Deep Learning for Algorithmic Sports Betting

Ben Schlagman[1]

MSc Machine Learning

Academic Supervisor: Prof. Philip Treleaven

Technical Supervisor: John Goodacre

Submission date: 23/09/2024

Abstract

This thesis investigates Deep Learning (DL) for algorithmic sports betting. The original research presented in this thesis was conducted in collaboration with Quant Sports Trading (QST) and offers applied models for trading in real-time betting markets. This thesis focuses on sports betting through online exchanges. These exchanges, such as Betfair, revolutionised sports betting by adopting a structure closer to that of financial exchanges, enabling direct peer-to-peer trading, algorithmic access and allowing for order matching at millisecond speeds. While financial markets are characterised by assets with intrinsic value, stringent regulatory frameworks, and high liquidity, sports markets present a different landscape. These exchanges facilitate betting between participants using back and lay mechanisms, creating opportunities to profit from either side of a wager. While sports markets have traditionally been overlooked by major financial traders and hedge funds due to their smaller scale, lower liquidity, and slower velocity compared to financial markets, they also have the potential to become a major asset class. The rise of online exchanges has bridged the gap with the trading world, creating new avenues for original research.

The thesis consists of the following three experiments:

1. **Data Exploration and Infrastructure** The first experiment outlines the development of a research and machine learning platform. It details the integration of pre-game, in-play, and exchange data, the implementation of a cloud-native time series AWS database architecture, and design choices that optimise the researcher workflow, from data querying latency to feature creation and backtesting.

2. **Baseline Models for Pre-Game Football Predictions** This experiment develops a series of baseline value betting models to classify the outcome of upcoming football matches based on historical pre-game data. These models serve two purposes: first, to validate the research platform's capacity to handle large-scale data requests, and second, to provide a benchmark that will guide further refinements in the final experiment.

3. **Deep Learning for In-play Momentum Betting** The third and final experiment uses deep learning to implement an in-play momentum betting strategy. Leveraging the integrated data framework, stacked Long Short-Term Memory (LSTM) networks, attention-based transformers, and Convolutional Neural Networks (CNNs) analyse real-time exchange data to identify shifts in match momentum, enabling dynamic and fast-paced in-play predictions.

This thesis is believed to present the following original contributions to science:

1. **A Comprehensive Data Framework and Research Platform** This thesis establishes a robust framework for capturing, storing, and analysing sports exchange data, identifying unique patterns in sports trading. By combining in-play state-of-game data with exchange data, the framework creates synthetic features tailored for algorithmic trading models. The research platform enables easy data querying, feature creation, and extensive backtesting, providing a simulated environment for model training and testing, thus serving as a valuable tool for future research.

2. **Understanding of the Betting Market Microstructure** The microstructure of sports betting, unlike traditional financial markets, has been underexplored. This thesis provides a detailed examination of nuances such as artificial order lag, odds behaviour, and liquidity factors. This research enhances understanding of real-time bet matching and execution,

offering insights to optimise these processes. It sheds light on the unique characteristics of sports betting markets, contributing to more effective trading strategies and better market dynamics.

3. **Value Betting and Momentum Models** This thesis adapts machine learning to sports betting, first by developing a series of baseline pre-game value betting models to identify mispriced odds, and then by applying deep neural networks for in-play momentum betting. These approaches open new avenues for both research and practical application in the betting industry.

Publication

The research presented in this thesis has produced the paper "Review of Algorithmic Trading in Financial and Sports (Exchanges)" available on SSRN [15]. The paper provides an in-depth analysis of how algorithmic trading principles in financial markets can be adapted to the evolving landscape of sports exchanges.

Impact Statement

This thesis is believed to present the following contributions to industry:

1. **Development of a Commercial-Ready Software Tool** Beyond the research and contributions to science, the thesis outlines the implementation of a robust software tool that can be leveraged as a commercial product. In working with QST, the project remains focused on the design, implementation and testing of all infrastructure which comprises the ability to confidently begin to deploy strategies for the automated algorithmic trading of sports markets on Betfair. This means buying data, testing strategies, showing Betfair exchange book ladders, visualising trades, running strategies and seeing profit and loss. Beyond this, showcasing not just the data but statistics, analytics, backtesting tools as well as the ability for researchers to easily query data and iterate around strategy development.

2. **Proven Value Betting and In-Play Strategies** In addition to the machine learning platform, this thesis presents proven value betting and in-play strategies for football, derived using classification algorithms and deep learning. A handful of these models were tested using historical betting markets, generating profits similar to those measured during backtesting. By integrating high-frequency exchange data with in-play data, as demonstrated in the first experiment, this approach can assist punters and betting businesses like QST in more effectively managing inventory imbalances, thereby reducing risk.

3. **Industry-Wide Impact and Practical Applications** This research not only contributes to academic knowledge but also offers practical tools and insights for industry professionals. By refining trading strategies and increasing market efficiency, it has the potential to significantly enhance the sports trading industry.

Contents

1 Introduction **8**
- 1.1 Research Motivations . . . 8
- 1.2 Research Objectives . . . 9
- 1.3 Research Experiments . . . 9
- 1.4 Scientific Contributions . . . 9
- 1.5 Thesis Structure . . . 10

2 Background & Literature Review **11**
- 2.1 Algorithmic Trading . . . 11
  - 2.1.1 Data . . . 11
  - 2.1.2 Pre-trade Analysis . . . 12
  - 2.1.3 Execution Model . . . 12
- 2.2 Sports Betting . . . 13
  - 2.2.1 Bookmakers . . . 14
  - 2.2.2 Betting Exchanges . . . 14
  - 2.2.3 Characteristics of Trading on Betfair . . . 15
- 2.3 Football Betting Prediction . . . 16
  - 2.3.1 Value Betting . . . 16
  - 2.3.2 In-play Betting . . . 17
- 2.4 Deep Learning . . . 18
  - 2.4.1 Key Components of Deep Learning . . . 18
  - 2.4.2 Types of Deep Learning . . . 19

3 Data Exploration and Infrastructure **21**
- 3.1 Background . . . 21
- 3.2 Data . . . 21
  - 3.2.1 Data Collection & Analysis . . . 22
  - 3.2.2 Price Discovery . . . 24
- 3.3 Database Architecture . . . 26
  - 3.3.1 Pre-game Data Storage . . . 27
  - 3.3.2 In-play Data Storage . . . 28
  - 3.3.3 Exchange Data Storage . . . 28
- 3.4 Backtesting . . . 29
- 3.5 Results . . . 29
- 3.6 Discussion . . . 29

4 Baseline Models for Pre-Game Football Predictions **31**
- 4.1 Background . . . 31
- 4.2 Data . . . 31
  - 4.2.1 Data Exploration . . . 32
  - 4.2.2 Data Pre-Processing . . . 34
- 4.3 Training and Methodology . . . 38
  - 4.3.1 Model Selection . . . 38
  - 4.3.2 Training Methodology . . . 39
- 4.4 Testing and Evaluation . . . 39
  - 4.4.1 Evaluation . . . 39

4.4.2 Feature Selection . . . 40
4.5 Results . . . 41
4.6 Discussion . . . 42

5 Deep Learning for In-play Momentum Betting **44**
5.1 Background . . . 44
5.2 Data . . . 45
5.2.1 Data Exploration . . . 46
5.2.2 Data Pre-Processing . . . 47
5.3 Training and Methodology . . . 48
5.3.1 Model Selection . . . 49
5.3.2 Training Methodology . . . 51
5.4 Testing and Evaluation . . . 52
5.4.1 Evaluation . . . 52
5.5 Results . . . 53
5.6 Discussion . . . 55

6 Conclusion and Future Work **57**
6.1 Summary . . . 57
6.2 Contributions . . . 58
6.3 Future Work . . . 58

List of Figures

2.1 Algorithmic Investment Platform [25] . . . 12
2.2 Screenshot of a Betfair Match Odds . . . 15
2.3 Shallow Neural Network [31] . . . 19

3.1 Pre-game Footystats Data Visualisations . . . 23
3.2 Liquidity and Total Traded Volume for Tottenham, showing Back, Lay, and Total Volume in the Aston Villa vs. Tottenham Match . . . 26
3.3 Liquidity and Traded Volume for Tottenham, Illustrating Market Dynamics and Betting Activity Over Time during the Aston Villa vs. Tottenham Match. . . . 26
3.4 Diagram of Database Architecture: Integration of MongoDB and MongoDB Atlas Cluster on AWS, illustrating VPC Peering Connection across Multiple Availability Zones with NAT Gateways. . . . 27

4.1 Cumulative Goals vs Expected Goals (xG) . . . 33
4.2 Pearson Correlation Matrix . . . 34
4.3 Spearman Correlation Matrix . . . 35
4.4 Comparison of Historical Win Streaks for Manchester City and Everton . . . 37
4.5 Rolling Goal Difference for Last 3, 5, and 10 Games for Manchester City and Everton 37
4.6 Accuracy, F1, Precision and Recall results from models LR, XGB, SVM, RFC, MLP 40
4.7 Top 10 Features Importance from XGBoost . . . 41
4.8 Confusion Matrix for SVM on P10 Dataset: Performance Breakdown of Home Wins, Away Wins, and Draws Predictions . . . 43

5.1 In-play Best Available Back Price (ATB) Progression for Tottenham, Aston Villa, and The Draw during the Aston Villa vs. Tottenham Match. . . . 46
5.2 Training and Validation Loss During Model Training. . . . 51
5.3 Temporal Predictions for each Model and Runner in the Test Data. . . . 54
5.4 Residual Plots for each Model and Runner in the Test Data . . . 55

List of Tables

2.1 Example Odds, Payout and Probabilities 13

3.1 Data Providers Categorised by Sport and Type (Pre-game, In-play, and Exchange) for Football, Tennis, and Horse Racing. 23
3.2 Sample Selection of Markets Available on Betfair, Detailing the Number of Runners and Example Selections for Each. 24
3.3 Betfair Prices and Tick Sizes 25

4.1 Summary Statistics for EPL Matches 32
4.2 Dataset Split Across Different Periods for Model Training, Validation, and Testing 36
4.3 Model performance comparison across different datasets 42

5.1 Fields Required for Constructing the Order Book in Exchange Data, Including Their Codes and Descriptions 45
5.2 Limit Order Book Price Ladder 46
5.3 Dataset Distribution and Purpose 48
5.4 Performance Metrics for LSTM, AS LSTM, and Transformer Models 53
5.5 CNN Performance Metrics for each Runner. 54

Chapter 1

Introduction

This first chapter provides an overview of the thesis by examining all motivations for the research problem, outlining the objectives, experiments, and contributions of the study, then concluding with the thesis structure. The text provides background information on the main topics: financial markets, algorithmic trading, sports betting and machine learning.

1.1 Research Motivations

This thesis is motivated by the unsaturated nature of the betting market, providing unique opportunities for both original research and commercial endeavours. The first stock exchange was established in Amsterdam in 1602, facilitating the trading of shares of the Dutch East India Company. Now, modern financial markets are characterised by advanced technology, regulatory frameworks, and a diverse range of financial instruments. In the data-rich and relatively idiosyncratic exchange traded classes like equities and futures, all areas of computational statistics, AI and ML have been well-applied [22]. With the sheer volume of data, computing power, and competition in financial markets, it has become increasingly difficult to gain a competitive edge.

While sports markets have traditionally been overlooked by major financial traders and hedge funds due to their smaller scale, lower liquidity, and slower velocity compared to financial markets, they also have the potential to become a major asset class. Sports betting markets remain relatively underexplored and underresearched, offering fresh opportunities for innovation and discovery. Given enough liquidity and participants then in the future they also offer an uncorrelated asset class, recession proof and unaffected by anything financial markets related.

Over the last 20 years, sports trading has experienced dramatic changes, transitioning from traditional bookmakers to modern exchanges like Betfair. Betfair, established in 2000, revolutionised sports betting, providing the simple mechanism of matching back and lay bets live between market participants. Prices reflect implied probabilities rather than intrinsic values. This distinction results in unique trading dynamics, risk profiles, and regulatory challenges. From a trader's viewpoint there are some obvious implications, such as the ability to trade 'runners' in-play, similar to intra-day trading in finance. Unlike traditional sportsbooks, which may have adversarial relationships with skilled traders, exchanges theoretically offer a degree of independence, being more interested in the liquidity of the event than reducing their exposure.

Financial applications like algorithmic trading pioneered machine learning research and offered great promise for applications in market-making and automated trading in capital markets. Although there is extensive academic research on financial markets, the application of machine learning in real-time trading for sports betting is still largely unexplored. This thesis will explore the evolution of machine learning models in sports trading, contrasting early models with advanced techniques such as deep neural networks, highlighting how modern algorithms can leverage timely data to predict market trends and optimise trading strategies.

1.2 Research Objectives

The main research objective is to implement a platform where researchers can easily query data, create features for model training and both back and bulk backtest trading strategies. On completion the next objective is to use the platform to implement and evaluate pre-game value betting models and deep learning in-play momentum betting. The research platform should include abstraction layers to manage large volumes of data from various sources seamlessly. For researchers, this facilitates rapid ideation by providing efficient and expressive data querying and filtering capabilities when constructing models.

A significant objective in constructing a robust data infrastructure is to gather and unify football in-play state-of-game data, high-frequency exchange data, and historical match information. This consolidated dataset serves as the foundation for model training, allowing models to gain an in-depth understanding of market dynamics and specific match events, such as momentum shifts, drifts, and abrupt changes in odds. Although these three data sources are distinct and come from different providers, they must be seamlessly integrated. Additionally, when trading live, this integration must be performed quickly enough to inform live models effectively.

With a solid grasp of the various betting markets in football and the development of the necessary data and research infrastructure, the next steps involve implementing a series of baseline pre-game models for value betting prior to kick-off, followed by a set of in-play deep learning models. The baseline models are established firstly to test the data platform and secondly to provide a benchmark for comparison against more advanced machine learning models, assessing metrics such as the models' relative Sharpe Ratio, Sortino Ratio, and PnL. The focus should be on the adaptability of machine learning models to the dynamic betting environment, as well as their effectiveness in managing risk and inventory.

1.3 Research Experiments

This thesis is split into three systematically ordered investigations, each investigating a different area of algorithmic sports trading, with each study building upon the findings of the previous one.

1. **Data Exploration and Infrastructure.** In the first experiment, we developed a research platform that integrates in-play state-of-game data with Betfair's exchange data. To facilitate efficient data retrieval and analysis, we employed a cloud-based NoSQL time series database using MongoDB, hosted on Amazon Web Services. This database underpins the machine learning platform, allowing QST researchers to process high-frequency data and develop initial features for their models.

2. **Baseline Models for Pre-Game Football Predictions** This experiment uses the research platform to build a series of pre-game baseline models that classify the outcome of upcoming football matches based on historical match data. This tests the robustness of the machine learning platform and provides an opportunity for further refinement before moving to the final experiment.

3. **Deep Learning for In-play Momentum Betting** This study implements an in-play momentum betting strategy using deep learning, specifically stacked LSTM networks, attention-based transformers, and CNNs. These insights inform the development of deep learning models that identify shifts in match momentum, enabling dynamic in-play betting strategies.

1.4 Scientific Contributions

This thesis is believed to present the following original contributions to science:

1. **A Comprehensive Data Framework.** Establishing a robust framework for capturing and storing sports exchange data, while identifying distinctive characteristics and patterns unique to sports trading. This thesis integrates in-play data with millisecond-level exchange data to

create derived features for trading models. This data framework provides a novel approach to collecting and storing pre-game, in-play, and exchange data, offering fresh perspectives and new opportunities for other researchers to build on.

2. **Research and Machine Learning Platform.** The platform will allow the easy querying of data, creation of features for model training and both back and bulk backtesting. This simulated environment specifically designed for sports trading is pivotal for training and testing models and something other researcher will use to advance the field further.

3. **Understanding of the Betting Market Microstructure** As a relatively unexplored area, the microstructure of sports betting has not been thoroughly studied. This thesis offers a detailed analysis of the subtleties and irregularities within the betting market, such as artificial order lag, to improve and streamline order execution.

4. **Value Betting and Momentum Models** While Machine Learning models have long been used to trade financial instruments, this thesis applies them to a new field by adapting ML specifically for sports betting. We start by defining a series of baseline models for pre-game value betting, then use the previous insights and tools to develop and evaluate an in-play momentum betting model using deep neural networks.

1.5 Thesis Structure

The structure of this thesis is organised as follows:

- **Chapter 2 - Background and Literature Review** This chapter explores the background and literature relevant to the thesis, focusing on algorithmic trading, sports exchanges, betting, and the application of machine learning for football predictions. The literature review aims to provide a comprehensive understanding of the relevant academic literature and concepts in sports betting whilst examining existing methodologies.

- **Chapter 3 - Data Exploration and Infrastructure.** This chapter presents experiment one, the development of a research and machine learning platform. This includes the integration of different data sources, the AWS database infrastructure, the implementation process and design choices that optimise the researcher workflow, data querying latency and usability of the pipeline.

- **Chapter 4 - Baseline Models for Pre-Game Football Predictions.** In this chapter, we test the platform created in the prior experiment by modelling team performance and developing a range of baseline Machine Learning models to accurately predict the outcome of matches pre-game.

- **Chapter 5 - Deep Learning for In-play Momentum Betting.** In this chapter we drive into the final experiment, bring together everything learnt from the prior two to implement an in-play betting model. We use exchange data to forecast in-play events and build momentum based betting model. We explore the implementation of stacked LSTM networks, attention based tranformers and CNN architectures commonly used in trading.

- **Chapter 6 - Conclusion and Future Work.** The final chapter provides an overall conclusion of the research and main results from each chapter with respect to key findings from each model. We highlight the achievements and contributions of this thesis, and provide recommendations for future work to be carried on by other researchers.

Chapter 2

Background & Literature Review

This chapter explores the background and literature relevant to the thesis, focusing on algorithmic trading, sports exchanges, betting, and the application of machine learning for football predictions. The literature review aims to provide a comprehensive understanding of the foundational theories and concepts in sports betting whilst examining existing methodologies. By exploring how machine learning techniques have been applied, this chapter identifies research gaps and sets the stage for subsequent experimental investigations and discussions.

2.1 Algorithmic Trading

Algorithmic trading automates trading processes by leveraging computational power, mathematical models, algorithms, and machine learning to make optimal transaction decisions [28].

A typical algorithmic investment platform determines which securities to buy or sell, the optimal timing, and their quantities. This process follows a five-stage pipeline [22], illustrated in Figure 2.1. The first stage, 'Real-time/Historical Data' involves acquiring, refining, and structuring the necessary data. 'Pre-trade Analysis' then focuses on analysing this data to identify potential trading opportunities. The 'Trading Signal' stage determines what to buy or sell given the analysis from the previous stage. 'Trade Execution' involves placing orders to construct the portfolio derived from the trading signal then lastly, 'Post-trade Analysis' evaluates the results of the trading activities using metrics such as P&L. [37].

Within these, the Alpha, Risk, and Transaction Fee models inform the Pre-trade Analysis, with their results being integrated into the Portfolio Construction Model, which then is responsible for generating trading signals. The output is then used by the Execution Model to carry out the trades. For the purpose of this thesis, the focus is on Data, Pre-trade Analysis and the Execution Model.

2.1.1 Data

Data collection in algorithmic trading involves gathering information from various sources that can dictate prices, manage risk, and develop a more precise model of the trading environment. Narang [26] highlights that the success of a trading system depends highly on the availability of abundant and frequently updated 'clean' data for analysis . Traditional sources of data, such as exchange or order book data captures the state of the exchange at millisecond timestamps. The order book acts as the public interface for tracking buy and sell offers submitted by market participants for listed securities. Orders are aggregated and sorted into a price ladder using three key data points: price, volume, and time of arrival

Beyond financial data, economic data includes factors like consumer confidence, inflation rates, trade balances, and government fiscal policies. Social data includes sentiment scraped from social media platforms (e.g., Reddit, X). Similarly, this data can be real-time, such as live feeds from

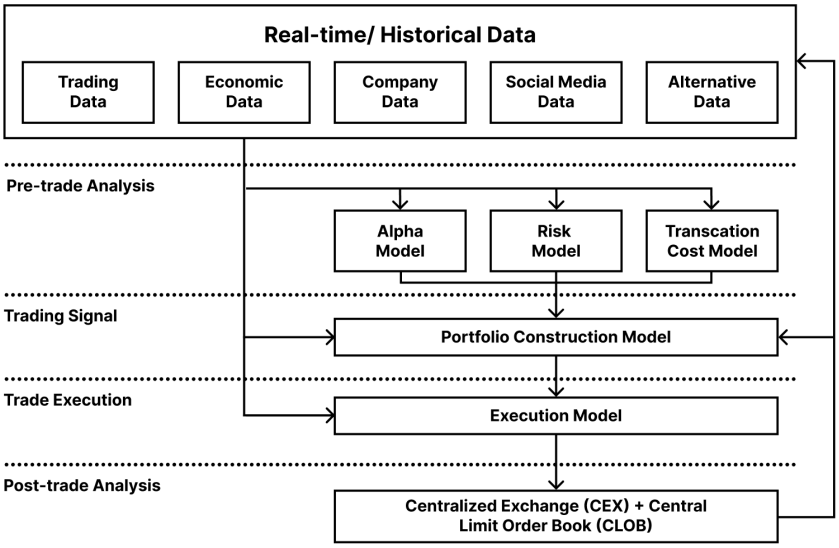

Figure 2.1: Algorithmic Investment Platform [25]

ECNs, financial news services, and social media streams, or it can be historically collected and purchased from data providers. Additionally, data can be classified based on its processing state: raw data, which is unprocessed and may contain errors or be unanalysed; cleaned data, which has been corrected for errors and inconsistencies; and analysed data, which has been processed to emphasise principal components relevant to specific contexts.

Both real-time and historical data sources are crucial for the research, design, and backtesting of trading algorithms. They play a significant role in guiding the decision-making processes of all components within an algorithmic trading system.

2.1.2 Pre-trade Analysis

The Alpha model takes cleaned data from the aforementioned sources to identify market inefficiencies that might be exploited in live trading [37]. The three principal techniques are, fundamental analysis, technical analsyis and quantivitve analysis. The latter applies a range of computational statistics, physics, and ML to exploit features of the data collected. Generally, this includes differential calculus in models such as the Black-Scholes for option pricing and stochastic differential equations for modeling asset prices [3].

The Risk model [4] assesses potential risks associated with each trading position by analysing factors such as market volatility, inter-market correlations, and exposure to various sectors or asset groups. It oversees the strategy's risk exposure through risk management techniques, such as those found in Monte Carlo Simulations [13]. These are sometimes integrated directly into the Alpha model to enhance predictive accuracy and operational safety.

Finally, the Transaction Fee model calculates the costs of trading activities, factoring in elements like bid-ask spreads and order sizes, as well as potential hidden costs, like market impact and slippage. By incorporating these costs, the trading strategy aims to optimise profitability while minimising expenses.

2.1.3 Execution Model

The execution model handles the implementation of trades necessary to form the portfolio devised by the portfolio construction model. It takes into account factors like market liquidity and focuses on reducing both market impact and transaction costs. The execution model decides the optimal

timing, size, and method of trade placement to reduce slippage and ensure the trades are carried out efficiently [1]. This model considers various execution strategies, such as VWAP (Volume Weighted Average Price), TWAP (Time Weighted Average Price), and implementation shortfall, to achieve the best possible execution prices while managing the trade-off between speed and cost.

2.2 Sports Betting

One way in which sports betting is different from many other types of betting, from a statistical point of view, is that the probabilities cannot be fully specified [2]. In many casino games, both the gambler and the bookmaker have complete knowledge of the probabilities involved. However, in the context of sporting events, there are numerous factors that influence the probabilities, making it difficult to precisely quantify their impact. These dynamics mean that there isn't an inherent bias in favor of the bookmakers; consequently, gamblers have the potential to profit over the long term.

Punters (participants) place bets on their perceived outcomes, either before or during an event, on predefined markets, wagering money based on those predictions. Markets are linked with specific events (e.g., football or tennis match) and include odds for all types of predictions such as match winners, goal scorers, and specific scores. In betting, these prices are called 'odds', with the most frequent being fixed-odds bets, where the bookmakers offer their own odds on any range of markets [19]. Odds indicate both the potential winnings from a bet and the estimated chance of the event taking place.

Odds are inversely related to the probabilities of specific outcomes. Bookmakers will attempt to evaluate the likelihood of these various outcomes to then determine their corresponding odds [39]. Odds can be expressed in various formats, including decimal (e.g., 0.4), fractional (e.g., 3/2), and moneyline (e.g., +150 or -200) [12]. Throughout this thesis, odds are presented in decimal format. Fractional odds are more commonly used in the UK, while moneyline odds are prevalent in the US. Moneyline odds can be less intuitive, positive values indicate the potential profit from a $100 wager, and negative values indicate the amount needed to wager to win $100. Despite these different notations, all formats can be easily converted into one another and convey the same underlying information.

Consider the following example involving a football match between Home and Away teams. The match odds are given in Table 2.1, along with total payout and implied probability. The formulas for calculating the total payout (TP) and implied probability (IP) are given by $\text{TP} = \text{odds} \times \text{wager}$ and $\text{IP} = \frac{1}{\text{odds}}$, respectively.

Runner	Odds	Total Payout on £100	Implied Probability
Home	5.0	£500	20%
Away	1.25	£125	80%
The Draw	3.0	£300	33.33%

Table 2.1: Example Odds, Payout and Probabilities

The following chapters are in line with the research that has examined a range of quantitative approaches for modeling betting markets and creating predictive statistical models. To mention a few, Dixon and Coles applied statistical models to assess scoring in football and explore inefficiencies in betting markets [10]. In 2010, Croxon and Reade conducted the first study into price (i.e., odds) efficiency for 'in-play' markets and found that the betting exchange leads the process of information aggregation and consequently price discovery [7]. Since the 2010 Croxon and Reade study, a significant portion of research on systematic sports betting has focused on betting exchanges, including data, efficiency, and microstructure.

2.2.1 Bookmakers

Up until the late 1990s, the vast majority of bets were placed with traditional bookmakers who would quote prices for events of popular interest [7]. Bookmakers accept bets on events at predetermined odds and function similarly to market makers in financial markets by facilitating liquidity. They do this by consistently quoting prices at which they are willing to place bets [35]. Odds are considered fair when the total implied probabilities of all possible outcomes in a market add up to one. In such a case, if a bettor wagers on every possible outcome, they would receive their entire stake back. However, to ensure a profit, bookmakers set the odds so that the combined implied probabilities exceed one. The excess percentage is known as the "Overround" (in the UK) or "Vigorish" (in the US) and it is the reason for the long-term profit of the bookmaker [38]. Bookmakers set the odds to create an overround, ensuring they maintain a margin on the total amount wagered, typically between 5% and 10%. They do this by offering odds that are slightly lower than the true probabilities of the outcomes. This margin covers their operating costs and generates profit.

2.2.2 Betting Exchanges

Over the last 20 years, sports trading has experienced dramatic changes, transitioning from traditional bookmakers to modern exchanges like Betfair. Betfair revolutionised sports betting by moving the traditional betting environment to one akin to financial exchanges, thus enabling direct peer-to-peer trading, algorithmic access and allowing for order matching at millisecond speeds. There are a variety of exchanges, such as Betfair, Betdaq and Smarkets. At present Betfair are the most liquid and therefore for the purposes of this thesis we will concentrate on Betfair as an archetypal sports exchange.

Betting exchanges have many similarities with which traditional financial markets [23], are built on. In general trading platforms allow participants to execute trades anonymously, including buying or selling equities and commodities, through computer networks. This has lead to transformative change in trading strategies, and market efficiency. Today, these exchanges support algorithmic trading, where algorithms can submit orders in milliseconds, model market behavior statistically, and operate within predefined risk parameters. Recently, there has been heightened interest in ML models for their capability to simulate trading environments, adapt dynamically, and improve through historical data analysis. This has led to a competitive drive for advanced data collection and trading infrastructures. The success of advanced algorithms in capital markets has thus driven significant competition and research in other order book-oriented markets, including cryptocurrencies and sports betting.

The introduction of betting exchanges, especially Betfair, has created new opportunities for bettors and small-scale traders. These exchanges have also attracted high-volume quantivitve traders who buy and sell odds much like one would handle stocks or commodities. The exchange allows individual punters to bet with each other directly. Given two participants: A and B. Participant A bets that a certain event will occur, while Participant B bets against it. In a fixed-odds bet, if the event doesn't happen, A pays B an agreed sum. Conversely, if the event does happen, B pays A a predetermined multiple of the stake based on the fixed odds they set. This allows participants to bet on market outcomes, as in traditional bookmaking, but also to bet against them. Exchanges differ from bookmakers in that they do not bear direct risk or determine the odds. Instead, they serve as intermediaries, connecting individual bettors who wish to place opposing bets. Their revenue model is based on charging commission fees on the net winnings of successful bets, which ensures that they remain neutral.

To represent the scale, Betfair processes over six million trades a day—six times the number of trades on the London Stock Exchange and more than all the European stock exchanges combined [16]. Betfair has become the largest in terms of transaction volume, frequency and international presence, resulting in generally high liquidity levels across its markets. Figure 2.2 shows a image of a typical Betfair market, specifically a Match Odds market, for the "Germany vs Scotland" football match.

Figure 2.2: Screenshot of a Betfair Match Odds

2.2.3 Characteristics of Trading on Betfair

There are some clear market differences between financial and sport trading worth mentioning. For most financial markets, such as equities, assets themselves typically hold intrinsic values and may generate dividends. Ownership of these equities also offers investors certain legal protections. The assets themselves exist in a competitive marketplace, where values may be influenced by macroeconomic trends or other external factors. On the other hand, sports markets lack an underlying asset with intrinsic value. Here, the prices are purely reflective of the implied probabilities of outcomes, leaning more towards speculation than investment, and are therefore governed by a different set of regulations. A typical sports exchange, such as Betfair, has a relatively simple set of participants: retail traders, bots, professionals (including bookmakers), and the exchange itself. Participants must be aware of liquidity within their markets to effectively scale their products and mitigate market impact. While financial markets like the S&P 500 can see daily volumes of 2 to 6 billion shares, worth hundreds of billions to over a trillion dollars, sports exchanges operate on a much smaller scale. In terms of liquidity measures like immediacy, depth, resilience, breadth, and tightness, sports exchanges are significantly smaller and thus less attractive to institutional players.

The greatest difference compared to bookmakers is the back and lay mechanism which allows bettors to wager either in favor of or against a particular runner, and to place both market and limit orders. In this study, a market is defined as a platform where individuals and entities trade fungible items at prices determined by supply and demand, representing numerous interactions among participants. For a trade to be completed, two opposing bets must be placed and matched. Once matched, the trade is finalised, resulting in either a profit or a loss. One bet corresponds to a winning outcome and the other to a losing outcome. If the stake is S, and back odds B, then one can represent the profit using Eq. (2.1), as a tuple (profit if runner wins, profit if runner loses):

$$\text{BackProfit} = (S \times (B-1), -S) \tag{2.1}$$

Lay betting is the opposite of back betting, where a bettor bets against a specific outcome. If the outcome does not occur, the bettor wins the bet and receives a payout from the person who placed the back bet. This is given by Eq. (2.2).

$$\text{LayProfit} = (-S \times (B-1), S) \tag{2.2}$$

Profit or loss is calculated using Eq. (2.3), representing the profit from the winning bet minus the loss from the losing bet.

$$\text{PL} = \text{Profit of winning bet} - \text{Liability of losing bet} \tag{2.3}$$

This capability enables participants to actively trade outcomes while an event is ongoing, facilitating dynamic risk management and profit-locking strategies in response to live price fluctuations. This leads to the concept of greening. Greening is a popular trading strategy on sports exchanges that involves hedging a position to secure the same guaranteed profit regardless of the event's outcome. This technique can be likened to hedging strategies used in financial markets, where

traders aim to lock in profits or limit losses by balancing positions in different assets.

If v_2 is the volume of a back order and p_2 the price, with v_1 the volume of a lay order and p_1 the price, then the combined exposure of the orders is:

$$\text{SpreadProfit} = (v_2 \times (p_2 - 1) - v_1 \times (p_1 - 1), v_1 - v_2) \tag{2.4}$$

Thus if a trader knew the back price and volume he was willing to enter and wished to achieve an x% gain regardless of outcome. Then the lay volume he would need to place would be:

$$v_1 - v_2 = x \times v_2 \Rightarrow v_1 = v_2 \times (x + 1) \tag{2.5}$$

In a lay bet, the liability refers to the amount owed to the bettor who placed a back bet on the same event if the outcome occurs. The liquidity of the market influences how easily punters can place back or lay bets at various prices. When sufficient liquidity exists at a specified price and a bet is entered into the order book, the bet is considered matched, meaning another participant has taken the opposite position in the trade. The back/lay spread, which is the difference between the highest available back price and the lowest available lay price, is similar to the bid/ask spread in financial markets. This spread represents the cost of executing a trade and indicates the potential for profit or loss.

In contrast to traditional financial markets, betting exchanges like Betfair are uniquely influenced by market suspensions and the finite nature of sports events. Market suspensions can be triggered by significant in-game events such as goals or fouls. Such suspensions temporarily freeze trading, fixing the current positions and preventing any alterations or new bets. This can significantly influence trading strategies, particularly for traders who rely on real-time, in-play betting to capitalise on event developments. Additionally, the definitive timeframes associated with sports events set a clear boundary for trading activities. Unlike financial markets where positions can be modified over extended periods, bets on Betfair must be concluded by the end of the event.

2.3 Football Betting Prediction

For years, forecasting the result of a full-time football match (whether a home win, away win, or draw) or creating models to predict expected goals and other related statistics has been researched. These models are generally divided into two types: pre-game and in-play. Pre-game models, which are often employed for strategic betting before the game starts, focus on historical data aggregations such as team line-ups, average shots on target, number of cards, or average distance covered per game [34]. In contrast, in-play models use a continuous data stream that could include real-time updates on key match events like goals and substitutions, or even more detailed data such as microsecond order book information from betting exchanges. These models benefit from the capability to dynamically adjust to market conditions and enhance price discovery as fresh data comes in [8]. Consequently, in-play models are capable of placing thousands of bets during a match and securing a profit before the game ends. Although traditional value betting strategies lack this dynamic capability, recent studies have shown that combining pre-game and in-play models can effectively manage risks and detect when odds are misaligned.

2.3.1 Value Betting

Value betting involves identifying bets where the odds offered by the bookmaker are higher than the true probability of the outcome. This is analogous to value investing in financial markets, where investors seek undervalued stocks that are priced below their intrinsic value

Early statistical models in sports betting focused on modelling price behaviour. The Dixon and Coles [10] model is a recognised approach for predicting football match outcomes using a Poisson arrival process, which demonstrated predictable price drifts as time passed without a goal. This approach uses a Poisson regression method to adjust for variations in team strengths and home-field advantage. While addressing common issues in Poisson models, such as the underestimation

of low scores and overestimation of high scores, the model aims to exploit inefficiencies in the betting market. Despite its simplicity, the model effectively generates accurate predictions for closely contested matches. However, its assumption of static parameters over time has prompted a shift towards incorporating more dynamic models that can adapt to the evolving nature of teams' performances throughout a season. The double Poisson model proposed by Karlis and Ntzoufras [21] considered how the opponents' XGA (Expected Goals Against) or defensive record can influence the number of goals scored. The model accounts for factors like team strength, home-field advantage, and the relationship between goals scored by both home and away teams. This addition leads to better model fitting and more accurate predictions, especially in representing the frequency of draws, an outcome often underrepresented in other models.

Extending these original algorithms, more recent research explores deep learning, bayesian networks, and the integration of bookmakers' odds. Zhang et al. [40] proposed the AS LSTM model, which integrates an LSTM model with an attention mechanism. They used this model to predict football match outcomes by employing a sliding time window and incorporating historical match data from the teams. This approach showcases the potential of more sophisticated machine learning models in forecasting sports results. Constantinou et al. introduced the pi-football model, employing a Bayesian network to construct and update team ratings [6]. Conversely to other approaches relying solely on pre-game data, this model integrates synthetic features that capture game nuances and dynamics. Originally developed for chess, the Elo rating system has been adapted for sports like football to gauge team skill levels, proving effective in match outcome prediction [18]. Constantinou and Fenton expanded upon this with the Pi ratings system, which considers variables such as match location (home, away, or neutral), goal difference, and recent match history. Egidi et al. were the first to integrate bookmakers' odds to enhance match outcome predictions. Their hierarchical Bayesian Poisson model estimated scoring rates for each team using both historical data and bookmaker odds [11].

To further develop our understanding, we need models that can adeptly manage the inherent unpredictability and randomness in football games, leading to more sophisticated value betting strategies. Szymanski and Kuypers developed a strategy that leverages discrepancies between the actual probabilities of match outcomes and the odds set by bookmakers [36]. Their approach involves betting on outcomes that are undervalued by the market, with the goal of achieving a positive expected return. This strategy is rooted in the efficient market hypothesis, which suggests that consistently beating the market average return without taking additional risks is unlikely. Despite its straightforward premise, this strategy has proven to be effective in certain betting markets.

2.3.2 In-play Betting

In-play betting adds a dynamic component to wagering, where odds and opportunities shift throughout the course of the match. Momentum betting is a type of in-play betting that involves betting on teams or players that are currently performing well, with the expectation that their good form will continue. Alternatively one may examine in-play prices and analyse them looking for price and volume trends. This is similar to momentum trading in financial markets, where traders buy stocks that have been rising in price, expecting the upward trend to continue. Vizard [11] provides a comprehensive study of momentum betting on sports exchanges.

The literature on in-play betting often parallels that of algorithmic trading in financial markets, reflecting a common structured and high-frequency approach used in both the betting and financial exchange limit order books. This similarity makes it conducive to applying time series analysis and machine learning algorithms in both contexts. Deep learning methods have been used in algorithmic trading, as demonstrated by Dixon et al., who developed a Deep Neural Network (DNN) featuring five hidden layers to trade foreign exchange (FX) and commodity instruments using 5-minute interval prices [9]. They also employed stacked LSTM networks, a variant of recurrent neural networks (RNNs), to forecast the mid-price movements in limit order books. Their findings indicated that LSTM networks surpassed other machine learning models in both predictive accuracy and trading performance.

Øvregård explored the use of Neural Networks (NN) for in-play tennis, using a Multilayer Perceptron (MLP) architecture [29]. Instead of predicting specific prices, Øvregård's study aimed to identify profitable trading opportunities using the MLP to approach in-play betting as a regression problem, rather than a classification one. This method allowed the MLP to quantify the extent and confidence of price fluctuations, rather than merely indicating whether prices would rise or fall. Divos was one of the first to provide a comprehensive analysis of the in-play football betting market, demonstrating that many principles from financial mathematics are transferable to betting. Notably, he introduced the Constant Intensity Model, a risk-neutral method akin to the Black-Scholes model for pricing and hedging in-play bets. Divos also evaluated the effectiveness of K-Nearest Neighbour, Linear, and Neural Network (NN) models in predicting full-time scores based on half-time in-play statistics [8]. His analysis showed that while all three models had similar performance in terms of log-likelihood, the NN model, with its large number of parameters, was considered the least preferable due to its complexity. In addition, Rahman [32] proposed a football game prediction framework based on LSTM architecture, achieving a 63.3% accuracy. Johnson et al. [20] used tactical feature engineering to compress space-time and promote the fine-tuning of three pre-trained CNNs.

2.4 Deep Learning

Deep neural networks are a category of machine learning models that, when trained with data, utilise a process called deep learning [31]. These models have become highly valuable in predicting events due to their capability to handle complex, nonlinear scenarios. They are particularly effective in identifying intricate relationships among variables and in optimising various objectives [27]. Deep learning models are widely used in everyday technologies, significantly contributing to advancements in natural language processing, computer vision, and reinforcement learning.

2.4.1 Key Components of Deep Learning

Deep learning is centred around Neural Networks. A NN is a class of algorithms composed by MP-Units (i.e. neurons) firstly introduced by Mcculloch and Pitts [24]. The formula of a single neuron C is written as Eq. (2.6)

$$C(X, \theta) = \phi\left(\sum_{n=1}^{N}(w_n \cdot x_n)\right) \tag{2.6}$$

In this equation, C is the output of the neuron, X represents the input vector, θ denotes the parameters of the neuron, w_n are the weights, x_n are the input features, N is the number of inputs, and ϕ is the activation function applied to the weighted sum of the inputs.

The activation function allows the model to describe non linear relations between input and output. The ReLU function has been used as far back as 1969, however in the early days of NNs, it was more common to use the logistic sigmoid of tanh activation functions. A neural network consists of multiple layers, each containing several neurons. The main goal of a neural network is to create a mapping function, $\hat{f}$, which accurately approximates the true target, f, by minimising the error function, J. In deep learning, the core method for determining the optimal parameters, θ, is known as backpropagation [33]. Backpropagation aims to calculate the derivatives of the loss l concerning each weight and bias, and then propagate the error from the output layer back to the input layer, adjusting the weights according to an optimisation rule.

The loss function, denoted as J, varies depending on the specific problem being addressed. In this study, which involves time series classification with N output classes, the model assigns a probability to each class n. A common loss function used in this context is categorical cross-entropy, also known as logarithmic loss.

$$J(f, \hat{f}) = -\sum_{n=1}^{N} f_n \ln(\hat{f}_n) \tag{2.7}$$

The function returns a number that describes the mismatch between the model predictions and their corresponding group truth values. During training we seek parameter values that minimise the loss and hence map the input to output as closely as possible.

Neural Networks become 'deep' when they have more than one hidden layer. The hidden layers are between the input layer and the output layer, the more of them there are the more deep the NN becomes. A shallow network consists of just on input layer, a hidden layer and an output layer. This is illustrated in Figure 2.3

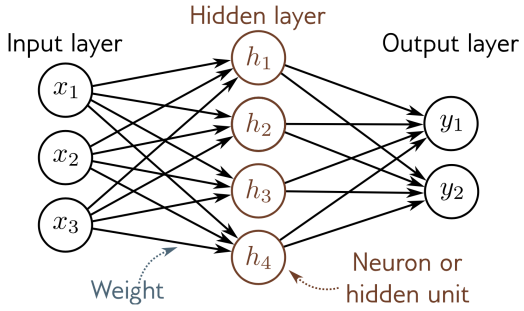

Figure 2.3: Shallow Neural Network [31]

One of the challenges in neural networks is overfitting, where the network excessively memorises the training set details, resulting in poor performance on the test set. Overfitting occurs when there are too many parameters compared to the number of training samples. Regularisation techniques are employed to prevent overfitting. One significant method is dropout,which applied to intermediate layers, allows the network to ignore a random set of neurons from each training update, reducing over-dependence.

2.4.2 Types of Deep Learning

A Deep Neural Network Classifier (DNNC) is a form of Feed-Forward Neural Network (FFNN) that includes multiple fully connected intermediate layers. In an FFNN, the layers are sequentially arranged, with each unit in a layer connected to the units in the preceding layers. The DNNC serves primarily as a standard reference point when compared to other deep learning neural network architectures.

RNNs, in particular, Long Short-Term Memory (LSTMs) are modern approaches for time series forecasting [14]. To overcome the vanishing gradient problem inherent in traditional RNNs, LSTMs use a unique cell state mechanism complemented by three specific gates: input, forget, and output. The efficiency of RNNs is due to their recurrent connections, which enable the network to utilise the history of previous time series values. These networks are distinguished by looped connections that introduce feedback and memory, allowing them to learn and generalise across sequences of inputs rather than focusing on individual patterns. The GRU, introduced by Cho et al. [5], is a variant of the LSTM that aims to improve computational efficiency by merging the forget and input gates into a single 'update gate'. Both LSTM and GRU architectures offer a deep learning framework capable of capturing complex temporal dependencies in football match data.

CNNs are a class of deep learning models particularly effective for tasks involving spatial data, such as image and video recognition. Unlike traditional feed-forward neural networks, CNNs employ convolutional layers to autonomously and adaptively learn spatial feature hierarchies from input data. These layers utilise a sequence of filters to identify patterns, edges, and textures within the data. The resulting features are then processed through pooling layers, which reduce dimensionality and decrease computational complexity. In the context of time series, CNNs operate differently from their application in image processing. Instead of detecting pixel correlations, they focus on identifying relationships between indicators and time segments. By doing so, CNNs can generate

filters that encapsulate particular patterns within the time series data, enabling them to effectively forecast future values based on these learned patterns.

Chapter 3

Data Exploration and Infrastructure

This chapter outlines the development of a research and machine learning platform. It includes the integration of pre-game, in-play, and exchange data, as well as the AWS database infrastructure. Additionally, the implementation process and design choices are discussed, focusing on optimising the researcher workflow, data querying latency, and the usability of the pipeline. This also covers the creation of features for model training, along with both backtesting and bulk backtesting.

3.1 Background

The foundation of any successful algorithmic trading platform in the domain of sports betting hinges on the effective utilisation and management of diverse data streams. In this context, data is not merely a component but the driving force behind the entire trading process. The quality, structure, and accessibility of data directly influence the accuracy of predictive models and the success of trading strategies. This chapter delves into the essential components required to build a robust research platform, focusing particularly on the integration of pre-game, in-play, and exchange data. These data types form the backbone of the analytical processes, allowing researchers to develop, test, and refine predictive models that are essential for deploying profitable trading strategies. By creating a system that seamlessly incorporates these varied data sources, the research platform enables the comprehensive analysis needed for both pre-game and in-play predictions.

A key aspect of this integration is the underlying data infrastructure, which is designed to handle large volumes of data efficiently while maintaining the flexibility to adapt to the specific needs of different sports. The chapter explores the implementation of a dual-database approach, utilising both SQL and NoSQL databases, to accommodate the structured and unstructured nature of the data. This setup not only ensures the rapid querying of historical data but also facilitates real-time data processing, which is crucial for in-play betting strategies. By establishing a scalable and secure environment, the platform supports the rigorous demands of backtesting and bulk testing, thereby providing a solid foundation for subsequent chapters that focus on model development.

3.2 Data

A typical workflow for a researcher would involve using data from a historical database to test market efficiencies and attempt to create predictive features based on an envisaged strategy. Models would then be trained, and the testing process for these models would not just involve historical statistics but also a full backtest, as if replaying the historical data live. Running a series of backtests and analysing the distribution of outcomes will provide insights into the strategy's effectiveness, ready to then be deployed live.

To bring this workflow to life, the first stage of any algorithmic trading platform focuses on the data. In sports betting, data can be sourced from three primary categories: pre-game, in-play, and exchange data. This chapter details the creation of an environmental test bed that allows researchers to quickly query pre-game data or filter specific aspects of in-play games across different matches. This setup is designed to facilitate the easy development of features and machine learning models, which can then be backtested and bulk backtested. The data and its underlying infrastructure are crucial for pre-game research, post-game analysis, in-game strategy execution, queries, and analytics. They form the backbone of the system and will determine the success of the following chapters.

Pre-game Data

Pre-game data contains historical matches where each match has a set of basic event data and post-processed statistics. It is referred to as pre-game because the post-processed data becomes available for querying before the next match that is being modelled takes place. Pre-game is more structured than the other two, it can be thought of as one table, where each row represents a match, and columns are complied from event data (e.g., shots on target).

In-play Data

In-play data refers to the real-time state of the game as it unfolds during an event. In football, this can include details such as shots or passes, the distance travelled, and the position of the ball, along with player interactions. This data typically has lower latency than broadcast data and is used for evaluating individual player and match performance, by sportsbooks to adjust odds based on game dynamics, and by media outlets to deliver live scores and statistical updates.

In-play data can be either live or historical, with both types having the same exact structure. The key difference lies in their consumption. Historic is used to train models, which can then be implemented into trading strategies and backtested. Live data typically comes through a PUSH feed directly from the provider. There is a race for such data, for example drones have been employed at both football matches and horse races in this race for the lowest latency state of game data. In general a research platform will integrate in-play state of game data with millisecond-level exchange data. The resulting dataset provides an in-depth insight into the market microstructure, along with match-specific events such as momentum shifts, drifting odds, and volatility enabling the algorithmic trader to take advantage in a manner that no retail bettor would be capable.

Exchange Data

Exchange data is composed of Betfair's microsecond-level order book updates. These updates provide the best available prices for backing and laying, show liquidity at each price point, and include the most recent traded price— all of which are crucial for building the live price ladder. Similar to in-play data, exchange data can be obtained either in real-time or as historical records and is available for purchase from the exchange.

Live data from Betfair can be accessed through their REST API or via a JSON-RPC connection, which activates a subroutine on the client's side when an update is detected. Historical exchange data, accessible through the Betfair Exchange Historical Data service with a PRO subscription, is stored within the research platform's database. This PRO subscription offers the same detailed granularity as the live updates, making it essential for backtesting and deploying strategies that have been trained using the historical data.

3.2.1 Data Collection & Analysis

Although this thesis primarily focuses on football betting, it was crucial to design the research platform to be adaptable for future applications and to be compatible with other sports like tennis and horse racing. Consequently, data for all three sports has been gathered and integrated into the platform. Each sport inherently requires a distinct data structure, showcasing that the platform is both viable and flexible, with the capability to be easily expanded and maintained over time. Table

3.1 lists the data providers that were incorporated into the research platform. It's important to mention that while these specific options were selected, a comprehensive evaluation of alternatives was carried out, confirming that these choices offered the best results.

	Pre-game	In-play	Exchange
Football	FootyStats	StatsBomb	Betfair
Tennis	Opta	RunningBall	Betfair
Horse racing	RacingPost	Total Performance Data	Betfair

Table 3.1: Data Providers Categorised by Sport and Type (Pre-game, In-play, and Exchange) for Football, Tennis, and Horse Racing.

Pre-game: FootyStats

FootyStats offers a thorough collection of football data, covering more than 1,500 leagues worldwide. This includes major competitions such as the Premier League, La Liga and international tournaments such as the World Cup and Euro Cup. Their database features detailed statistics on individual players, teams, leagues, and matches, with over 219 data fields available for games dating back to 2010. Additionally, FootyStats tracks odds for 68 popular betting markets, allowing for the creation of models that combine historical statistics with bookmaker odds, as discussed by L. Egidi et al. [11]. Since the 2016 season, FootyStats has also provided pre-match Expected Goals (xG) calculations for each team. The xG metric estimates the average number of goals a team is likely to score based on historical data.

In an attempt to analyse and showcase the data we can present pre-game statistics in an series of visualisations and this can be integrated with the gpt-4 API, to provide pre-match commentary. Figure 3.1 shows a mock-up web application built on the back of FootyStats.

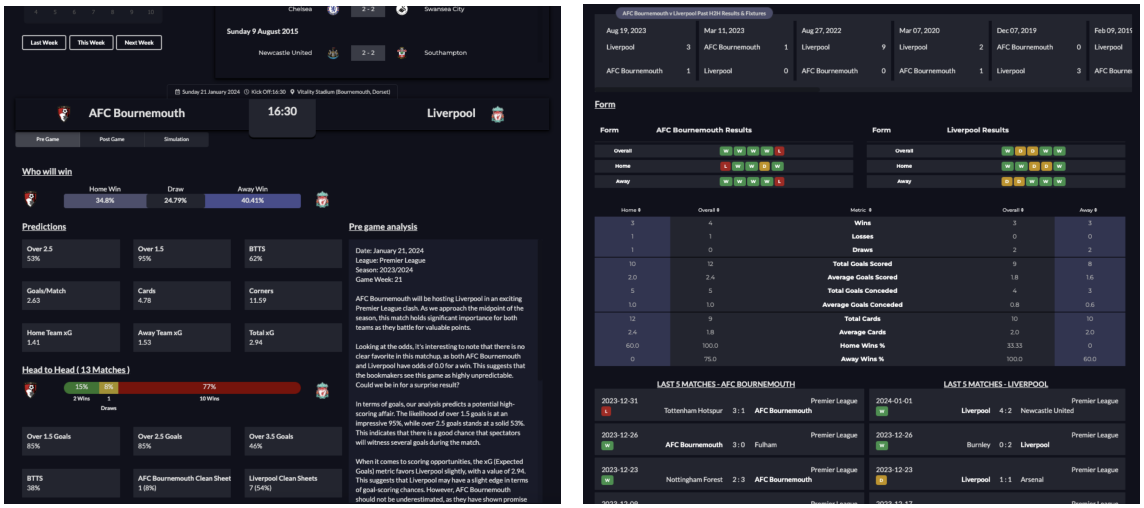

Figure 3.1: Pre-game Footystats Data Visualisations

FootyStats was selected over other options like 'Football-Data' due to its extensive dataset, technical specifications and database architecture. The FootyStats Football Data API offers straightforward access to its distributed MySQL database, delivering data in JSON format. Since the database stores structured data, the API responses can be easily converted into a tabular format, which is ideal for most machine learning applications. 5,320 EPL matches, covering all seasons from 2010 to 2023, were downloaded using the League Matches endpoint, this endpoint is available through the FootyStats Premium subscription.

In-play: StatsBomb

StatsBomb are a live and in-play football data platform, widely used by team coaches and scouts for player evaluation. Additionally, it serves the media and betting industry by offering live updates and dynamic models during matches. Statsperform are another provider employing a large number of professional watchers at the match itself supplying data. Statsbomb collect detailed data on position of each player, pass initiation and completion points, and finer details such as the foot used for passes or shot velocity. This data comprises an array of 2,400 data points for in-play analytics. They also build their own metrics on top of this data such as Expected Goals (xG) which factors in not only the ball and goalkeeper positions but also the portion of the goal obstructed by defenders and those potentially able to block the shot but not yet in the attacker's path. This is some of the most comprehensive data in the industry and will play a key role in the upcoming chapters.

In accessing this data statsBomb has created libraries for Python and R that allow clients to stream data directly into their applications. Using the statsbombpy package, data for Champions League seasons from 2010 to 2018, as well as the 2015 EPL season, was collected, totaling 2630 matches. Given the high cost of StatsBomb data, it is not feasible for purely research purposes, so all acquired data came from StatsBomb's open-source database.

Exchange: Betfair

Betfair's exchange data is crucial for in-play betting across all sports markets. At its core, this data represents the price ladder, providing punters with a precise view of the market and the event's current state. For instance, jumps in the best available back and lay prices for any runner can be used to reconstruct the full-time score and timing of goals. On a more advanced level, exchange data supports tasks such as backtesting, post-trade strategy evaluation, and trading signal generation through statistical analysis or machine learning. To perform these tasks effectively, a comprehensive understanding of the exchange data's structure and the roles of volatility, liquidity, and other order book features in price discovery is essential.

Market Name	# of Runners	Example Runner
Player To Provide an Assist?	5	Kevin De Bruyne
Total Match Goals Over/Under 3.5	2	Over 3.5 Goals
Team To Win Both Halves	4	Manchester United
Handicap Betting +1.5	3	Liverpool -1.5
First Player To Be Carded	25	Declan Rice
Half-Time Correct Score	6	1-0 to Chelsea
Total Match Cards	3	Under 4.5 Cards

Table 3.2: Sample Selection of Markets Available on Betfair, Detailing the Number of Runners and Example Selections for Each.

There are only 351 prices available on Betfair and it should be noted that the tick sizes vary by price. In effect when the market closes the runner who wins is priced at 1.0 and those who lost at 1000. This is quite different to financial markets where for example in US equities the tick size is \$0.01, and indeed may reduce in some cases to fractions of a cent, whereas in foreign exchange markets prices are often quoted in pips, the 4th decimal place of the price quote.

3.2.2 Price Discovery

The movement of prices and how prices drift, experience jumps, and exhibit volatility in sports markets is influenced by the nature of the sport, the structure of the event, and the real-time developments during the match. In football, matches have a fixed duration of 90 minutes, with prices naturally drifting as time progresses. As time progresses without a goal, the probability of certain outcomes (e.g., a win for either team) naturally decreases. This can be modeled using a time-decay function.

Prices	Tick Size
1.01 to 2.00	0.01
2.02 to 3.00	0.02
3.05 to 4.00	0.05
4.1 to 6.0	0.1
6.2 to 10.0	0.2
10.5 to 20.0	0.5
21.0 to 30.0	1.0
32.0 to 50.0	2.0
55.0 to 100.0	5.0
110.0 to 1000.0	10.0

Table 3.3: Betfair Prices and Tick Sizes

$$P(t) = P_0 e^{-\lambda t} \tag{3.1}$$

where P_0 is the initial probability and λ is the decay rate.

Goals cause immediate jumps in prices, reflecting the significant change in the likelihood of outcomes.

$$P_{\text{new}} = \frac{P_{\text{goal}}}{P_{\text{total}}} \tag{3.2}$$

Volatility increases in the final minutes as the match outcome becomes more certain. Key events like goals, penalties, or red cards significantly impact volatility.

Algorithmic traders must be keenly aware of liquidity within their markets to effectively scale their products and mitigate market impact. The most popular sports on Betfair, such as football, horse racing, and tennis, may see trading volumes reaching tens of millions of pounds for large events. 'Match Odds' is usually the most liquid market, offering punters the opportunity to bet on the outcome of the match. In the context of football, where a draw is a possible outcome, there are three betting options: Team A, Team B, or The Draw. Often, during the 105 minutes that encompass the match's active period (90 minutes of gameplay plus a 15-minute halftime interval), the Match Odds market can experience more than 70,000 updates to the order book. This equates to around 11 updates per second, providing constant information for the punters.

Figure 3.2 illustrates the liquidity trends for both back bets and lay bets in a Match Odds market. Liquidity noticeably rises on both betting sides before the match goes live. However, the back bet side consistently exhibits significantly higher liquidity levels. At its peak, the back bet liquidity reaches £130,000, while the lay bet liquidity caps at £63,000. This discrepancy highlights that punters primarily use the betting exchange to place wagers on the expected winner. This difference is crucial for risk management. A strategy that heavily depends on back bets might be lucrative. However, due to the lower liquidity on the lay side, it could face challenges in quickly rebalancing its inventory. Slow rebalancing can lead to higher risks, as the price may drift further from the optimal level during the balancing process, complicating the strategy's ability to exit positions and potentially resulting in losses.

Figure 3.4 illustrates the best available back prices in the Match Odds market for each of the three runners. Football matches typically feature gradual price shifts, where back prices for the leading team decrease as the match progresses, while lay prices for the losing teams also decline due to the fixed duration of the game. This trend is evident in Figure 3.4, which shows a gradual change in the best back prices, except during periods of post-goal price adjustment.

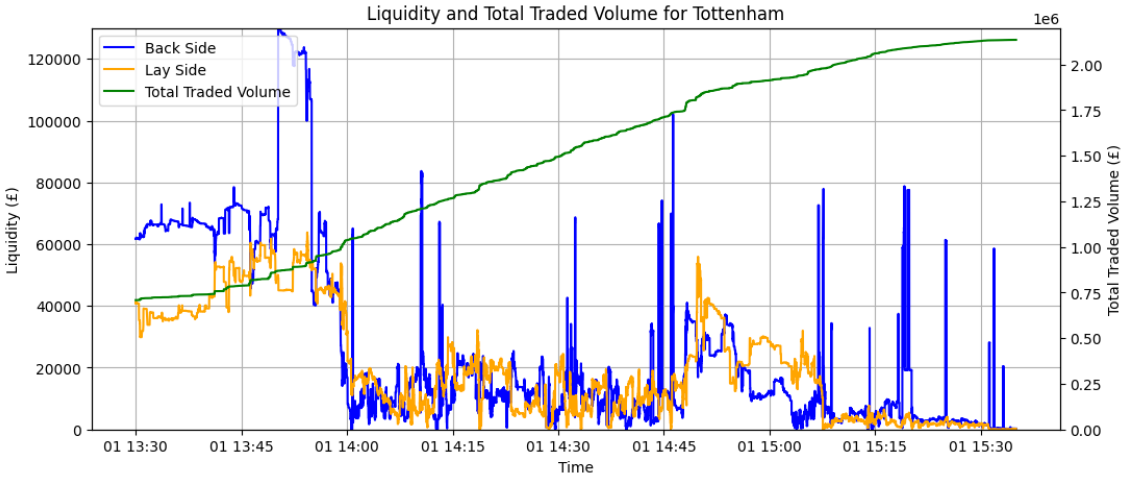

Figure 3.2: Liquidity and Total Traded Volume for Tottenham, showing Back, Lay, and Total Volume in the Aston Villa vs. Tottenham Match

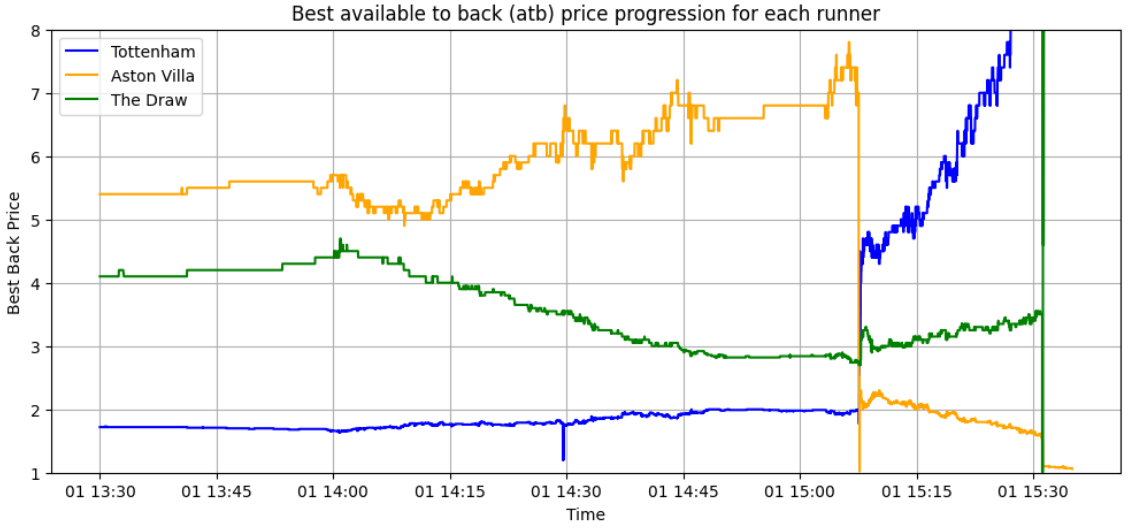

Figure 3.3: Liquidity and Traded Volume for Tottenham, Illustrating Market Dynamics and Betting Activity Over Time during the Aston Villa vs. Tottenham Match.

3.3 Database Architecture

Several key design decisions have been prioritised to achieve robust, scalable, and efficient data management. The system must handle large volumes of data and scale seamlessly without compromising latency, ensuring that data is always readily available for querying. Additionally, it needs to be robust enough to manage frequent data entries and exits while maintaining high security standards. Given the potential value of this data to traders seeking a competitive edge, the system must be designed to withstand potential cyber threats. To meet these objectives, a combination of MongoDB and Amazon Web Services (AWS) Relational Database was selected to establish the final architecture. The architecture integrates seamlessly with the existing QST infrastructure, which also operates within the AWS ecosystem and was utilised in the experiments described in Chapters 4 and 5. Although the current setup is built on AWS, similar database services are available on Google Cloud Platform and Microsoft Azure, making it relatively straightforward to migrate the architecture to other cloud providers if needed.

This duel database approach combining a NoSQL MongoDB with SQL AWS Relational Database MariaDB was established due to the varying data shapes that exist across the three data sources.

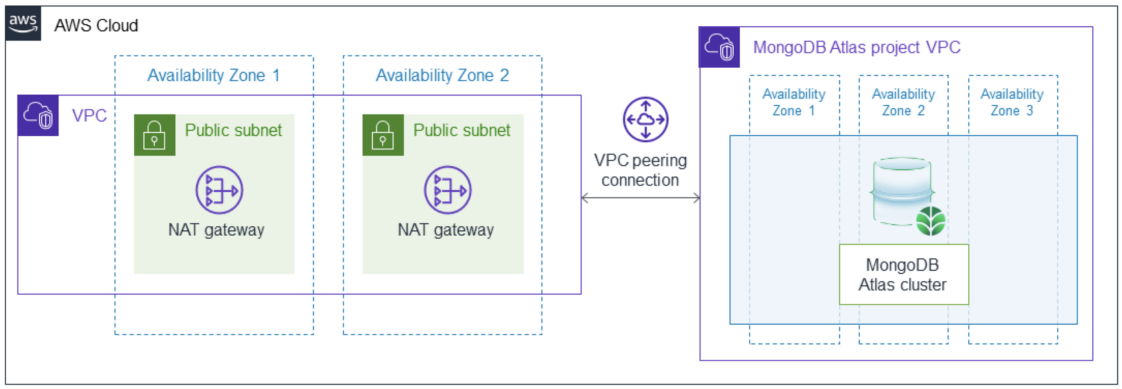

Figure 3.4: Diagram of Database Architecture: Integration of MongoDB and MongoDB Atlas Cluster on AWS, illustrating VPC Peering Connection across Multiple Availability Zones with NAT Gateways.

Structured Query Language (SQL) is particularly well-suited for organising data into tables. Once a table is established, its columns or fields are fixed and cannot be changed, which is why it is termed 'structured'. SQL facilitates relationships between tables through the use of primary and foreign keys on specific columns. When data from two different tables needs to be combined, the join operator goes through pairs of records, linking data where a foreign key in one table matches the primary key in the other. SQL data is managed and queried within a Relational Database Management System (RDBMS). Amazon RDS, using the MariaDB SQL engine, is an AWS service that offers this functionality. While there are various SQL engines like MySQL, PostgreSQL, and Oracle, they generally offer similar capabilities without major differences.

Another option is MongoDB, a NoSQL database that utilises a dynamic data structure, eliminating the need for a fixed schema. This feature makes it highly effective for working with unstructured or semi-structured data types, such as JSON documents and key-value pairs. Unlike traditional SQL databases, NoSQL systems such as MongoDB do not enforce data relationships with foreign keys. This allows for more dynamic data storage and retrieval, accommodating various file formats and supporting agile, scalable development. NoSQL databases are designed to scale horizontally, meaning they can efficiently distribute computational loads by adding new nodes to the system. In contrast, SQL databases typically scale vertically, which involves upgrading a single node's CPU or memory, a method that can be less sustainable long-term. MongoDB specifically uses a document-oriented model where data is organised into collections akin to SQL tables, and each collection contains documents, which are JSON-like structures. The integration of MongoDB with AWS provides developers with the flexibility to leverage MongoDB's capabilities without being locked into the AWS ecosystem, while still taking advantage of features offered by MongoDB and its Atlas database manager.

3.3.1 Pre-game Data Storage

Pre-game data consists of historical events where each event includes basic event data and post-processed statistics. This data is termed 'pre-game' because it becomes available for querying before the next modelled match occurs. These statistics follow a predefined set of measures, making pre-game data well-suited for storage in an SQL database. An Amazon RDS instance using MariaDB was implemented to manage pre-game data for football, tennis, and horse racing, with appropriate SQL schemas. The data for each sport is maintained under 100,000 records to optimise the efficiency of join operations. The main tables include pre-game, pre-match, and pre-race, along with additional tables for details such as league, tournament, or competition information.

3.3.2 In-play Data Storage

For football, Statsperform and StatsBomb are the two mentioned providers, in the former case, employing a large number of professional watchers at the match itself supplying data and in the latter utilising advanced computer vision algorithms for detailed data like player positions, the ball's 3D trajectory, and specific metrics such as pass footedness or shot velocity. In tennis, StatsPerform delivers WTA in-play data, offering insights into player performance and match statistics with real-time scoring updates that can be quicker than the time the umpire takes to officially update the score. In horse racing, Total Performance Data (TPD) place GPS's on the horses to enable the live monitoring of horse positions and sectional times. In-play data tends to be less structured compared to pre-game data due to the wide range of update types that can happen during an event. To handle this complexity, a MongoDB database was set up with a separate collection for each sport. The documents in these collections are JSON files generated from in-play updates, or in the case of tennis, XML files that have been converted to JSON. Storing this data in a SQL database would require managing hundreds of fields in a single table to accommodate all potential update types and properties. This approach would result in many empty cells for infrequent events like cards or substitutions, making MongoDB a more efficient choice.

3.3.3 Exchange Data Storage

Exchange data, similar to in-play data, lacks a predefined schema and is less structured. Fields are included only when there is an update to a previously recorded value, ensuring data is not duplicated. Additionally, a single field, such as 'atb', may contain multiple Price-Volume tuples if volumes have changed for several ticks. These characteristics necessitate the use of a MongoDB architecture. However, storing and retrieving this data is complex. Market data JSONs can include over 70,000 updates per market, leading to file sizes exceeding 16MB, which is MongoDB's limit for a single document. The difficulty is further compounded by the existence of more than 1.8 million exchange data files. As a result the following three options we're explored: AWS S3 bucket with SQL, MongoDB GridFS and a MongoDB Time Series Collection.

The initial and simplest approach involved uploading Betfair's compressed exchange data files into an S3 bucket, with metadata and file paths stored in an Amazon RDS MariaDB instance. Researchers could then sift through the metadata to locate relevant markets before downloading and decompressing the files locally for training or backtesting strategies. However, this method has two significant drawbacks. First, integrating Amazon S3 and RDS, which are separate services, adds complexity. Second, querying exchange data requires downloading and decompressing files to a local directory each time they are needed, which is neither scalable nor efficient in terms of time or space.

An alternative, and seemingly more practical solution, was using MongoDB's GridFS. GridFS is designed to manage files larger than 16MB by dividing them into chunks and storing each chunk as a separate document. When exchange data is queried, GridFS automatically reassembles these chunks. The primary downside of this approach is that GridFS stores files in a binary-encoded format, which is not ideal for dynamic querying and retrieval, making it less suited for applications requiring real-time data access. This approach as result acts more like an indexing system and thus more similar to the previous approach to one more dynamic.

Ultimately, Time Series Collections proved to be the optimal solution, as the pt field could serve as the timestamp for each update (or document). Time series databases generally reduce file sizes by leveraging the common structure of entries to apply compression algorithms effectively. Time Series Collections offer significant compression benefits, successfully reducing file sizes to below 16MB. Crucially, these collections store each update as a separate document with the pt field, enabling efficient querying, visualisation with MongoDB Charts, and the ability to skip over arbitrary time periods in the exchange data.

3.4 Backtesting

The platform remains a researcher tool for examining data, forming views, showing analysis, and deploying backtests on historic data. A full backtest is as if the listener was replaying the historic data live. This is done by running a history on these backtests to gain an idea of the distribution of outcomes of a strategy.

With this data infrastructure and MongoDB's Time Series Collections, backtesting strategies have become a more streamlined and automated process. By converting raw market data into live ladders, we can clearly index each update using the pt field, which represents the time of data updates. This method allows each update to be treated as an individual document, linking updates together and inputting them into the backtesting framework. Previously, exchange data was fetched from S3, processed locally on the client side, and gradually loaded into the backtest. Now, with the data already processed and residing within MongoDB clusters, backtests have direct access to the Time Series Collections. This transition moves the entire backtesting workflow to the server side, allowing researchers to test and verify strategies with much greater speed and efficiency.

3.5 Results

The database developed in this chapter brings together pre-game, in-play, and exchange data, each fulfilling a unique role. Pre-game data provides a historical perspective and crucial statistics necessary for constructing predictive models before the start of games. In-play data offers a real-time snapshot of the game, vital for evaluating player performance and adjusting betting odds. Betfair's exchange data delivers insights into betting market trends, which are critical for conducting financial analysis in sports betting. By integrating these diverse data sources, the database offers a comprehensive tool for analysis and decision-making. By bringing these different data types together in a scalable, robust and efficient way, the platform enables a researcher to use the data in the database to test market efficiencies, attempt to create predictive features based on the type of strategy envisaged, train models then the test these on a full backtest.

To demonstrate a researcher's workflow and how they would use this data platform, consider the following example which uses pre-game team statistics, in-play goal counts, and minute-by-minute odds. This would involve a series of database queries. Imagine an over-under basic strategy in football; a researcher might find that if the pre-game xG for a match was below 1.7, and the total number of goals at, say, 62 minutes was 1, then if the over-under 4.5 goals at any point was above 1.06 decimal odds, this might prove to be a profitable strategy with low volatility across 200 games, with an exit strategy if a goal was scored in the next 10 minutes. In this very simple example, this demonstrates the ability to query the data easily and quickly.

The process begins by retrieving the pre-game football data, generating a list of relevant IDs. These IDs are then matched with corresponding Betfair events stored in MongoDB, which leads to the extraction of specific marketIndex.json files. These files are further filtered within MongoDB to focus on Betfair market IDs related to the 'Over/Under 3.5 Goals' market, producing a targeted list of market IDs. Next, the exchange data files linked to these market IDs are retrieved from S3 storage and either loaded into memory or a temporary MongoDB collection, resulting in a collection of JSON files containing historical exchange data. This data is then filtered again to pinpoint instances where the 'Over 3.5 Goals' market was backed at odds below 4.0 before the match, and at odds above 1.5 at the 62nd minute of play. This refined dataset is subsequently used for bulk backtesting to evaluate the effectiveness of the sports betting strategy.

3.6 Discussion

Returning to our originally outlined contributions, this chapter builds a comprehensive data framework. Data is pre-processed from the database, establishing a robust system for capturing and storing sports exchange data with the identification of key characteristics. This infrastructure serves as the backbone of the research platform and helps achieve a research environment that intuitively

integrates data and enables quick construction of statistics and data retrieval for backtesting. This data framework offers a new way to collect and store pre-game, in-play, and exchange data, providing new perspectives and opportunities for other researchers to create features for model training and testing.

This integration of pre-game and in-play exchange data, with a focus on machine learning applications, sets the scene for the next two experiments. The incorporation of 11 years worth of pre-game EPL data creates an extensive dataset that sets up the analysis and the modelling of a series of pre-game baseline models. The integrate of in-play data with millisecond exchange data then feeds into the development of the third experiment where we attempt to develop deep learning models that identify shifts in match momentum, enabling dynamic in-play betting strategies.

Chapter 4

Baseline Models for Pre-Game Football Predictions

This chapter presents the implementation of baseline classification models designed to predict the full-time outcomes of football matches. Building on the research platform established in Chapter 3, we extract and process pre-game data, followed by a pipeline of feature engineering, model selection, training, and evaluation.

4.1 Background

The most popular sports on Betfair have typically been football, horse racing, and tennis, which may see trading volumes reaching tens of millions of pounds for large events. For football in particular, the Match Odds market, where punters can bet on the three outcomes—home, away, and draw—is usually the most liquid. With this being the most liquid, the financial incentive to accurately predict full-time results has subsequently led to a high degree of research.

Early statistical models in sports betting focused on modelling price behaviour. One of the most well known papers came from Dixon and Coles [10] who proposed using a Poisson arrival process, which demonstrated predictable price drifts as time passed without a goal. This method was employed to estimate the number of goals each team would score in a match and took advantage of inefficiencies within the betting market, leading to profitable outcomes. Karlis and Ntzoufras [21] later enhanced this work by introducing a dynamic double Poisson model, which proposed a bivariate Poisson distribution to account for the relationship between the attacking and defensive strengths of the teams involved. More contemporary research in football betting focuses on feature engineering and supervised learning techniques. Constantinou et al. [6] developed the pi-football model, which utilises a Bayesian network to establish and update team ratings.

This chapter explores the implementation and evaluation of various machine learning classification models, conducting a thorough comparative analysis to evaluate their performance across multiple accuracy metrics on a wide range of datasets. The aim of this investigation is to stress-test the research platform developed in Chapter 3 and to build a series of baseline models that can be used as benchmarks for the later experiment. The chapter will cover a series of steps, from pre-processing and enhancing the input features to exploring the implications of feature selection and transformation techniques, as well as dimensionality reduction and label/one-hot encoding. We will examine which methods increase the accuracy and applicability of ML models in live betting scenarios.

4.2 Data

At the start of any machine learning pipeline is the data. The availability of vast and increasingly comprehensive datasets is a key factor influencing the success of a predictive model. In this chapter, we focus solely on pre-game data. Pre-game data consists of historical events, where each

event includes a set of basic event data and post-processed statistics.

The data used in this chapter comes from the research platform and database developed in Chapter 3. The platform's programming interface and low-latency database architecture enable the quick retrieval of structured Footystats pre-game data, preparing it for subsequent transformation and processing steps. The models are trained on 10 seasons of English Premier League (EPL) data, rather than data from international club and country competitions such as the Champions League, FA Cup, or Euros, as these tournaments require different modelling techniques. Focusing just on EPL simplifies the modeling environment however you do lose the extra information on form, and time between games as these competitions are played simultaneous to each other.

4.2.1 Data Exploration

To analyse trends, changes in team performance, and player rotations in the EPL, data from the past 10 seasons was incorporated into the modelling environment. To understand the structure and characteristics of the data Table 4.1 illustrates some general statistics.

Statistics	**Mean**	**Min**	**25%**	**50%**	**75%**	**Max**	**Std**
Home Goals	1.53	0.0	1.0	1.0	2.0	9.0	1.32
Away Goals	1.22	0.0	0.0	1.0	2.0	9.0	1.19
Home Shots	12.37	0.0	9.0	12.0	16.0	37.0	5.26
Away Shots	10.35	0.0	7.0	10.0	13.0	31.0	4.66
Home Possession %	51.20	12.0	43.0	51.0	60.0	83.0	11.90
Away Possession %	48.65	8.0	40.0	49.0	57.0	83.0	11.88
Home Fouls	10.40	0.0	8.0	10.0	13.0	23.0	3.45
Away Fouls	10.71	0.0	8.0	11.0	13.0	26.0	3.56
Home xG	1.09	0.52	0.0	1.39	1.67	3.59	0.84
Away xG	0.94	0.94	0.0	1.20	1.48	2.84	0.72

Table 4.1: Summary Statistics for EPL Matches

The summary statistics of EPL matches reveal distinct patterns that underscore the significant advantage often held by home teams. On average, home teams score 1.53 goals per match, compared to 1.22 goals by away teams. This home advantage is further highlighted by the higher mean number of shots taken by home teams, with 12.37 shots per match, compared to 10.35 shots by away teams. Additionally, home teams have a slightly greater possession percentage, averaging 51.20%, while away teams hold possession for 48.65% of the time. The standard deviation in goals scored and shots taken indicates variability, yet the consistently higher averages for home teams suggest that playing on home turf provides a substantial benefit, likely influenced by factors such as being accustomed to the playing conditions, support from the home crowd, and less fatigue from travel. Away teams, on average, commit slightly more fouls, with 10.71 fouls per match compared to 10.40 by home teams. This trend could suggest that away teams, often under more pressure, resort to fouling more frequently as a defensive mechanism, particularly when the game is slipping away from them, necessitating more defensive discipline.

The Expected Goals (xG) metric, provides an indication of the quality of chances created, we observe a similar trend favouring home teams. The mean xG for home teams is 1.09, compared to 0.94 for away teams. This slight edge in xG suggests that home teams not only create more scoring opportunities but also generate higher-quality chances. Figure 4.1 illustrates that pre-game xG serves as a strong indication of actual goals, with Manchester United's scoring rate generally aligning with what would be expected based on the quality of their chances. We can also extract form from these metrics as towards the end of the season, there is a noticeable divergence, indicating a period where the team underperformed relative to the quality of chances they created.

To enhance the understanding of the data the statistics beyond Table 4.1 and uncover relationships between specific features, Figure 4.2 and Figure 4.3 show two different correlations. Understanding these correlations is crucial prior to machine learning model training because it helps identify

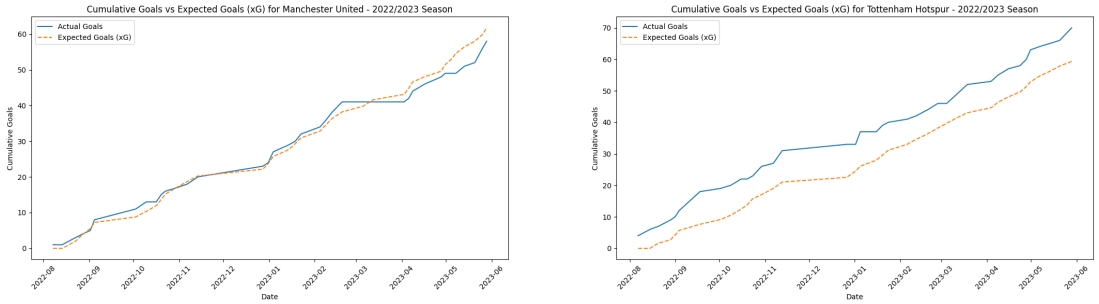

Figure 4.1: Cumulative Goals vs Expected Goals (xG)

multicollinearity among features, which can lead to redundant information being fed into the model. Additionally, knowing the strength and direction of correlations can guide feature selection, ensuring that the most relevant variables are included, thereby improving model accuracy and interpretability. Correlation analysis also assists in diagnosing potential overfitting issues by highlighting features that may have strong linear relationships with the target variable, enabling more informed decisions during the feature engineering process.

Figure 4.2 shows the Pearson correlation coefficient which is used to measure the linear relationships between two normally distributed variables with the equation

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

- r = correlation coefficient
- x_i = values of the x-variable in a sample
- $\bar{x}$ = mean of the values of the x-variable
- y_i = values of the y-variable in a sample
- $\bar{y}$ = mean of the values of the y-variable

Figure 4.3 shows the Spearman correlation coefficient, which, unlike Pearson, measures the strength and direction of the monotonic relationship between two variables. The Spearman correlation does not assume that the data is normally distributed or that the relationship between the variables is linear. Instead, it assesses how well the relationship between two variables can be described using a monotonic function. The equation can be expressed as

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

- ρ = Spearman's rank correlation coefficient
- d_i = difference between the two ranks of each observation
- n = number of observations

In both Pearson and Spearman correlations, the values obtained range from -1 to +1. Values greater than 0 indicate a positive correlation, while values less than 0 indicate a negative correlation. In the case of Pearson, this refers to a linear relationship, whereas for Spearman, it refers to a monotonic relationship. Both plots clearly show that the strongest relationships are observed between the number of goals (full-time and half-time), shots on and off target, possession percentages, match outcomes, and odds.

A strong positive correlation between full-time home goals and half-time home goals suggests that teams leading at half-time are more likely to maintain or extend their lead by full-time.

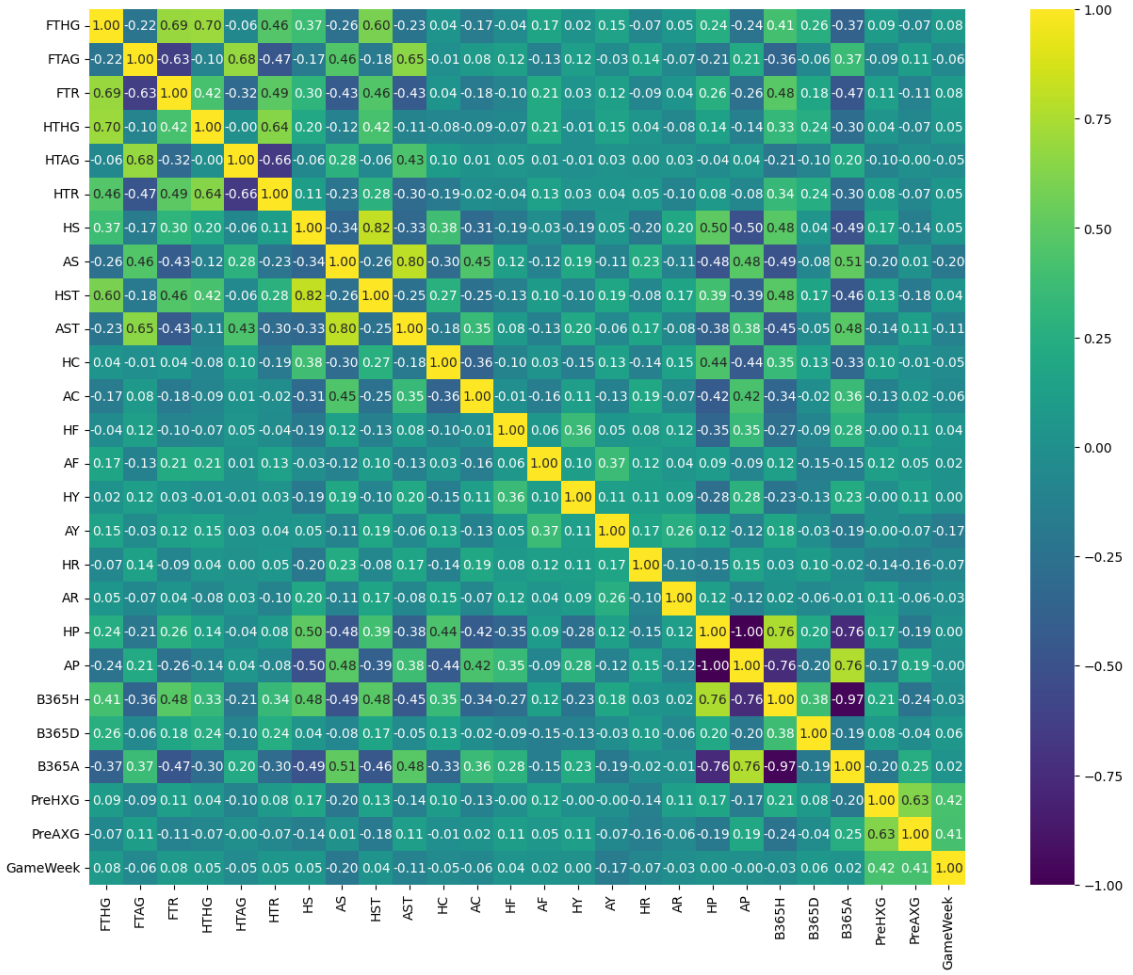

Figure 4.2: Pearson Correlation Matrix

Similarly, home shots are highly correlated with shots on target, indicating that a greater volume of attempts generally leads to more accurate shooting. Pre-match expected goals (xG) also show a positive relationship with actual goals scored, highlighting xG's utility as a predictive metric. On the other hand, away possession percentage is moderately negatively correlated with full-time home goals, implying that when the away team controls more possession, the home team tends to score fewer goals. Betting odds for a home win are strongly negatively correlated with full-time home goals, indicating that more favourable odds are associated with higher goal tallies for the home side. Additionally, the full-time result correlates strongly with both the number of goals scored and the betting odds, underscoring the importance of these metrics in determining match outcomes. Finally, the matrix shows low or negligible correlations with variables such as game week, suggesting that certain factors have minimal predictive power in isolation. These insights are crucial for feature selection in predictive modelling, where focusing on strongly correlated variables can improve model accuracy.

4.2.2 Data Pre-Processing

Data pre-processing is a crucial step in preparing the dataset for modelling, ensuring the quality and suitability of the input data is in a refined format amenable to ML algorithms. This phase involves tasks such as data cleaning to handle missing or inconsistent values, normalisation to scale features appropriately, and label encoding for categorical variables. By identifying and addressing inherent patterns, outliers, and correlations, pre-processing enhances the data's structure, facilitating more accurate and efficient model training. These steps are essential to mitigate potential biases and improve the performance and generalisability of the predictive models.

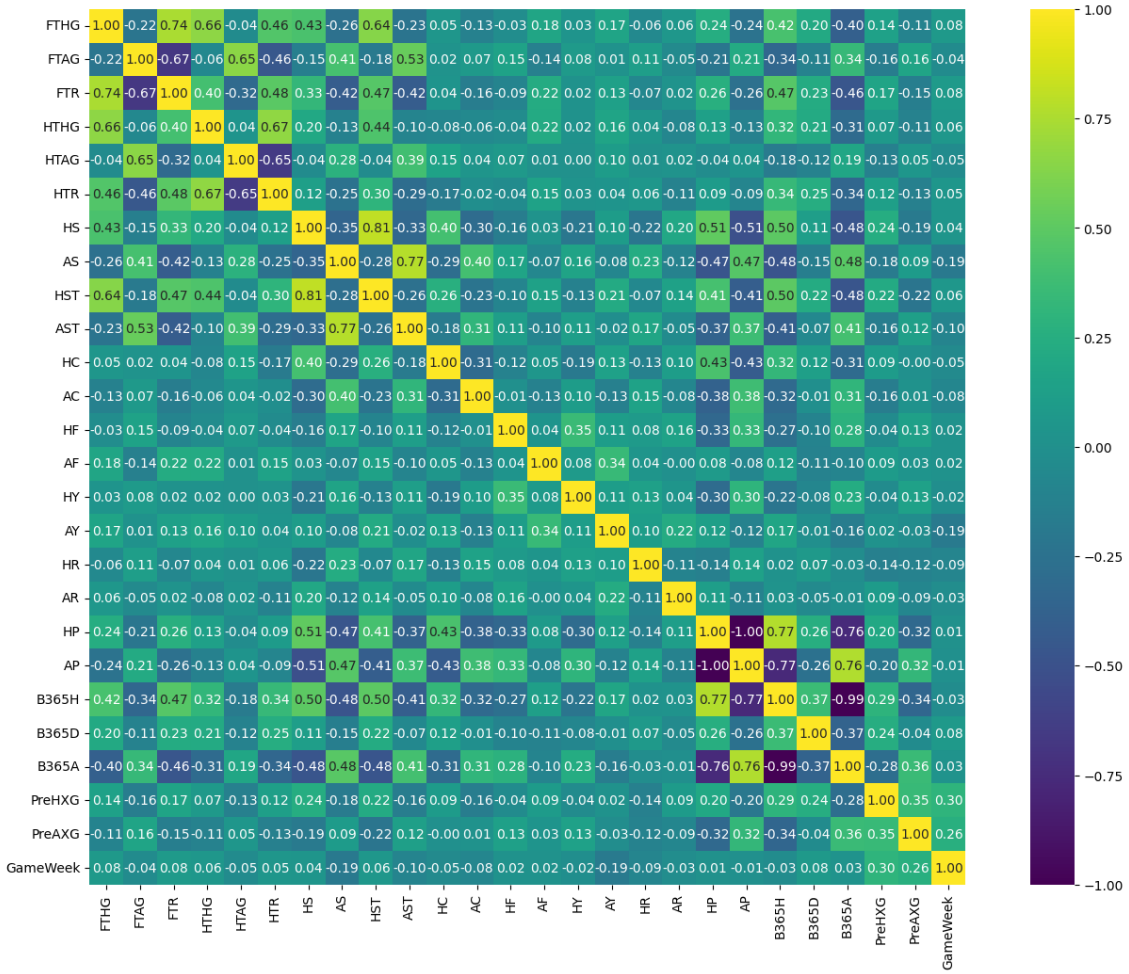

Figure 4.3: Spearman Correlation Matrix

Data Cleaning

The first step in the data pre-processing pipeline is data cleaning, where the raw football data is transformed and organised into a structured format suitable for analysis and modelling. This process involves filtering out incomplete or irrelevant matches, removing duplicates, and converting date formats. Only matches marked as 'complete' have been kept, effectively removing any incomplete or upcoming games that could add noise or inaccuracies to the dataset. This step also addresses missing values, reducing the risk of errors or biases during the training of algorithms. The data is then organised chronologically and converted from UNIX timestamps to a human-readable datetime format. Proper ordering is crucial for any analysis that examines trends or temporal changes, enabling time series analysis or the generation of features based on time-based patterns. Additionally, rows with missing values and duplicates are removed to ensure data integrity.

FootyStats compiles an extensive dataset with over 219 fields, many of which are tailored for broadcasting or media usage. For our analysis, which targets the 'Match Odds' market, we identified that out of the 68 columns related to odds, only three are actually useful for predicting the Full-Time Result. The process of selecting these specific columns is vital as it eliminates superfluous data from the FootyStats dataset. By narrowing down the focus, we ensure that the data exploration and model training stages concentrate on the most influential features, such as match outcomes or goal counts, which are directly relevant to the prediction task at hand.

Another important step in data pre-processing involves applying one-hot encoding to categorical variables, which converts them into a binary format that is easier for machine learning models to

interpret. For example, categorical columns like the home team and away team are transformed into numerical data. By creating separate binary columns for each category, one-hot encoding helps prevent the models from incorrectly treating categorical data as ordinal, thereby enhancing the accuracy of predictions.

Data Splitting

In the data splitting process for this project, the datasets were not randomly divided into training and test sets, as is common in many machine learning applications. Instead, the data was split based on chronological order, which is crucial for sporting event predictions where the model's primary task is to forecast future match outcomes. Random splitting could introduce temporal inconsistencies, where information from future matches might inadvertently influence the training process, leading to data leakage and overly optimistic performance. For example, factors such as team form, recent matches, and historical results significantly inform the model's predictions, and thus, maintaining the chronological sequence of events ensures that the model only learns from past data and is tested on unseen future events.

The code provided allows for different methods of data splitting. One approach is the basic train-test split, where the dataset is divided into training and test sets based on a specified ratio (e.g., 80% training and 20% testing). The testing set remains unseen by the model during training and serves as an indicator of how well the model generalises to new data. Additionally, an optional validation split can be introduced, where the test set is further divided into validation and test subsets. This validation set is used during the model development phase to fine-tune hyperparameters without touching the final test set, preserving its integrity for final evaluation. Table 4.2 shows this data split for each period with varying about of historical data. Each baseline model will be trained and tested on the varying dataset sizes and evaluated carefully.

Dataset Name	Period	Split	Seasons	# of Matches
P1	Past Season	Training	2022/23	190
		Validation	2022/23	95
		Test	2022/23	95
P2	Past Two Seasons	Training	2021/22	380
		Validation	2022/23	190
		Test	2022/23	190
P5	Past Five Seasons	Training	2018/21	950
		Validation	2021/22	475
		Test	2022/23	475
P10	Past Ten Seasons	Training	2013/19	1900
		Validation	2019/21	950
		Test	2021/23	950

Table 4.2: Dataset Split Across Different Periods for Model Training, Validation, and Testing

Feature Engineering

After analysing the FootyStats data using a correlation matrix then going through a series of cleaning, pre-processing, steps the next stage is to select features to identify and construct the most important characteristics for training the model. The findings from the correlation matrix, which reveal the relationships between the FootyStats fields and the target variable, serve as the foundation for the feature selection process.

Match-by-match data is aggregated to construct detailed statistical profiles for each team within the dataset. This process utilises the information we have on how teams have performed across a range of statistics to calculate various metrics. These metrics include possession percentages,

goals scored, fouls committed, draws, yellow and red cards received, corner kicks, shots on goal, shots on target, wins, losses, and goals conceded. The data is collected on a match-by-match basis and is categorised into home and away games to reveal any potential home-field advantages or difficulties faced in away matches. Additionally, the dataset tracks the number of seasons each team has competed in, offering valuable insights into their experience, longevity in the league, and potential financial advantages. Performance trends are also monitored by recording the number of goals and the goal differences over the last several matches, providing an indication of recent form. This information is crucial for making short-term predictions and conducting trend analysis.

As mentioned briefly, form is a vital feature that can significantly influence the predictive accuracy of pre-game football models. Within the context of machine learning tracking a team's recent form—particularly through features like win streaks—offers a robust indicator of a team's momentum. When a team loses a match, their streak resets to 0. Figure 4.4 illustrates the form of high and mid performing teams. By examining Manchester City's win streaks, it's evident that when they manage to string together a handful of victories in a row, they often maintain that form, resulting in extended unbeaten runs. In contrast, Everton's form, as depicted in the figure, reveals a more inconsistent pattern.

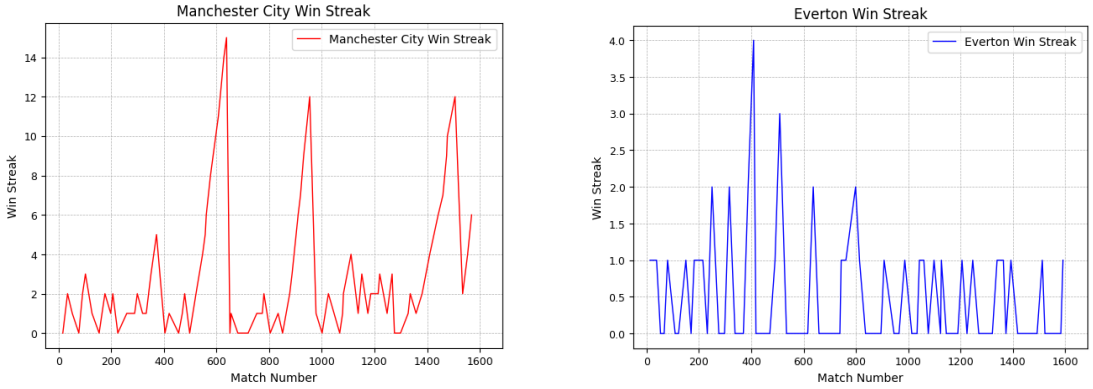

Figure 4.4: Comparison of Historical Win Streaks for Manchester City and Everton

Unlike the Win Streak metric, which only tracks consecutive victories, the Last n-Match Form provides a more comprehensive overview of a team's recent performance. It does this by consistently updating key statistics, such as the goal difference $G_{\text{diff}} = G_{\text{scored}} - G_{\text{conceded}}$, the total number of goals scored, and the average accuracy of shots taken, over the last n matches. This rolling metric offers a deeper insight into a team's overall form beyond just their winning record. Similarly to figure 4.4, figure 4.5 shows the rolling goal difference over the last 3, 5, and 10 matches for Manchester City and Everton. Manchester City exhibits larger fluctuations and more pronounced peaks, indicative of their high-scoring games and generally consistent performance. In contrast, Everton's graph shows a more inconsistent pattern, reflecting their struggles to maintain form over the same periods.

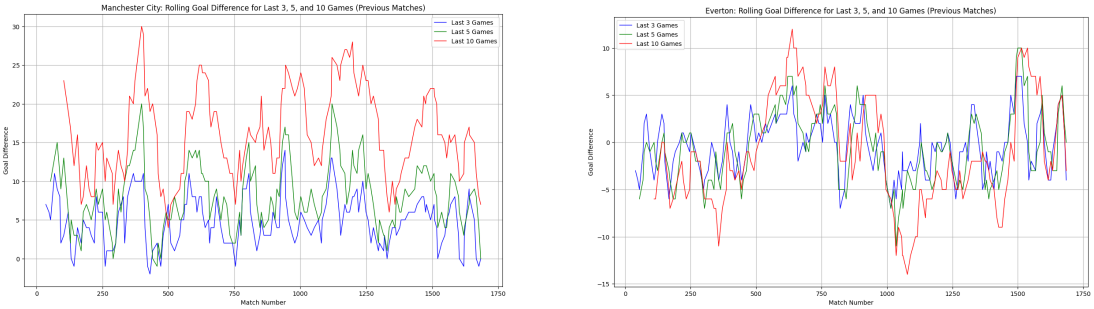

Figure 4.5: Rolling Goal Difference for Last 3, 5, and 10 Games for Manchester City and Everton

4.3 Training and Methodology

This section outlines the comprehensive approach taken to develop, train, and evaluate a series of pre-game baseline models aimed at predicting the outcome of football matches—specifically, whether the home team wins, the away team wins, or the match ends in a draw. As a multi-class classification problem, the objective is to develop models that not only accurately predict match outcomes but also identify and exploit potential inefficiencies in the odds set. These odds represent implied probabilities, and by comparing model-generated confidence levels with these probabilities, we can devise a betting strategy that seeks to capitalise on mispriced odds. The following sections will detail the methodology employed, model selections and training.

4.3.1 Model Selection

When addressing the multiclass classification challenge, the selection of models is influenced by key factors such as prediction accuracy, handling of class imbalance, computational performance, and model interpretability. The selected models must provide accurate full-time result predictions and assess the certainty of each output, transforming these into probabilities. Therefore, the algorithms must also account for the variability within football data, such as hidden feature interactions and the biased distribution toward certain results, like the home advantage.

Logistic Regression

Logistic Regression is a simple yet effective model for binary and multiclass classification tasks. It estimates the probability of a match outcome by assuming a linear relationship between the input features and the log-odds of the outcomes. Its strength lies in its interpretability and ease of implementation, making it a good baseline for comparison. However, it may struggle with capturing complex, non-linear relationships often present in football match data.

XGBoost

XGBoost is a powerful and efficient gradient boosting algorithm that excels in handling structured data with numerous features. It builds an ensemble of decision trees, each correcting the errors of the previous ones, which makes it effective at capturing complex non-linear interactions. XGBoost's ability to prevent overfitting through regularisation and its scalability make it particularly suitable for predicting football outcomes. Its flexibility and strong performance in various scenarios have made it a popular choice in the machine learning community.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a robust classification algorithm that finds the optimal hyperplane to separate different classes in the data. It is especially useful in cases where the data is not linearly separable, as it can use kernel functions to map the data into higher dimensions. A SVM is well-suited for football prediction tasks where class boundaries are not easily defined. However, it can be computationally intensive, requiring careful tuning for best results.

Random Forest Classifier

Random Forest Classifier is an ensemble method that combines multiple decision trees to improve prediction accuracy and reduce overfitting. It works well with a variety of data types and is particularly effective at handling complex feature interactions, which are common in football match prediction. Random Forest is also valuable for its ability to provide insights into feature importance, helping to identify key factors influencing match outcomes. Its robustness and versatility make it a strong candidate for this task.

Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) is a type of artificial neural network that excels at modelling non-linear relationships in data. With multiple layers of neurons, MLP can capture intricate patterns in football match features, making it powerful for prediction tasks. Although MLPs require careful

tuning and can be computationally demanding, their ability to learn from large datasets and generalise well makes them a valuable tool in this context.

4.3.2 Training Methodology

The training methodology in this study involves splitting the dataset into training and test sets, with an optional validation split depending on the scenario. Following the data pre-processing pipeline, we obtain a dataset that is cleaned, restructured, and augmented with specific team features. To access this data in a way that reflects the historical context—meaning going back in time and accessing these statistics as they were at that moment—we first compute the relevant statistics for each match using all available data up to that point in time. These computed statistics are then appended to every row in the dataset to ensure that the training environment closely resembles the testing environment. The data is split chronologically to ensure that the model is trained on past data and tested on future, unseen events, which is critical for time-dependent predictions like football matches. Each model is trained on the training set, and performance is evaluated on the test set to measure the model's generalisation ability. Additionally, K-fold cross-validation is used during training to assess model stability and prevent overfitting, ensuring that the chosen model performs consistently across different subsets of the data.

4.4 Testing and Evaluation

To obtain preliminary results, the models (XGB, SVM, RFC, MLP) were tested on the full set of features for each dataset defined in Table 4.2. Additionally, a benchmark Logistic Regression (LR) model was tested.

4.4.1 Evaluation

Following the training process, K-fold cross-validation was used to assess the models using various metrics, including Accuracy, F1 Score, Precision, and Recall. Each of these metrics provides unique insights into the models' performance, helping to gauge their effectiveness and reliability within a predictive framework.

Accuracy is a straightforward metric that measures the proportion of correct predictions out of the total number of predictions. It is calculated using the formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.1}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

Precision measures the accuracy of positive predictions, defined as the ratio of true positives to the sum of true positives and false positives:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.2}$$

It indicates how many of the predicted positive outcomes were actually correct.

Recall (also known as Sensitivity or True Positive Rate) assesses how well the model captures actual positive cases, calculated as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.3}$$

It shows the ability of the model to find all the relevant positive instances in the dataset.

F1 Score is the harmonic mean of Precision and Recall, providing a balanced measure when the class distribution is uneven:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

This score is particularly useful in scenarios where both precision and recall are important, as it combines both metrics into a single number.

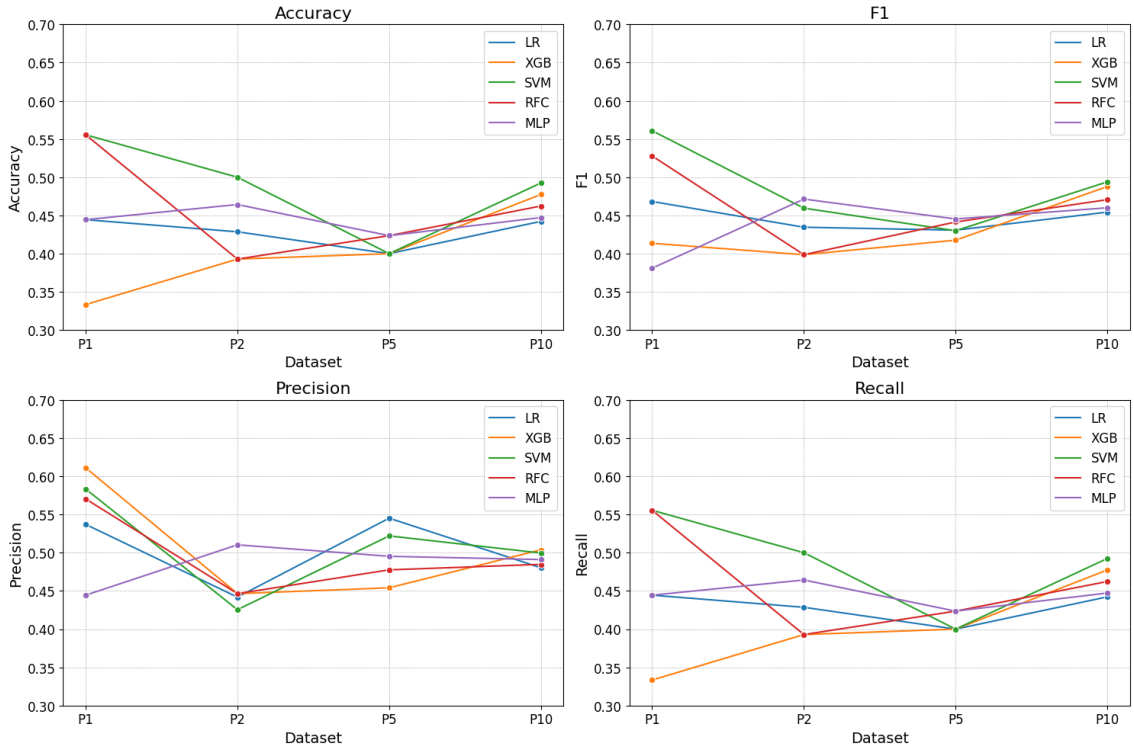

Figure 4.6: Accuracy, F1, Precision and Recall results from models LR, XGB, SVM, RFC, MLP

Figure 4.6 presents the preliminary accuracy, F1, precision, and recall metrics from these models on every dataset. The accuracy results clearly indicate that LR performs worse compared to models that incorporate a wider range of features. Specifically, LR's accuracy peaks at around 45% on the P1 dataset and declines to approximately 40% on the P5 dataset, highlighting its limitations in capturing the complexities of the data as the dataset size increases. Among the more advanced models, XGB and SVM consistently outperform others, particularly on the larger dataset P10. For instance, SVM achieves an accuracy of around 56% on the P1 dataset but drops to nearly 50% on the P10 dataset. Similarly, RFC shows a strong performance, with its accuracy ranging from 56% on P1 but then fails to achieve the same on the P10. These models also excel in other metrics; for example, XGB's F1 score reaches around 0.56 on P1, while RFC follows closely with a score of 0.53 on the same dataset. SVM and MLP also demonstrate competitive results, with SVM particularly excelling in recall on the P2 dataset, achieving a recall of 0.5. MLP, on the other hand, maintains consistent performance across all datasets, with accuracy hovering around 45%.

4.4.2 Feature Selection

Feature selection is a critical process carried out to enhance the accuracy and efficiency of the preliminary models by identifying and removing potentially redundant or harmful features. By systematically evaluating and excluding low-impact variables from the training set, the plan is to distil the dataset to only the most influential features, thereby improving model interpretability and predictive power. This step not only simplifies the model by reducing the dimensionality but also optimises computational resources, ensuring that the model focuses on the most relevant data for more accurate and robust predictions.

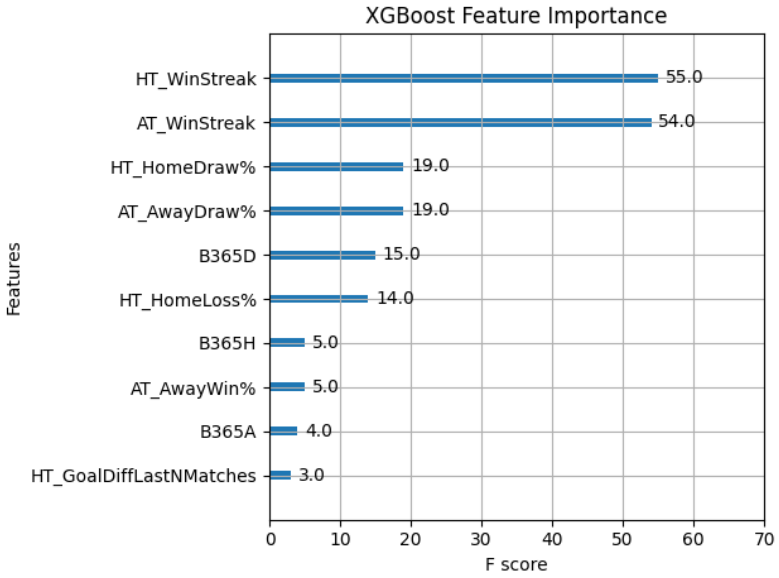

Figure 4.7: Top 10 Features Importance from XGBoost

Applying this methodology to the XGBoost model, the plot in Figure 4.7 highlights the significance of newly developed team statistics, such as win streaks, as well as traditional betting odds. The high importance of features like `HT_WinStreak` and `AT_WinStreak` underscores the predictive power of a team's recent performance, this aligns with the chapter's discussion on the importance of form as an indicator of match outcomes. Additionally, including betting odds (`B365D`, `B365H`, and `B365A`) as significant features confirms their role as implied probabilities, further demonstrating their value in predicting match results. This plot not only validates the selection of these features but also reinforces the decision to exclude less impactful variables, which aren't present in the top 10 features and include namely, Shots on Goal Per Match, Goal Accuracy and Shots On Target Accuracy.

4.5 Results

The final training and testing of the models was done with the selected features in Figure 4.7. The results, including accuracy, F1 score, precision, recall and crossvalidation for each model on all four data splits, are presented in Table 4.3.

The RFC marked a significant improvement from the 56% accuracy observed in the preliminary results to 67% on the selected features. This suggests that RFC is particularly effective on smaller datasets where recent data might have a higher relevance. However, as the dataset size increases, RFC's performance diminishes, with an accuracy of only 51% on the P10 dataset. For the larger dataset P10, SVM outperformed other models, achieving an accuracy of 53%. On average, the models performed with a mean accuracy of 54.1% across all datasets, demonstrating moderate predictive capability. Overall, while RFC and SVM showed strong performance in specific scenarios, no single model consistently outperformed others across all datasets.

Figure 4.8 presents a confusion matrix that visualises how well the SVM model performed on the P10 dataset, categorising predictions into Home Win, Away Win, and Draw. The cells along the diagonal—showing counts of 67 for Home Wins, 37 for Away Wins, and 2 for Draws—indicate where the model correctly predicted the outcome, aligning with the actual result (true positives). The intensity of the cell colour corresponds to the frequency of these correct predictions, with darker shades representing a higher number of correct matches, as highlighted by the accompanying colour scale on the right side of the matrix.

Model	Dataset	Accuracy	F1	Precision	Recall	CrossVal
LR	P1	0.67	0.62	0.62	0.67	0.58
	P2	0.54	0.48	0.44	0.54	0.62
	P5	0.49	0.48	0.49	0.49	0.60
	P10	0.51	0.49	0.49	0.51	0.59
XGB	P1	0.65	0.57	0.55	0.65	0.58
	P2	0.56	0.52	0.52	0.56	0.59
	P5	0.51	0.48	0.49	0.51	0.53
	P10	0.47	0.43	0.42	0.47	0.58
SVM	P1	0.63	0.55	0.56	0.63	0.59
	P2	0.54	0.48	0.44	0.54	0.61
	P5	0.51	0.46	0.47	0.51	0.60
	P10	0.53	0.48	0.56	0.53	0.59
RFC	P1	0.67	0.62	0.62	0.67	0.58
	P2	0.54	0.48	0.44	0.54	0.62
	P5	0.48	0.44	0.43	0.48	0.53
	P10	0.51	0.46	0.47	0.51	0.60
MLP	P1	0.65	0.57	0.55	0.65	0.58
	P2	0.54	0.48	0.44	0.54	0.61
	P5	0.48	0.44	0.42	0.48	0.53
	P10	0.42	0.43	0.46	0.42	0.55

Table 4.3: Model performance comparison across different datasets

The model shows strong predictive capability for Home Wins, as indicated by the relatively dark and dominant diagonal cell corresponding to this class. Away Wins are predicted moderately well, with 37 correct predictions, but there is a notable level of misclassification where some Away Wins were incorrectly predicted as Home Wins (26) or Draws (12). Draws are significantly underpredicted, with only 2 correct predictions, and a large number of actual Draws were misclassified as either Home Wins (28) or Away Wins (12). This underperformance in predicting Draws suggests a potential class imbalance or model bias towards predicting the more frequent outcomes of Home or Away Wins, likely due to the lower representation of Draws in the training data. Consequently, the model's ability to predict Draws is limited, which could impact its overall performance in scenarios where Draws are more prevalent.

4.6 Discussion

In this chapter, we undertook a comprehensive approach to developing baseline models for pregame football predictions, focusing on the full-time result of matches. The process began with a detailed data exploration and transformation phase, where we pre-processed and engineered features from up to 10 seasons of English Premier League (EPL) data. These steps were crucial in ensuring the quality and relevance of the input data, which involved cleaning the dataset, splitting it chronologically to avoid data leakage, and performing feature engineering to extract key statistical profiles for each team. The correlation matrices generated during data exploration played a pivotal role in identifying significant relationships between variables, guiding our feature selection process, and ensuring that the most influential factors, such as team form and recent performance metrics, were included in the models.

The models were then trained and evaluated using a set of established metrics, including accuracy, F1 score, precision, and recall. The Random Forest Classifier (RFC) initially demonstrated strong performance on smaller datasets, achieving an accuracy of 67% on the P1 dataset. However, as the dataset size increased, the Support Vector Machine (SVM) model emerged as the most robust, with an accuracy of 53% on the larger P10 dataset. This highlights the importance of dataset size

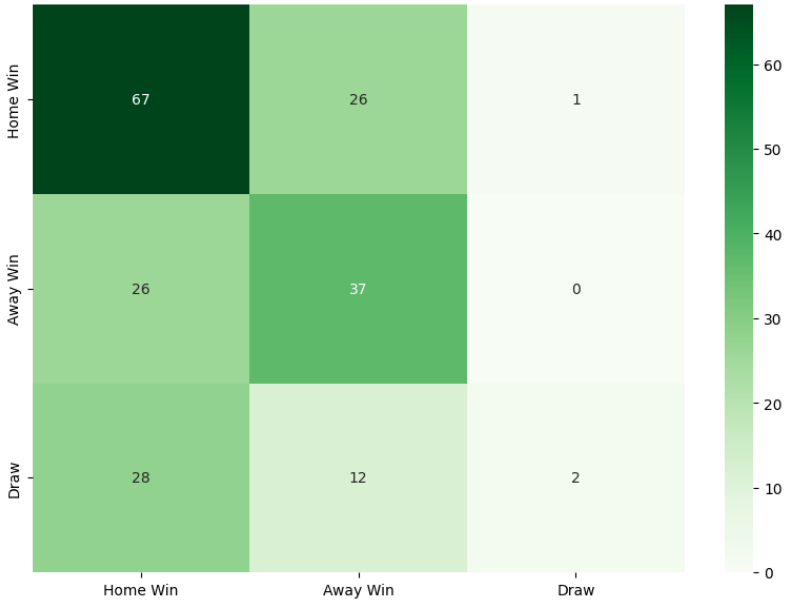

Figure 4.8: Confusion Matrix for SVM on P10 Dataset: Performance Breakdown of Home Wins, Away Wins, and Draws Predictions

and complexity in model performance, with different models excelling under varying conditions. The results also underscored the critical role of feature selection, where the model's focus on recent performance indicators like win streaks and goal differences significantly improved predictive accuracy. The discussion of model results reflects the pragmatic approach of the chapter, where iterative testing and evaluation informed the final model selection, providing a solid foundation for future experiments in football match predictions.

Ultimately, the chapter illustrated the efficacy of the research and machine learning platform developed in Chapter 3, which was instrumental in dynamically retrieving and processing the FootyStats data. The platform's capabilities allowed for an efficient and thorough analysis, demonstrating how well-engineered machine learning pipelines can be leveraged to create reliable baseline models in sports betting scenarios. The chapter's findings not only contribute to the ongoing research in sports analytics but also lay the groundwork for more sophisticated models in subsequent studies, where these baselines will serve as benchmarks.

Chapter 5

Deep Learning for In-play Momentum Betting

This chapter explores the implementation of in-play momentum betting in football by leveraging deep learning models such as stacked LSTMs, Transformers, and CNNs to predict real-time odds movement based on betting exchange data. We detail the process of feature engineering and transformation, followed by model selection, training, and evaluation, highlighting the strengths and weaknesses of each approach.

5.1 Background

The in-play betting environment stands out as both a thrilling and demanding area. This fast-paced style of betting, where wagers are made during the ongoing event, demands quick and well-informed decisions. This experiment uses the in-play data captured through our custom-built data infrastructure, as detailed in Chapter 3. After undergoing a series of transformations, pre-processing steps, and feature engineering, the data is used to train various predictive models. Special emphasis is placed on stacked LSTM networks, known for their effectiveness in time series forecasting. We compare the performance of these networks against a attention based transformer model and a convolutional neural network.

This experiment is driven by three key motivations. Firstly, the inherently unpredictable nature of live football matches, characterised by their frequently changing dynamics, presents a strong foundation for predictive modelling. Precise, real-time predictions can improve betting strategies, providing bettors with an advantage in the rapidly changing world of in-play betting. Thirdly, the challenge of efficiently processing live data streams and making timely predictions makes this a particularly intriguing problem for time series based deep learning models.

Traditionally, betting was limited to forecasts made before a match began. However, advancements in technology and the advent of live broadcasting have transformed this approach, enabling bettors to interact with games in real-time. High-frequency trading (HFT) dramatically changed financial markets by facilitating rapid transactions, supported by exchanges providing co-location services. In a similar vein, speed is a critical factor in sports betting, especially for in-play scenarios, where wagers are placed on unfolding events. The momentum of a game can shift quickly, causing betting odds to change just as fast, thereby presenting attentive bettors with multiple opportunities to place strategic bets.

In-play models use live data streams, including real-time game updates and order book data from betting exchanges. Divos [8] was one of the first to provide a comprehensive view of the in-play football betting market. Divos demonstrated the applicability of financial mathematics concepts to betting. Notably, he introduced the Constant Intensity Model, a risk-neutral framework for pricing and hedging in-play bets, similar to the Black-Scholes model. Additionally, by recognising the presence of an implied intensity smile in football betting, Divos introduced the Local Intensity

model, inspired by the local volatility model from finance. In general, Neural networks have shown good performance in predicting problems. Rahman [32] proposed a football game prediction framework based on LSTM architecture, achieving a 63.3% accuracy. Johnson et al. [20] used tactical feature engineering to compress space-time and promote the fine-tuning of three pre-trained CNNs. Zhang et al. [40] introduced the AS LSTM model, integrating an attention mechanism with LSTM to capture the team's short-term state and better explore the team's potential characteristics, enhancing the prediction of result outcomes. Then beyond football, Yu et al. [41] proposed a method combining deep Bi-LSTM with Mixture Density Network (MDN) to real-world basketball trajectory data to help players decide when and where to shoot. Oytun [30] used a backpropagation neural network and LSTM to predict the performance of female handball players.

At the centre of this experiment's lies the implementation of an LSTM. LSTM networks [17] are a type of recurrent neural network (RNN) capable of learning order dependence in sequence prediction problems by retaining information about past inputs for a variable amount of time and capable of extracting non-linear relationships between features. This retention period is not fixed but depends on their weights, the number of stacked layers, and the input data. LSTMs were a solution introduced to expand recurrent neural networks where learning to store pertinent information over extended time intervals through recurrent backpropagation can be problematic, primarily due to problems with error gradients over long sequences. LSTMs offer an efficient gradient-based method which learns when to forget previous hidden states and when to update hidden states given new information. Incorporating memory units enables LSTM architectures to learn complex long term temporal dynamics that a standard RNN is not capable of.

5.2 Data

In its most basic form, exchange data is used to represent the price ladder, providing bettors with a clear view of the market's status, and by extension, the status of the underlying event. Exchange data is crucial to in-play betting across any sports market. In more advanced scenarios, exchange data facilitates backtesting, post-trade evaluation of strategies, and the generation of trading signals using statistical analysis or machine learning. Updates, including the best prices available for backing and laying, liquidity at each tick, and the last traded price, are essential for constructing the live price ladder. It is important to note that the structure of exchange data remains consistent regardless of the type of event (e.g., sport) it relates to. Table 5.1 outlines the fields required to build the order book and provides their descriptions.

Code	Field	Details
pt	Published Time	Time of update
tv	Traded Volume	Cumulative volume matched in the market
ltp	Last Traded Price	Most recent price at which a trade occurred
trd	Traded	Change in price and volume during a trade
atb	Available to Back	Best available back prices and volumes
atl	Available to Lay	Best available lay prices and volumes

Table 5.1: Fields Required for Constructing the Order Book in Exchange Data, Including Their Codes and Descriptions

Betfair Premium offers a PRO Exchange Historical Data service, which provides an extensive collection of historical data across various sports markets. This data is stored within the research platform's database, as detailed in Chapter 3. The PRO subscription delivers the same level of detail as live updates, which is crucial for backtesting and for strategies that were developed using historical exchange data. In figure 5.1 we see a typical Betfair Exchange ladder. The favourite is priced with a best back of 1.70 and a best lay of 1.71, giving a spread of 0.01 and a percentage spread of approximately 0.59%. For model training purposes we convert the market updates into live ladder prices at each timestamp, that way we can see the real state of the market at any point in time.

Back Volume	Price	Lay Volume	Traded
1983.7	1.65		0
2283.58	1.66		34.46
335.5	1.67		64.76
1562.43	1.68		0
1507.72	1.69		544.54
2134.17	1.70		16555.34
	1.71	1996.48	15871.26
	1.72	1281.7	23738.6
	1.73	1095.18	17688.39
	1.74	208.78	3539.11
	1.75	4.21	234.73

Table 5.2: Limit Order Book Price Ladder

5.2.1 Data Exploration

The movement of prices in sports markets is influenced by the nature of the sport, the structure of the event, and the real-time developments during the match. Prices will drift, experience jumps, and exhibit volatility. In football, matches have a fixed duration of 90 minutes, with prices naturally drifting as time progresses. As time progresses without a goal, the probability of certain outcomes (e.g., a win for either team) naturally decreases. Goals cause immediate jumps in prices, reflecting the significant change in the likelihood of different outcomes. Volatility increases in the final minutes as the match outcome becomes more certain. Key events like goals, penalties, or red cards significantly impact volatility. Taking the Tottenham vs Aston Villa game as illustration, figure 5.1 shows the fluctuation of odds that provide a clear narrative of the matches progression. As Aston Villa scored the first goal, a pronounced spike in the odds for a Tottenham win reflected the diminished probability of that outcome.

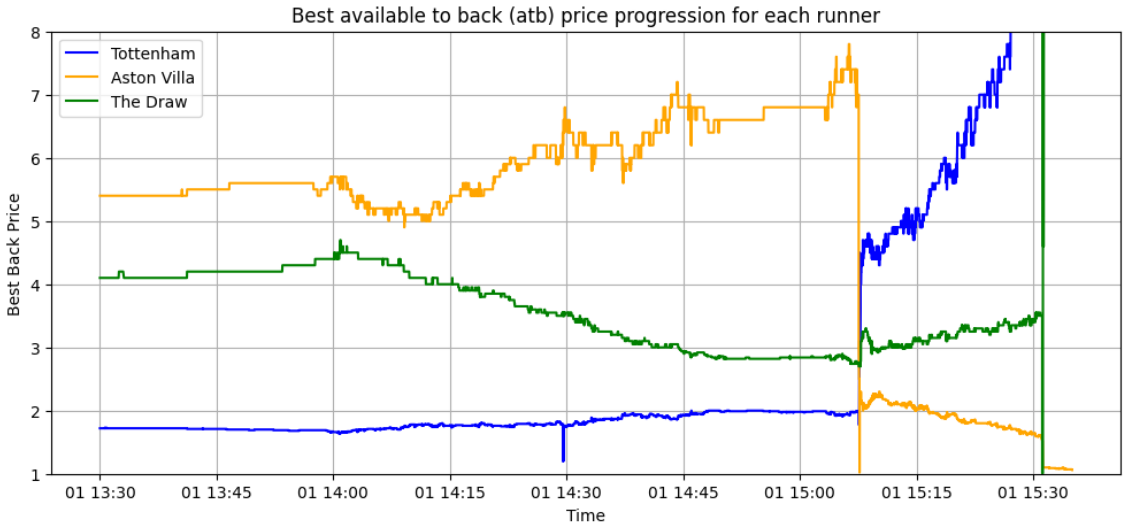

Figure 5.1: In-play Best Available Back Price (ATB) Progression for Tottenham, Aston Villa, and The Draw during the Aston Villa vs. Tottenham Match.

During a live match, the fluctuations in odds generate inherent noise in the data. This noise results from various unpredictable factors within the game, such as unexpected player actions, referee decisions, or crowd reactions. While major events like goals lead to noticeable changes in odds, the ongoing fluctuations between these events stem from the market's collective effort to predict the unpredictable aspects of the game in real-time. This noise complicates predictive modeling by obscuring genuine patterns and diminishing the clarity of signals needed for accurate predictions.

5.2.2 Data Pre-Processing

Similarly to Chapter 4, data pre-processing forms a crucial part of any machine learning pipeline. It involves refining raw data into a format that can be readily consumed by deep learning models. This process includes data cleaning to handle missing values, outlier detection and rectification, normalisation to scale features appropriately, and feature engineering. These steps are essential for mitigating potential biases and improving the performance and generalisability of the predictive models.

Data Cleaning

Data cleaning is the initial step post-collection, crucial for ensuring the quality and reliability of the dataset used for model training. The raw market data, as captured by our custom infrastructure, often contains irregularities such as missing values, outliers, and inconsistencies in the time series index. To address these, we begin by converting the timestamps into a standardised datetime format, ensuring temporal accuracy across the dataset. The data is then restructured, with each runner's Last Traded Price (LTP) represented as a distinct column, facilitating a more granular analysis of each runner's performance. Resampling the data at consistent intervals—every ten seconds—mitigates issues arising from irregular time gaps between observations, while forward-filling ensures continuity by interpolating missing values based on the last available data point. This method maintains the integrity of the time series, which is vital for the sequential nature of the models being employed. Subsequently, rows containing any residual missing values are eliminated, further refining the dataset. Finally, the detection and removal of outliers using the Interquartile Range (IQR) method are implemented to exclude anomalous data points that could skew the model's predictions. These steps collectively transform the raw market data into a clean, structured, and reliable input, optimising the accuracy and generalisability of the deep learning models.

Normalisation is a critical step in the data pre-processing pipeline, particularly when working with machine learning models like deep neural networks that are sensitive to the scale of input features. In this experiment, Min-Max Scaling was employed to normalise the dataset, ensuring that all features are scaled to a consistent range, between 0 and 1. This process prevents features with larger numerical ranges from disproportionately influencing the model's learning process. The scaler was fitted on the training data to establish the appropriate scaling parameters, which were then applied consistently across the training, validation, and test datasets. This approach preserves the integrity of the data distribution while enhancing the performance and stability of the deep learning models.

Data Splitting

The models employed in this experiment are designed to handle sequences of data, making them particularly well-suited for the time series nature of in-play betting markets. In our context, these sequences represent contiguous time windows of exchange data, capturing the dynamic fluctuations in market prices and volumes as the match progresses. Given that the raw historical data arrives at irregular intervals, it is crucial to standardise the data into consistent sequences to facilitate effective model training.

After the data has been cleaned, we generate these sequences by sliding a fixed-length window over the time-indexed data. Each sequence is paired with a corresponding label that represents the target prediction—in our case the next LTP for each runner. This sliding window approach ensures that each sequence captures the temporal dependencies within the market data, which are critical for making accurate predictions.

The data is then split into three distinct sets: training, validation, and test sets. The training set, comprising the majority of the data, is used to fit the model. To assess the model's ability to generalise and to fine-tune hyperparameters, a separate validation set is used. Then, a test set, which remains unseen during the training and validation phases, is used to evaluate the model's performance on new, unseen data, providing an indication of how well the model is likely to perform in a real-world betting scenario. Throughout the experiment, the dataset is divided such that 10%

of the data is allocated to the test set. The remaining 90% is further divided, with 15% assigned to the validation set and 75% to the training set. This splitting ensures that each set retains the temporal structure of the original data, which is vital for sequence-based models. Table 5.3 illustrates this breakdown.

Dataset	Percentage of Total Data	Purpose
Training	75%	Model training
Validation	15%	Hyperparameter tuning
Testing	10%	Model evaluation

Table 5.3: Dataset Distribution and Purpose

Feature Engineering

With the initial data cleaning complete, we proceeded to the feature creation phase. Although most of our primary features were derived directly from the data, we also explored the creation of synthetic features to capture underlying patterns or relationships.

One of the key features engineered was the Rate of Change (RoC), which measures the percentage change in data over a specified period. RoC is calculated as:

$$\text{RoC}_p = \left(\frac{P_t - P_{t-p}}{P_{t-p}} \right) \times 100 \tag{5.1}$$

where P_t represents the value at the current time t and P_{t-p} is the value p periods ago. This formula expresses how much the value has changed relative to its value p periods in the past, thus providing a percentage measure of the momentum. This feature provides insights into the momentum and strength of the data's movement, which is crucial for understanding short-term market dynamics.

Another feature was the Moving Average (MA), which smooths the data by averaging it over a specified window, reducing noise and helping to identify trends more clearly. The MA is computed as:

$$\text{MA}_t = \frac{1}{N} \sum_{i=0}^{N-1} P_{t-i} \tag{5.2}$$

where N is the window size and P is the price at time t. This feature helps to mitigate the impact of short-term fluctuations and highlights longer-term trends.

Additionally, the Moving Average Convergence Divergence (MACD) was used, which compares short-term and long-term exponential moving averages (EMAs) to identify momentum shifts. MACD is calculated as:

$$\text{MACD} = \text{EMA}_{\text{short}} - \text{EMA}_{\text{long}} \tag{5.3}$$

The signal line, a moving average of the MACD, is used to identify buy or sell signals based on the crossover points between the MACD and the signal line. This feature is particularly useful for detecting changes in the momentum and potential reversals in the trend.

5.3 Training and Methodology

In this section, we detail the approach taken to train and evaluate the predictive models for in-play momentum betting. The training process is designed to leverage the extensive in-play exchange betting data by employing a variety of advanced machine learning models, including stacked Long Short-Term Memory (LSTM) networks, Transformers, and Convolutional Neural Networks

(CNNs). Each model is carefully selected based on its suitability for handling the complex, time-dependent nature of the data. We outline the methodology for model selection, hyperparameter tuning, and the evaluation metrics used to assess performance.

5.3.1 Model Selection

Determining the outcome of football matches requires analysing sequences where previous events play a crucial role in shaping predictions. Advanced deep learning techniques, such as Recurrent Neural Networks (RNNs), excel at processing and understanding data that unfolds over time. Among these, Long Short-Term Memory (LSTM) networks are particularly effective in retaining and utilising long-range patterns within the data, making them especially useful for this type of analysis. Beyond RNNs, we also employ a Transformer model, which processes the entire input sequence simultaneously rather than one step at a time. This is achieved through a mechanism called 'self-attention', which allows the model to focus on different parts of the sequence when making predictions. We also explore a combination of these approaches in the form of an Attention-based LSTM (AS-LSTM), which integrates the benefits of both LSTM and attention mechanisms.

While the aforementioned models are primarily used for regression tasks—predicting the exact movement of odds—we also implement a Convolutional Neural Network (CNN) to classify future movements of odds into binary categories, such as "up" or "down." In the context of football betting, the objective is to predict odds for three distinct runners: Home win, Away win, or Draw.

Long Short Term Memory (LSTM)

A stacked LSTM model is employed, where multiple LSTM layers are arranged to increase the depth of the model. This architecture enhances the model's ability to capture and learn more complex temporal relationships within the data. Stacked LSTMs are widely used in fields such as financial forecasting, particularly in predicting stock prices, due to their capability to identify intricate patterns across different time scales. In this application, the stacked LSTM processes sequences of past odds and momentum-related features to predict the odds for three possible outcomes: Home win, Away win, or Draw. By retaining information from previous matches and odds, the model is able to detect patterns and trends that may influence future odds. The hierarchical structure of stacked LSTMs proves particularly advantageous, as each layer can learn different temporal aspects—lower layers focus on short-term dependencies, while higher layers capture long-term trends.

The model architecture implemented takes in sequences of features, each with a dimension of 15, corresponding to the real-time market indicators. This input is then processed by three LSTM layers, each with 50 hidden units. The model is structured to operate in a batch-wise manner, where the LSTM layers are stacked to form a deep recurrent network. The first LSTM layer captures the immediate temporal dependencies, and its hidden states are passed as input to the subsequent LSTM layers, each further refining the temporal patterns learned from the data. The output from the final LSTM layer, which encapsulates the comprehensive sequence information, is then passed through a fully connected layer (GEMM operation) to produce the final output. This output layer, instead of predicting a probability distribution, directly forecasts the next LTP for each runner in the market. The architecture also incorporates operations like slicing and squeezing to handle the specific tensor shapes required by the LSTM and linear layers, ensuring that the final predictions are appropriately formatted. This design allows the model to efficiently manage and transform the sequential betting data into accurate real-time price predictions, capturing both short-term fluctuations and long-term trends inherent in live football matches.

Transformer

The Transformer model implemented is designed to handle the sequential nature of betting exchange data while leveraging the advanced capabilities of self-attention mechanisms. Unlike recurrent networks, the Transformer processes the entire sequence simultaneously, allowing it to model dependencies across all time steps without the need for sequential computation.

The architecture begins with an embedding layer that transforms the input features (with a dimension of 15) into a higher-dimensional space (with a dimension of 64), suitable for processing by the Transformer. Following the embedding, positional encoding is applied to the input data, which helps the model retain information about the relative position of each element in the sequence, a crucial aspect when dealing with time series data. The positional encoding is added to the embedded input, providing the model with a sense of order in the sequence, which is otherwise absent in purely attention-based models. The core of the Transformer consists of multiple layers of encoders and decoders, each equipped with multi-head attention mechanisms and feed-forward neural networks. In this implementation, the model includes two encoder layers and two decoder layers, with four attention heads in each layer. The encoder processes the input sequence, capturing the relationships between all elements in the sequence through self-attention. The decoder then uses this processed information to predict the next LTP for each runner.

A key feature of this model is the use of masking within the Transformer layers, which ensures that the predictions for a particular time step depend only on the past and current time steps, maintaining the causality necessary for time series prediction. After the sequence passes through the Transformer, the output corresponding to the last time step is extracted, as it contains the most relevant information for predicting the next LTP. This output is then passed through a fully connected layer to generate the final prediction.

Attention-based LSTM (AS-LSTM)

The AS-LSTM (Attention-based LSTM) model offers several advantages over a traditional LSTM network by incorporating an attention mechanism that allows the model to focus on the most relevant parts of the input sequence when making predictions. In a standard LSTM, the model processes sequential data by passing information from one time step to the next, relying solely on the final hidden state to make predictions. This can lead to the loss of information, especially in long sequences where earlier data points might be crucial for accurate prediction. By integrating an attention layer, the AS-LSTM model overcomes this limitation by computing attention weights for each time step, which allows the model to dynamically select and weigh different parts of the sequence based on their relevance. This selective focus not only enhances the model's ability to capture important temporal patterns but also improves interpretability, as the model's attention weights can be examined to understand which parts of the sequence influenced the prediction.

The model starts with an LSTM layer that processes the input sequence of features, each with a dimension of 15 followed by two additional LSTM layers, each with 50 hidden units. As the sequence moves through these layers, the model generates hidden states that encapsulate learned temporal patterns. An attention layer is then applied to the output of the final LSTM layer. This layer computes attention scores, which are normalised using a softmax function to produce attention weights. These weights create a context vector—a weighted sum of the LSTM outputs—that highlights the most relevant time steps for prediction. This context vector is passed through a fully connected layer to predict the next LTP for each runner. By combining LSTM and attention mechanisms, the AS-LSTM model efficiently transforms complex sequential data into accurate and interpretable predictions.

Convolution Neural Network (CNN)

The Convolutional Neural Network (CNN) model is designed to classify future movements of odds into binary categories "up" or "down." Unlike traditional time series models like LSTMs, which process data sequentially, the CNN is adept at capturing spatial patterns within the data by applying convolutional filters over the input sequences. This makes it particularly effective for detecting local patterns in time series data, such as sudden shifts or spikes in market indicators, which are crucial for predicting short-term movements in betting odds.

The model architecture begins with an input layer that takes in sequences of features for each time step, where the input dimension corresponds to the number of features. The model consists of two convolutional blocks, each composed of a 1D convolutional layer followed by a max-pooling layer. The first convolutional layer applies 32 filters with a kernel size of 3, which extracts local features

from the input sequences. The output is then downsampled by the max-pooling layer, which reduces the sequence length by half, retaining the most prominent features. This is followed by a second convolutional block with 64 filters, further refining the feature extraction process. After the convolutional layers, the output is flattened and passed through a fully connected layer with 128 units, which transforms the extracted features into a higher-level representation. A dropout layer is applied to this fully connected layer to prevent overfitting, ensuring the model generalises well to unseen data. The final output layer is a fully connected layer that maps the features to the predicted classes for each runner—"up" or "down"—indicating the expected movement of the next Last Traded Price (LTP) for each runner.

5.3.2 Training Methodology

The training process is guided by a series of hyperparameters, which are predefined settings that dictate how models learn. For the LSTM, AS-LSTM, and Transformer models, the training setup is consistent, given their shared focus on regression tasks. These models use Mean Squared Error (MSE) as the loss function, which is ideal for minimising the difference between predicted and actual values. They are optimised using the Adam optimiser with a learning rate of 0.001, and each is trained over 1000 epochs. Additionally, AdamW—an extension of the Adam optimiser with weight decay—is used to help regularise the models by preventing overfitting.

On the other hand, the CNN model, designed for binary classification, employs Cross-Entropy Loss, which is more appropriate for classification tasks. The CNN is trained with a lower learning rate of 0.00001 over 3000 epochs to effectively capture and learn from the data. After processing all batches for an epoch, the model's performance is evaluated on a validation set to gauge its generalisation capability. This process is repeated for all epochs, allowing each model to continually refine its weights and improve its predictions.

To optimise model performance, grid search was employed for hyperparameter tuning, systematically exploring a range of parameters to identify the best configuration for each model. This automated approach ensured that the chosen hyperparameters, such as learning rate and batch size, were optimal for the learning process. Additionally, a one-cycle learning rate policy was implemented, where the learning rate starts low, increases during the first half of training, and then decreases in the second half. This dynamic adjustment promotes faster convergence and helps the model avoid getting trapped in local minima. To further enhance generalisation and prevent overfitting, early stopping was applied, monitoring validation performance and halting training when improvement ceases, ensuring that the model remains robust on new, unseen data.

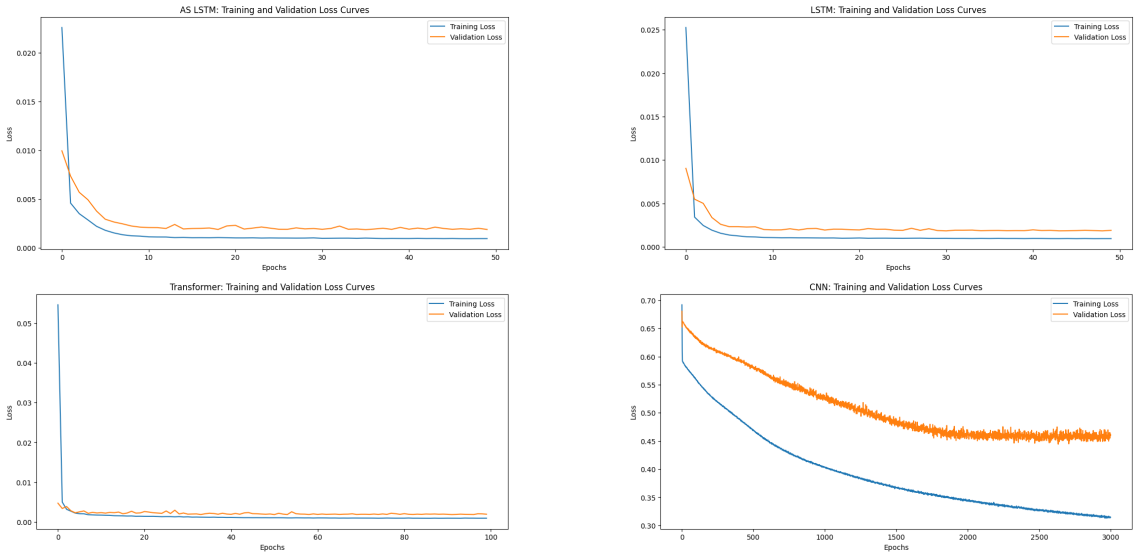

Figure 5.2: Training and Validation Loss During Model Training.

5.4 Testing and Evaluation

To obtain results, the models (LSTM, Transformer, AS LSTM, CNN) were tested on the dataset. The models developed offer crucial insights into predicting the potential outcomes of football matches, whether it's a home victory, an away win, or a draw. Our objective is to align these predictions as closely as possible with the actual betting market odds. By accurately predicting these key outcomes, the models open up a wide array of betting possibilities for each game. The predicted odds not only assess the probability of each scenario—be it a win, loss, or draw—but also quantify the confidence level of these predictions. This thorough approach allows for a deeper understanding of match dynamics, equipping bettors with practical insights to develop more effective betting strategies.

5.4.1 Evaluation

Since we are training both regression and classification models, there exists a distinct set of evaluation metrics tailored to each. These metrics provide a comprehensive view of each model's performance, allowing us to make informed decisions for real-world football betting strategies. For the regression models, the evaluation begins with the Mean Squared Error (MSE), which measures the average of the squared differences between the predicted and actual values. MSE is calculated as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.4}$$

where y_i represents the true value, and $\hat{y}_i$ represents the predicted value. Lower values of MSE indicate better model performance by minimising prediction errors.

Beyond MSE, additional metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were calculated to offer a more granular view of model performance. RMSE is the square root of MSE, providing error measurements in the same units as the output variable, making it more interpretable in the context of market price predictions:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.5}$$

MAE measures the average magnitude of the errors in the predictions, offering insight into the absolute differences between predicted and actual values without considering the direction of the errors. It is computed as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.6}$$

To further assess the model's ability to predict market outcomes, the R-squared (R^2) metric was employed. R^2 provides a measure of the proportion of variance in the dependent variable that is predictable from the independent variables. This metric is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{5.7}$$

where $\bar{y}$ is the mean of the observed data. R^2 values closer to 1 indicate a strong predictive ability, reflecting how well the model's predictions align with the actual outcomes.

Additionally, the Explained Variance Score was computed to evaluate the proportion of variance in the true values that is captured by the model's predictions. This score is calculated as:

$$\text{Explained Variance Score} = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)} \tag{5.8}$$

where $\text{Var}(y)$ is the variance of the true values, and $\text{Var}(y - \hat{y})$ is the variance of the prediction errors. Higher values closer to 1 indicate that the model explains a significant portion of the variance in the target variable.

Together, these metrics—MSE, RMSE, MAE, R^2, and the Explained Variance Score—provide a comprehensive evaluation framework, offering both a broad and detailed assessment of model performance. They enable us to quantify the accuracy of our predictions and understand the nuances of model behaviour, which are crucial for refining our betting strategies in live, in-play football markets.

For the CNN, the evaluation process begins by testing the model using Cross-Entropy Loss,m calculated as:

$$\text{Cross-Entropy Loss} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \tag{5.9}$$

where n is the number of classes, y_i represents the true label for class i (which is 1 if the class is the correct one, and 0 otherwise), and $\hat{y}_i$ is the predicted probability for class i.

In line with the metrics discussed in Chapter 4, additional measures include Accuracy, Precision, Recall, and F1-Score. These metrics collectively offer a detailed assessment of the model's proficiency in categorising the outcomes for each participant in a football match. Accuracy reflects the overall correctness of the model's forecasts, while Precision, Recall, and F1-Score provide a more nuanced understanding of the trade-offs between true positives, false positives, and false negatives.

5.5 Results

Table 5.4 presents the performance metrics of the three deep learning models—LSTM, AS LSTM, and Transformer. The LSTM model outperforms the others across all metrics, demonstrating the lowest RMSE (0.0811) and MAE (0.0375), indicating its superior ability to minimise prediction errors. Additionally, its R-squared value of 0.9457 and EVS of 0.9507 suggest that the LSTM model captures a high proportion of the variance in the betting market data, making it highly effective at predicting the movement of odds. The AS LSTM, which integrates an attention mechanism with the LSTM architecture, shows slightly lower performance than the basic LSTM. While its RMSE (0.0863) and MAE (0.0433) are marginally higher, the AS LSTM still demonstrates strong predictive power, with an R-squared value of 0.9384 and an EVS of 0.9457. This suggests that while the attention mechanism adds value, it may also introduce complexity that slightly reduces the model's overall performance compared to the simpler LSTM.

Model	RMSE	MAE	R-squared	EVS
LSTM	0.0811	0.0375	0.9457	0.9507
AS LSTM	0.0863	0.0433	0.9384	0.9457
Transformer	0.0913	0.0436	0.9321	0.9376

Table 5.4: Performance Metrics for LSTM, AS LSTM, and Transformer Models

Figure 5.3 shows the temporal predictions of each model for the test data runners. It provides insights into how closely the predictions align closely with the real market odds over time. The LSTM model is particularly effective at reaching the height of the peaks more closely than the other models, reflecting its ability to capture sharp changes in market dynamics.

Figure 5.4 presents the residual plots, which depict the differences between the actual and predicted values for the three models. Analysing these residuals can provide insights into areas where the model's performance might be enhanced. With the residuals being consistently close to zero, this suggests that the model's predictions are largely accurate, indicating a good fit.

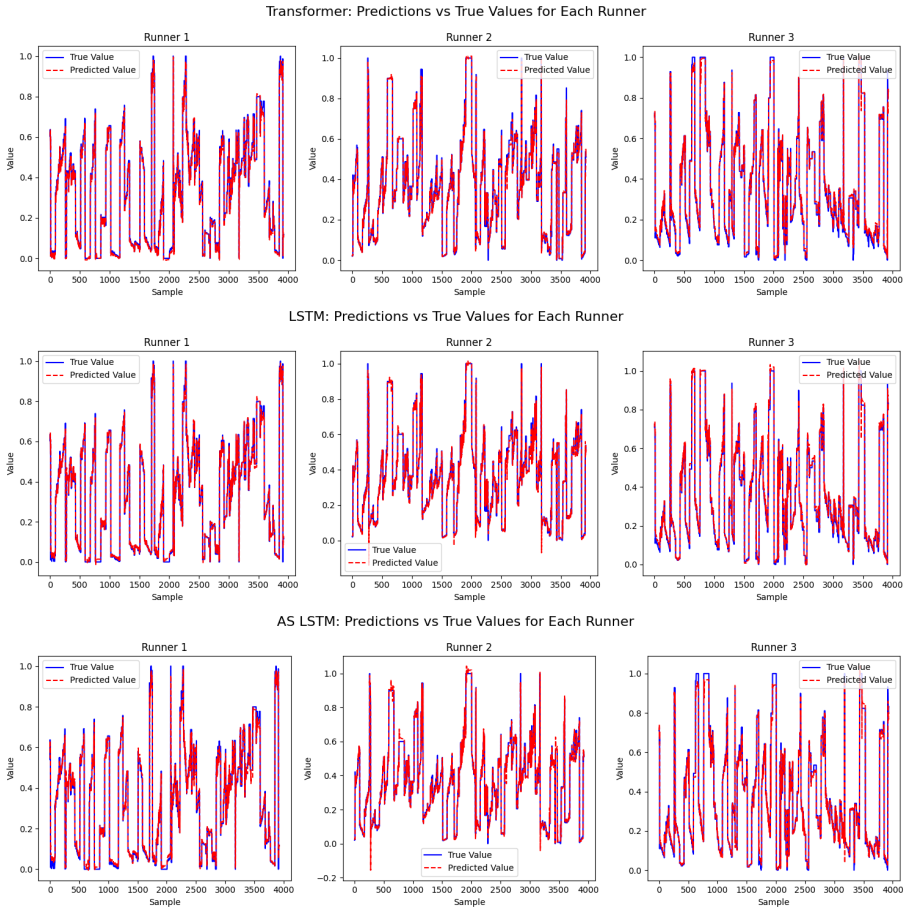

Figure 5.3: Temporal Predictions for each Model and Runner in the Test Data.

Table 5.5 outlines the performance metrics for the model's predictions of price movements across different runners: Home Win, Away Win, and Draw.

Runner	Accuracy	Precision	Recall	F1-score
Home Win	0.6903	0.6435	0.5310	0.5819
Away Win	0.4851	0.4985	0.6067	0.5473
The Draw	0.7635	0.6168	0.5968	0.6066

Table 5.5: CNN Performance Metrics for each Runner.

The model demonstrates the highest accuracy in predicting the price movements for the Draw runner, with an accuracy of 76.35%, precision of 0.6168, recall of 0.5968, and an F1-score of 0.6066. This suggests that the model is particularly effective in capturing market momentum when it comes to predicting draws. For the Home Win runner, the model achieves an accuracy of 69.03%, but with a lower recall and F1-score. This indicates that while the model is precise in its predictions, it may not fully capture all instances of market shifts towards home wins. The prediction of price movements for the Away Win runner presents the greatest challenge, with the model achieving an accuracy of only 48.5%, a precision of 0.4985, and a recall of 0.6067, leading to an F1-score of 0.5473. These metrics reflect the inherent difficulty in forecasting market movements for the Away Win runner, suggesting that this aspect of the model could benefit from further refinement and optimisation. Overall, these classification results do not reach the level of accuracy seen in the regression models, with the lower accuracy across all runners highlighting this discrepancy.

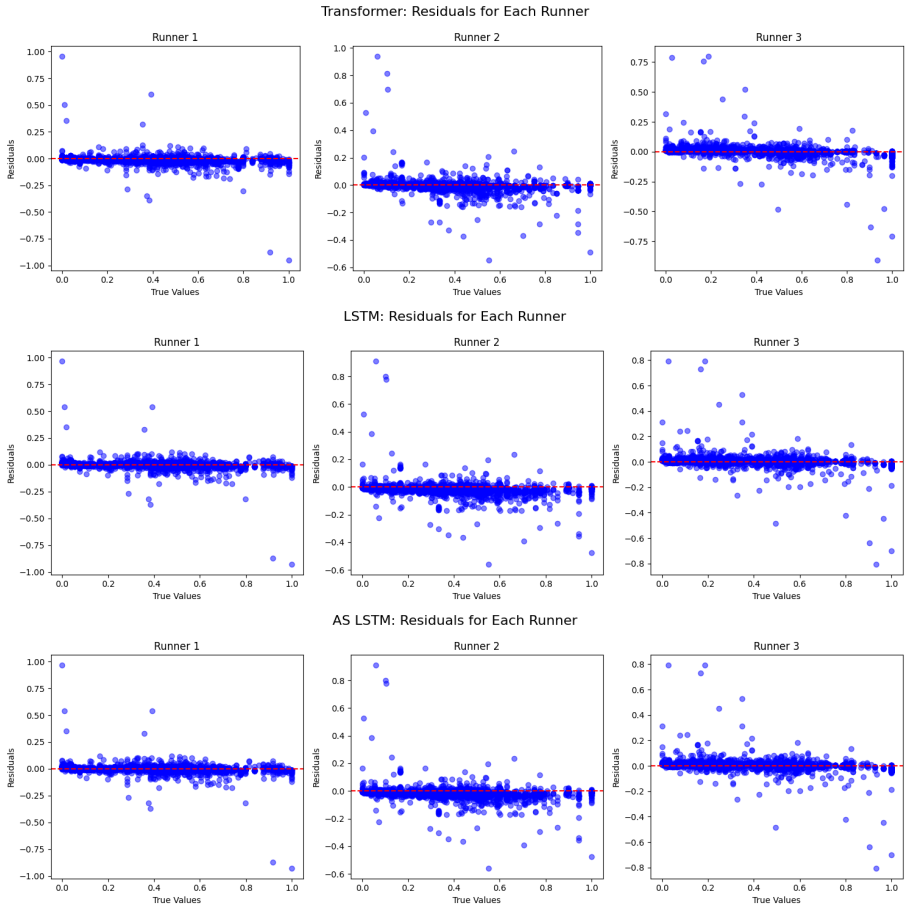

Figure 5.4: Residual Plots for each Model and Runner in the Test Data

5.6 Discussion

In-play betting is characterised by a high level of unpredictability and noise, especially due to the brief 10-second window used for recording the LTP. This short timeframe can cause notable volatility, as it captures the immediate uncertainties and swift changes in how the market perceives a match's outcome. In the realm of football, and particularly with in-play betting, having a deep understanding of how odds can drift—meaning how they progressively shift in one direction over time—can offer a significant strategic advantage. This shift in odds isn't random; it's closely linked to the decreasing time left for an event, such as a goal, to change the course of the game. As the match progresses and the likelihood of specific outcomes becomes clearer, the odds adjust accordingly. This adjustment process is especially pronounced following crucial events such as goals, red cards, or injuries, where the market quickly recalibrates its expectations across all potential outcomes, whether it be a Home Win, Away Win, or Draw. These moments of recalibration can present traders with valuable opportunities if they can accurately anticipate how the market will settle after the event. Within the framework of our models, these trends are effectively captured and utilised, boosting the models' predictive accuracy and potentially adding another layer of complexity to betting strategies. Although our models might occasionally experience brief misalignments during sudden market fluctuations, they are adept at swiftly recalibrating, aligning closely with the market as it steadies after these critical moments.

In this chapter, we explored the complexities and unpredictability of in-play football betting, tackling these challenges using advanced deep learning techniques. We applied a variety of models, including LSTM, AS-LSTM, and Transformer networks, all specifically designed to handle the time-sensitive nature of in-play data. Inspired by methodologies from financial markets, we also integrated additional synthetic features like Rate of Change (RoC), Moving Average (MA), and

Moving Average Convergence Divergence (MACD) to improve the model's capability for making predictions over short intervals. These features, commonly used in financial trading to capture momentum and trend shifts, were particularly effective in improving the model's sensitivity to the nuances of odds movements in real-time betting markets. By integrating these techniques, our models could better understand and predict the ebb and flow of in-play odds, offering a sophisticated approach to the challenging task of in-play momentum betting.

Chapter 6

Conclusion and Future Work

This chapter concludes the thesis by revisiting the key findings, highlighting the achievements, and outlining the industry and scientific contributions presented throughout. It also discusses opportunities for future work to be carried out by other researchers.

6.1 Summary

The betting market, in contrast to traditional financial markets, remains relatively underexplored, presenting a wealth of opportunities for both academic research and commercial ventures. While there are notable similarities between the two domains—such as liquidity provision, limit order book dynamics, and volatility characteristics—algorithmic trading has primarily driven the development and application of machine learning in financial markets. Despite the extensive body of literature on ML applications in finance, its adoption in sports trading has been minimal. This disparity highlights a significant gap and forms the basis for the second and third experiments in this thesis, which focus on advancing predictive modeling in football betting by exploring both pre-game and in-play strategies.

To achieve these aims, the research objectives were meticulously defined. The primary objective focused on creating a comprehensive research and machine learning framework that streamlines data management, model development, and analytical processes. This framework combines in-play football event data at millisecond intervals with historical match data and pre-game statistics, allowing for the training of models that accurately reflect the intricate dynamics of market microstructures. The second objective centred on a thorough exploration of the football betting environment through an extensive literature review, leading to the creation of a pre-game betting model designed to identify value opportunities. This process included training multiclass classifiers to forecast final match outcomes, validating the models against bookmaker odds, and setting up benchmark models for comparative analysis. Lastly, the research progressed from pre-game analysis to in-play momentum betting, employing deep learning models that leverage exchange data to seize real-time betting opportunities.

The thesis is structured around three key experiments, each aligned with the overarching research objectives. The first experiment focuses on data exploration and infrastructure, employing a dual database approach that integrates in-play and exchange data using a cloud-native NoSQL time series architecture alongside a SQL database for pre-game data. The second experiment advances this foundation by accurately modeling football betting markets and developing baseline models for pre-game value betting. Building on these insights, the third experiment explores the use of deep learning models to capture in-play momentum betting opportunities, leveraging the integrated data framework to enhance predictive accuracy and strategy development.

6.2 Contributions

The research objectives defined in Chapter 1 were reached by successfully carrying out the experiments in Chapter 3 - Data Exploration and Infrastructure, Chapter 4 - Baseline Models for Pre-Game Football Predictions and Chapter 5 - Deep Learning for In-play Momentum Betting. These experiments produced significant findings and original contributions that hold valuable implications for both industry and academic research.

A Comprehensive Data Framework and Research Platform

The first experiment developed a strong system for collecting and archiving sports exchange data, pinpointing essential features and distinct patterns unique to sports trading. By combining in-play data with millisecond exchange data, the thesis generates features for trading models. The framework provides a new approach to collecting and storing pre-game, in-play, and exchange data, and with that a research platform facilitates easy data querying, feature creation for model training, and both back and bulk back-testing. This simulated environment, specifically designed for sports trading, is crucial for training and testing models and serves as a valuable tool for other researchers to further advance the field.

Understanding of the Betting Market Microstructure

As an underexplored market, the microstructure of sports betting has not received the same level of attention as traditional financial markets. This thesis provides a detailed examination of the nuances and anomalies within the betting market, focusing on aspects such as artificial order lag, the behaviour and movement of odds, and the factors influencing liquidity. By analysing these elements, the research enhances the understanding of how bets are matched and executed in real-time, and how these processes can be optimised. This comprehensive analysis not only sheds light on the unique characteristics of sports betting markets but also offers practical insights that can inform and improve efficient order execution, ultimately contributing to more effective trading strategies and better market dynamics.

Value Betting and Momentum Models

This thesis significantly advances the understanding and application of value betting and momentum models in sports trading. By developing pre-game value betting models, Experiment 2 demonstrates how machine learning can effectively identify bets where odds are mispriced. These models provide a solid foundation for refining sports betting strategies. Additionally, in Experiment 3 the exploration of in-play momentum betting models applies deep learning to capture real-time shifts during live events, enabling dynamic betting strategies that adapt to changing match conditions. This dual approach enhances predictive accuracy and opens new avenues for both academic research and practical application in the betting industry.

6.3 Future Work

While Experiment 2 successfully developed robust value betting strategies for football markets, there remain several avenues for further exploration and enhancement. The baseline models could benefit from the inclusion of additional features, such as a Pi-rating system, which would provide a more nuanced assessment of team strengths over time. Moreover, incorporating external statistics, such as point-in-time player statistics and sentiment analysis, could yield valuable insights. Adding contextual information, such as player transfers, injuries, and the stakes of a match, could also enhance predictive accuracy. Additionally, a deeper understanding of team formations and tactical approaches would allow for more sophisticated models, leading to the development of richer datasets that better capture the complexities of football betting.

In Experiment 3, there is significant potential for exploring alternative models and further enhancing predictive capabilities. One promising direction is the integration of pre-game state data from Experiment 2 with in-play data, creating a holistic model that leverages both types of information for more accurate in-play predictions. Additionally, the development of event-driven models could

focus on predicting the likelihood of specific in-game events, such as goals, penalties, or red cards, using event-driven architectures. This approach would allow for more targeted and dynamic betting strategies, responding in real-time to the evolving circumstances of a match.

The research platform developed in this thesis, which integrates pre-game, in-play, and exchange data, establishes a solid groundwork for ongoing experimentation and innovation in sports trading. A promising direction for future exploration is the investigation of in-play market-making strategies leveraging reinforcement learning techniques. These methods could significantly enhance the efficiency of liquidity provision and overall trading performance. Additionally, there is potential in developing arbitrage strategies that exploit discrepancies across different markets, leading to considerable profit opportunities. Future studies might also explore the broader application of machine learning models across various sports or consider cross-market strategies that integrate insights from multiple sports concurrently. The platform's flexibility and scalability open up a broad spectrum of possibilities for advancements in sports trading.

Bibliography

[1] R. Almgren and N. Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–40, 2001.

[2] A. Altmann. *A statistical approach to sports betting*. PhD thesis, City University London, 2004.

[3] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

[4] E. P. Chan. *Quantitative trading: how to build your own algorithmic trading business*. John Wiley & Sons, 2021.

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] A. C. Constantinou, N. E. Fenton, and M. Neil. pi-football: A bayesian network model for forecasting association football match outcomes. *Knowledge-Based Systems*, 36:322–339, 2012.

[7] K. Croxson and J. J. Reade. *Exchange vs. dealers: a high-frequency analysis of in-play betting prices*. Department of Economics, University of Birmingham Birmingham, 2011.

[8] P. Divos. *Modelling of the In-Play Football Betting Market*. PhD thesis, UCL (University College London), 2020.

[9] M. Dixon, D. Klabjan, and J. H. Bang. Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance*, 6(3-4):67–77, 2017.

[10] M. J. Dixon and S. 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.

[11] L. Egidi, F. Pauli, and N. Torelli. Combining historical data and bookmakers' odds in modelling football scores. *Statistical Modelling*, 18(5-6):436–459, 2018.

[12] E. D. Feustel and G. S. Howard. *Conquering Risk: Attacking Vegas and Wall Street*. Academic Publications, 2010.

[13] P. Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

[14] R. Gonçalves, V. M. Ribeiro, F. L. Pereira, and A. P. Rocha. Deep learning in exchange markets. *Information Economics and Policy*, 47:38–51, 2019.

[15] J. Goodacre and B. Schlagman. Algorithmic trading in financial and sports (exchanges). *Available at SSRN*, 2024.

[16] C. Henderson. escenic, 11 2009.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] L. M. Hvattum and H. Arntzen. Using elo ratings for match result prediction in association football. *International Journal of forecasting*, 26(3):460–470, 2010.

[19] R. Insley, L. Mok, and T. Swartz. Issues related to sports gambling. *Australian & New Zealand Journal of Statistics*, 46(2):219–232, 2004.

[20] W. R. Johnson, J. Alderson, D. Lloyd, and A. Mian. Predicting athlete ground reaction forces and moments from spatio-temporal driven cnn models. *IEEE Transactions on Biomedical Engineering*, 66(3):689–694, 2018.

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

[22] A. Koshiyama, N. Firoozye, and P. Treleaven. Algorithms in future capital markets. *Available at SSRN 3527511*, 2020.

[23] R. N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems*, 11:193–197, 1999.

[24] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[25] A.-H. Mihov, N. Firoozye, and P. Treleaven. Towards augmented financial intelligence. *Available at SSRN 4148057*, 2022.

[26] R. K. Narang. *Inside the Black Box: The Simple Truth about Quantitative Trading.-Description Based on Print Version Record*. J. Wiley et Sons, 2009.

[27] M. A. Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015.

[28] G. Nuti, M. Mirghaemi, P. Treleaven, and C. Yingsaeree. Algorithmic trading. *Computer*, 44:61 – 69, 12 2011.

[29] Ø. N. Øvregård. Trading" in-play" betting exchange markets with artificial neural networks. Master's thesis, Institutt for datateknikk og informasjonsvitenskap, 2008.

[30] M. Oytun, C. Tinazci, B. Sekeroglu, C. Acikada, and H. U. Yavuz. Performance prediction and evaluation in female handball players using machine learning models. *IEEE Access*, 8:116321–116335, 2020.

[31] S. J. Prince. *Understanding Deep Learning*. MIT press, 2023.

[32] M. A. Rahman. A deep learning framework for football match prediction. *SN Applied Sciences*, 2(2):165, 2020.

[33] D. E. Rumelhart and D. Zipser. Feature discovery by competitive learning. *Cognitive science*, 9(1):75–112, 1985.

[34] G. Shahtahmassebi and R. Moyeed. An application of the generalized poisson difference distribution to the bayesian modelling of football scores. *Statistica Neerlandica*, 70(3):260–273, 2016.

[35] P. Shen and R. M. Starr. Market-makers' supply and pricing of financial market liquidity. *Economics letters*, 76(1):53–58, 2002.

[36] S. Szymanski. Kuypers. t.(1999). winners and losers: The business strategy of football.

[37] P. Treleaven, M. Galas, and V. Lalchand. Algorithmic trading review. *Communications of the ACM*, 56:76–85, 11 2013.

[38] P. Tsirimpas. *Specification and performance optimisation of real-time trading strategies for betting exchange platforms*. PhD thesis, Imperial College London, UK, 2014.

[39] J. Wolfers and E. Zitzewitz. Interpreting prediction market prices as probabilities, 2006.

[40] Q. Zhang, X. Zhang, H. Hu, C. Li, Y. Lin, and R. Ma. Sports match prediction model for training and exercise using attention-based lstm network. *Digital Communications and Networks*, 8(4):508–515, 2022.

[41] Y. Zhao, R. Yang, G. Chevalier, R. C. Shah, and R. Romijnders. Applying deep bidirectional lstm and mixture density network for basketball trajectory prediction. *Optik*, 158:266–272, 2018.