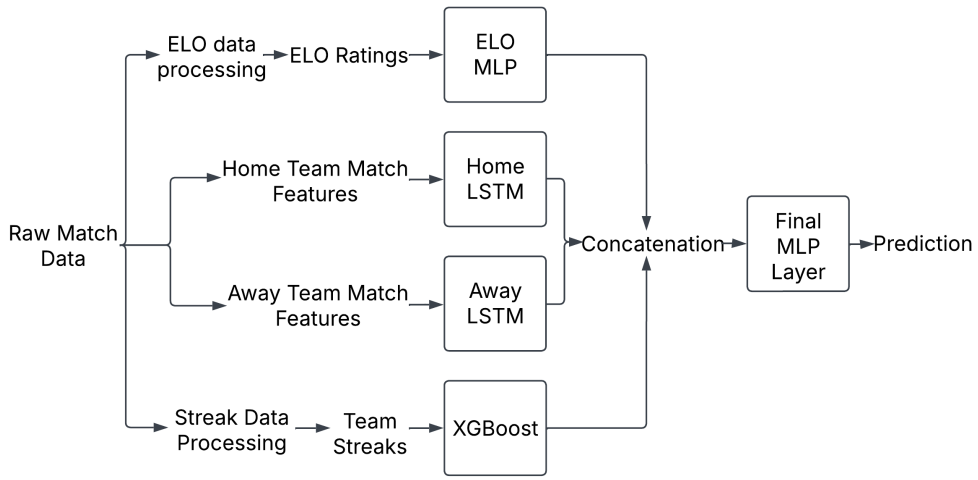

Figure 3.1: A basic abstraction of the structure of the model.

3.3.2 Optimiser and Loss Function

The optimiser chosen for this model was Stochastic Gradient Descent (SGD). This was chosen over other commonly used optimisers such as Adam due to its better generalisation ability in deep learning tasks [42], making it effective for this scenario.

For the loss function, Cross Entropy Loss was employed, as it is highly effective for multi-class classification problems [27] like this one. To address class imbalance, class weights were utilised, ensuring that the model learns effectively from all classes. This was crucial, given the unbalanced distribution of the match results in the dataset (43% home win, 32% away win, 25% draw), which could lead to poor prediction performance, especially for the least frequent class (draw).

4 Implementation Tools and Environment

4.1 Language and Environment

Python was selected as the language to develop this project, as it "has become the prime language for application in Data Science and Machine Learning Domains" [43]. This is down to a rich number of libraries designed for data processing, visualisation, and machine learning making it an ideal choice for building predictive deep learning models and performing data analysis. The following Python libraries were used in development:

- Pandas: A library that provides data manipulation and analysis, providing flexible data structures which are essential for cleaning and transforming data. [44].
- NumPy: The fundamental package for scientific computing in Python, providing a range of data structures, objects, and routines including efficient array operations and mathematical functions [45].
- PyTorch: A library that provides the framework for building and training deep learning models [46].
- Scikit-Learn: A machine learning library that provides a range of tools for predictive data analysis such as data pre-processing, model training, and model evaluation[47].
- XGBoost: The XGBoost library used to implement the XGBoost algorithm in Python [42].
- MatPlotLib: A library with the ability to create a range of complex and insightful visualisations from data[48].
- Seaborn: A data visualisation library based on MatPlotLib that provides a high-level interface for drawing informative statistical graphics[49].

The development environment chosen was Jupyter Notebook, chosen because of its documentation abilities, allowing markdown or LaTeX com-

ments alongside code snippets to provide a clear structure and explanation of each section of code. Once the code was finalised, it was exported as a `.py` file for execution.

4.2 Hardware

To ensure fast and efficient execution of the models, the code was run on the Viking supercomputer, utilising the GPU partition with nVidia A40 GPUs. The use of GPUs significantly accelerated the training and evaluation of the deep learning models, making it possible to handle large datasets and perform complex computations more efficiently. Additionally, running the code on Viking enabled the setup of a dedicated virtual environment that contained all the necessary libraries and dependencies, ensuring smooth execution of the code, and removing the need for package installations on local machines.

4.3 Data Split

Before the data was split, it was ordered by the date of the match each element of data was for, to be able to give a better insight into a team's form over time. The data was then split, with the first 70% of the data allocated as training data, the next 20% (70%-90%) used as a validation set, and the remaining 10% (90% - 100%) used as the test dataset. This meant that the evaluation and testing was fed from sequences of games before the games being evaluated and tested on, preventing data leakage and aiming to improve predictive accuracy.

4.4 Hyperparameters

The next stage of development was to determine the hyperparameters of the model. This stage is crucial, as achieving an optimal hyperparameter setup can lead to higher performance in terms of predictive accuracy. To achieve this, the Grid Search algorithm was used. This algorithm evaluates a model in a brute-force style: trying every possible permutation of specified hyperparameter values, aiming to find the best performing set of hyperparameter values for that model. This method was chosen, due to its ability to improve classification accuracy, regardless of the type of machine learning model used [50]. It is also guaranteed to find the optimal

combination of hyperparameters from its hyperparameter input pool, but this does come at the cost of increased processing time and workloads.

Before the grid search algorithm for the model was run, some hyperparameters and model setup were finalised based on the findings of those who tackled similar problems. For example, the batch size of 1 used by Tiwari et al. [29] in their LSTM implementation of a similar task allowed them to achieve a high level of accuracy. The batch size of 1 means that each match is processed one at a time, preventing temporal mismatches. On top of this, the `shuffle` parameter of the `DataLoader` functions for all of the datasets are set to `False`, meaning that each match is processed in chronological order, allowing a team's form to be accurately mapped over time.

Two separate grid search algorithms were run. One was initially run on just the LSTM + MLP portion of the model, with the XGBoost parameters set to constant values considered "default". Once the optimal setup for this portion was achieved, another grid search was run on just the XGBoost algorithm to find its ideal set of hyperparameters. The top 10 hyperparameter combinations from both these searches are shown in Appendix B with both Tables 4.1 and 4.2 respectively, with both the top hyperparameter combinations from each used for the final testing and analysis of the model detailed in the following final sections.

5 Results and Evaluation

With the optimal hyperparameter set decided, the model was run and evaluated. This produced a final overall accuracy across the test set, as well as accuracy across subsets of the test set, with each subset dedicated to matches for a specific league. The model was run 10 times to capture any variations in results, with the results of each run in Table 5.1 below, with the peak predictive accuracy for each subset highlighted in bold.

Table 5.1: Overall and league specific predictive accuracy from multiple runs.

Run No.	Overall	ENG	GER	ITA	FRA	SPA	BEL	NED	POR
1	52.61%	50.44%	52.17%	55.70%	52.61%	55.70%	**57.65%**	43.50%	52.17%
2	52.86%	**51.32%**	52.17%	56.14%	53.08%	55.26%	**57.65%**	44.07%	52.17%
3	52.64%	50.88%	51.96%	55.87%	53.14%	55.46%	57.47%	44.27%	51.93%
4	52.73%	50.44%	52.72%	**57.02%**	52.13%	55.46%	57.14%	43.50%	52.06%
5	52.48%	50.88%	50.54%	56.14%	**53.55%**	55.26%	56.47%	44.07%	51.63%
6	52.73%	50.44%	53.26%	56.22%	52.61%	55.70%	55.29%	**45.76%**	52.17%
7	52.92%	50.88%	53.26%	**57.02%**	52.13%	55.70%	57.06%	44.07%	51.93%
8	52.36%	51.18%	51.09%	55.26%	53.08%	55.26%	56.47%	44.63%	51.09%
9	**53.04%**	**51.32%**	**53.80%**	**57.02%**	52.13%	55.70%	57.06%	44.63%	**52.72%**
10	52.48%	**51.32%**	51.63%	56.14%	53.08%	**55.82%**	55.88%	43.50%	51.09%

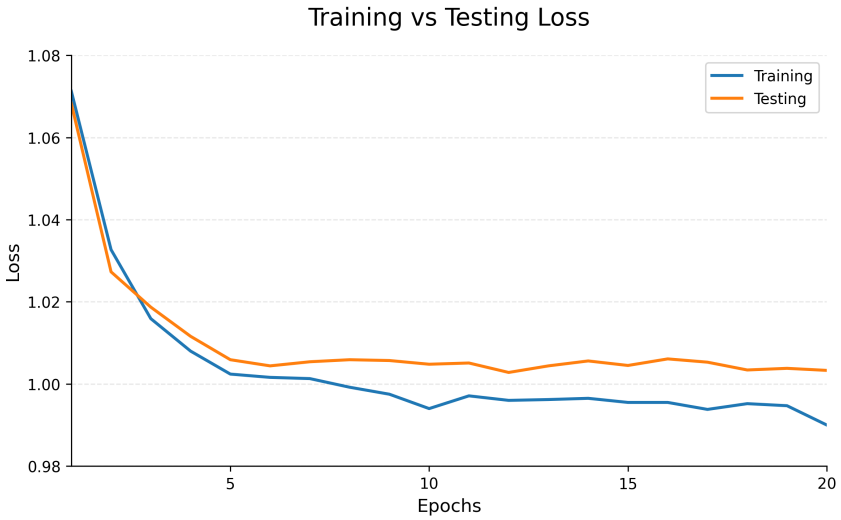

Figure 5.1: Training vs Testing Loss over 20 epochs of runtime.

The best run was seen with run 9: achieving a test accuracy of 53.02% on the entire dataset. A graph showing the training and testing loss for this run can be seen above in Figure 4.1. This graph shows both losses decreasing

for roughly the first 6 or 7 epochs, before the testing loss plateaus at a slightly higher level compared with the training loss, which continues to slowly trend downward.

The results for each subset and overall from all runs were then averaged and compared with the predictive accuracy of two leading UK bookmakers: Bet365 and William Hill, with the prediction of these bookmakers taken as their odds favourite for each game. The results of this can be seen in Table 5.2 and Figure 5.1 below.

Table 5.2: Overall and league specific average predictive accuracy compared with bookmakers.

League	Model	Bet365	William Hill
English Premier League	50.91%	56.18%	56.23%
German Bundesliga	52.26%	52.78%	57.24%
Italian Serie A	56.25%	54.61%	54.43%
French Ligue 1	52.75%	53.40%	50.93%
Spanish La Liga	55.53%	52.54%	52.63%
Belgian Pro League	56.81%	53.41%	51.66%
Dutch Eredivise	44.20%	57.15%	57.04%
Portuguese Primeira Liga	51.90%	56.15%	56.26%
Entire Dataset	52.68%	54.24%	52.90%

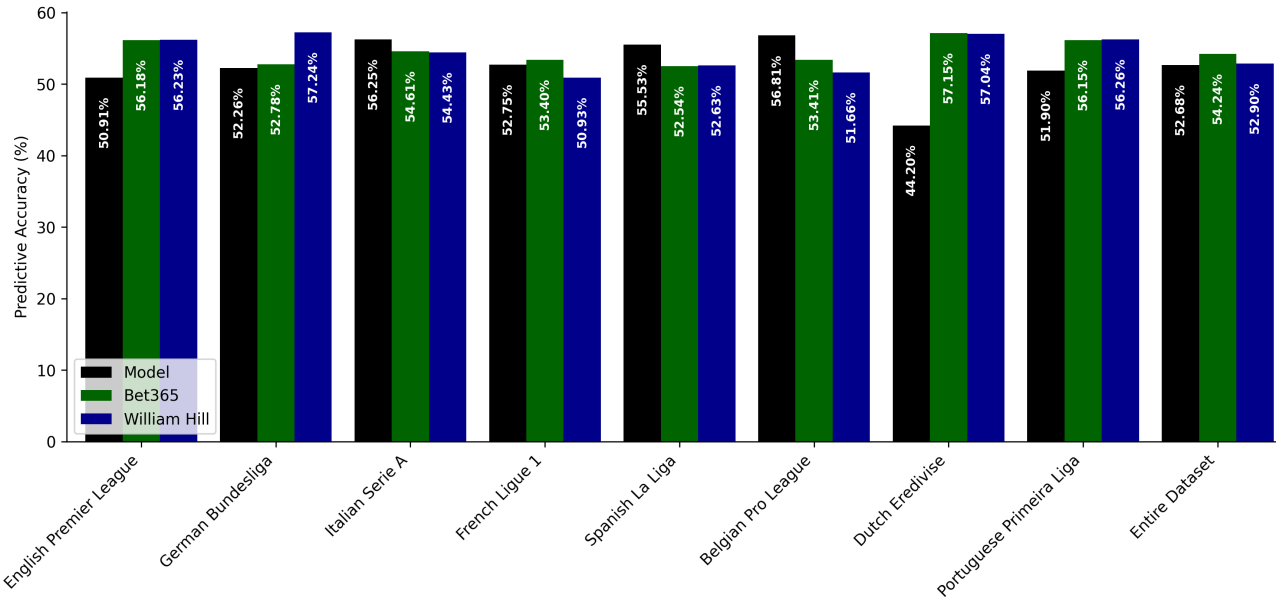

Figure 5.2: Overall and league specific predictive accuracy compared with bookmakers.

These results show that, overall, the model performed marginally worse than William Hill in terms of average predictive accuracy, but marginally

better if considering peak performance accuracy. However, the model did fall over a percentage shorter than Bet365 in terms of both average and peak predictive accuracy. The model performed exceptionally in predicting the Spanish La Liga, Belgian Pro League and Italian Serie A, comfortably outperforming both bookmakers by a few percent of accuracy. However, the model did underperform by a larger margin for both the English Premier League and the Portuguese Primeria Liga and fell drastically short for the Dutch Eredivise (over 10% worse) which is certainly an area for future investigation.

6 Conclusion and Future Work

6.1 Conclusion

The original aims of this project were to create a machine learning model capable of predicting football match outcomes, that is able to outperform bookmakers' predictions. Whilst it cannot be said that this model outright outperforms bookmakers' predictions, it can certainly be seen to be highly competitive, outperforming the data on the selected bookmakers accuracy on select leagues, despite falling marginally short overall. Additionally, it can be argued that the model is able to provide a more complete set of predictions than compared with bookmakers as it is able to predict draws, something very rarely done by bookmakers due to the way odds are set.

One of the further objectives initially established was to identify the most influential factors in predicting match outcomes. This was achieved by the use of the Pearson Correlation Coefficient and Mutual Information, which identified the number of shots and shots on target as one of the main indicators to the outcome of a match. Another secondary aim was to compare the accuracy of predicting match outcomes across different leagues, something achieved in the previous section, with Belgian Pro League and Italian Serie A identified as the leagues easier to predict, and the Dutch Eredivise the hardest.

6.2 Future Work

There are a multitude of ways this model could be improved in terms of future work. One key way could be the inclusion of further advanced statistics such as Expected Goals (xG), Expected Assists (xA) and Expected Threat (xT) when they become more widely available, as mentioned previously. To supplement this, including line-ups for each team for each game and devising a player importance or ranking system could also improve the model's accuracy. This would account for the inclusion or omission of influential key players that could sway the outcome of the match.

As mentioned in the results section, due to the large disparity in predictive quality between the model and the bookmakers in predicting the Dutch Eredivise, this is certainly an area for future investigation. Was the data collected from this league poor quality or missing values? Or was the model just not able to create accurate predictions compared with the other leagues? Diagnosing and fixing this could definitely help the model improve in its overall ability to predict.

Something else that could be considered to improve the predictions of the model is external factors. This could include whether they are home or away, the time of day or the weather conditions they are playing in, the number of rest days between matches, or even the distance they've had to travel to an away match. These are additional influences on a team that can affect their playing ability, and consequently, the outcome of the match, and the inclusion of these could only help the quality of predictions.

Another method of improvement, in terms of measuring the success of the model could be implementing a method similar to that used by Rodrigues and Pinto [25]. By calculating the potential profit that could be made if say £2 was staked on the predictions from the model, it's accuracy could be better evaluated for practical uses in attempting to "beat the bookies", something also highlighted in the initial aims and objectives.

Appendices

Appendix A

Table 3.1: The Pearson Correlation Coefficient between original match statistics and a Home Win.

Feature	Name In Code	Pearson Value
Full Time Home Goals	FTHG	0.635
Half Time Home Goals	HTHG	0.424
Home Shots On Target	HST	0.396
Home Shots	HS	0.208
Away Red Cards	AR	0.102
Home Corners	HC	0.031
Away Yellow Cards	AY	0.022
Away Fouls Committed	AF	-0.016
Home Fouls Committed	HF	-0.020
Away Corners	AC	-0.061
Home Yellow Cards	HY	-0.097
Home Red Cards	HR	-0.132
Away Shots	AS	-0.242
Away Shots On Target	AST	-0.404
Half Time Away Goals	HTAG	-0.416
Full Time Away Goals	FTAG	-0.638

Table 3.2: Mutual Information (MI) relationship between original match statistics and Full Time Result (FTR).

Feature	Name In Code	MI value
Full Time Home Goals	FTHG	0.308
Full Time Away Goals	FTAG	0.284
Half Time Home Goals	HTHG	0.111
Half Time Away Goals	HTAG	0.104
Away Shots On Target	AST	0.096
Home Shots On Target	HST	0.09
Away Shots	AS	0.039
Home Shots	HS	0.036
Home Red Cards	HR	0.012
Home Yellow Cards	HY	0.008
Home Fouls Committed	HF	0.005
Away Red Cards	AR	0.005
Home Corners	HC	0.003
Away Fouls Committed	AF	0
Away Corners	AC	0
Away Yellow Cards	AY	0

Table 3.3: Original feature coverage.

Metric	Coverage (%)
Full Time Goals	100
Half Time Goals	99.975
Shots On Target	99.969
Shots	99.969
Corners	99.969
Fouls Committed	98.476
Red Cards	99.975
Yellow Cards	99.975

Table 3.4: The Pearson Correlation Coefficient between streaks and form statistics and a Home Win.

Feature	Name In Code	Pearson Value
Home Points Last 20 Games	Home_Points_Last_20	0.237
Home Points Last 10 Games	Home_Points_Last_10	0.220
Home Points Last 5 Games	Home_Points_Last_5	0.189
Home Unbeaten Streak	Home_Unbeaten_Streak	0.150
Home Win Streak	Home_Win_Streak	0.130
Away Winless Streak	Away_Winless_Streak	0.101
Away Loss Streak	Away_Loss_Streak	0.084
Home Loss Streak	Home_Loss_Streak	-0.084
Home Winless Streak	Home_Winless_Streak	-0.096
Away Win Streak	Away_Win_Streak	-0.128
Away Unbeaten Streak	Away_Unbeaten_Streak	-0.145
Away Points Last 5 Games	Away_Points_Last_5	-0.193
Away Points Last 10 Games	Away_Points_Last_10	-0.227
Away Points Last 20 Games	Away_Points_Last_20	-0.240

Table 3.5: The Pearson Correlation Coefficient between select other engineered statistics and a Home Win.

Feature	Name In Code	Pearson Value
Home Net Full Time Goals	Home_Net_FTG	0.855
Home Net Shots on Target	Home_Net_ST	0.526
Home Net Shots	Home_Net_S	0.276
Home ELO	Home_ELO	0.265
Away ELO	Away_ELO	-0.273
Away Net Shots	Away_Net_S	-0.276
Away Net Shots on Target	Away_Net_ST	-0.526
Away Net Full Time Goals	Away_Net_FTG	-0.855

Table 3.6: Mutual Information (MI) relationship between select other engineered statistics and Full Time Result (FTR).

Feature	Name In Code	MI Value
Home Net Full Time Goals	Home_Net_FTG	1.071
Away Net Full Time Goals	Away_Net_FTG	1.071
Home Net Shots on Target	Home_Net_ST	0.17
Away Net Shots on Target	Away_Net_ST	0.163
Home Net Shots	Home_Net_S	0.043
Away Net Shots	Away_Net_S	0.039
Home ELO	Home_ELO	0.038
Away ELO	Away_ELO	0.037

Table 3.7: Every feature in the dataset, their names in the code and which part of the model they will be fed into.

Feature	Name In Code	Part of Model
Full Time Result	FTR	All
Full Time Home Goals	FTHG	LSTM
Full Time Away Goals	FTAG	LSTM
Home Shots	HS	LSTM
Away Shots	AS	LSTM
Home Shots On Target	HST	LSTM
Away Shots On Target	AST	LSTM
Home ELO	Home_ELO	MLP
Away ELO	Away_ELO	MLP
Home Win Streak	Home_Win_Streak	XGBoost
Away Win Streak	Away_Win_Streak	XGBoost
Home Loss Streak	Home_Loss_Streak	XGBoost
Away Loss Streak	Away_Loss_Streak	XGBoost
Home Unbeaten Streak	Home_Unbeaten_Streak	XGBoost
Away Unbeaten Streak	Away_Unbeaten_Streak	XGBoost
Home Winless Streak	Home_Winless_Streak	XGBoost
Away Winless Streak	Away_Winless_Streak	XGBoost
Home Points Last 5 Games	Home_Points_Last_5	XGBoost
Away Points Last 5 Games	Away_Points_Last_5	XGBoost
Home Points Last 10 Games	Home_Points_Last_10	XGBoost
Away Points Last 10 Games	Away_Points_Last_10	XGBoost
Home Points Last 20 Games	Home_Points_Last_20	XGBoost
Away Points Last 20 Games	Away_Points_Last_20	XGBoost
Home Net Goals	Home_Net_FTG	LSTM
Away Net Goals	Away_Net_FTG	LSTM
Home Net Shots	Home_Net_S	LSTM
Away Net Shots	Away_Net_S	LSTM
Home Net Shots On Target	Home_Net_ST	LSTM
Away Net Shots On Target	Away_Net_ST	LSTM
Home Goals Conceded	FTHG_Conceded	LSTM
Away Goals Conceded	FTAG_Conceded	LSTM
Home Shots On Target Conceded	HST_Conceded	LSTM
Away Shots On Target Conceded	AST_Conceded	LSTM
Home Shots Conceded	HS_Conceded	LSTM
Away Shots Conceded	AS_Conceded	LSTM

Appendix B

Table 4.1: Top 10 Grid Search results by test accuracy from the LSTM+MLP branch of the model.

Seq Len	LSTM size	LSTM layers	MLP size	Dropout	LR	Momentum	Weight Decay	Epochs	Test Acc
10	64	5	16	0.2	0.001	0.7	0.01	20	0.5222
10	128	3	64	0.5	0.001	0.99	0.001	20	0.5215
10	64	5	16	0.5	0.001	0.9	0.001	20	0.5206
20	64	5	16	0.2	0.001	0.7	0.01	40	0.5203
10	32	4	16	0.5	0.0001	0.9	0.001	20	0.5200
10	64	5	32	0.5	0.001	0.99	0.01	20	0.5196
20	32	4	64	0.5	0.0001	0.7	0.001	40	0.5194
10	32	4	64	0.5	0.001	0.7	0.001	20	0.5194
10	32	4	16	0.2	0.0001	0.99	0.001	40	0.5189
20	128	4	64	0.5	0.001	0.99	0.001	20	0.5185

Table 4.2: Top 10 Grid Search results by test accuracy from the XGBoost branch of the model.

n Estimators	LR	Max Depth	Subsample	Colsample Bytree	Gamma	Min Child Weight	Test Acc
200	0.01	5	0.8	0.6	0.1	3	0.5253
50	0.001	7	0.6	0.6	0	3	0.5245
200	0.01	9	0.8	0.8	0.1	3	0.5240
300	0.01	5	1	1	0.1	5	0.5234
100	0.01	7	0.8	0.8	0	3	0.5227
300	0.01	7	0.6	1	0	5	0.5227
50	0.1	5	0.8	1	0.1	3	0.5225
100	0.05	3	0.8	0.8	0.1	1	0.5223
200	0.1	3	0.8	0.8	0.1	1	0.5221
200	0.001	5	0.6	0.8	0.2	5	0.5220

Bibliography

[1] Sport For Business, *The world's most watched sports*. [Online]. Available: https://sportforbusiness.com/the-worlds-most-watched-sports/#:~:text=Soccer&text=With%203.5%20billion%20fans%20worldwide,viewed%20sport%20in%20the%20world..

[2] Gambling Commission, *Industry statistics - febraury 2024 - correction: Official statistics.* [Online]. Available: https://www.gamblingcommission.gov.uk/statistics-and-research/publication/industry-statistics-february-2024-correction.

[3] technavio, *Gambling market analysis uk - size and forecast 2024-2028*. [Online]. Available: https://www.technavio.com/report/gambling-market-in-uk-industry-analysis.

[4] The Times, Tom Kershaw, *The secretive analytics company helping clubs unearth next moisés caicedo*. [Online]. Available: https://www.thetimes.com/sport/football/article/jamestown-analytics-data-football-moises-caicedo-gxztr67hk.

[5] Barney Corkhill, SportsMole, *Chelsea complete record moises caicedo signing from brighton hove albion*. [Online]. Available: https://www.sportsmole.co.uk/football/chelsea/transfer-talk/news/chelsea-complete-record-moises-caicedo-signing_521138.html.

[6] Dan Pritchard, Analytics FC, *Data in context: How did graham potter's brighton achieve a 'big six' style of play with a bottom six budget?* [Online]. Available: https://analyticsfc.co.uk/blog/2022/09/20/data-in-context-how-did-graham-potters-brighton-achieve-a-big-six-style-of-play-with-a-bottom-six-budget/.

[7] Sam Williams, Liverpool Football Club, *Behind the badge: The physicist who leads liverpool's data department*. [Online]. Available: https://www.liverpoolfc.com/news/behind-the-badge/398645-ian-graham-liverpool-fc-behind-the-badge.

[8] Mohammed Dougramaji, Rockbourne, *Data analytics in football: How lfc used data to gain the edge*. [Online]. Available: https://rockborne.com/graduates/blog/data-analytics-in-football-lfc/#:~:text=The%20use%20of%20data%20analysis,club%27s%20financial%20stability%20and%20sustainability..

[9] FotmMob, *Mohammed salah stats*. [Online]. Available: https://www.fotmob.com/en-GB/players/292462/mohamed-salah.

[10] Miguel Delaney, The Independent, *Liverpool win premier league for first time in club's history.* [Online]. Available: https://www.independent.co.uk/sport/football/premier-league/liverpool-premier-league-title-winners-2019-20-manchester-city-chelsea-result-a9585276.html.

[11] Feargal Brennan, *How many trophies have liverpool won? a complete list of all major silverware in the reds trophy case.* [Online]. Available: https://www.sportingnews.com/uk/football/news/how-many-trophies-liverpool-won-list-silverware/slfx3wmamhgaqqd1t38uo4idh.

[12] Karun Singh, *Introducing expected threat (xt)*. [Online]. Available: https://karun.in/blog/expected-threat.html.

[13] Hudl Statsbomb, *Introducing on-ball value (obv)*. [Online]. Available: https://statsbomb.com/news/introducing-on-ball-value-obv/.

[14] Betting Kingdom, *The impact of ai and machine learning in sports betting*. [Online]. Available: https://www.bettingkingdom.co.uk/blog/betting-advice/the-impact-of-ai-and-machine-learning-in-sports-betting/#:~:text=Machine%20learning%20models%20can%20analyze,as%20new%20information%20becomes%20available..

[15] K. P. Murphy, 'Machine learning - a probabilistic perspective,' in *Adaptive computation and machine learning series*, 2012, p. 1. [Online]. Available: https://api.semanticscholar.org/CorpusID:17793133.

[16] B. Choi, L.-K. Foo and S.-L. Chua, 'Predicting football match outcomes with machine learning approaches,' *MENDEL*, vol. 29, pp. 229–236, 2023.

[17] D. Prasetio and D. Harlili, 'Predicting football match results with logistic regression,' in *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, 2016, pp. 1–5.

[18] Statista, *Share of premier league matches ending in a draw from 1992/93 to 2024/25.* [Online]. Available: https://www.statista.com/statistics/1498537/premier-league-draws/.

[19] M. A. Raju, M. S. Mia, M. A. Sayed and M. Riaz Uddin, 'Predicting the outcome of english premier league matches using machine learning,' in *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2020, pp. 1–6.

[20] L. Breiman, 'Random forests,' *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] J. Hucaljuk and A. Rakipović, 'Predicting football scores using machine learning techniques,' in *2011 Proceedings of the 34th International Convention MIPRO*, 2011, pp. 1623–1627.

[22] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System,' in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. [Online]. Available: https://doi.org/10.1145/2939672.2939785.

[23] D. Berrar, P. Lopes and W. Dubitzky, 'Incorporating domain knowledge in machine learning for soccer outcome prediction,' *Machine Learning*, vol. 108, no. 1, pp. 97–126, 2019. DOI: 10.1007/s10994-018-5747-8. [Online]. Available: https://doi.org/10.1007/s10994-018-5747-8.

[24] C. Cortes and V. Vapnik, 'Support-vector networks,' in *Proceedings of the 7th annual ACM workshop on Computational learning theory*, ACM, 1995, pp. 273–297.

[25] F. Rodrigues and Â. Pinto, 'Prediction of football match results with machine learning,' *Procedia Computer Science*, vol. 204, pp. 463–470, 2022, International Conference on Industry Sciences and Computer Science Innovation, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.08.057. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922007955.

[26] S. S. Haykin, *Neural networks and learning machines*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:60504238.

[27] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, pp. 164–199. [Online]. Available: https://www.deeplearningbook.org.

[28] G. Franchini, 'Greennas: A green approach to the hyperparameters tuning in deep learning,' *Mathematics*, vol. 12, p. 850, Mar. 2024. DOI: 10.3390/math12060850.

[29] E. Tiwari, P. Sardar and S. Jain, 'Football match result prediction using neural networks and deep learning,' in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 229–231. DOI: 10.1109/ICRITO48877.2020.9197811.

[30] B. Vaziri, S. Dabadghao, Y. Yih and T. L. Morin, 'Properties of sports ranking methods,' *Journal of the Operational Research Society*, vol. 69, no. 5, pp. 776–787, 2018. DOI: 10.1057/s41274-017-0266-8.

[31] A. Elo, *The Rating of Chess Players, Past and Present*. New York: New York: Arco Publishing, 1978, ISBN: 978-0-668-05493-6.

[32] FIFA, *Fifa/coca-cola men's world ranking procedures*. [Online]. Available: https://inside.fifa.com/fifa-world-ranking/procedure-men.

[33] L. M. Hvattum and H. Arntzen, 'Using elo ratings for match result prediction in association football,' *International Journal of Forecasting*, vol. 26, no. 3, pp. 460–470, 2010. DOI: 10.1016/j.ijforecast.2009.10.002.

[34] M. E. Glickman, 'The glicko system,' *Boston University Department of Mathematics Technical Report*, 1995. [Online]. Available: http://www.glicko.net/glicko/glicko.pdf.

[35] H. Mathien, *European soccer database*, 2016. [Online]. Available: https://www.kaggle.com/hugomathien/soccer.

[36] Football Data, *Football data - historical soccer results and betting odds*. [Online]. Available: https://www.football-data.co.uk.

[37] G. Y. Lee, L. Alzamil, B. Doskenov and A. Termehchy, 'A survey on data cleaning methods for improved machine learning model performance,' *arXiv preprint arXiv:2109.07127*, 2021. [Online]. Available: https://arxiv.org/abs/2109.07127.

[38] U. Khurana, H. Samulowitz, F. Nargesian, D. Turaga and E. B. Khalil, 'Learning feature engineering for classification,' in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 2529–2535. DOI: 10.24963/ijcai.2017/352. [Online]. Available: https://www.ijcai.org/proceedings/2017/352.

[39] R. J. Janse, T. Hoekstra, K. J. Jager *et al.*, 'Conducting correlation analysis: Important limitations and pitfalls,' *Clinical Kidney Journal*, vol. 14, no. 11, pp. 2332–2337, 2021. DOI: 10.1093/ckj/sfab085. [Online]. Available: https://doi.org/10.1093/ckj/sfab085.

[40] J. Walters-Williams and Y. Li, 'Estimation of mutual information: A survey,' in *Rough Sets and Knowledge Technology*, Springer, 2009, pp. 389–396.

[41] M. A. Rahman, 'A deep learning framework for football match prediction,' *SN Applied Sciences*, vol. 2, no. 2, pp. 1–10, 2020. DOI: 10.1007/s42452-019-1821-5. [Online]. Available: https://link.springer.com/article/10.1007/s42452-019-1821-5.

[42] XGBoost, *Xgboost python package*. [Online]. Available: https://xgboost.readthedocs.io/en/stable/python/.

[43] O. Castro, P. Bruneau, J.-S. Sottet and D. Torregrossa, 'Landscape of high-performance python to develop data science and machine learning applications,' *ACM Computing Surveys*, vol. 56, no. 3, Mar. 2024. DOI: 10.1145/3617588. [Online]. Available: https://doi.org/10.1145/3617588.

[44] Pandas, *Pandas: Powerful data structures for data analysis and statistics*. [Online]. Available: https://pandas.pydata.org.

[45] NumPy, *Numpy: The fundamental package for scientific computing with python*. [Online]. Available: https://numpy.org.

[46] PyTorch, *Pytorch: An open-source machine learning library*. [Online]. Available: https://pytorch.org.

[47] scikit-learn, *Scikit-learn: Machine learning in python*. [Online]. Available: https://scikit-learn.org.

[48] J. D. Hunter, 'Matplotlib: A 2d graphics environment,' *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[49] M. Waskom, *Seaborn: Statistical data visualization*, https://seaborn.pydata.org/, Version 0.13.2, 2024.

[50] L. Zahedi, F. G. Mohammadi, S. Rezapour, M. W. Ohland and M. H. Amini, 'Search algorithms for automated hyper-parameter tuning,' *arXiv preprint arXiv:2104.14677*, 2021. [Online]. Available: https://arxiv.org/abs/2104.14677.