Optimising Cricket Team Selection for One Day International Series Based on Match Conditions

A Thesis Submitted for the Degree of Master of Computer Science

L.G.U.P. Gunawardhana

University of Colombo School of Computing

2021

DECLARATION

I hereby declare that the thesis is my original work, and I have written it in its entirety. I have duly acknowledged all the sources of information that have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: Gunawardhana L.G.U.P.

Registration Number: 2018/MCS/027

Index Number: 18440271

11/30/2021

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr Gunawardhana L.G.U.P. under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by,

Supervisor Name: Dr. H.A. Caldera

H.A.Caldera 29-11-2021

Signature of the Supervisor & Date

I would like to dedicate this thesis to my parents, who supported me throughout the research period and facilitated me to focus on my research work

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Dr H.A. Caldera for their immense support, guidance and encouragement throughout the research. This thesis would not have been possible without their mentorship.

Also, I would like to thank my mentors and co-workers at LSF for their support. Especially Mrs Sherazad Hamit, Mr Nuwan Senaratna and Mr Tharindu Madushanka for their support and encouragement towards completing this thesis.

I extend my sincere gratitude to all the lecturers at the University of Colombo School of Computing for their valuable advice, comments and encouragement.

Finally, I would like to express my heartfelt thanks to my family for their support and inspiration through the many days and nights dedicated to completing this research. And also to my friends who shared their opinions to make this research a success.

ABSTRACT

This thesis focuses on predicting an optimal Sri Lankan cricket team for One Day International (ODI) matches by analysing player performance under different conditions, including weather conditions, opponents and venue. We try to maximise the overall team performance by predicting the best team combination from existing players. The selectors generally perform team selection considering recent performance, including batting and bowling averages of the players. These metrics provide limited insight into players' potential performance, which leads to drop-ups of qualified. Therefore, consideration of more factors and robust machine learning is required. Our study considers overall performance, consistency, venue, opposition, and recent form of players to predict the players' performance using Random Forest Regression. Then, use the predicted performance to evaluate the player rating of each player towards the team by using Neural Networks. Previous studies have proved that Neural Networks can solve team selection problems successfully [1]. Then we select the team based on the predicted winning contributions to maximise the overall team winning probability. The study concludes by predicting the last 45 matches the Sri Lankan cricket team played during 2017-2019 with the actual playing 11 and the optimal playing 11 selected using our proposed system. We observed that the winning rate of the Sri Lankan cricket team could be improved from 37.77% to 77.77% (105% improvement) if teams were selected using our proposed system.

Keywords: Random Forest Regression, Neural Network, Performance Prediction, Team Combination

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CHAPTER 1

1. INTRODUCTION

Cricket is recognised internationally as one of the most entertaining, competitive, and popular sports. It involves two teams consisting of fifteen players, each including four substitute players. These substitute players can replace on-field players if they have to walk out of the field due to an injury or any other reason. Each of these two teams consists of batsmen, bowlers and wicket keepers and both teams get the chance to bat and bowl against the opposition. A maximum of 11 players can bat from each team, and the goal of the batsmen is to score the highest number of marks against the opposition bowlers in the number of overs given or until all batsmen get out. The team which scores the highest marks is the winner of the match. Many factors affect the performance of the teams and individual performance. The weather, the pitch, opposition team, day/night, batting first/second are some of those factors. Apart from player statistics, the selection committee must consider these factors since they affect individual players' performance. Hence, contributing to the overall performance of the team towards winning the match.

There are three main cricket formats at the international level: Test matches, One-Day Internationals and Twenty20 Internationals. International Cricket Council acts as the governing body and upholds the game's rules and regulations while providing match officials. These different game formats require different playing styles and skills by the players, and hence the selection committee will have to select the player pool accordingly.

1.1. Research Aims and Objectives

We plan to investigate how player performance get affected by various weather conditions and how player combinations can improve the team's overall performance. Furthermore, use those results to train a machine learning model to predict player performance and the best team of players, given the match conditions and opponents.

1.1.1. Aim

To investigate the impact of weather conditions on cricket player performance and construct an optimal team prediction model using machine learning techniques.

1.1.2. Objectives

We break down our study into the following objectives in order to systematically approach our research aim.

- Critical Review of Literature on player performance analysis, prediction and team selection
- Collect match record details and player performance details
- Collect weather data related to each match venue
- Synthesise the data from both sets to establish if correlation points exist between weather conditions and player performance
- Develop a Machine Learning model to predict the player performance
- Develop a Machine Learning model to select the best combination of players
- Measure the accuracy of the prediction model with actual match data

1.2.Motivation

There was a time where every cricket playing nation in the world feared to play against the Sri Lankan Cricket Team. The team consisted of world-renowned players who could play under pressure, play aggressively and understand and read the game well. Most importantly Sri Lankan Cricket team was known as a team with courage, confidence, and a team who would fight until the last ball to steal the opposition's victory at the slightest chance they got. However, unfortunately, those glory days have come to an end for Sri Lanka Cricket. At present, Sri Lanka cricket is suffering from poor performance consistently. While cricket experts claim that players' political, religious, and personal disputes are the culprit for this continuous poor performance, there is much controversy in social media regarding the players' management and selection by the Sri Lankan Cricket Board.

Having been playing cricket at school, academy, and university level for over 13 years, The author has become familiar with the domain of cricket and has a sound understanding of physical, mental, and environmental factors affecting players' performance. Therefore, the author thought of combining his experience in cricket with his knowledge in Computer Science. To make a reliable and accurate systematic approach for analysing player performance. Furthermore, to explore the possibility of using advanced data mining and machine learning techniques into predicting player performance and the best pool of players for playing international cricket under a set of given conditions.

1.3.Problem Definition

Throughout cricket, players and commentators believe that weather conditions play a significant role in player performance. Hence, towards the results of the matches. Even though scientists have tried to explain the impact of weather conditions such as temperature and humidity on cricket ball dynamics, they have not been able to make much success in scientifically explaining the effects of weather conditions on player performance. [2] [3]. Nevertheless, there are some scenarios where players complain or struggle to perform well in certain weather conditions. Let us break down those scenarios based on Batting, Fielding and Bowling performance.

Before starting a match, both captains of two teams toss a coin under the match referee's supervision. The captains consider the toss as an essential factor in the decision of the match. The captain who wins the toss gets to decide whether to bat first or bowl first. The captain then weighs his team's strengths and weaknesses, pitch condition and outfield condition, opponents bowling and batting strengths and weaknesses to decide whether to bat first or bowl first. The captain also pays attention to the weather forecast of the day in making his decision. The captain must consider how the humidity and wind conditions would change throughout the match to gain an advantage. For example, if there is much dew expected (due to humidity) in the ground, the captain would avoid bowling or fielding under dew conditions. Because it would make the ball more slippery, and bowlers would struggle to grip the ball properly. The fielders would also struggle to catch the ball. Especially in playing day and night matches, dew is likely to affect the team bowling in the night. Also, inexperienced players would struggle to field under artificial light conditions. Lack of experience and practice of playing under night light conditions affect domestic players making their way into international cricket since most domestic cricket matches are day matches. Even under daylight conditions, some players might struggle in taking high catches under cloudy conditions since they have difficulty seeing the ball in the cloudy white background, with the white ball or even due to the change of the ball's trajectory due to wind. The cricketers also believe that humidity, wind speed and direction also affect the dynamics of the ball. If fast bowlers are bowling with the wind, they will gain more speed with the wind's support. If the bowlers are bowling against the wind or perpendicular to the wind, there is a chance it will help the bowler swing the ball in the air and confuse and keep the batsmen guessing. Batmen should be aware of these conditions to perform better. As weather conditions change, the behaviour of the pitch will also change.

Therefore, the ball will bounce higher after hitting the pitch, bounce lower than expected, or even the speed loss after hitting the pitch will differ. So, without understanding these scenarios, a batsman will not time his shots and quickly get out. Therefore, it becomes evident that we cannot disregard the impact of different weather conditions in a cricket match.

Before a cricket tournament, the squad's selection is tedious that the Cricket Selection Committee should perform. We should consider some vital factors in team selection: current form, consistency, past performance statistics, team balance, fitness conditions, weather conditions, team contribution, and opportunity for younger players to gain experience. This process is mainly performed manually by the selection committee. Most of the time, most of the factors mentioned above get overlooked during the selection process, leading to poor team selection and losing matches. Furthermore, it is crucial to note that a player's overall average performance is not an accurate metric that the selection committee can use to predict his performance in the upcoming tournaments. Therefore, a more detail-oriented, systematic and precise performance analysis based on the players' statistics combined with other factors mentioned above seems an inevitable requirement for optimum team selection in the modern competitive game of cricket.

Most research related to player performance analysis uses mathematical modelling or machine learning to analyse their overall performance. And then, using the performance results, they rank the players they predict the outcome of upcoming matches—most researches conducted towards player selection focus on selecting the highest performing players for the team. In contrast, they should select the players to maximise the team performance and overall winning probability. Also, none of the researches has considered all the factors affecting performance, such as match conditions, Form, Consistency, and opponents, to predict a more suitable team for given match conditions. They predict the best team based on the overall performance of the players disregarding the specific match conditions.

1.4.Scope

The performance analysis method proposed in this research can be adapted to any international or domestic cricket team in any format with some modifications to suit each game's format. It is a mammoth task to collect data from all cricket players from all international cricket playing countries and analyse it. Therefore, in the scope of this research, we will be only using the dataset of all One Day International matches played by the Sri Lankan cricket team between the period 2010-2019. The dataset is publicly available at https://stats.espncricinfo.com/. We will aggregate this dataset with weather data from https://www.worldweatheronline.com/ and study the impact of weather data, and other player performance attributes on predicting players' performance. The research will focus on analysing the player performance based on the following attributes. To collect data on the number of catches dropped/ run out opportunities missed, we will be going through the ball-by-ball commentary of each inning. *Table 1* shows the list of factors that we are considering in analysing the performance of the players.

Table 1: List of Factors, Attributes to Consider for Performance Analysis

Batting	Bowling	Fielding / Wicket Keeping
1. Runs scored 2. Bowls faced 3. Minutes on the ground 4. Number of Fours 5. Number of Sixes 6. Strike Rate 7. Opponent 8. Batting Position 9. Day/Night Condition 10. Temperature 11. Wind Speed 12. Rain 13. Humidity 14. Cloud percentage 15. Atmospheric Pressure 16. Form 17. Consistency 18. Inning (First/Second) 19. Toss 20. Match outcome 21. Ground	1. Overs bowled 2. No. of maiden overs 3. Runs conceded 4. Wickets taken 5. Economy 6. Dot balls 7. No. of Fours conceded 8. No. of Sixes conceded 9. Number of no balls 10. Number of wide 11. Opponent 12. Day/Night Condition 13. Temperature 14. Wind Speed 15. Rain 16. Humidity 17. Cloud percentage 18. Atmospheric Pressure 19. Form 20. Consistency 21. Inning (First/Second) 22. Toss 23. Match outcome 24. Ground	1. No. of catches taken 2. No. of stumps/ run-outs taken 3. No. of catches dropped 4. No. of run-outs missed 5. Outfield condition 6. Opponent 7. Day/Night Condition 8. Temperature 9. Wind Speed 10. Rain 11. Humidity 12. Cloud percentage 13. Atmospheric Pressure 14. Form 15. Consistency 16. Inning (First/Second) 17. Toss 18. Match outcome 19. Ground

1.5. Solution

Using mathematical modelling and machine learning approaches help us in identifying players with relative high-performance potential. However, it would be much more helpful for the selection committee to understand each player's weaknesses and strengths when playing under different conditions. For example, a batsman might have performed well under daylight but might not have performed well in night light conditions. Understanding each players' strengths and weaknesses would help the selection committee to understand players' performance better.

On the other hand, coaches would benefit from the same information, allowing them to pay special attention to each player's weaknesses and change their training programmes to improve their skills. Therefore, in this research, we would approach the problem by empirically identifying and selecting the most suitable approach to analyse the players' statistics and understand how the player performance varies based on different conditions.

The dataset of matches from https://www.espncricinfo.com/ includes scores, strike rates, number of fours, number of sixes, strike rate of batsmen, and overs bowled, runs conceded, wickets taken by bowler, economy, dot balls, number of fours, sixes, wides, and no balls conceded by bowlers against the opponent team. We will process the dataset to acquire player performance against opponent teams. Also, we will gather weather data for each match venue via https://www.worldweatheronline.com/. The website provides an interface to select the date and the stadium to get a detailed weather report of the match venue. By aggregating weather data with previous player performance data, we expect to determine how the weather impacts the players' performance and help selectors filter out the most suitable pool of players given the venue and expected weather conditions for a cricket tournament.

Furthermore, we will systematically combine the individual player performance towards selecting an optimal team to improve the team's winning potential. Rabah Al-Shboul et al. [1] have proposed a neural network-based model for team prediction, which takes players' performance from both teams playing as inputs and combines the player performance to predict the team's optimum performance in basketball. In our study, we will adopt this neural network-based team prediction model to combine and predict the performance of cricket teams.

1.6.Structure of the Thesis

The remaining of the thesis is structured as follows. Chapter 2 will give a critical Literature review of the related works on player performance analysis and team prediction, and the impact of weather conditions. Chapter 3 will describe the technologies we adopt in our study to achieve the research objectives and implement an experimental environment. Chapter 4 will extensively describe the research methodology we have designed to analyse and predict player performance and combine players for optimal team performance. Also, it will explain how we implemented the design, collected the data, processed and trained the machine learning algorithms, fine-tuned the models, and predicted the final team combinations. Chapter 5 will evaluate the results and outcomes gained from the research, and Chapter 6 will discuss the limitations of the research, future improvements that we can make to improve the system.

1.7. Summary

We started this chapter by giving a brief introduction to the game of cricket and how different factors affect the performance of individual players and hence towards the overall team performance of the team. Also, we have defined the aims and objectives of the research, how we are determined to achieve objective by objective towards the final result. Then we explained how our experience of the game of crickets helps us understand the dynamics and how it motivated us to resolve player performance prediction and optimum team selection. After that, we defined the scope and explained the structure of the study we have conducted.

CHAPTER 2

2. BACKGROUND AND RELATED WORKS

2.1. Introduction

This section will discuss the research published by different authors to solve player performance prediction in sports and team selection. First, we will look at different approaches adopted by different studies on predicting and analysing player performance. Then we will discuss the research conducted on understanding the impact of weather conditions on sports. After that, we will look into studies that have focused on combining players for optimum team predictions by combining the individual performance of players. We will review the current studies and their solutions and discuss the challenges that have remained unsolved in this field of study. Towards the latter of this chapter, we will summarise our literature review and define the problem we are trying to solve in our study.

2.2. Performance Analysis Based on Mathematical Approaches

The number of articles published online related to player performance prediction and team selection seems relatively low. Lemmer has been a consistent research contributor in the research area of player performance analysis. In his article, he proposes a systematic approach toward the performance analysis of players. In his study, he has suggested that depending only on traditional statistics such as strike rate, the average of batsmen or economy rate, the number of wickets taken by a bowler is not adequate. Therefore, he has proposed formula with a few additional factors such as the batting position to analyse player performance. Lemmer has not considered the form, consistency of the players or weather conditions under which the players performed.

D. Bhattacharjee and H. Saikia [4] have proposed a composite performance index irrespective of whether the considered player is a bowler or batsman. They then use a binary programming method to select a balanced team consisting of 15 players for IPL (Indian Premier League). Once again, the author does not discuss or consider the impact of different weather and conditions on player performance.

Wickramasinghe [5] proposed a hierarchical linear model for predicting batsmen's performance and the possibility of using a neural network for predicting the number of wickets a bowler will take. The number of wickets taken by a bowler is a good metric for measuring the bowlers' performance. However, it cannot be considered the only attribute that we should use to evaluate a player's performance. In ODI cricket, conceding fewer runs to the opponent team is more important than taking wickets. Taking wickets is one way of slowing down the opponents from scoring many runs, but sometimes taking wickets means allowing the batsmen to take risks to go for scoring shots to result in batsmen scoring more runs. Therefore, the runs conceded, wickets taken by a bowler, combined with other attributes, will provide a good performance metric for bowlers.

2.3. Performance Analysis Based on Machine Learning.

Jhanwar and Pudi [6] propose a method to predict a match's outcome by analysing the two teams' past performance. They first calculate each players' performance and then use an algorithm developed by themselves to model batsmen and bowlers' performance, giving weight to the players' more recent performance. They then calculate the overall performance index for the team by summing each player's performance indexes. Then they use a supervised learning approach to predict match outcomes. They have implemented supervised classification models including SVM, Random Forests, Logistic Regression, Decision Trees and KNN classifier.

They achieved the highest accuracy by using a KNN classifier with k=4. The author gives more weight to the recent performance of the players to make more accurate predictions. He also considers the batting position of the batsmen and their form in analysing the performance of players. However, once again, playing weather conditions have been omitted from this study as well.

Passi and Pandey [7] propose a method based on supervised learning and machine learning techniques to predict the players' performance. First, they rate the performance of players concerning batting and bowling performance. They define five performance levels for each consistency, form, opposition, venue, batting average, batting strike rate, bowling average, bowling strike rate and five-wicket hauls. They use a formula to calculate the form and consistency of players. This formula and the weights they use to calculate form and consistency are derived using the analytic hierarchy process developed by Thomas L. Saaty. Form and consistency are two attributes which the researchers mostly ignore. They have also considered additional parameters such as batting position, match type, match time, venue(home/away/neutral), tournament, toss, pressure (importance of the match). Then they rated them into five levels as similar to how they rated batting and bowling performance. However, the suggested approach does not consider the ground's weather condition when predicting the team. Then they use four machine learning algorithms: Naïve Bayes, Decision

Trees, Random Forest and Support Vector Machine to predict the player performance. They achieved the highest accuracy by using the Random Forest.

More comprehensive and recent research done by Kapadiya and Adhvaryu [8] is the first research that includes weather data. They claim that weather conditions are an essential factor to consider when analysing player performance and team selection. They combine player performance data with weather data to provide more accurate performance prediction using machine learning techniques. They have considered the importance of the match as an attribute in weighting the players' performance. These attributes help in understanding the players' performance under different pressure conditions. They have used supervised machine learning algorithms: Naïve Bayes, Decision Trees, Random Forest and Support Vector Machine, Weighted Random Forest to predict player performance. Their study has achieved a higher accuracy rate by combining weather data for their predictions than other studies conducted without considering weather data. However, they have not proposed a method to combine the players to the team for optimal performance.

Sinha [9] has developed machine learning models for predicting the cricket matches' outcome, taking ground advantage, past performance of the players into consideration. They have considered the additional factors in evaluating the player performance: Toss, Home Ground, Captains, Favourite Players, Opposition, the Fifties, the Hundreds, Fours, Sixes. They have implemented SGD Regressor, KNN-Regressor, Linear Regression using Least-Square Estimates, Weighted KNN-Regressor and compared the accuracy of results. They have observed that all the machine learning models implemented provided identical results—further, the author analyses how the team's performance varies under home and away conditions.

As discussed above, most of these researches propose performance analysis methods based on mathematical modelling and machine learning approaches. None of the studies can predict player squad considering all attributes such as form, consistency, weather condition, venue, and other factors. Most studies separately consider different sets of features. Still, none of them has combined all weather conditions, form, consistency factors and predicted the most suitable squad for a given set of match conditions and opponent. Our research aims to aggregate attributes such as form, consistency, weather conditions, venue, day/night conditions, toss, the importance of the match, which impacts analysing players' performance more accurately.

2.4. Impact of Weather in Sports

Some studies have aimed to analyse how weather conditions affect the performance of players' in sports. [10] has studied how heat stress contributes to decreasing match performance in football players. They evaluated activity patterns and thermal responses of players during football matches. The considered teams have played the matches under different environmental conditions. They have measured the physical performance of players using a telemetric sensor and a global positioning system. They have also considered the ambient temperature and relative humidity in evaluating the performance of the players. Their study concluded that the players' physical performance might reduce when the heat stress increases.

Another aspect is how the weather conditions impact the objects and their dynamics in sports. Hence if they impact the players' performance, one such research conducted to analyse how temperature changes affect golf ball dynamics is [11]. They have conducted impact testing on golf balls and measured inbound velocity, outbound velocity, impact duration and maximum deformation using high-speed cameras. Their results have concluded that the dynamics of the golf ball change with temperature. Even though the golf ball has a different physical composition from the cricket ball, we can safely assume a similar impact on the cricket ball under different temperatures.

When it comes to batting in cricket, one factor that puts batsmen in trouble is the cricket ball's Swing. More often, batsmen are deceived by Swing more than they get deceived by the speed of the delivery. In their research, James A Scobie, Simon G Pickering, and others, Fluid dynamics of cricket ball swing [2], state that Swing is the lateral deviation of the ball while travelling towards the batsmen after being released from the bowler's hand. Therefore, a batsman should get accustomed to the ball's Swing under different conditions before successfully scoring runs without getting out. Adapting to these various conditions allows players to anticipate the deliveries' swing movement and play accordingly. Adapting is a skill required by professional cricket players to succeed and perform well in international cricket. David James and Others [3] studied atmospheric conditions' impact on the cricket ball swing. Based on their study, they have concluded that humidity and any other atmospheric conditions do not affect the cricket ball swing. However, their study primarily focused on analysing physical changes to the ball due to various changes in atmospheric conditions. They hypothesise that any change in the Swing has to be due to physical changes caused to the ball by atmospheric conditions. The physical changes to the cricket ball might be insignificant due to atmospheric conditions. However, that does not mean we can rule out that the wind's speed and direction can impact changing the ball's direction in mid-air. Moreover, humidity can change the moisture level of the cricket ball and thereby soften the leather cover of the cricket ball. This added softness to the leather can change the speed of the ball off the pitch once the ball is bounced and can deviate the ball's trajectory.

2.5. Team Selection and Overall Team Performance Prediction

As Lemmer has [12] mentioned in his paper in team sports, the most critical factor affecting winning is teamwork. As he further shows in his study, even though India won the Twenty20 World Cup in 2007, they had only one batsman in the top 10 rankings. Therefore, he highlights the importance of team effort over individual performance as the key to winning matches. In their research, H. Saikia, D. Bhattacharjee [13] has suggested a performance index for players to evaluate each player's performance considering their score in batting, bowling, and fielding aspects. And then, they rank the players based on the performance score calculated for each player. Then they select the final team from the ranked player list. While this approach helps rank players and identify individual players with high performance, it does not reflect the combination of players that would yield the optimum performance. For example, a batsman in good form and scoring many runs in a match might not win the match for his team if the bowlers did not bowl well and restrict the opposition team to a lower total. Similarly, bowlers will not concede fewer runs to the opposition if the fielder performs poorly and lets the opposition score runs without pressure. If the fielders are dropping catches and missing run-out chances in the field during the match, it will negatively affect the bowlers' mentality during the match, and bowlers will not be able to get wickets and limit the opposition to a lower total score.

R. Al-Shboul, T. Syed, J. Memon, and F. Khan [1] has proposed a Competitive Neural Network-based Team Selection Approach where they consider the combined performance of the players to select the best team in football. Instead of rating the players individually, they rank the players with a relative performance index. The relative player performance index is based on each players contribution towards winning a match. They also suggest that a naïve approach of using the winning ratio to evaluate the players is not good enough. Because the winning ratio will be similar for every player in the team, disregarding their contribution towards the win. So, they suggest a semi-supervised neural network model that analyses all the players' input features to predict win/loss. At the same time, they claim that we can adopt this approach in other team games too. However, in the game of cricket, it can get more complicated. In football, all the players perform on the ground at the same time towards the same objective. Even though they might have different roles and responsibilities in the team, their ultimate objective is to score goals and avoid the opposing team from scoring goals. However, when it comes to cricket, the team has two main phases. In one phase, the players have to score runs by batting. In the other phase, the players have to bowl and field against the opposition batting. Therefore, the team should consist of batsmen and bowlers, and everyone should be good at fielding. The bowlers' primary objective is to concede fewer runs to the opposition while bowling, take wickets, and restrict the opposition to a lower total. On the other hand, Batsmen have to score many runs to give a defendable score to the bowlers. Therefore, in cricket, the team consist of more diverse roles than in football. Therefore, it is a challenge to analyse the different skills of players and compare players in batting, bowling and fielding aspects and balance the team's overall skill. They have achieved an overall accuracy of around 54% for the player rating neural network. Then, an accuracy of 60% for the team prediction.

2.6.Summary of Literature

Table 2 summarises the literature review based on the approach and parameters considered for performance analysis.

Table 2: Literature Review Summary by Performance Analysis Approach

No	Study	Factors Considered	Performance Analysis Approach
1	(Lemmer, 2008) [12]	*Conventional factors*[1], Batting Position	Mathematical Equation to Calculate a Performance Index
2	(Bhattacharjee and Saikia, 2014) [4]	*Conventional factors*	The composite performance index, Binary Programming
3	(Wickramasinghe, 2014) [5]	*Conventional factors,*	Hierarchical linear model
4	(Jhanwar and Pudi, 2016) [6]	*Conventional factors*	Supervised machine learning algorithms, SVM, Random Forests, Logistic Regression, Decision Trees and KNN
5	(Passi and Pandey, 2018) [7]	*Conventional factors,* Highest score, Centuries, The fifties, Five Wicket Hauls,	Using machine learning algorithms Naïve Bayes, Decision Trees, Random Forest and Support Vector Machine

[1] Conventional factors: Runs Scored, Strike Rate, Average, Economy, Wickets taken

		batting position, match-type, match time, venue(home/away/neutral), the tournament, toss, pressure (importance of the match)	
6	(Kapadiya and Adhvaryu, 2020) [8]	*Conventional factors, the* importance of the match, Humidity, Wind flow, Rain, cold, Day/night condition	Supervised machine learning algorithms to predict player performance. Naïve Bayes, Decision Trees, Random Forest and Support Vector Machine, Weighted Random Forest.
7	(Sinha, 2020) [9]	*Conventional factors,* Toss, Home Ground, Captains, Favourite Players, Opposition, the Fifties, Fours, Sixes	Supervised machine learning: SGD Regressor, KNN-Regressor, Linear Regression using Least-Square Estimates, Weighted KNN-Regressor
8	R. Al-Shboul, T. Syed, J. Memon, and F. Khan 2017 [1]	Player Relative Performance Rating	Neural Network

As discussed above, most previous studies have not considered weather condition attributes in predicting player performance. Also, these studies focus on selecting players based on individual performance rather than combining players to optimize team performance.

2.7.Summary

This chapter discussed the different approaches used in previous studies towards analysing and evaluating player performance and team predictions. While most of the research has focused on ranking players based on individual performance, some studies have approached the combination of player performance for optimum combinations in other sports (Rabah Al-Shboul et al. [1]). Our study will combine the approaches used in the reviewed work and propose a complete system that can predict the optimum team to improve the match-winning rate for cricket based on combining individual performance and overall team performance of players under given match conditions.

CHAPTER 3

3. TECHNOLOGY

3.1. Introduction

This chapter will discuss the technologies we have adopted to analyse and predict player performance and select the optimum team. From the initial phase of gathering data to the final team selection outcome, we have used different technologies to process the data and implement the experimental setups to analyse the data. We will provide an in-depth understanding of the technologies and techniques adopted to achieve this study's final goal.

3.2. Web Scraping

The World Wide Web (WWW) consists of a vast amount of data stored in different formats. Most of this data is stored in human-readable formats like web pages or publicly available in human-readable formats. Direct access to a database or a spreadsheet can make it easy to gather, process and analyse information. However, when data is not available as a database or spreadsheet, we have to convert the information from web pages into a more processable format. When we use web scraping, we extract data from web pages and organise them for semantic processing. Web we scrape data using the Hyper Text Transfer Protocol (HTTP) or using a web browser. The scrapping process can be done manually by a human or automated by writing a bot or a web crawler programme. Due to the ability of web scraping to collect data efficiently from enormous web sources, it is considered a powerful technique. [13]

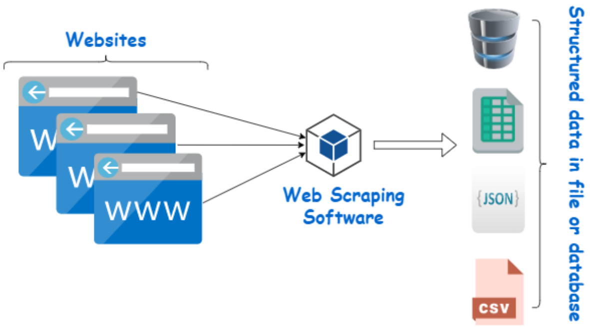


Figure 1: Web Scraping[2]

[2] Image Courtesy of https://www.webharvy.com/articles/what-is-web-scraping.html

3.3. Machine Learning

Machine Learning is the concept of adapting human learning techniques into computers. Hence this field of technology is derived from neuroscience, biology, statistics and mathematics. Machine Learning is also considered a subdomain of Artificial Intelligence. We can allow the computer to feed on data and machine learning techniques to provide generic models and predictions based on statistical calculations. Researchers have developed various machine learning algorithms over the years. These algorithms offer different approaches towards building a generic model based on data input. Therefore, it is essential to determine the best machine learning algorithms based on the nature of the problem we are trying to solve.

We can divide Machine Learning Algorithms into several categories based on learning method and prediction method; Machine Learning algorithms can learn Supervised or Unsupervised. Each of these different types of problem-solving require different kinds of algorithms. Moreover, some algorithms work for both classification and regression problems. We will discuss a few machine learning algorithms that we consider to solve our player performance prediction problem.

3.3.1. Supervised Learning

Supervised machine learning is where the output of the dataset is provided to the machine learning algorithm to learn. So, the algorithm will look into both input and output data from a dataset and then define a model to predict the output of datasets where the output is unknown.

We can separate supervised learning into two types based on the prediction method: classification and regression. [14]

3.3.1.1. Classification

Classification problems use a supervised machine learning algorithm to assign a category based on a given set of input parameters. Classification solutions are used where it is necessary to categorize or label a set of input attributes. For example, a classification problem can be identifying an animal from an image and labelling whether the animal is a cat or a dog.

3.3.1.2. Regression

We use Regression problems to understand the relationships between dependent and independent variables. Regression models can output a numerical value. In contrast to classification, regression is used where the output value should be continuous and numerical.

3.3.1.3. Naïve Bayes

Naïve Bayes classifiers are supervised machine learning algorithms with statistical classifiers. They can predict the probability where the output falls into a predefined class based on probability. One assumption made when using Naïve Bayes is that each input attribute contributes to the model's final output class. This assumption is also referred to as class conditional independence. Therefore, we need to be very careful in determining the input attributes when training a Naïve Bayes algorithm.

Bayes Theorem: Let X be a data tuple, and C be a class label. Let X belong to class C, then

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{3-1}$$

where;

• P(C|X) is the posterior probability of class C given predictor X.

• P(C) is the prior probability of class.

• P(X|C) is the posterior probability of X given class C.

• P(X) is the prior probability of the predictor.

The classifier calculates the probability P(C|X) for every class Ci for a given tuple X. It will then predict the class to which X belongs based on the highest posterior probability.

X belongs to class Ci if and only if $P(Ci|X) > P(Cj|X)$ for $1 \leq j \leq m, j \neq i$.

3.3.1.4. Decision Trees

A decision tree creates hierarchical decisions (a tree) for a class-labelled training dataset. Each node of a decision tree represents a decision that has to be made by the algorithm to move to one out of two decisions in the path to predict the final output value or class. Over the years, researchers have suggested improvements to improve the accuracy and shortcomings of using this algorithm. ID3 and C4.5 are two such decision tree algorithms, whereas C4.5 is a more improved algorithm with the ability to handle numeric and nominal data values. Also, it can deal with missing attribute values. In a decision tree algorithm, each new data record starts from the tree's root node and then moves downwards through each node according to the decision threshold of the node. These records are then partitioned recursively based on selected attributes. A heuristic procedure is used to determine the splitting criterion for the selected attributes. The algorithm ends if all the data records end up being classified into the same class, all data records are consumed or until no more attributes remain for further partitioning.

3.3.1.5. Random Forest

Random forest is a machine learning algorithm that we can use for both classification and regression problems. A random forest is a set of decision trees combined. Each tree depends on a random vector sampled independently, which is the same for all the trees in the random forest. Each tree is built by randomly selecting attributes at each node of the tree to determine the splits. The basic procedure of building a decision tree is to start with the dataset and iterate with subsets of the initial dataset. When constructing a classifier, several attributes are selected from the list of all attributes randomly. Then the trees are grown by adding more attributes to them.

3.3.1.6. Support Vector Machine (SVM)

In their research paper, Vladimir Vapnik, Bernhard Boser, and Isabell Guyon initially introduced the concept of Support Vector Machines [15]. SVM is more accurate and overcomes the problem of overfitting. Same as some of the random forest algorithms discussed before, SVMs also be used to solve regression and classification problems. First, the SVM transforms the original dataset into a higher dimensional dataset using nonlinear mapping. In the next step, the algorithm searches for the optimal linear hyperplane, separating the dataset into different classes accurately. With a suitable mapping in the higher dimension, the dataset is guaranteed to be separated by a hyperplane. The algorithm uses support vectors and margins defined by the support vectors in defining the hyperplane. The support vectors generated by the algorithm provide a comprehensive description of the trained machine learning model.

We can write the equation of a hyperplane as:

$$W. X + b = 0 \tag{3-2}$$

W is a weight vector. W = {w1, w2, w3, w4, …., wn} where n is the total number of attributes and b is a scalar. b is also referred to as bias. The hyperplane denoted by the above equation draws the separation between two classes.

Initially, SVM was used for binary classifications. Later on, SVM algorithms were improved to support multiclass classifications.

3.3.1.7. Neural Networks

A Neural Network (NN) or an Artificial Neural Network (ANN) is a classifier algorithm designed to mimic the behaviour of the human brain. A human brain consists of an enormous amount of nerve cells and neurons. Each of these cells is interconnected, forming a very complex web-like structure. These cells transmit signals between each other. Each cell receives a transmission from all the cells connected to it. However, it only sends output only if the input signals reach a certain threshold level. If it reaches the threshold, it transmits the output signal to all the connected cells.

In an ANN, this behaviour is implemented using perceptron. A perceptron takes several weighted inputs and combines and summarise them into one. If the combined input exceeds the threshold value, it will send an output signal. The activation function determines the value of the output. The activation function is often chosen to be between 0 and 1 or -1 and 1. Usually, the derivative of the activation function is used during the training phase of the ANN. Therefore, the derivative is often expressed in terms of original function values. Hence we can write the equation for a perceptron as below.

$$y = \emptyset\left(\sum_{i=1}^{n} \omega_i x_i + b\right) \tag{3-3}$$

"y" is the output signal and $\emptyset$ is the activation function. "n" is the number of other perceptrons connected to the perceptron. ω_i is the weight corresponding to the ith connection and "x_i" is the input value from the i^{th} connection. Finally, b represents a constant threshold value. The network can modify the weight associated with b.

These perceptrons are then organised into layers. Each layer is connected to the previous layer and get the inputs of earlier layers. Any classifier model should learn from training data and learn and adjust the model to predict the correct classification for new data. In an ANN, as the model learns, it modifies the weights associated with each connection between perceptrons between the layers. The most common way to train an ANN is to set initial weights for the network and then feed training data to the system. Then the output error is calculated and fed back into the system in reverse, and the weights are modified to reduce the error. This process is known as back propagation. By repetitive back propagation, we can optimise the weights to minimise the error.

To make the learning process and back propagation more efficient, sometimes the momentum technique is used. The momentum technique helps to determine the optimum stepping of weights. If the steps are tiny, it will take more time to converge. On the other hand, if the step size is too significant, it will never converge and keep on oscillating. In the momentum technique, the step size is calculated and changed dynamically.

A significant problem with ANN is that it can overfit the training data if not well trained. An overfitted model will predict more accurately on the training data but fails to classify new data accurately. We can avoid this issue by using the cross-validation technique. In cross-validation, the training set and test sets are varied, and the evaluation is done on each set to determine the best set of hyper parameters. If the test set/ validation set error is higher than the error in the test dataset, the model is overfitting. If the training dataset and test dataset error are approximately equal, the model is trained more accurately to predict a general input.

An excellent example of regression would be predicting the sales revenue based on previous months sales revenue data.

3.3.2. Unsupervised Learning

Unsupervised learning contrasts supervised learning where no output is provided to the machine learning algorithm to learn. We can use unsupervised algorithms to analyse and cluster unlabelled data without manually labelling or categorising it. We can use these algorithms to identify hidden patterns in data without any human intervention. [14]

3.3.2.1. Clustering

Clustering is an unsupervised data mining technique for categorising data based on similarities and dissimilarities. Clustering is the task of dividing the dataset into several clusters, such that data points in the same cluster are more similar to other data points in the same cluster and different to the data points in other groups. An excellent example of a clustering algorithm is the K-means clustering algorithm. It groups data into clusters based on similarities. The K is the number of clusters into which the algorithm will categorise the data.

3.3.2.2. K-Means Clustering

K-means is a popular and one of the simplest algorithms that we can use for clustering unsupervised data. K defines the number of clusters in data. Moreover, the algorithm will put the data into different clusters based on the vector distances. Initially, the algorithm starts with randomly selected centroids and iterate through each data point, aiming to optimise the positions of the centroids with every data point added to the cluster with the closest centroid to the data point.

3.4.Programming Languages and Tools

We chose the Python language and associated libraries to implement the experimental setup, results, comparisons and to build machine learning models to predict player performance and optimum team combinations. Python provides powerful tools to train, evaluate and implement machine learning models with ease. We will give a brief introduction to the technologies we have adapted to implement our proposed system.

3.4.1. Python

Python is a general-purpose, open-source programming language. The language is optimised for productivity, portability and integration. This programming language has a large community of developers worldwide with rich and powerful libraries to support machine learning, internet scripting, user interfaces and much more. Python also facilitates Object-Oriented Programming (OOP). It also has a straightforward, readable and maintainable syntax. Even though Python is a general-purpose language, it is often used as a scripting language because of its ability to utilise and direct software components written in other languages with ease. Python provides a rapid software development experience.

3.4.1.1. Scikit-Learn

Scikit-Learn [16] is an open-source machine learning library written in Python. It facilitates easy and fast integration of machine learning algorithms in Python. The library consists of various machine learning algorithms for classification, regression, and model evaluation functionality. It also provides functionality for data pre-processing. Some algorithms in the Scikit-Learn library are implemented in C language to improve efficiency. The use of the C language is possible with the static compiler available for Python, which can compile C code for Python. [17]

3.4.2. PyCharm

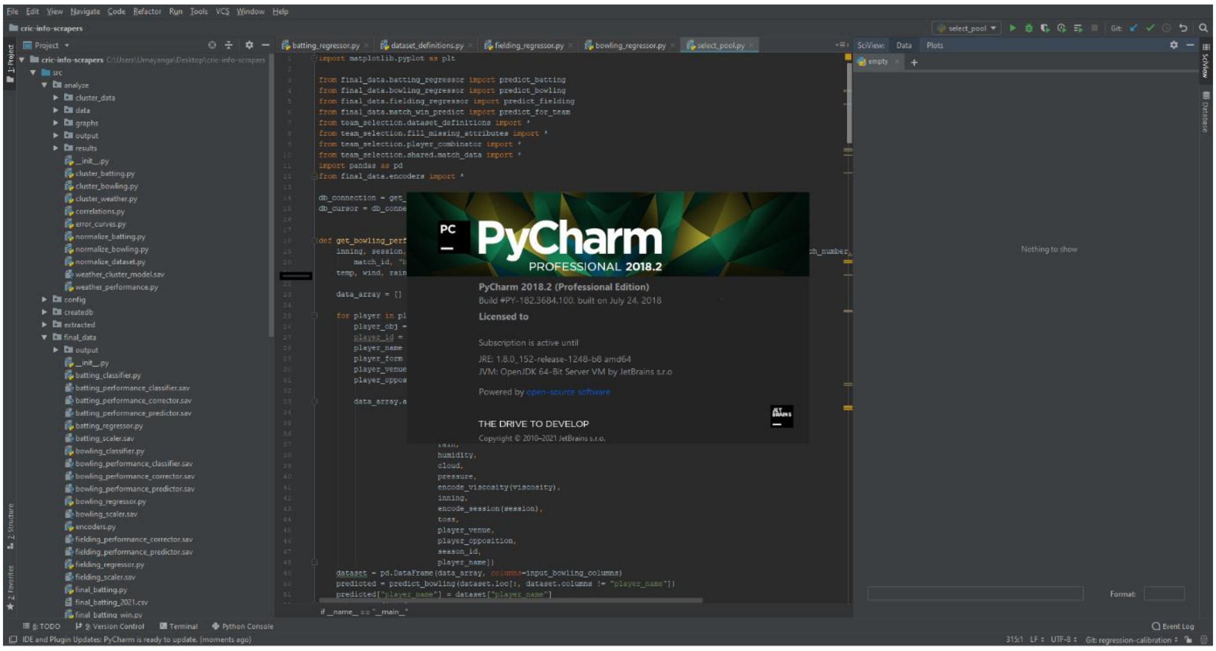


Figure 2: PyCharm 2018.2 Interface

PyCharm is an IDE that has been specifically designed for web and application development using Python Language. It further improves the efficiency of developing applications with Python by providing features such as auto-completion of code, debugging tools, project directory navigation and searching, database tools, version controlling and many more features. There are two main editions of PyCharm available—namely, Community Edition and Professional Edition. The community edition is free and can be downloaded and installed free of charge in our systems. Professional edition, on the other hand, comes with more advanced features for application development. Developers can try the Professional edition for free for a trial period.

3.5.Summary

This chapter discussed the leading technologies that we adopted to proceed with our study and the implementation. We briefly introduced the web scraping technology we used to scrape cricket match details and player performance details. Then we looked into details of the machine learning technology, different types of machine learning algorithms and different machine learning algorithms and how they work.

CHAPTER 4

4. DESIGN & METHODOLOGY

4.1.Introduction

This chapter aims to outline the proposed methodology for performance analysis of players described in research questions. We will explain the architectural design and the methodology used in this research with in-depth explanations. Also, we will describe the setup we use for experimenting with our methodology. The ultimate goal is to lay a strong foundation towards a solution with high accuracy of team prediction based on past player performance analysis. We will walk through the assumptions and decisions based on the dataset as we progress towards the final solution. Based on the literature research on past research related to the problem, we will follow the approaches adopted by those studies as the base methodology for this study. We will improve the existing and already proposed methodologies and concepts towards achieving an efficient and reliable prediction model as we progress.

4.2.Approach

Our study aims to provide an efficient and reliable predictive model for analysing the Sri Lankan cricket team's performance, assuming that different weather conditions affect each player's performance differently. We can split the main problem into two main subproblems. Namely, building a performance prediction model for all players based on batting, bowling and fielding performance [8]. And then how to find the optimum combination of players for the squad with the constraints of having at least one wicket-keeper and at least five bowlers in the team [1]. We found that most previous studies [7], [8], [9] have yielded more accurate results with machine learning classifiers such as Random Forest and Neural Network compared with other methodologies from the intense literature review. Therefore, our research will initially analyse the dataset and experiment with different machine learning approaches suggested by previous studies to test the baseline accuracy we can achieve with our prediction model.

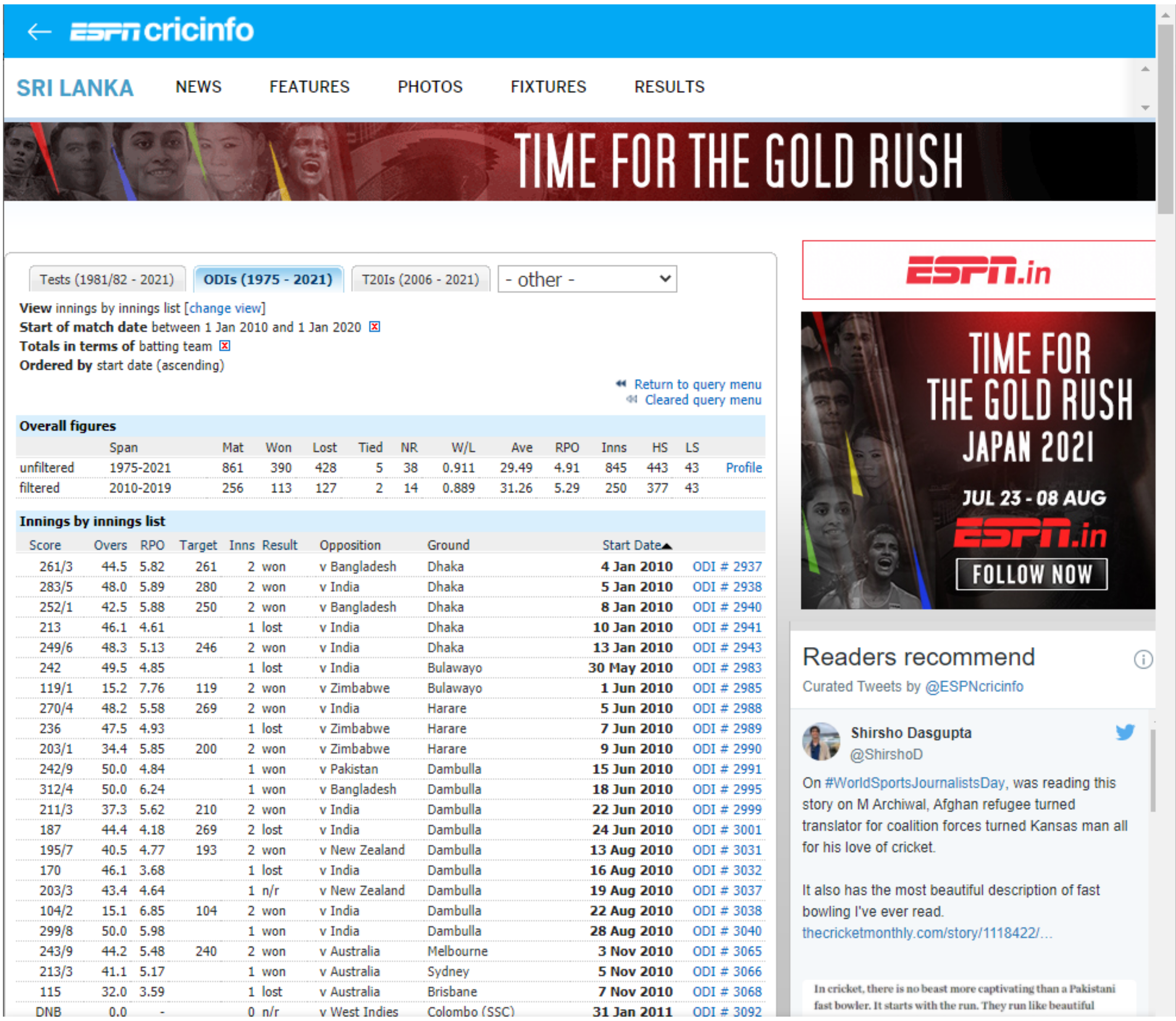
One of the primary concerns in our study is whether player performance is affected by different weather conditions. Most of the previous studies related to this problem have omitted weather conditions and analysed player performance overall, disregarding different playing conditions players perform. However, some studies have considered the effect of weather data by considering the humidity, wind flow, rain and day/night conditions [8]. Moreover, they have concluded that the evaluated weather condition data helped them improve the accuracy of the performance prediction model. Our study expands the weather attributes by including temperature, viscosity, cloud percentage, and atmospheric pressure. Also, we consider the calculation of form and consistency of players based on the equations developed in [7]. Most of the previous studies have not considered all of these attributes combined to analyses players' performance and build performance prediction models.

Regarding the team selection problem, most studies have evaluated the individual player performance and selected the players with the highest predicted performance and included them in the team. Our research defines a metric as a player performance rating for each player, derived based on the player's player performance and overall contribution to the team. With this new metric, we hope that it would help to combine player performance to achieve the maximum winning rate for the team.

4.2.1. Web Scraping and Data Collection

First, we needed to extract match details and player performance details from the https://stats.espncricinfo.com/ website. Figure 3 shows the web pages with the details we need to extract from the website.

We wrote the web scraper to iterate through each cricket match record and load the match details page. A sample match details page is shown in Figure 4. Once the match details are loaded, we can extract Toss results and Toss decisions from the table as shown in the image. Also, we can extract the season of the match, the time of play for the first and second sessions, and the date the match was played.



Overall figures

	Span	Mat	Won	Lost	Tied	NR	W/L	Ave	RPO	Inns	HS	LS	
unfiltered	1975-2021	861	390	428	5	38	0.911	29.49	4.91	845	443	43	Profile
filtered	2010-2019	256	113	127	2	14	0.889	31.26	5.29	250	377	43	

Innings by innings list

Score	Overs	RPO	Target	Inns	Result	Opposition	Ground	Start Date	
261/3	44.5	5.82	261	2	won	v Bangladesh	Dhaka	4 Jan 2010	ODI # 2937
283/5	48.0	5.89	280	2	won	v India	Dhaka	5 Jan 2010	ODI # 2938
252/1	42.5	5.88	250	2	won	v Bangladesh	Dhaka	8 Jan 2010	ODI # 2940
213	46.1	4.61		1	lost	v India	Dhaka	10 Jan 2010	ODI # 2941
249/6	48.3	5.13	246	2	won	v India	Dhaka	13 Jan 2010	ODI # 2943
242	49.5	4.85		1	lost	v India	Bulawayo	30 May 2010	ODI # 2983
119/1	15.2	7.76	119	2	won	v Zimbabwe	Bulawayo	1 Jun 2010	ODI # 2985
270/4	48.2	5.58	269	2	won	v India	Harare	5 Jun 2010	ODI # 2988
236	47.5	4.93		1	lost	v Zimbabwe	Harare	7 Jun 2010	ODI # 2989
203/1	34.4	5.85	200	2	won	v Zimbabwe	Harare	9 Jun 2010	ODI # 2990
242/9	50.0	4.84		1	won	v Pakistan	Dambulla	15 Jun 2010	ODI # 2991
312/4	50.0	6.24		1	won	v Bangladesh	Dambulla	18 Jun 2010	ODI # 2995
211/3	37.3	5.62	210	2	won	v India	Dambulla	22 Jun 2010	ODI # 2999
187	44.4	4.18	269	2	lost	v India	Dambulla	24 Jun 2010	ODI # 3001
195/7	40.5	4.77	193	2	won	v New Zealand	Dambulla	13 Aug 2010	ODI # 3031
170	46.1	3.68		1	lost	v India	Dambulla	16 Aug 2010	ODI # 3032
203/3	43.4	4.64		1	n/r	v New Zealand	Dambulla	19 Aug 2010	ODI # 3037
104/2	15.1	6.85	104	2	won	v India	Dambulla	22 Aug 2010	ODI # 3038
299/8	50.0	5.98		1	won	v India	Dambulla	28 Aug 2010	ODI # 3040
243/9	44.2	5.48	240	2	won	v Australia	Melbourne	3 Nov 2010	ODI # 3065
213/3	41.1	5.17		1	won	v Australia	Sydney	5 Nov 2010	ODI # 3066
115	32.0	3.59		1	lost	v Australia	Brisbane	7 Nov 2010	ODI # 3068
DNB	0.0	-		0	n/r	v West Indies	Colombo (SSC)	31 Jan 2011	ODI # 3092

Figure 3: Match Records List from https://stats.espncricinfo.com/[3]

[3] Web url: https://stats.espncricinfo.com/ci/engine/team/8.html?class=2;spanmax1=01+Jan+2020; spanmin1=01+Jan+2010;spanval1=span;template=results;type=team;view=innings

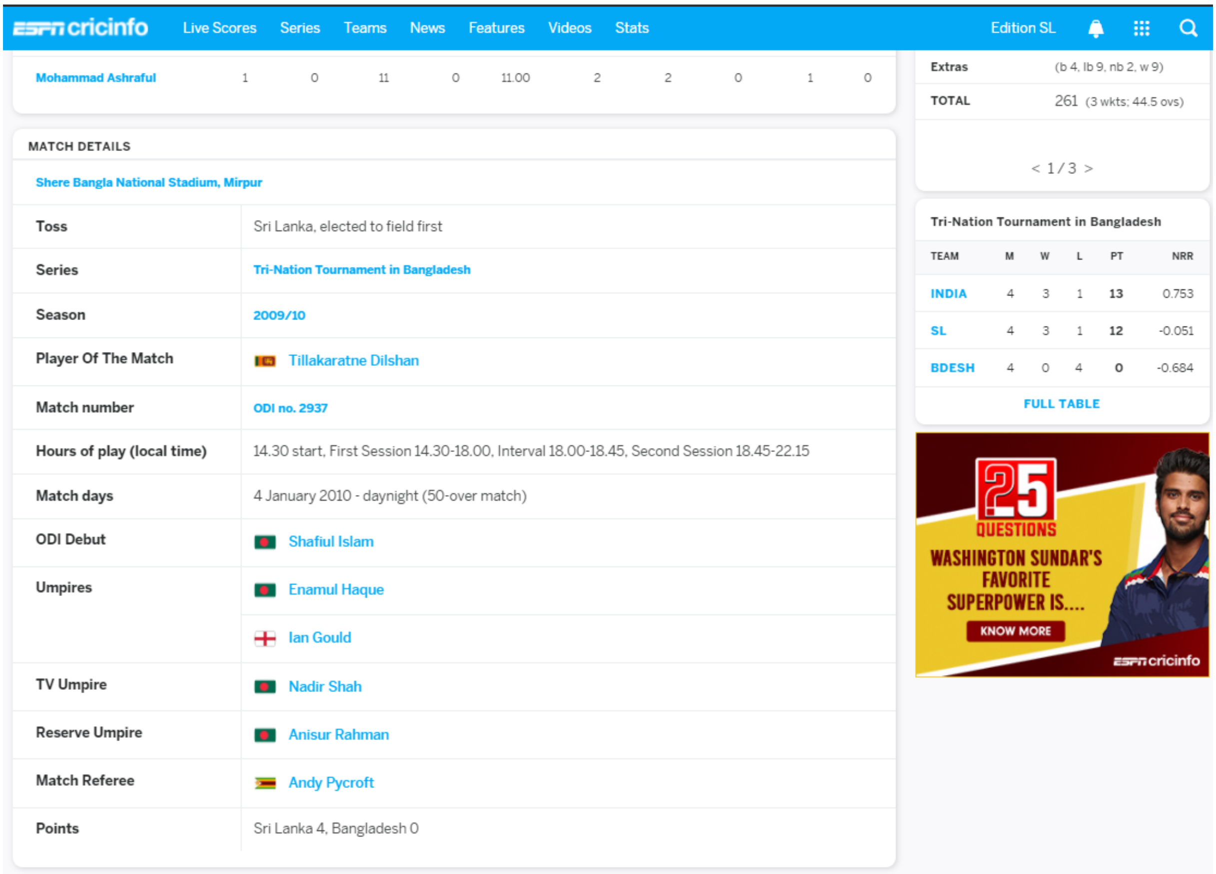
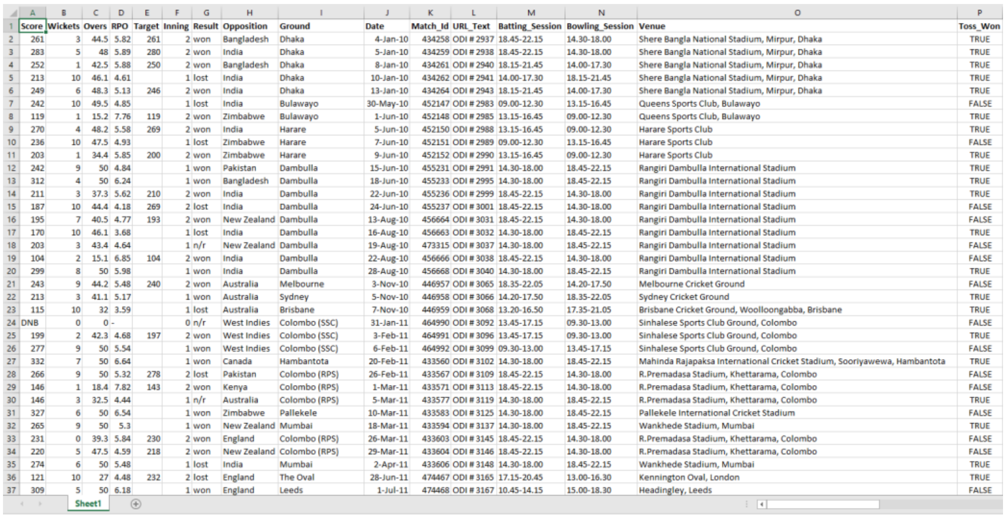


Figure 4: Match Details from https://stats.espncricinfo.com/[4]

Combining this data with the match list page data of Score, Overs, RPO, Target, Batting Inning, Result, Opposition, Ground and Date, a Comma Separated Values File (CSV File) was created. A Snippet from the CSV file of match details is shown in Figure 5.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Score	Wickets	Overs	RPO	Target	Inning	Result	Opposition	Ground	Date	Match_Id	URL_Text	Batting_Session	Bowling_Session	Venue	Toss_Won
2	261	3	44.5	5.82	261	2	won	Bangladesh	Dhaka	4-Jan-10	434258	ODI # 2937	18.45-22.15	14.30-18.00	Shere Bangla National Stadium, Mirpur, Dhaka	TRUE
3	283	5	48	5.89	280	2	won	India	Dhaka	5-Jan-10	434259	ODI # 2938	18.45-22.15	14.30-18.00	Shere Bangla National Stadium, Mirpur, Dhaka	TRUE
4	252	1	42.5	5.88	250	2	won	Bangladesh	Dhaka	8-Jan-10	434261	ODI # 2940	18.15-21.45	14.00-17.30	Shere Bangla National Stadium, Mirpur, Dhaka	TRUE
5	213	10	46.1	4.61		1	lost	India	Dhaka	10-Jan-10	434262	ODI # 2941	14.00-17.30	18.15-21.45	Shere Bangla National Stadium, Mirpur, Dhaka	TRUE
6	249	6	48.3	5.13	246	2	won	India	Dhaka	13-Jan-10	434264	ODI # 2943	18.15-21.45	14.00-17.30	Shere Bangla National Stadium, Mirpur, Dhaka	TRUE
7	242	10	49.5	4.85		1	lost	India	Bulawayo	30-May-10	452147	ODI # 2983	09.00-12.30	13.15-16.45	Queens Sports Club, Bulawayo	FALSE
8	119	1	15.2	7.76	119	2	won	Zimbabwe	Bulawayo	1-Jun-10	452148	ODI # 2985	13.15-16.45	09.00-12.30	Queens Sports Club, Bulawayo	TRUE
9	270	4	48.2	5.58	269	2	won	India	Harare	5-Jun-10	452150	ODI # 2988	13.15-16.45	09.00-12.30	Harare Sports Club	TRUE
10	236	10	47.5	4.93		1	lost	Zimbabwe	Harare	7-Jun-10	452151	ODI # 2989	09.00-12.30	13.15-16.45	Harare Sports Club	FALSE
11	203	1	34.4	5.85	200	2	won	Zimbabwe	Harare	9-Jun-10	452152	ODI # 2990	13.15-16.45	09.00-12.30	Harare Sports Club	TRUE
12	242	9	50	4.84		1	won	Pakistan	Dambulla	15-Jun-10	455231	ODI # 2991	14.30-18.00	18.45-22.15	Rangiri Dambulla International Stadium	TRUE
13	312	4	50	6.24		1	won	Bangladesh	Dambulla	18-Jun-10	455233	ODI # 2995	14.30-18.00	18.45-22.15	Rangiri Dambulla International Stadium	TRUE
14	211	3	37.3	5.62	210	2	won	India	Dambulla	22-Jun-10	455236	ODI # 2999	18.45-22.15	14.30-18.00	Rangiri Dambulla International Stadium	TRUE
15	187	10	44.4	4.18	269	2	lost	India	Dambulla	24-Jun-10	455237	ODI # 3001	18.45-22.15	14.30-18.00	Rangiri Dambulla International Stadium	FALSE
16	195	7	40.5	4.77	193	2	won	New Zealand	Dambulla	13-Aug-10	456664	ODI # 3031	18.45-22.15	14.30-18.00	Rangiri Dambulla International Stadium	FALSE
17	170	10	46.1	3.68		1	lost	India	Dambulla	16-Aug-10	456663	ODI # 3032	14.30-18.00	18.45-22.15	Rangiri Dambulla International Stadium	TRUE
18	203	3	43.4	4.64		1	n/r	New Zealand	Dambulla	19-Aug-10	473315	ODI # 3037	14.30-18.00	18.45-22.15	Rangiri Dambulla International Stadium	FALSE
19	104	2	15.1	6.85	104	2	won	India	Dambulla	22-Aug-10	456666	ODI # 3038	18.45-22.15	14.30-18.00	Rangiri Dambulla International Stadium	FALSE
20	299	8	50	5.98		1	won	India	Dambulla	28-Aug-10	456668	ODI # 3040	14.30-18.00	18.45-22.15	Rangiri Dambulla International Stadium	TRUE
21	243	9	44.2	5.48	240	2	won	Australia	Melbourne	3-Nov-10	446957	ODI # 3065	18.35-22.05	14.20-17.50	Melbourne Cricket Ground	FALSE
22	213	3	41.1	5.17		1	won	Australia	Sydney	5-Nov-10	446958	ODI # 3066	14.20-17.50	18.35-22.05	Sydney Cricket Ground	TRUE
23	115	10	32	3.59		1	lost	Australia	Brisbane	7-Nov-10	446959	ODI # 3068	13.20-16.50	17.35-21.05	Brisbane Cricket Ground, Woolloongabba, Brisbane	TRUE
24	DNB	0	0	-		0	n/r	West Indies	Colombo (SSC)	31-Jan-11	464990	ODI # 3092	13.45-17.15	09.30-13.00	Sinhalese Sports Club Ground, Colombo	FALSE
25	199	2	42.3	4.68	197	2	won	West Indies	Colombo (SSC)	3-Feb-11	464991	ODI # 3096	13.45-17.15	09.30-13.00	Sinhalese Sports Club Ground, Colombo	TRUE
26	277	9	50	5.54		1	won	West Indies	Colombo (SSC)	6-Feb-11	464992	ODI # 3099	09.30-13.00	13.45-17.15	Sinhalese Sports Club Ground, Colombo	FALSE
27	332	7	50	6.64		1	won	Canada	Hambantota	20-Feb-11	433560	ODI # 3102	14.30-18.00	18.45-22.15	Mahinda Rajapaksa International Cricket Stadium, Sooriyawewa, Hambantota	TRUE
28	266	9	50	5.32	278	2	lost	Pakistan	Colombo (RPS)	26-Feb-11	433567	ODI # 3109	18.45-22.15	14.30-18.00	R.Premadasa Stadium, Khettarama, Colombo	FALSE
29	146	1	18.4	7.82	143	2	won	Kenya	Colombo (RPS)	1-Mar-11	433571	ODI # 3113	18.45-22.15	14.30-18.00	R.Premadasa Stadium, Khettarama, Colombo	FALSE
30	146	3	32.5	4.44		1	n/r	Australia	Colombo (RPS)	5-Mar-11	433577	ODI # 3119	14.30-18.00	18.45-22.15	R.Premadasa Stadium, Khettarama, Colombo	TRUE
31	327	6	50	6.54		1	won	Zimbabwe	Pallekele	10-Mar-11	433583	ODI # 3125	14.30-18.00	18.45-22.15	Pallekele International Cricket Stadium	FALSE
32	265	9	50	5.3		1	won	New Zealand	Mumbai	18-Mar-11	433594	ODI # 3137	14.30-18.00	18.45-22.15	Wankhede Stadium, Mumbai	TRUE
33	231	0	39.3	5.84	230	2	won	England	Colombo (RPS)	26-Mar-11	433603	ODI # 3145	18.45-22.15	14.30-18.00	R.Premadasa Stadium, Khettarama, Colombo	FALSE
34	220	5	47.5	4.59	218	2	won	New Zealand	Colombo (RPS)	29-Mar-11	433604	ODI # 3146	18.45-22.15	14.30-18.00	R.Premadasa Stadium, Khettarama, Colombo	FALSE
35	274	6	50	5.48		1	lost	India	Mumbai	2-Apr-11	433606	ODI # 3148	14.30-18.00	18.45-22.15	Wankhede Stadium, Mumbai	TRUE
36	121	10	27	4.48	232	2	lost	England	The Oval	28-Jun-11	474467	ODI # 3165	17.15-20.45	13.00-16.30	Kennington Oval, London	TRUE
37	309	5	50	6.18		1	won	England	Leeds	1-Jul-11	474468	ODI # 3167	10.45-14.15	15.00-18.30	Headingley, Leeds	FALSE

Sheet1

Figure 5: Match Details CSV File Snippet

[4] Web url: https://www.espncricinfo.com/series/tri-nation-tournament-in-bangladesh-2009-10-434245/bangladesh-vs-sri-lanka-1st-match-434258/full-scorecard

The match details were scrapped and put into a CSV file; the next step was to scrape the batting details of the Sri Lanka Players from the webpage. A sample from the webpage with batting performance can be shown as in Figure 6.

SRI LANKA INNINGS (TARGET: 261 RUNS FROM 50 OVERS)

BATTING		R	B	M	4s	6s	SR
Upul Tharanga	c †Mushfiqur Rahim b Rubel Hossain	**14**	15	17	3	0	93.33
Tillakaratne Dilshan	c Naeem Islam b Mahmudullah	**104**	122	183	12	0	85.24
Kumar Sangakkara (c)†	c †Mushfiqur Rahim b Shafiul Islam	**74**	73	105	10	0	101.36
Thilan Samaraweera	not out	**41**	54	66	5	0	75.92
Chamara Silva	not out	**4**	7	11	0	0	57.14
Extras	(b 4, lb 9, nb 2, w 9)	**24**					
TOTAL	**(44.5 Ov, RR: 5.82)**	**261/3**					

Did not bat: Thilina Kandamby, Muthumudalige Pushpakumara, Suraj Randiv, Suranga Lakmal, Nuwan Kulasekara, Chanaka Welegedara

Fall of wickets: 1-35 (Upul Tharanga, 3.5 ov), 2-183 (Kumar Sangakkara, 29.1 ov), 3-242 (Tillakaratne Dilshan, 41.5 ov)

Figure 6: Batting Performance Data from https://stats.espncricinfo.com/[5]

With our batting data scraper, we could scrape the player's name, how and whether they got out, runs scored, balls face, minutes spent in the field batting, the number of fours, sixes scored and the strike rate. Once the scraped data is put into a CSV file, it looks as shown in Figure 7.

[5] Web url: https://www.espncricinfo.com/series/tri-nation-tournament-in-bangladesh-2009-10-434245/bangladesh-vs-sri-lanka-1st-match-434258/full-scorecard

	A	B	C	D	E	F	G	H	I	J
1	Name	Desc	Runs	Balls	Minutes	Fours	Sixes	Strike_Rate	Batting_Position	Match_Id
2	Upul Tharanga	c †Mushfiqur Rahim b Rubel Hossain	14	15	17	3	0	93.33	1	434258
3	Tillakaratne Dilshan	c Naeem Islam b Mahmudullah	104	122	183	12	0	85.25	2	434258
4	Kumar Sangakkara	c †Mushfiqur Rahim b Shafiul Islam	74	73	105	10	0	101.37	3	434258
5	Thilan Samaraweera	not out	41	54	66	5	0	75.93	4	434258
6	Chamara Silva	not out	4	7	11	0	0	57.14	5	434258
7	Upul Tharanga	c & b Harbhajan Singh	30	48	78	4	0	62.5	1	434259
8	Lahiru Thirimanne	c Gambhir b Sreesanth	22	24	38	4	0	91.67	2	434259
9	Kumar Sangakkara	c Raina b Harbhajan Singh	60	80	127	4	0	75	3	434259
10	Thilan Samaraweera	not out	105	106	146	11	0	99.06	4	434259
11	Thilina Kandamby	c Sreesanth b Harbhajan Singh	8	11	16	1	0	72.73	5	434259
12	Suraj Randiv	run out (†Dhoni/Nehra)	4	9	12	0	0	44.44	6	434259
13	Thisara Perera	not out	36	15	18	6	1	240	7	434259
14	Upul Tharanga	not out	118	126	179	18	0	93.65	1	434261
15	Mahela Jayawardene	c †Mushfiqur Rahim b Naeem Islam	108	117	159	13	0	92.31	2	434261
16	Kumar Sangakkara	not out	17	14	19	2	0	121.43	3	434261
17	Upul Tharanga	c Karthik b Tyagi	0	4	3	0	0	0	1	434262
18	Tillakaratne Dilshan	c Gambhir b Khan	33	17	26	8	0	194.12	2	434262
19	Kumar Sangakkara	c Raina b Yuvraj Singh	68	78	127	9	0	87.18	3	434262
20	Mahela Jayawardene	c Kohli b Khan	5	13	16	1	0	38.46	4	434262
21	Thilan Samaraweera	lbw b Sreesanth	0	6	6	0	0	0	5	434262
22	Thilina Kandamby	run out (Khan/Karthik)	1	11	16	0	0	9.09	6	434262
23	Thisara Perera	c Yuvraj Singh b Mishra	11	17	22	2	0	64.71	7	434262
24	Suraj Randiv	b Mishra	56	76	103	5	0	73.68	8	434262
25	Thilan Thushara	c Yuvraj Singh b Khan	28	44	56	3	0	63.64	9	434262
26	Chanaka Welegedara	st †Dhoni b Mishra	1	4	9	0	0	25	10	434262
27	Suranga Lakmal	not out	0	8	8	0	0	0	11	434262
28	Upul Tharanga	c Kohli b Nehra	0	3	2	0	0	0	1	434264
29	Tillakaratne Dilshan	c †Dhoni b Yuvraj Singh	49	54	82	8	0	90.74	2	434264
30	Kumar Sangakkara	c Sehwag b Harbhajan Singh	55	51	91	8	0	107.84	3	434264

Figure 7: Batting Data CSV File Snippet

Similarly, we wrote a scraper to extract bowling data from the opposition team's scorecard. Figure 8 shows how the bowling statistics of the Sri Lankan Bowlers are shown in the scorecard.

Fall of wickets: 1-65 (Imrul Kayes, 12.5 ov), 2-71 (Tamim Iqbal, 13.6 ov), 3-71 (Raqibul Hasan, 14.6 ov), 4-74 (Shakib Al Hasan, 16.4 ov), 5-132 (Mushfiqur Rahim, 31.4 ov), 6-227 (Mahmudullah, 47.3 ov), 7-238 (Mohammad Ashraful, 48.3 ov)

BOWLING	O	M	R	W	ECON	0s	4s	6s	WD	NB
Nuwan Kulasekara	10	1	46	1	4.59	28	2	0	1	0
Chanaka Welegedara	8	0	39	0	4.87	27	5	0	1	0
Tillakaratne Dilshan	3	0	16	1	5.33	11	1	0	1	0
Suranga Lakmal	9	1	63	2	7.00	32	8	2	5	0
Suraj Randiv	10	0	51	2	5.09	28	3	1	2	0
Thilina Kandamby	5	0	21	0	4.20	15	1	0	0	0
Muthumudalige Pushpakumara	5	0	21	0	4.20	12	0	0	1	0

Figure 8: Bowling Performance Data from https://stats.espncricinfo.com/[6]

With the bowling data scraper, we were able to extract: the bowler's name, number of overs bowled, number of maiden overs bowled, runs conceded, wickets-taken, economy, number of

[6] Web url: https://www.espncricinfo.com/series/tri-nation-tournament-in-bangladesh-2009-10-434245/bangladesh-vs-sri-lanka-1st-match-434258/full-scorecard

dot balls bowled, number of fours, sixes, wides and no balls conceded to the opposition team. The final bowling data CSV file looks like shown in Figure 9.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Name	Overs	Maidens	Runs	Wickets	Econ	Dots	4s	6s	Wd	Nb	Match_Id
2	Nuwan Kulasekara	10	1	46	1	4.6	28	2	0	1	0	434258
3	Chanaka Welegedara	8	0	39	0	4.88	27	5	0	1	0	434258
4	Tillakaratne Dilshan	3	0	16	1	5.33	11	1	0	1	0	434258
5	Suranga Lakmal	9	1	63	2	7	32	8	2	5	0	434258
6	Suraj Randiv	10	0	51	2	5.1	28	3	1	2	0	434258
7	Thilina Kandamby	5	0	21	0	4.2	15	1	0	0	0	434258
8	Muthumudalige Pushpakumara	5	0	21	0	4.2	12	0	0	1	0	434258
9	Chanaka Welegedara	10	1	66	5	6.6	33	7	0	4	4	434259
10	Suranga Lakmal	8	1	48	0	6	23	7	0	1	0	434259
11	Thilan Thushara	10	0	62	2	6.2	29	7	1	1	0	434259
12	Suraj Randiv	10	0	36	0	3.6	35	2	0	0	0	434259
13	Thilina Kandamby	4	1	22	0	5.5	14	0	2	1	0	434259
14	Thilan Samaraweera	2	0	12	0	6	4	1	0	0	0	434259
15	Thisara Perera	6	0	27	2	4.5	21	3	0	1	0	434259
16	Nuwan Kulasekara	10	1	48	2	4.8	39	7	0	2	0	434261
17	Thilan Thushara	9	0	67	1	7.44	26	5	2	5	0	434261
18	Suraj Randiv	10	1	40	2	4	33	4	0	0	0	434261
19	Thisara Perera	10	0	32	2	3.2	34	0	0	1	0	434261
20	Malinga Bandara	10	0	44	0	4.4	27	1	0	1	0	434261
21	Thilan Samaraweera	1	0	17	0	17	0	0	2	0	0	434261
22	Chanaka Welegedara	5.4	0	51	0	9	17	8	0	0	2	434262
23	Suranga Lakmal	10	0	75	0	7.5	29	12	0	1	3	434262
24	Thilan Thushara	6	0	33	1	5.5	18	4	0	1	0	434262
25	Suraj Randiv	6	0	19	0	3.17	23	0	0	0	0	434262
26	Thisara Perera	5	1	32	1	6.4	15	4	0	0	0	434262
27	Nuwan Kulasekara	10	0	48	4	4.8	39	6	1	1	0	434264
28	Chanaka Welegedara	10	1	53	3	5.3	31	6	0	0	0	434264
29	Thilan Thushara	6	1	26	0	4.33	22	2	0	2	0	434264
30	Suraj Randiv	9	0	47	1	5.22	29	4	1	1	0	434264

Figure 9: Bowling Data CSV File Snippet

Once we finished collecting batting and bowling statistics of the players for each match. The next step was to collect fielding data for each player from the match details page. A sample of how the fielding data is available from the website is shown in Figure 10.

BANGLADESH INNINGS (50 OVERS MAXIMUM)								
BATTING		R	B	M	4s	6s	SR
Tamim Iqbal	c Lakmal b Dilshan	40	46	58	5	0	86.95
Imrul Kayes	c Samaraweera b Kulasekara	23	37	52	2	0	62.16
Mohammad Ashraful	run out (†Sangakkara)	75	94	148	6	0	79.78
Raqibul Hasan	c Samaraweera b Lakmal	0	6	6	0	0	0.00
Shakib Al Hasan (c)	c Welegedara b Lakmal	1	7	7	0	0	14.28
Mushfiqur Rahim †	lbw b Randiv	35	52	54	3	0	67.30
Mahmudullah	lbw b Randiv	45	47	66	3	1	95.74
Naeem Islam	not out	22	9	14	1	2	244.44
Abdur Razzak	not out	1	2	8	0	0	50.00
Extras	(lb 3, w 15)	18					
TOTAL	(50 Ov, RR: 5.20)	260/7					

Figure 10: Fielding Performance Data from https://stats.espncricinfo.com/[7]

As shown in Figure 10, in front of every batsman of the opposition team, how they got out is mentioned. If they got out by caught, the fielder who caught the catch mentions the prefix “c”. Also, it is mentioned with the ‘run out’ prefix followed by the fielder’s name within brackets if they got run out. For each match and each player, we collected the number of catches taken, run-outs taken by each player during the match. Also, we needed to collect the data on miss-fields by the players. While the miss-fields data are not directly available on the website, we attempted to collect the miss-fields data from the commentary logs available from the same website. The dropped catches and missed run-out opportunities were not directly available for extraction using scrapers; we manually read through all the commentary logs to identify where players have dropped catches or missed run-out opportunities. Different commentators can interpret a particular instance of a dropped catch or a missed run-out opportunity differently. Since it would be a tedious task to go through video footage to determine whether the dropped opportunity was a difficult chance or not, we decided to trust the commentator’s judgement on whether it was a real opportunity missed or not. We only considered a specific instance as a dropped catch or a missed run-out opportunity only if the commentators have explicitly

[7] Web url: https://www.espncricinfo.com/series/tri-nation-tournament-in-bangladesh-2009-10-434245/bangladesh-vs-sri-lanka-1st-match-434258/full-scorecard

mentioned as such and has mentioned the fielder involved with the instance. A sample from the commentary log is shown in Figure 11.

END OF OVER 28 7 runs — **IRE: 138/3** CRR: 4.92 • RRR: 7.55 • Need 166 runs from 22 overs

Kevin O'Brien	26 (21)	Nuwan Pradeep	5-0-34-0
William Porterfield	67 (80)	Seekkuge Prasanna	5-0-31-0

27.6 **1** Pradeep to K O'Brien, 1 run
Drops the ball into the covers and sets off for the run

27.5 **1** Pradeep to Porterfield, 1 run
Looks to cut the ball again, doesn't time it and Porterfield calls him through for the quick single

27.4 • Pradeep to Porterfield, no run
Looks to cut the ball and doesn't time it

27.3 • Pradeep to Porterfield, no run
Its a bouncer from Prasanna and O'Brien can't get anything on it

27.2 **4** Pradeep to Porterfield, FOUR runs
Dropped O'Brien gets another life. He hits the ball high in the air and Mathews gets under the ball at long-on, it swirls in the wind and Mathews can't hold on. To make things worse the ball runs to the boundary

27.1 **1** Pradeep to K O'Brien, 1 run
Its a good length ball from Pradeep. O'Brien plays a late cut and can't beat a diving de Silva at point

END OF OVER 27 4 runs — **IRE: 131/3** CRR: 4.85 • RRR: 7.52 • Need 173 runs from 23 overs

Figure 11: Misfielding instances in Commentary Log from https://stats.espncricinfo.com/

The final Fielding data CSV file looks like shown in Figure 12 below.

	A	B	C	D	E	F
1	Name	Catches	Run Outs	Dropped Catches	Missed Runouts	Match_Id
2	Suranga Lakmal	1	0	0	0	434258
3	Thilan Samaraweera	2	0	0	0	434258
4	Nuwan Kulasekara	0	0	1	0	434258
5	Kumar Sangakkara	0	1	1	1	434258
6	Chanaka Welegedara	1	0	0	0	434258
7	Lahiru Thirimanne	1	0	0	0	434259
8	Upul Tharanga	0	0	1	0	434259
9	Kumar Sangakkara	2	0	0	0	434259
10	Suraj Randiv	0	0	1	0	434259
11	Suranga Lakmal	1	0	0	0	434259
12	Thilina Kandamby	1	0	0	0	434259
13	Thisara Perera	1	0	0	0	434259
14	Thilan Thushara	0	1	0	0	434261
15	Mahela Jayawardene	0	0	2	0	434261
16	Suraj Randiv	1	0	0	0	434261
17	Upul Tharanga	1	0	0	0	434261

Figure 12: Fielding Data CSV File Snippet

Once we finalised all the match data scraping and storing into CSV files, we collected weather data for each match data we have collected. We collected the weather data from the https://www.worldweatheronline.com/ website. Unfortunately, weather data corresponding to each match was not scrapable by writing a web scraper. Therefore, we had to extract the weather data from the website manually. The website contains a page for each international stadium where we can navigate to and select the date of the match played and view the weather attributes data available in regular time intervals.

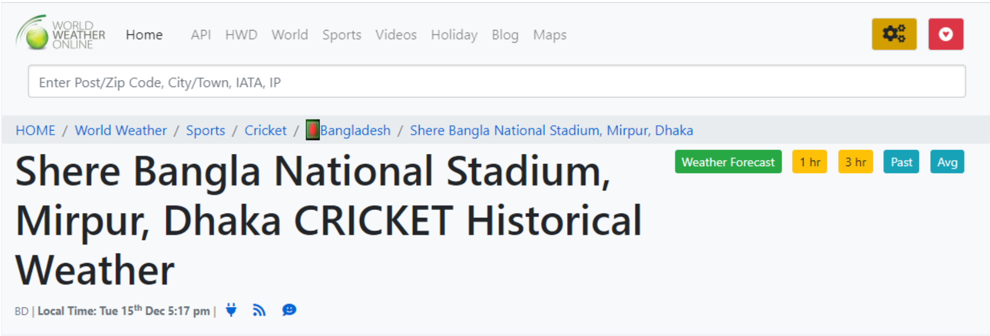


Figure 13: https://www.worldweatheronline.com/ has a page for each International Cricket Stadium[8]

Time	Weather	Temp	Feels	Wind	Gust	Rain	Humidity	Cloud	Pressure	Vis
00:00	Mist	20 °c	21 °c	8 km/h from N	11 km/h	0.0 mm	96%	100%	1007 mb	Poor
03:00	Fog	20 °c	20 °c	9 km/h from N	13 km/h	0.1 mm	98%	100%	1005 mb	Poor
06:00	Fog	20 °c	20 °c	9 km/h from NNE	15 km/h	0.0 mm	100%	100%	1006 mb	Poor
09:00	Partly cloudy	22 °c	22 °c	11 km/h from NNE	13 km/h	0.0 mm	90%	62%	1008 mb	Excellent
12:00	Cloudy	25 °c	27 °c	10 km/h from N	11 km/h	0.0 mm	77%	66%	1006 mb	Excellent

Figure 14: Weather data can be viewed for past days[9]

We collected weather data for batting sessions and bowling sessions separately. Usually, a batting or bowling session lasts around 3 hours. Weather data is available for every 3-hour window of the day from the website. Therefore, we mapped the weather data with batting and bowling sessions by selecting the most suitable and closest time frame from weather data.

[8] Web url: https://www.worldweatheronline.com/cricket/shere-bangla-national-stadium-mirpur-dhaka-weather/bd.aspx

[9] Web url: https://www.worldweatheronline.com/cricket/shere-bangla-national-stadium-mirpur-dhaka-weather-history/bd.aspx

Figure 15 shows the weather data mapped to the Sri Lankan team's batting sessions of each match. Similarly, we mapped weather data to bowling and fielding data as well.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Date	Match_Id	URL_Text	Batting_Session	Venue	Temp	Feels	Wind	Gust	Rain	Humidity	Cloud	Pressure	Vis
4-Jan-10	434258	ODI # 2937	18.45-22.15	Shere Bangla National Stadium, Mirpur, Dhaka	20°c	20°c	11 km/h from W	23 km/h	0.0 mm	50%	6%	1009 mb	Excellent
5-Jan-10	434259	ODI # 2938	18.45-22.15	Shere Bangla National Stadium, Mirpur, Dhaka	22°c	25°c	11 km/h from NW	21 km/h	0.0 mm	51%	0%	1010 mb	Excellent
8-Jan-10	434261	ODI # 2940	18.15-21.45	Shere Bangla National Stadium, Mirpur, Dhaka	24°c	25°c	5 km/h from W	11 km/h	0.0 mm	50%	0%	1012 mb	Excellent
10-Jan-10	434262	ODI # 2941	14.00-17.30	Shere Bangla National Stadium, Mirpur, Dhaka	30°c	29°c	5 km/h from NW	5 km/h	0.0 mm	35%	0%	1012 mb	Excellent
13-Jan-10	434264	ODI # 2943	18.15-21.45	Shere Bangla National Stadium, Mirpur, Dhaka	20 °c	20 °c	9 km/h from WNW	20 km/h	0.0 mm	42%	52%	1015 mb	Excellent
30-May-10	452147	ODI # 2983	09.00-12.30	Queens Sports Club, Bulawayo	14°c	12°c	21 km/h from ESE	31 km/h	1.2 mm	90%	63%	1025 mb	Excellent
1-Jun-10	452148	ODI # 2985	13.15-16.45	Queens Sports Club, Bulawayo	14°c	13°c	15 km/h from SE	17 km/h	0.0 mm	75%	74%	1024 mb	Excellent
5-Jun-10	452150	ODI # 2988	13.15-16.45	Harare Sports Club	22°c	22°c	8 km/h from SW	9 km/h	0.0 mm	38%	3%	1021 mb	Excellent
7-Jun-10	452151	ODI # 2989	09.00-12.30	Harare Sports Club	18 °c	18 °c	12 km/h from ENE	13 km/h	0.0 mm	45%	5%	1023 mb	Excellent
9-Jun-10	452152	ODI # 2990	13.15-16.45	Harare Sports Club	24 °c	25 °c	4 km/h from WNW	5 km/h	0.0 mm	39%	4%	1018 mb	Excellent
15-Jun-10	455231	ODI # 2991	14.30-18.00	Rangiri Dambulla International Stadium	27 °c	32 °c	3 km/h from SSW	6 km/h	5.1 mm	88%	48%	1010 mb	Good
18-Jun-10	455233	ODI # 2995	14.30-18.00	Rangiri Dambulla International Stadium	27 °c	31 °c	2 km/h from NW	4 km/h	6.5 mm	91%	56%	1009 mb	Good
22-Jun-10	455236	ODI # 2999	18.45-22.15	Rangiri Dambulla International Stadium	25 °c	28 °c	4 km/h from WSW	8 km/h	5.7 mm	99%	54%	1010 mb	Good
24-Jun-10	455237	ODI # 3001	18.45-22.15	Rangiri Dambulla International Stadium	25 °c	28 °c	4 km/h from SSW	6 km/h	11.0 mm	100%	57%	1008 mb	Average
13-Aug-10	456664	ODI # 3031	18.45-22.15	Rangiri Dambulla International Stadium	23 °c	26 °c	5 km/h from SW	10 km/h	0.6 mm	92%	21%	1010 mb	Excellent
16-Aug-10	456663	ODI # 3032	14.30-18.00	Rangiri Dambulla International Stadium	27 °c	31 °c	2 km/h from WNW	3 km/h	2.8 mm	82%	37%	1010 mb	Good
19-Aug-10	473315	ODI # 3037	14.30-18.00	Rangiri Dambulla International Stadium	26 °c	30 °c	4 km/h from SSW	6 km/h	4.2 mm	91%	37%	1010 mb	Good
22-Aug-10	456666	ODI # 3038	18.45-22.15	Rangiri Dambulla International Stadium	24 °c	27 °c	2 km/h from SSE	4 km/h	14.1 mm	98%	41%	1010 mb	Average
28-Aug-10	456668	ODI # 3040	14.30-18.00	Rangiri Dambulla International Stadium	27 °c	30 °c	9 km/h from SW	14 km/h	2.2 mm	77%	55%	1008 mb	Excellent
3-Nov-10	446957	ODI # 3065	18.35-22.05	Melbourne Cricket Ground	9 °c	8 °c	7 km/h from SSW	12 km/h	0.0 mm	80%	23%	1023 mb	Excellent
5-Nov-10	446958	ODI # 3066	14.20-17.50	Sydney Cricket Ground	17 °c	17 °c	21 km/h from SSE	28 km/h	0.3 mm	66%	31%	1022 mb	Excellent
7-Nov-10	446959	ODI # 3068	13.20-16.50	Brisbane Cricket Ground, Woolloongabba, Brisbane	23 °c	25 °c	17 km/h from E	23 km/h	0.4 mm	74%	27%	1018 mb	Excellent
31-Jan-11	464990	ODI # 3092	13.45-17.15	Sinhalese Sports Club Ground, Colombo	27 °c	30 °c	8 km/h from N	13 km/h	12.4 mm	88%	55%	1009 mb	Average
3-Feb-11	464991	ODI # 3096	13.45-17.15	Sinhalese Sports Club Ground, Colombo	26 °c	29 °c	24 km/h from N	36 km/h	0.5 mm	83%	68%	1009 mb	Good
6-Feb-11	464992	ODI # 3099	09.30-13.00	Sinhalese Sports Club Ground, Colombo	30 °c	34 °c	16 km/h from N	19 km/h	0.0 mm	63%	25%	1009 mb	Excellent
20-Feb-11	433560	ODI # 3102	14.30-18.00	Mahinda Rajapaksa International Cricket Stadium, Soc	26 °c	28 °c	9 km/h from ENE	15 km/h	0.5 mm	88%	26%	1012 mb	Excellent
26-Feb-11	433567	ODI # 3109	18.45-22.15	R.Premadasa Stadium, Khettarama, Colombo	26 °c	29 °c	3 km/h from WNW	6 km/h	0.0 mm	82%	12%	1011 mb	Excellent
1-Mar-11	433571	ODI # 3113	18.45-22.15	R.Premadasa Stadium, Khettarama, Colombo	26 °c	28 °c	18 km/h from NNE	30 km/h	0.6 mm	75%	22%	1010 mb	Excellent
5-Mar-11	433577	ODI # 3119	14.30-18.00	R.Premadasa Stadium, Khettarama, Colombo	28 °c	33 °c	12 km/h from W	16 km/h	0.0 mm	72%	20%	1009 mb	Excellent
10-Mar-11	433583	ODI # 3125	14.30-18.00	Pallekele International Cricket Stadium	30 °c	31 °c	5 km/h from N	6 km/h	0.0 mm	45%	0%	1008 mb	Excellent

Figure 15: Weather data mapped to batting sessions of Sri Lankan Team

After collecting all the data, we carefully examined the files and found that session details of a few matches were missing. We manually filled those data and mapped them with the corresponding weather data. Also, we cleaned the weather data manually to get rid of units such as Celsius, km/h, mm from the collected weather attributes.

4.2.2. Data Storage

Once all the scraped and collected data was stored as CSV files as the next step, we imported and stored the data into a Relational Database. We used MySQL to store data because storing data in a structured database allows us to query, pre-process, and manipulate data easily. The database schema used to store the data is shown in Figure 16.

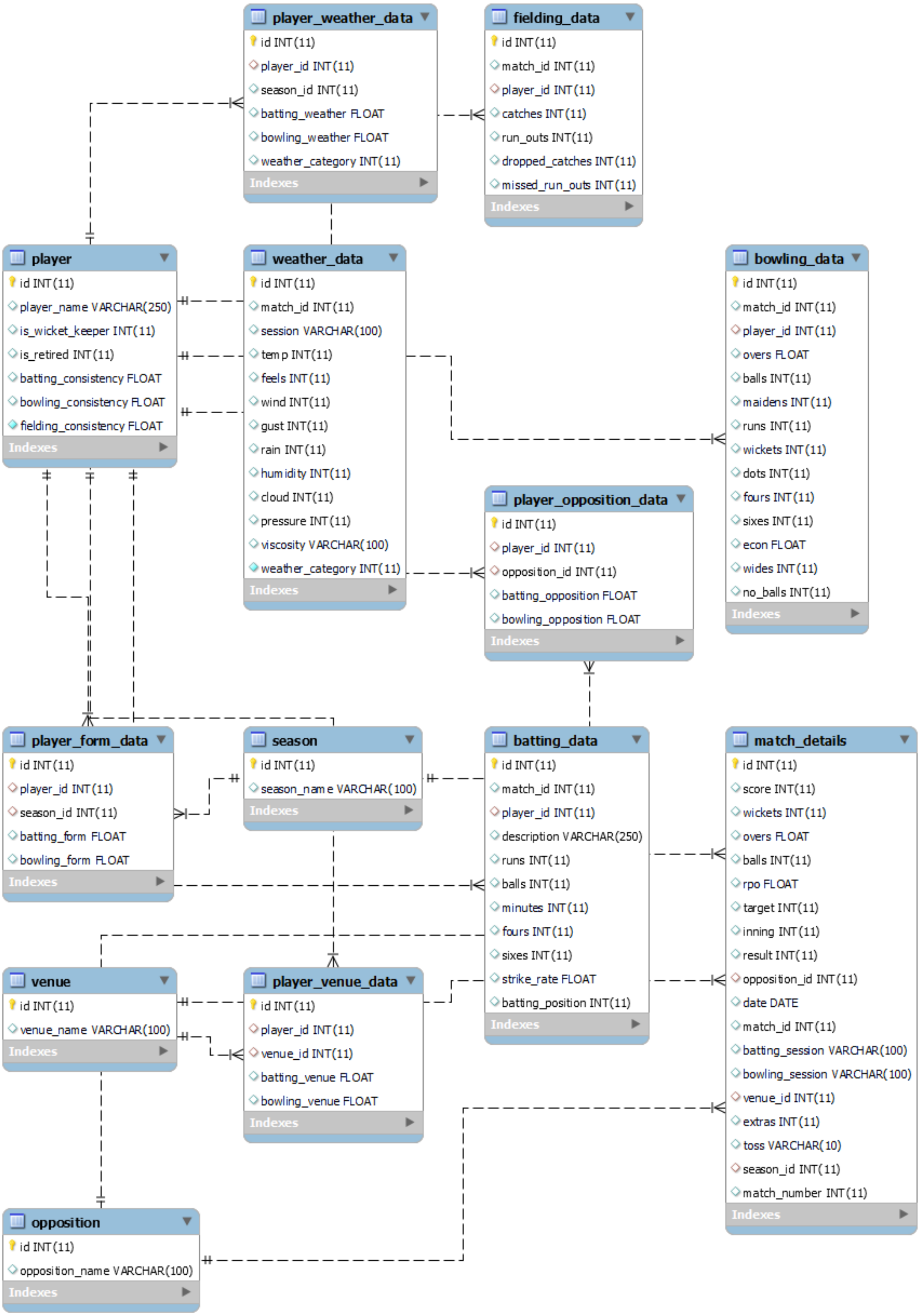


Figure 16: Database Schema

4.2.3. Data Pre-processing

Once we collected the dataset from the corresponding sources, match, player performance, and weather data were imported into a MySQL database using Python scripts. Then in the next step, we identified the match attributes we are considering for performance analysis of the players and find out any correlations between the data attributes to select the most suitable set of attributes for our prediction model. Figure 17 and Figure 18 shows the correlation matrix between the attributes for batting performance.

	runs	balls	minutes	fours	sixes	strike_rate	batting_position
runs	1	0.9375728	0.93473105	0.904511939	0.467780411	0.357647384	-0.386883811
balls	0.9375728	1	0.981167413	0.81127663	0.30454774	0.16889875	-0.417606171
minutes	0.93473105	0.981167413	1	0.808941935	0.306007206	0.184914172	-0.421946858
fours	0.904511939	0.81127663	0.808941935	1	0.287359706	0.371907124	-0.414521126
sixes	0.467780411	0.30454774	0.306007206	0.287359706	1	0.37689965	-0.053759416
strike_rate	0.357647384	0.16889875	0.184914172	0.371907124	0.37689965	1	-0.042522287
batting_positio	-0.386883811	-0.417606171	-0.421946858	-0.414521126	-0.053759416	-0.042522287	1

Figure 17: Correlation Matrix of Batting Performance Attributes

sixes	strike_rate	batting_position	temp	feels	gust	wind	rain	humidity	cloud	pressure
0.467780411	0.357647384	-0.386883811	-0.007625247	-0.00058935	0.011869495	0.010384453	-0.005878852	0.013750886	0.012687047	0.011588758
0.30454774	0.16889875	-0.417606171	0.005197289	0.010633289	0.008794666	0.008846016	-0.009947248	-0.000435086	0.005283425	0.004845115
0.306007206	0.184914172	-0.421946858	0.033001382	0.042904003	0.003252614	0.002427008	-0.000668567	0.002982356	-0.000765752	-0.010594545
0.287359706	0.371907124	-0.414521126	0.002423881	0.010664343	-0.010847936	-0.017246436	0.002971771	0.022334976	0.008518072	-0.009130498
1	0.37689965	-0.053759416	-0.042340557	-0.044446996	0.010594892	0.013696017	-0.021106359	-0.015372069	-0.014332868	0.04507877
0.37689965	1	-0.042522287	-0.022469993	-0.010968821	-0.011395104	-0.012280111	0.037088667	0.067703246	0.039791046	0.018530398
-0.053759416	-0.042522287	1	0.000673268	-0.007272157	0.007864559	0.013129177	-0.031301869	-0.030406743	-0.024936408	0.003653305
-0.042340557	-0.022469993	0.000673268	1	0.972806044	-0.120628736	-0.126099095	0.126520249	-0.26757401	-0.33725478	-0.52526101
-0.044446996	-0.010968821	-0.007272157	0.972806044	1	-0.093043125	-0.108248625	0.199870279	-0.09429305	-0.259920911	-0.554862125
0.010594892	-0.011395104	0.007864559	-0.120628736	-0.093043125	1	0.961229747	-0.241896522	0.074959689	-0.011387936	-0.00088829
0.013696017	-0.012280111	0.013129177	-0.126099095	-0.108248625	0.961229747	1	-0.265488353	0.016013496	-0.008767955	0.025429263
-0.021106359	0.037088667	-0.031301869	0.126520249	0.199870279	-0.241896522	-0.265488353	1	0.395660441	0.255678293	-0.236574726
-0.015372069	0.067703246	-0.030406743	-0.26757401	-0.09429305	0.074959689	0.016013496	0.395660441	1	0.48551571	-0.114708971
-0.014332868	0.039791046	-0.024936408	-0.33725478	-0.259920911	-0.011387936	-0.008767955	0.255678293	0.48551571	1	0.048761454
0.04507877	0.018530398	0.003653305	-0.52526101	-0.554862125	-0.00088829	0.025429263	-0.236574726	-0.114708971	0.048761454	1

Figure 18: Correlation Matrix of Batting Weather Attributes

We considered all the numeric data attributes that we have gathered and built a correlation matrix as shown in Figure 17 and Figure 18. As we can observe from Figure 17 and Figure 18, the number of runs a batsmen scores correlates with the number of balls he faces and the number of minutes he spends on the field. Also, it is observable that the number of fours that the batsmen scores also have a strong correlation (above 0.8). As we will be using these attributes to evaluate the players' performance, we kept all of the attributes except the number of minutes on the field that directly reflect the players' performance. Then when we are considering the weather data attributes, we can see that the temperature and feels, wind speed and gust also have correlations. Therefore, we decided to exclude feels and gust from the list of attributes. Similarly, we removed the feels and gust attributes from the bowling and fielding datasets as well.

4.2.3.1. Calculating Consistency and Form

Passi and Pandey, in their study [7], derived equations for calculating the consistency and form of players for batting and bowling performance. Also, they have proposed equations to calculate players' batting and bowling performance against opposition and in a specific venue. We adopted the equations proposed by them and, using a Python script, calculated the form, consistency, opposition and venue values for all the players and saved them to the database. The equations they have derived from their study are as follows. We only considered players who have played more than five matches when calculating Form, Opposition and Venue. For other players, we substituted the value with consistency.

Batting Consistency

$$\text{Consistency} = 0.4262*\text{Average} + 0.2566*\text{No. of innings} + 0.1510*\text{SR} + 0.0787*\text{Centuries} + 0.0556*\text{Fifties} - 0.0328*\text{Zeros} \quad (4\text{-}1)$$

Bowling Consistency

$$\text{Consistency} = 0.4174*\text{No. of overs} + 0.2634*\text{No. of innings} + 0.1602*\text{SR} + 0.0975*\text{Average} + 0.0615*\text{FF} \quad (4\text{-}2)$$

Batting Form

$$\text{Form} = 0.4262*\text{Average} + 0.2566*\text{No. of innings} + 0.1510*\text{SR} + 0.0787*\text{Centuries} + 0.0556*\text{Fifties} - 0.0328*\text{Zeros} \quad (4\text{-}3)$$

Bowling Form

$$\text{Form} = 0.3269*\text{No. of overs} + 0.2846*\text{No. of innings} + 0.1877*\text{SR} + 0.1210*\text{Average} + 0.0798*\text{FF} \quad (4\text{-}4)$$

Batting Opposition

$$\text{Opposition} = 0.4262*\text{Average} + 0.2566*\text{No. of innings} + 0.1510*\text{SR} + 0.0787*\text{Centuries} + 0.0556*\text{Fifties} - 0.0328*\text{Zeros} \quad (4\text{-}5)$$

Bowling Opposition

$$\text{Opposition} = 0.3177*\text{No. of overs} + 0.3177*\text{No. of innings} + 0.1933*\text{SR} + 0.1465*\text{Average} + 0.0943*\text{FF} \quad (4\text{-}6)$$

Batting Venue

$$\text{Venue} = 0.4262*\text{Average} + 0.2566*\text{No. of innings} + 0.1510*\text{SR} + 0.0787*\text{Centuries} + 0.0556*\text{Fifties} + 0.0328*\text{HS} \quad (4\text{-}7)$$

Bowling Venue

$$\text{Venue} = 0.3018*\text{No. of overs} + 0.2783*\text{No. of innings} + 0.1836*\text{SR} + 0.1391*\text{Average} + 0.0972*\text{FF} \quad (4\text{-}8)$$

The abbreviations used in the equations are as follows.

Average (Batting) – total runs scored divided by number of times got out
Average (Bowling) – total runs concede divided by the number of innings
No. of innings – number of innings batted, bowled
SR - Strike Rate (runs scored/ bowls faced)
FF – Number of five wickets taken in an inning in a given opposition or venue
Centuries – Number of innings 100 or more runs were scored
Fifties – Number of innings 50 to 99 runs were scored
Zeros – Number of time batsmen got out without scoring any runs
HS – Highest Number of Runs Scored in a given venue

4.2.3.2. Data Scaling

Often in any dataset, it can be frequently seen that attribute values spread in different scales. Machine learning algorithms perform better when all the input attributes are converted into the same scale. Therefore, normalisation can be used to convert all the numeric values using a standard scale. Normalisation places numeric attributes on the same scale (0,1) and prevents attributes with a large original scale from biasing the solution. The normalisation process was automated using the MinMaxScaler function available from the Scikit-Learn library. The MinMaxScaler function converts all the input values into a standard scale of values ranging from 0 to 1. The transformation done by the MinMaxScaler function can be represented using the following equations.

X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0)) 4-9

X_scaled = X_std * (max - min) + min 4-10

4.2.4. Individual Player Performance Prediction

Once we have the dataset cleaned and pre-processed Next step is to predict player performance under the given conditions. In our study, we are trying to predict multiple attributes to help evaluate the performance of players. Only considering the number of runs scored by a batsman would not be good enough. Because the number of balls consumed by each batsman to get that score is also essential in determining the number of batsmen who get the opportunity to bat in a specific inning. In our study, we will train a model to predict the runs scored, balls faced, number of fours, sixes scored and the ideal batting position for the batsmen.

Similarly, for the bowlers, we will train a model to predict the number of runs they concede to the opposition batsman, the number of overs the bowler will bowl, and the number of wickets the bowler will take.

Another vital role of every player is to be a good fielder. Fielding is an essential skill for every player disregard of the batting or bowling skills they possess. Therefore, we consider the attributes such as the number of catches, run-outs taken, catches missed, and run-outs missed to analyse the players' fielding performance. This data was collected from the commentary log of each match as explained in 4.2.1. At the same time, it would be valuable to consider the number of runs each fielder saved for the team while fielding; we could not consider that metric to evaluate fielders' performance since this data is not available. Also, the wicket-keeper of each team is a unique fielding position, so the same attributes as other fielders are collected for the wicket-keepers. The number of catches and run-outs taken by a fielder highly depend on the player's fielding position. Some fielding positions have a higher chance of getting catches/ run-outs, while other fielding positions will rarely get an opportunity to get a catch or a run-out. The critical factor is the success rate of each fielder at every opportunity they get at a catch or a run-out. Therefore, by considering the number of catches, run-outs, dropped catches, and missed run-out opportunities by each player, we can calculate the success rate of each fielder and use it as a metric for evaluating the player's fielding performance. Also, note that the fielding performance of a team is a significant factor affecting the number of runs conceded by the bowlers and hence towards the bowling performance of bowlers. Having only good bowlers does not help if the fielders are not up to the task equally and vice versa. A good combination of batting, bowling and fielding performance is the key to winning matches.

$$\textit{fielding success rate} = \frac{\textit{catches taken} + \textit{run outs taken}}{\textit{catches taken} + \textit{run outs taken} + \textit{catches dropped} + \textit{run outs dropped}} \qquad \text{4-11}$$

Using the performance metrics determined above for batsman, bowlers and fielders, we train predictive models for players' performance in each discipline; batting, bowling and fielding.

As the first step of training the prediction models, we evaluated the baseline accuracy of several regression algorithms towards predicting player performance. We chose these regression algorithms because they have been proved to predict player performance successfully in previous studies [7], [8], [9]. The regression algorithms we considered are Linear Regression, Support Vector Machine (SVM), Decision Tree and Random Forest. We chose the R-Squared (R^2) metric and Plot graphs to evaluate the prediction accuracy of the regression models.

Table 3 shows the R-Squared values for predicting runs scored by batsmen, runs conceded by bowlers and fielding success rate of players.

Table 3: Initial Prediction Accuracies for Regression Algorithms

ML Algorithm	**Runs Scored (R^2)**	**Runs Conceded (R^2)**	**Fielding Success Rate (R^2)**
Linear Regression	0.45	0.14	0.01
SVM Regression	0.14	0.12	-0.04
Decision Tree	-0.14	-0.36	-0.92
Random Forest	0.46	0.33	0.04

Based on the initial prediction results of the regression algorithms mentioned above, we observed that Random Forest Regression Algorithms performs relatively better when compared with other algorithms. Therefore, we decided to proceed with Random Forest Algorithm for performance predictions.

We graphed the variation of R^2 and Root Mean Squared Error (RMSE) against the maximum height of the trees and the maximum number of trees in the random forest for tuning the Random Forest Algorithm.

Figure 19 and Figure 20 show the variation of R2 and RMSE of predicted runs scored against the Max Height of the Trees.

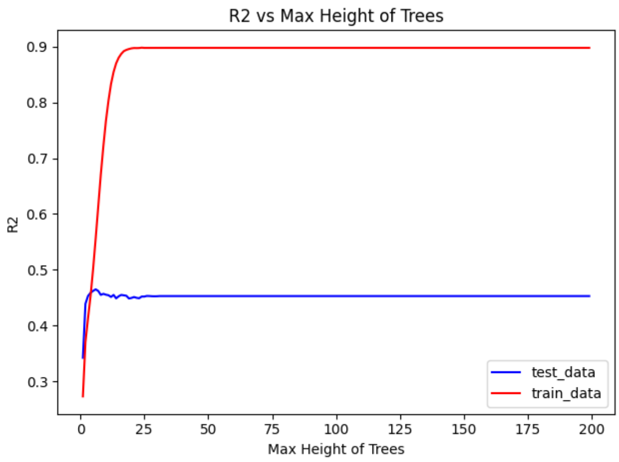


Figure 19: R^2 vs Max Height of Trees - Batting (Trees = 200)

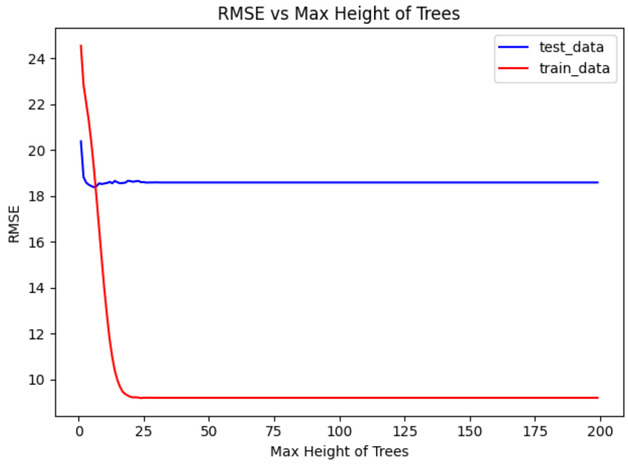


Figure 20: RMSE vs Max Height of Trees – Batting (Trees = 200)

Based on *Figure 19* and *Figure 20,* we can observe that if we increase the maximum height of the trees beyond a specific number, the model starts to overfit the training data. Therefore, we determined the optimum height of the trees by observing where the R^2 and RMSE values were approximately equal for both the training and test data sets.

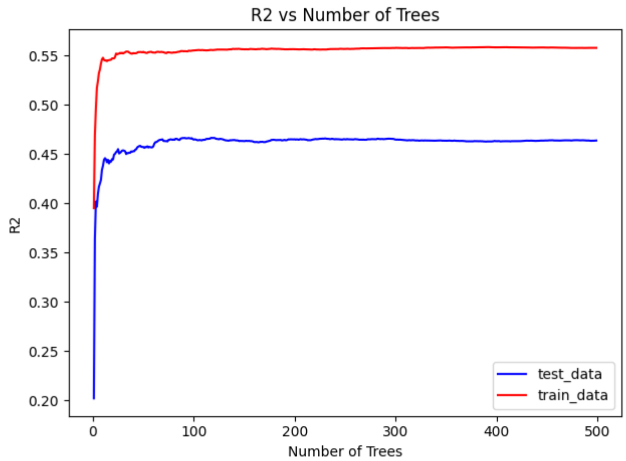


Figure 21: R^2 vs Number of Trees – Batting (Height = 6)

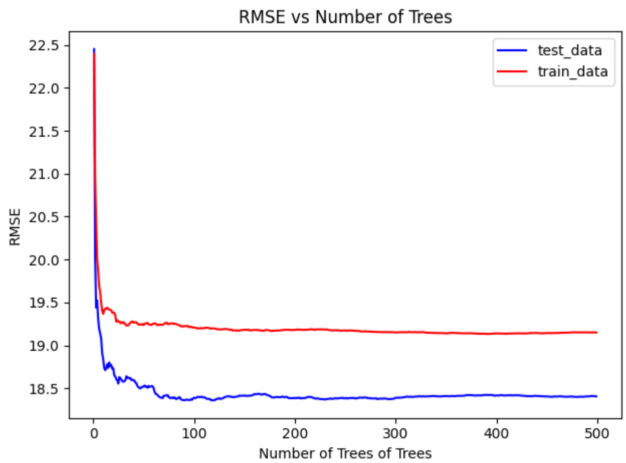


Figure 22: RMSE vs Number of Trees – Batting (Height = 6)

Figure 21 and *Figure 22* show the graphs of R^2 and RMSE against the Number of Trees with a Maximum height of 6. R^2 and RMSE keep improving until around 200 trees, and from there onwards, the values of R^2 and RMSE become steady. We continued to test for more trees up to 10000, but the change in R^2 and RMSE was not significant. Considering the R^2 and RMSE values, we decided to use six as the maximum height of trees and 200 as the number of trees. These values were consistent throughout the models for batting, bowling and fielding.

Initial Prediction Results for runs scored by batsmen and runs conceded by bowlers are shown in *Figure 23* and *Figure 24*, respectively. As observed from these two graphs, even though we tuned the model with optimum parameters for the maximum height of trees and number of trees, the model tends to predict with a significant bias error. One primary reason for this bias is the imbalance of data. In our dataset, we observed that the majority of runs scored by batsmen lies approximately below 25. Therefore, the model tries to fit more into the lower run predictions to minimise the variance and predict biased results.

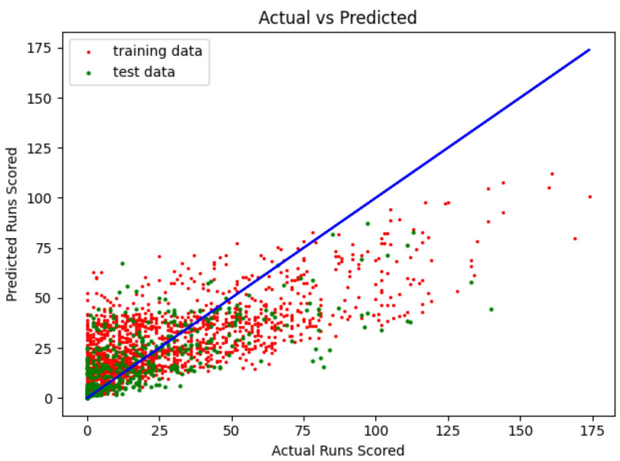


Figure 23: Predicted Runs Scored with Bias Error vs Actual Runs Scored

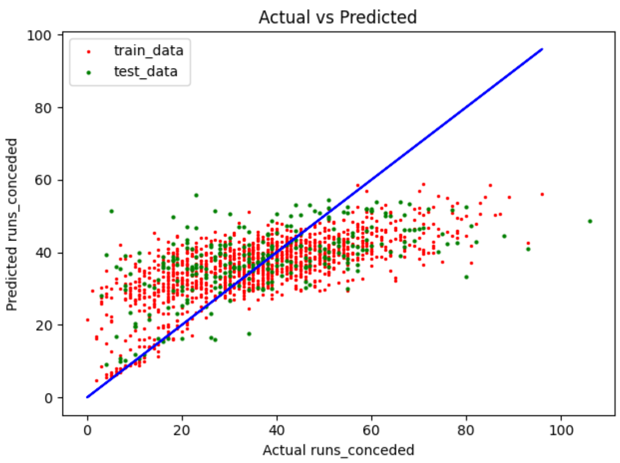


Figure 24: Predicted Runs Conceded with Bias Error vs Actual Runs Conceded

As Zhang and Lu have mentioned in their paper [18], this is a common issue in Random Forest Models. In their research paper, they have proposed a method using Random Forest to estimate the regression function. They suggest five different methods that we can use to estimate the bias effectively. One such method is to train a second Random Forest model to predict the error of the first model by taking the predicted value as input. So, we will be training a second model to predict the prediction error of the first Random Forest model. Then finally, by combining the two Random Forest models, we can significantly reduce the error and improve the accuracy of predictions. *Figure 25* shows a high-level diagram of the compound prediction model we designed based on the study of Zhang and Lu [18]. We made a slight modification to the approach suggested in the study of Zhang and Lu [18] to improve the bias correction.

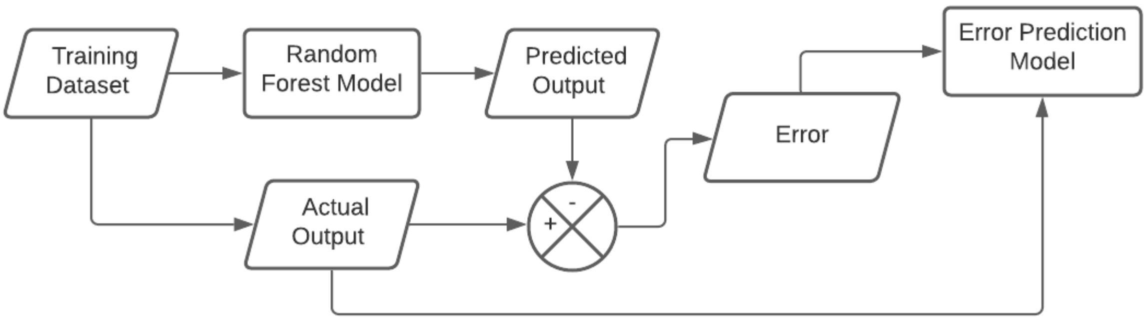


Figure 25: High-level diagram of Proposed Compound Prediction Model – Learning Phase

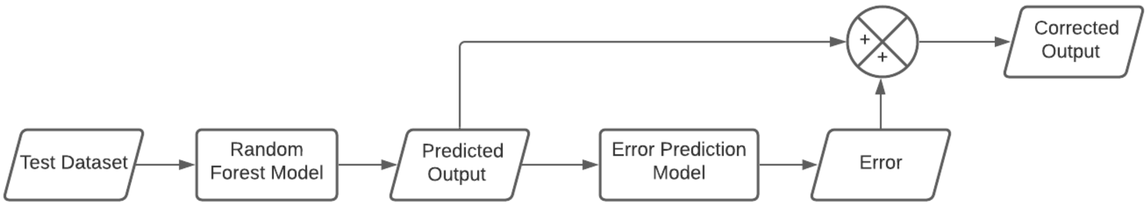


Figure 26: High-level diagram of Proposed Compound Prediction Model – Prediction phase

Using this combined model approach, we were able to reduce the bias of the prediction models significantly. Using the same approach, we trained multiple Random Forest Models to Predict different performance attributes of the players.

The following tables show the input attributes and output attributes of the batting, bowling, and fielding performance prediction modules.

Table 4: Input and Output attributes of the Batting Performance Prediction Module

Input Attributes	batting consistency, batting form, batting temp, batting wind, batting rain, batting humidity, batting cloud, batting pressure, batting viscosity, batting inning, batting session, toss, venue, opposition, season
Predicted Outputs	runs scored, balls faced, fours scored, sixes scored, batting position
Derived Outputs	strike rate = runs scored / balls faced

Table 5: Input and Output attributes of the Bowling Performance Prediction Module

Input Attributes	bowling consistency, bowling form, bowling temp, bowling wind, bowling rain, bowling humidity, bowling cloud, bowling pressure, bowling viscosity, batting inning, bowling session, toss, bowling venue, bowling opposition, season
Predicted Outputs	runs conceded, no. of deliveries bowled, no. wickets taken
Derived Outputs	economy = runs conceded / deliveries

Table 6: Input and Output attributes of the Fielding Performance Prediction Module

Input Attributes	fielding consistency, fielding temp, fielding wind, fielding rain, fielding humidity, fielding cloud, fielding pressure, fielding viscosity, fielding inning, fielding session, toss, season
Predicted Outputs	fielding success rate

All the values predicted using the performance prediction modules were saturated at 0 since negative values are not practical in performance attributes.

4.2.5. Player Rating Prediction Model

A batsman who bats in top order would get a higher opportunity to face many balls to score. However, a batsman batting in the middle order will get to bat with fewer deliveries to face in the last few overs. Therefore, even though those players might not get the opportunity to go for high scores, their contribution to the team by scoring quick runs with a higher strike rate is also valuable. It is of equal importance to the team performance as a top-order batsman who scores more runs with a relatively low strike rate. Similarly, some bowlers might perform well by taking wickets, and some bowlers might perform well by giving fewer runs to the opposition. Bowlers who get to bowl in the death overs (towards the end of the inning) are more likely to get wickets because they will probably be bowling to lower-order batsmen who are not well set and trying to score some runs quickly by risking the wicket.

On the other hand, some bowlers will get to bowl with fielding restrictions, where they can only put a limited number of fielders outside the 30 yards circle. Bowlers who bowl during these overs might have a more difficult time avoiding the batsmen from scoring boundaries and scoring more runs. Therefore, when evaluating players' performance, merely ranking the players based on runs scored, wickets taken, runs conceded would hinder the contribution of some players towards the overall team performance. Hence, in our study, we decided to develop a new attribute to rate players’ performance.

Once we predict the players’ performance under the given match conditions, we are training an Artificial Neural Network (ANN) to predict the probability that the team can win given the player’s contribution to the team. We created a dataset with 45 input attributes and labelled the match result (Win/Loss) to train the ANN Model. Then we take the probability of a player’s performance to be classified as a win as that player's rating. This player rating system was built based on the Neural Network based player rating system proposed by Al-Shboul et al. [1] to rate the performance of football players.

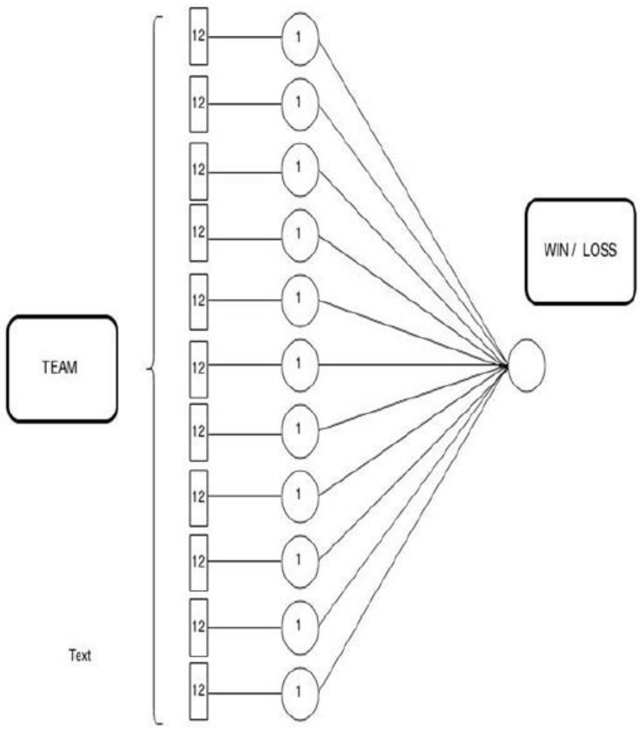


Figure 27: The player selection neural network architecture proposed by Al-Shboul et al. [1]

We modified the Neural Network architecture proposed by them by an additional layer and predicted cricket's player rating. While their Neural Network Architecture only has 11 input layers to represent 11 players, we introduce a new input layer on top of that by including a layer with 25 input nodes to represent different input attributes for the players.

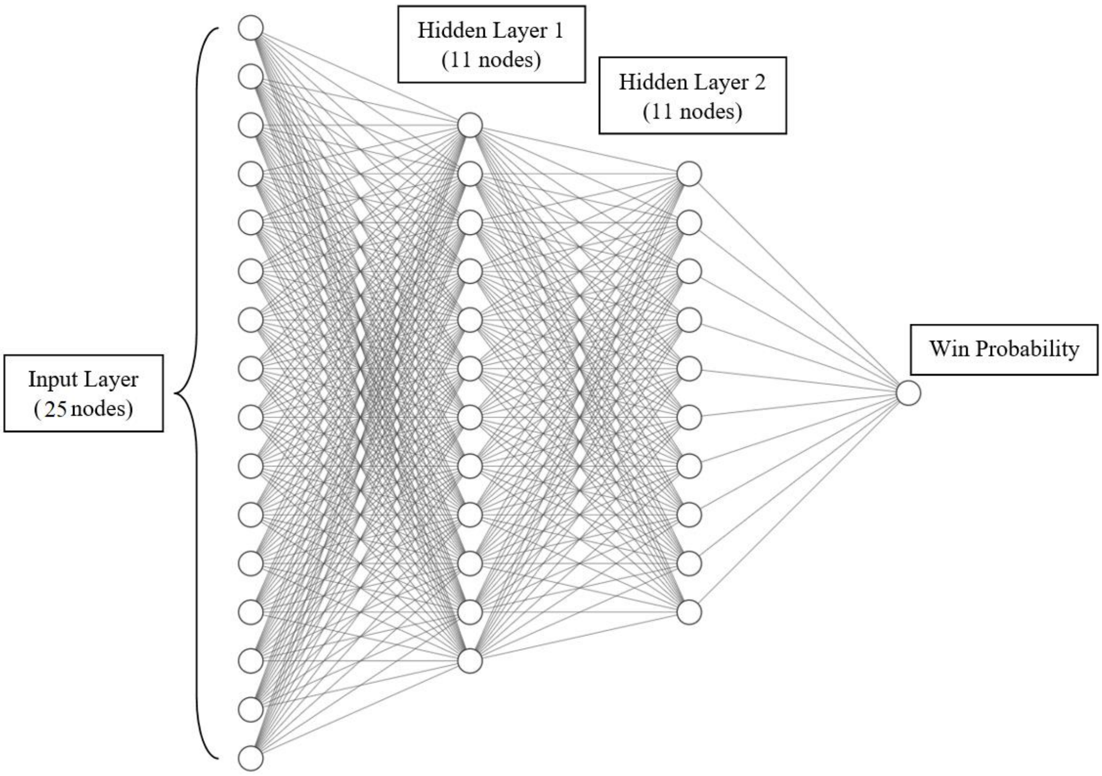


Figure 28: The modified player selection neural network architecture

Figure 27 shows the neural network architecture proposed in Al-Shboul et al., and *Figure 28* shows our study's modified neural network architecture.

The list of input attributes considered to the neural network and their source/derivations for training the neural network are shown in *Table 7*.

Table 7: Input Attributes and Their Source / Derivations for Training the Neural Network

No.	**Attribute**	**Source / Derivation**
1	batting consistency	4-1
2	batting form	4-3
3	batting temp	Batting Weather Dataset
4	batting wind	Batting Weather Dataset
5	batting rain	Batting Weather Dataset
6	batting humidity	Batting Weather Dataset
7	batting cloud	Batting Weather Dataset
8	batting pressure	Batting Weather Dataset
9	batting viscosity	Batting Weather Dataset

10	batting inning	Match Details Dataset
11	batting session	Match Details Dataset
12	toss	Match Details Dataset
13	batting venue	4-7
14	batting opposition	4-5
15	season	Match Details Dataset
16	runs scored	Batting Performance Dataset
17	balls faced	Batting Performance Dataset
18	fours scored	Batting Performance Dataset
19	sixes scored	Batting Performance Dataset
20	batting position	Batting Performance Dataset
21	batting contribution	Runs scored / total score
22	strike rate	Runs scored / balls faced
23	total score	Match Details Dataset
24	total wickets	Match Details Dataset
25	total balls	Match Details Dataset
26	target	Match Details Dataset
27	extras	Match Details Dataset
28	match number	Match Details Dataset
29	bowling consistency	4-2
30	bowling form	4-4
31	bowling temp	Bowling Weather Dataset
32	bowling wind	Bowling Weather Dataset
33	bowling rain	Bowling Weather Dataset
34	bowling humidity	Bowling Weather Dataset
35	bowling cloud	Bowling Weather Dataset
36	bowling pressure	Bowling Weather Dataset
37	bowling viscosity	Bowling Weather Dataset
38	bowling session	Match Details Dataset
39	bowling venue	4-8
40	bowling opposition	4-6
41	runs conceded	Bowling Performance Dataset
42	deliveries	Bowling Performance Dataset
43	wickets taken	Bowling Performance Dataset

44	bowling contribution	Runs conceded / target
45	economy	Runs conceded / deliveries
46	fielding consistency	4-11 (Overall)
47	success rate	4-11 (Per Match)

Out of the 47 input attributes listed above, we selected 25 significant features using p-values. We used 10-fold cross-validation to train and tune the neural network.

4.2.6. Team Performance Calculation Module

A batting team has only ten wickets at hand. Suppose the bowlers predicted performance suggests that they can all out the opposition without bowling the total number of overs. In that case, we have to limit the overall wickets taken to 10 wickets. Hence calculate the total number of runs conceded accordingly. Another essential aspect that most researchers have not considered in most researches is extra runs conceded to the opposition team by the bowling team. In a close game of cricket number of extra runs might be the key factor deciding between winning and losing. Since the number of extras conceded by a bowler to a batting side is already reflected in his bowling performance summary, we do not have to consider it explicitly when predicting the balling performance of bowlers. However, when it comes to predicting the overall total score of the batting team, if we only add the total number of runs scored by each batsman, we are making a mistake by ignoring the contribution to the total score made by extra runs. Also, we have to consider the total number of deliveries consumed by each batsman in their inning to determine how many deliveries are left for the other batsmen to bat. Without doing that, the total score predictions would be unreliable. While these types of prediction cases might be rare, we should not neglect the possibility of such cases while predicting the team's overall performance.

To find the optimum team, we have to combine 11 players from the available pool of players. Passi and Pandey, in their study [7], has mentioned that we need at least five-match records to calculate the form and consistency of a player. Our dataset identified 36 players who have played more than five ODIs. We decided only to consider those players for the selection pool since we do not have enough data to calculate the form and consistency for other players. Getting all combinations of 11 players from a pool of 36 players and calculating team performance for each combination of 11 players will take a mammoth amount of computational power. Therefore, we designed an algorithm to calculate the average team performance for the pool and used those average values to calculate each player's contribution to the team. The algorithms implement the mathematical functions given below to calculate the average team performance.

$$\text{team coefficient} = 11/ \text{ number of players in the pool} \qquad 4\text{-}12$$

$$\text{extras} = \text{average extra runs from the dataset (14.26)} \qquad 4\text{-}13$$

$$\text{average team score} = \begin{cases} 300 * \frac{total\ runs\ scored}{total\ balls\ faced} + extras; total\ balls\ faced > 300 \\ runs\ scored * team\ coefficient + extras; team\ coefficient < 1 \\ total\ runs\ scored + extras \end{cases} \qquad 4\text{-}14$$

$$\text{average target} = \begin{cases} 300 * \frac{total\ runs\ conceded}{total\ deliveries\ bowled}; total\ deliveries > 300 \\ 10 * \frac{total\ runs\ conceded}{total\ wickets\ taken}; total\ wickets\ taken > 10 \\ total\ runs\ conceded \end{cases} \qquad 4\text{-}15$$

$$\text{average balls faced} = \begin{cases} 300; total\ balls\ faced * team\ coefficient > 300 \\ total\ balls\ faced * team\ coefficient \end{cases} \qquad 4\text{-}16$$

$$\text{average wickets fallen} = \begin{cases} 10; total\ balls\ faced < 300 \\ \frac{10 * 300}{total\ balls\ faced} \end{cases} \qquad 4\text{-}17$$

4.2.7. Optimum Team Combination Module

After we have built: the player performance prediction model, player rating prediction model, and team performance calculation module, the next step is to combine the players to get the optimum total batting score and combine bowlers to concede the optimum number of runs opposition. A combination with a higher winning margin or a minimum losing margin would be the best possible team prediction. In combining the batsman and bowlers, there are a few constraints that we have to consider. One is the requirement of having at least one wicket-keeper in the playing eleven. Also, we need to have a minimum of five bowlers in the team. Having all-rounders will be an advantage to the team. In the early days of cricket, a player who could both bat and bowl were considered an all-rounder. However, an all-rounder should perform well in modern competitive cricket, similar to a specialised bowler and specialised batsman. Having such a player in the team would be a great advantage to the team as they can make room for another player, allowing the team to have more skills.

Taking the above-discussed constraints into consideration, first, we select a wicket-keeper from the pool with the highest performance rating. Secondly, we select five bowlers with the highest performance rating from the pool. Then we select the remaining five players from the rest of the pool. These last five players may consist of only specialised batsmen, all-rounders, or bowlers, depending on the players' corresponding player rating.

4.3.System Architecture

As we have discussed in detail in 4.2, we have four main components in our system.

1. Individual Player Performance Modules (Batting, Bowling and Fielding)
2. Player Rating Module
3. Team Performance Calculation Module
4. Optimum Team Combination Module

Once we have these four modules combined, the overall system will operate, as shown in *Figure 29*.

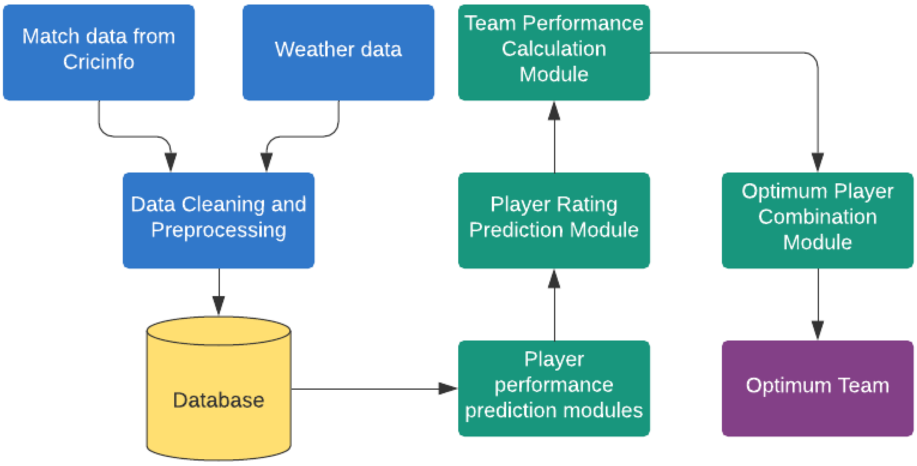


Figure 29: High-Level Architecture of Overall System

4.4.Methodology and Evaluation Plan

Based on our aims and objectives, the following diagram depicts the overall research methodology of our study and the evaluation milestones.

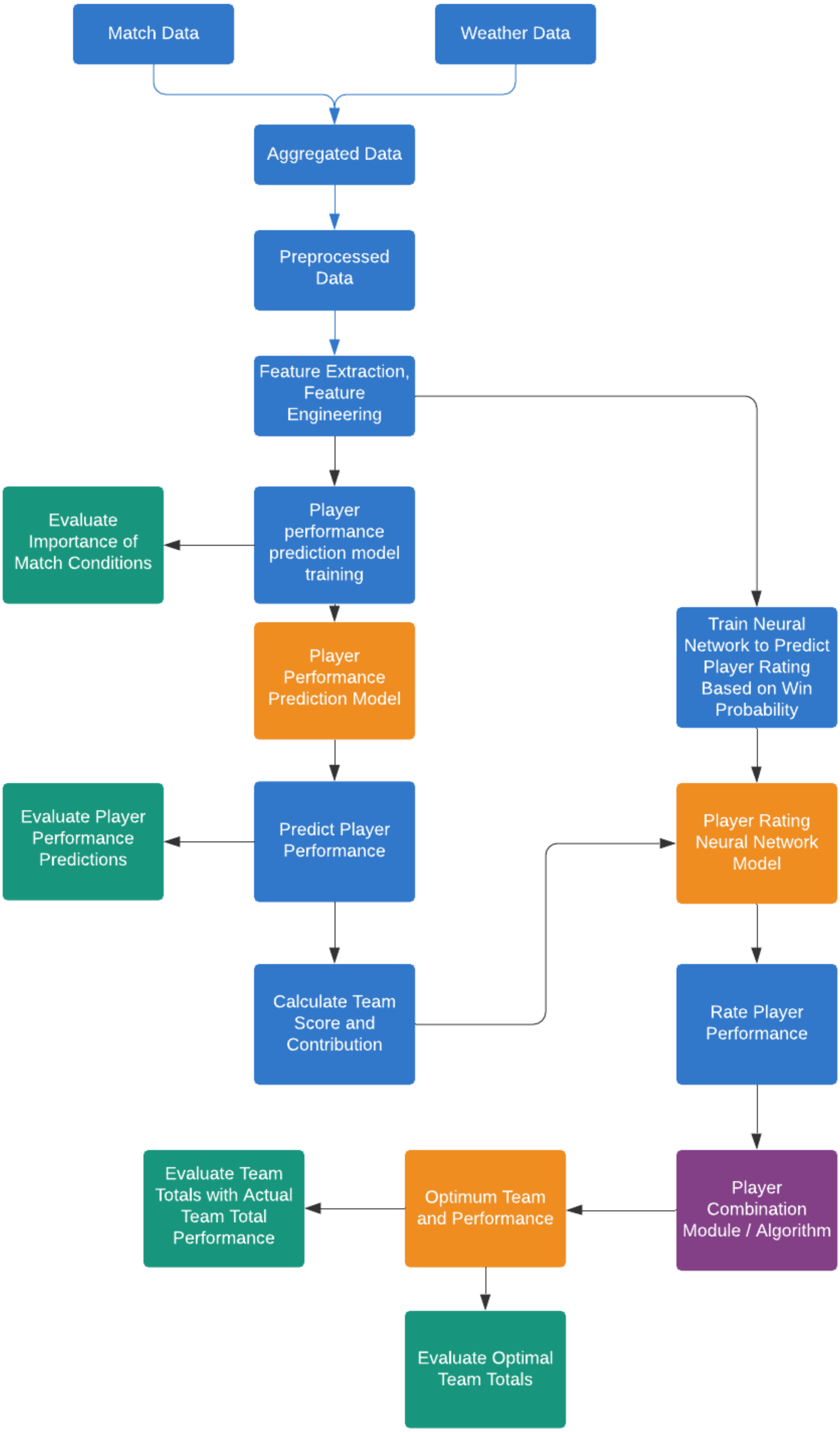


Figure 30: Overall Research Methodology and Evaluation Milestones

4.5. Summary

In this chapter, we have extensively described the research approach and implementation of the system. We have discussed data storage, pre-processing, and sub-components implementation starting from the data collection step. Then we discussed the final combination of different components to achieve our final system. Towards the latter of this chapter, we discussed the overall system architecture and the overall research methodology. The next chapter will discuss the study results, the detailed evaluation plan, and the system's performance we built using the methodology explained in this chapter.

CHAPTER 5

5. EVALUATION AND RESULTS

5.1. Introduction

This chapter discusses the result of the study in different sections. Each section will discuss how we set up the experimental design to test and evaluate each system module. Next, we will evaluate the results of each module and, finally, the system's overall performance. We used a test dataset consisting of 45 matches played during 2017-2019, which was not used for training any of the Machine Learning Models for overall evaluations.

5.2. Importance of Match Conditions and Player Performance Prediction

Figure 23 and Figure 24 in the previous chapter show that the initial prediction models we trained to predict players' performance had significant bias errors. Therefore, we trained a secondary model to predict the bias error and minimize bias by combining the two models. We will proceed with evaluating the accuracy of each performance prediction module. We used the feature importance technique and probability value (p-value) to identify the most significant features used as input features. We decided to use a significance level of 0.05 for the p-value as it is the generally accepted significance level for any dataset.

5.2.1. Batting Performance Prediction Model

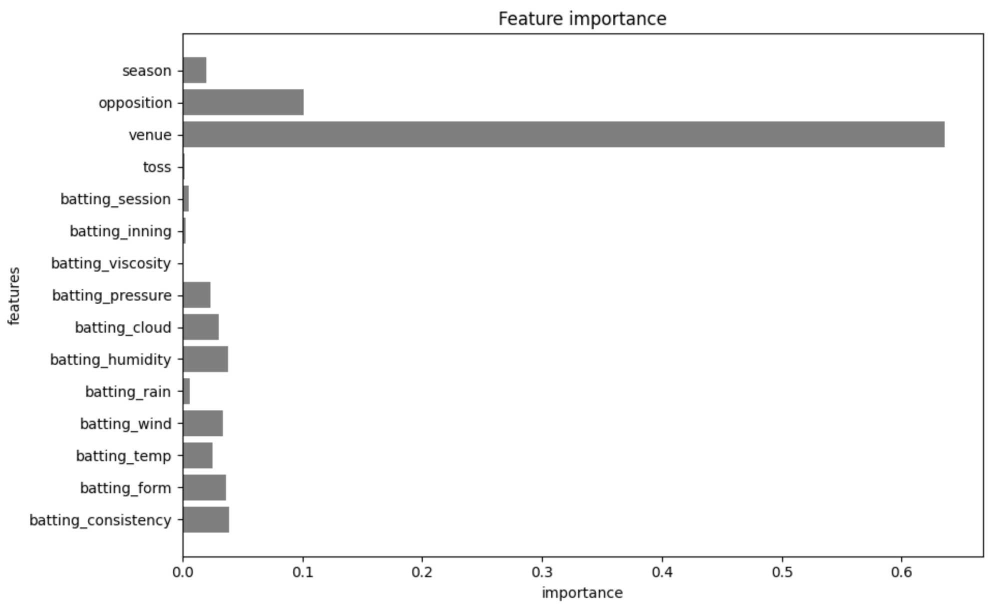


Figure 31: Feature Importance of Batting Performance Prediction

As shown in *Figure 31,* our batting performance model identifies venue and opposition as two significant factors for predicting a player's batting performance. As we tested with varying the number of input attributes, we observed that, even though the significance of weather attributes was relatively less significant than that of venue and opposition, the weather attributes still helped improve the accuracy of the model. To identify the significance of these attributes, we evaluated the input attributes p-values. *Table 8* shows the input attributes selected using the p-value for each output performance attribute.

Table 8: Selected features for Predicting Batting Performance

Output Attribute	**Selected Features**
Runs Scored	batting humidity, venue, opposition, season
Balls Faced	batting consistency, batting humidity, venue, opposition
Fours Scored	batting humidity, batting inning, venue, opposition, season
Sixes Scored	batting consistency, batting pressure, venue, opposition, season
Batting Position	batting consistency, batting form, batting inning, venue, opposition, season

As shown in *Table 8,* according to the feature selection, we can observe that humidity condition is considered a significant factor for predicting the batting performance of the players. Based on the selected input attributes, we graphed the predicted batting performances for each output attribute.

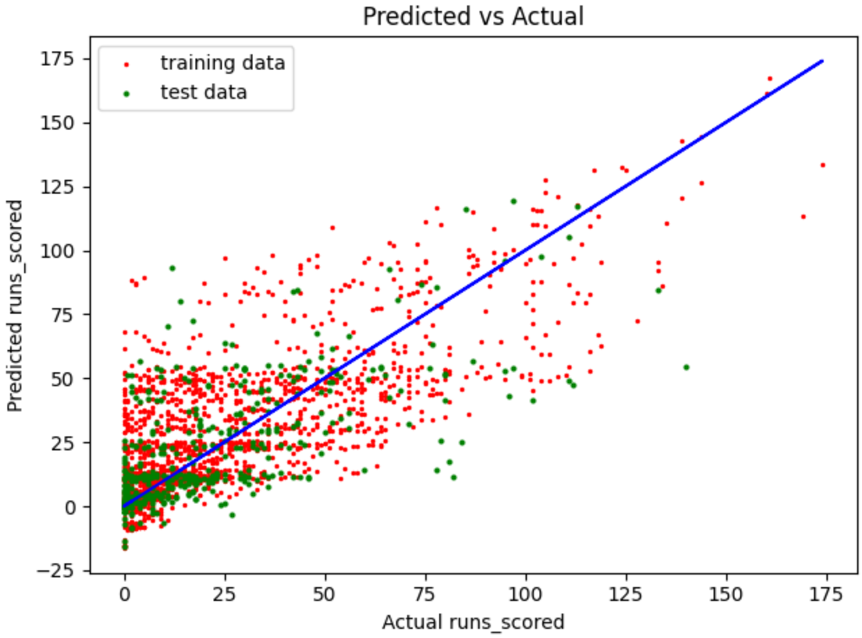


Figure 32: Predicted Runs Scored vs Actual Runs Scored

Figure 32 shows the predicted runs scored by the model we have trained. As we can observe, we have been able to improve the bias of the predictions significantly. The model had an RMSE value of 19.46 and an R^2 value of 0.40 against the test dataset. While 0.40 seems a low R^2 value for a prediction model given the unpredictive nature of cricket and human behaviour, this is acceptable. Falk and Miller (1992) [19] has recommended that an R^2 value greater than 0.10 is enough for soft model predictions. Therefore, we decided to use this model to predict the runs scored by a batsman to incorporate with the rest of the modules for optimum team selection.

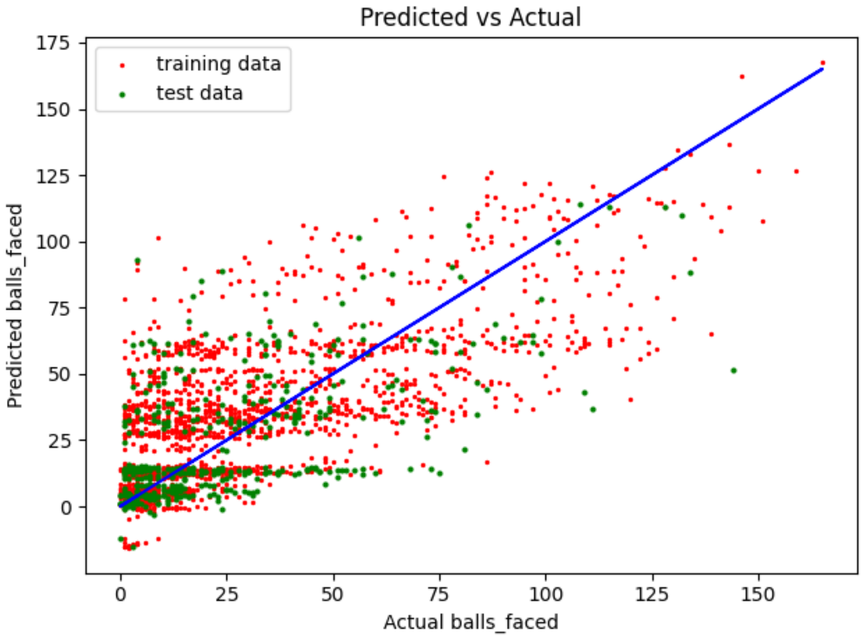


Figure 33: Predicted No. of Balls Faced vs Actual No. of Balls Faced

Figure 33 shows the predicted number of balls faced by a batsman. Like the predicted runs scored model, the ball faced model also has significantly improved bias and showed an RMSE value of 22.54 and an R^2 value of 0.26. Given that we can reasonably predict the number of balls a batsman faces, we selected this prediction model for optimum team selection.

Figure 34, *Figure 35* and *Figure 36* show the prediction results of the number of fours, sixes scored by batsmen and the batting position. We can observe that the prediction model has not improved significantly with the bias correction, and the bias is still present with the predictions for test data. Also, as discussed in the player rating module evaluation, these three features were declared not required based on the feature selection for our player rating ANN. Therefore, we excluded these three models from the player rating model and optimum team selection process.

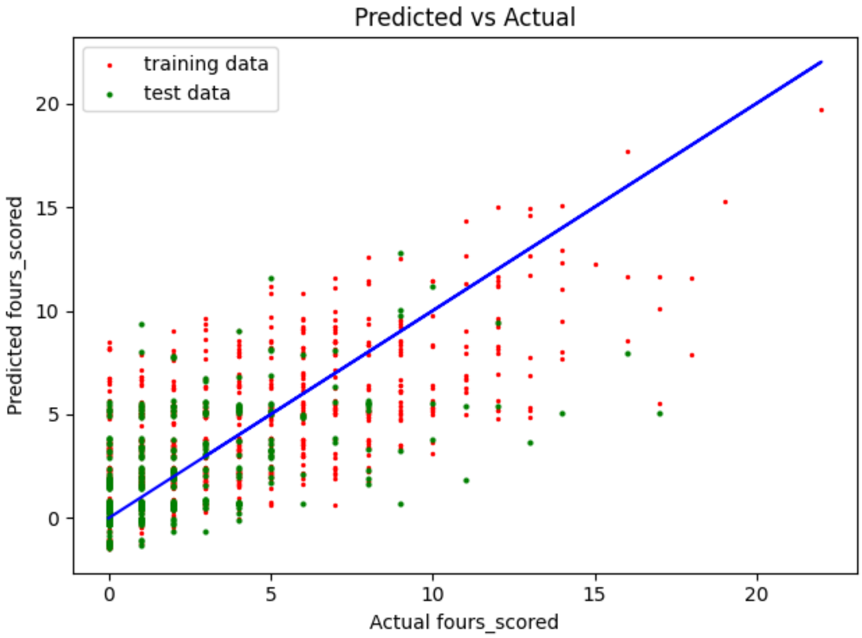


Figure 34: Predicted No. of Fours Scored vs Actual No. of Fours Scored

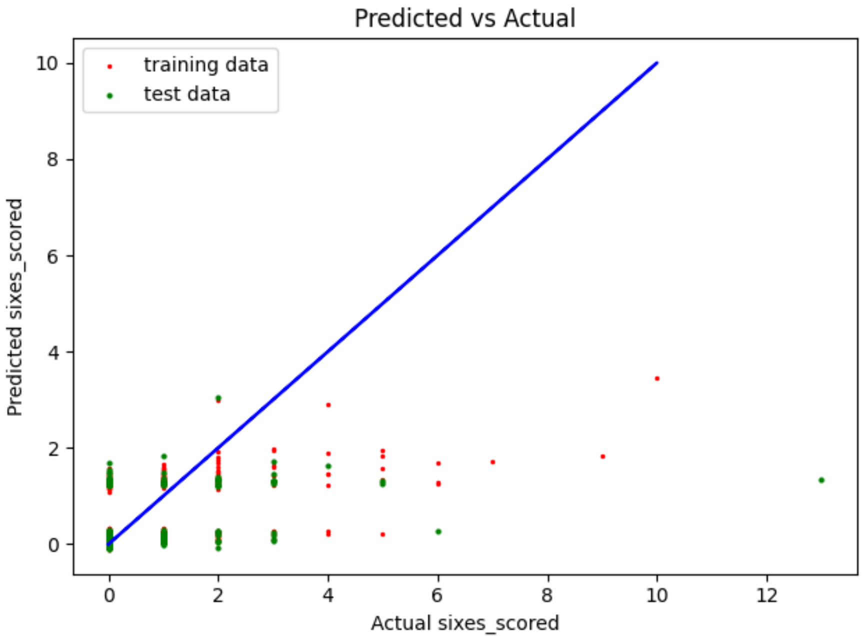


Figure 35: Predicted No. of Sixes Scored vs Actual No. of Sixes Scored

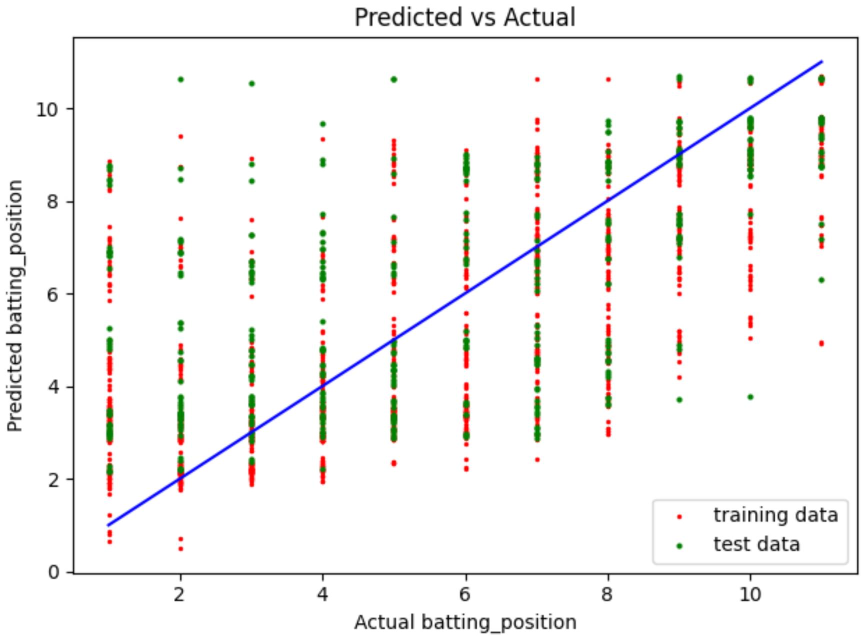


Figure 36: Predicted Batting Position vs Actual Batting Position

Table 9 summarises the RMSE and R^2 values for batting performance prediction modules.

Table 9: Evaluation Summary of Batting Performance Prediction Module

Output Attribute	**RMSE**	**R^2**	**Selected for Player Rating**
Runs Scored	19.46	0.40	Yes
Balls Faced	22.54	0.26	Yes
Fours Scored	2.34	0.25	No
Sixes Scored	0.95	0.13	No
Batting Position	2.59	0.27	No

5.2.2. Bowling Performance Prediction Model

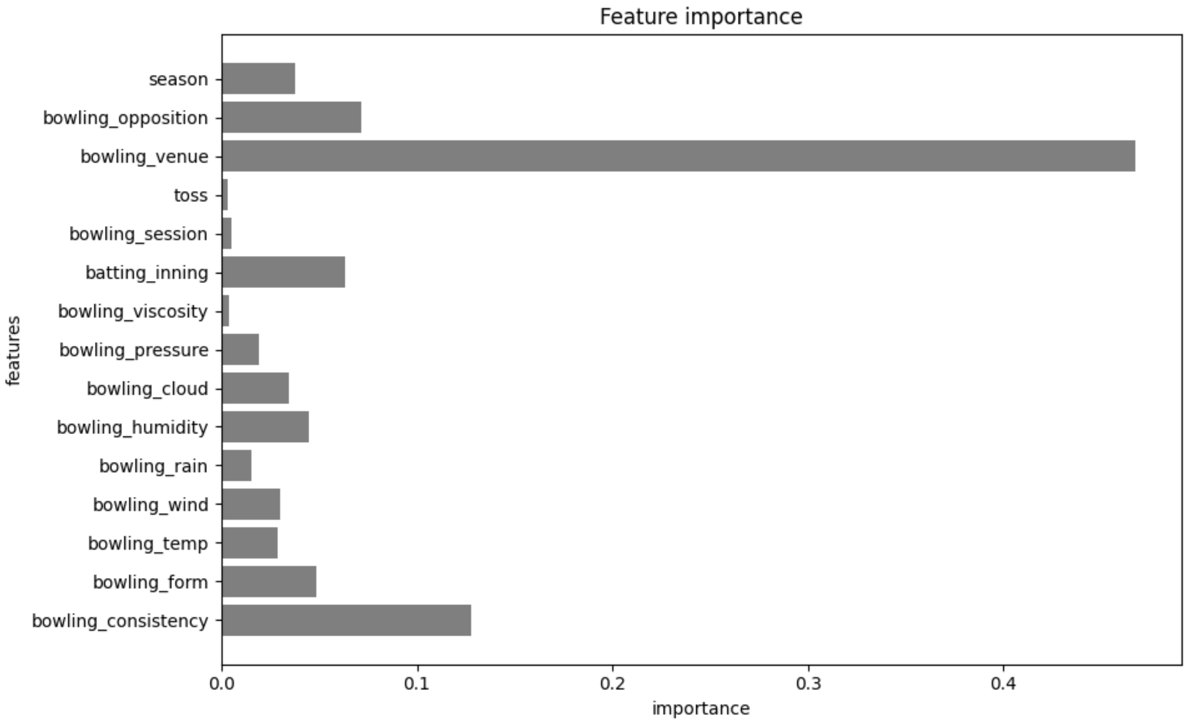


Figure 37: Feature Importance of Bowling Performance Prediction

Figure 37 shows the feature importance for bowling performance predictions. Similar to the batting performance prediction model, venue and opposition are identified as significant features for predicting bowling performance. Also, we can observe that bowling consistency and bowling inning (inverse of batting inning) are also given relatively higher importance than in the batting performance prediction models. We calculated the p-values for the above input features to eliminate any insignificant features from the model. *Table 10* shows the selected features for predicting each bowling performance attribute. As shown in the selected feature list, the humidity factor seems to be significant for bowling performance as well as for batting performance.

Table 10: Selected features for Predicting Bowling Performance

Output Attribute	**Selected Features**
Runs Conceded	bowling temp, bowling rain, bowling humidity, batting inning, bowling venue, bowling opposition, season
No. of Deliveries Bowled	bowling form, bowling rain, batting inning, bowling venue, bowling opposition, season
Wickets Taken	bowling consistency, bowling temp, bowling humidity, batting inning, bowling venue, bowling opposition, season

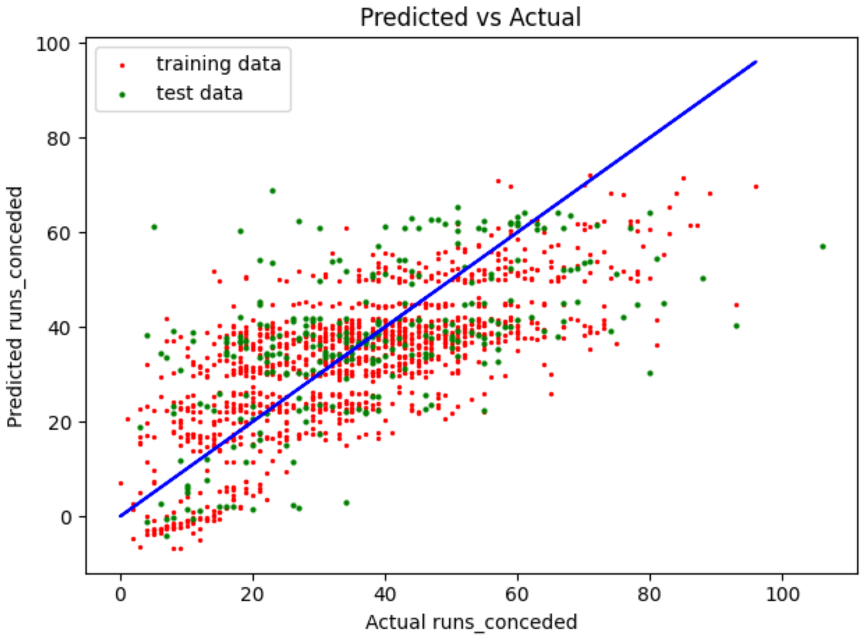


Figure 38: Predicted Runs Conceded vs Actual Runs Conceded

Figure 38 shows the performance of the runs conceded prediction module. The model has improved adequately with the bias correction and showed an R^2 value of 0.28. *Figure 39*, on the other hand, shows the performance of the number of deliveries prediction module. While the number of balls bowled by a bowler can be controlled, the idea of implementing this module was to identify full-time bowlers and part-time bowlers. This model helps our system to identify batsmen, bowlers, all-rounders and part-time bowlers. Full-time bowlers will bowl a higher number of overs while others bowl a relatively lesser number of overs.

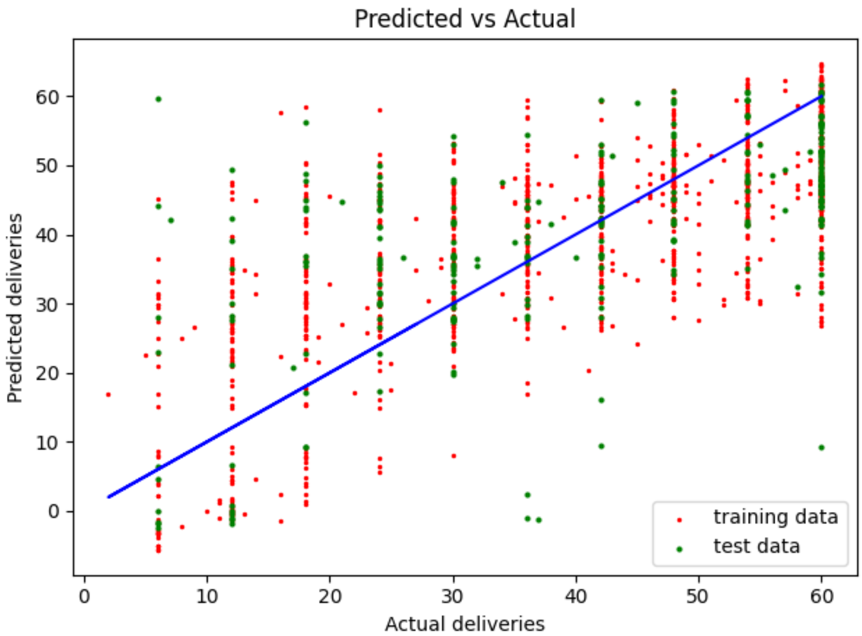


Figure 39: Predicted No. of Deliveries Bowled vs Actual No. of Deliveries Bowled

The bias correction module has not been able to correct the bias in this module adequately. Still, we decided to use this module to identify the average impact of bowlers, as it helps our player rating module to categorize bowlers and identify full-time bowlers.

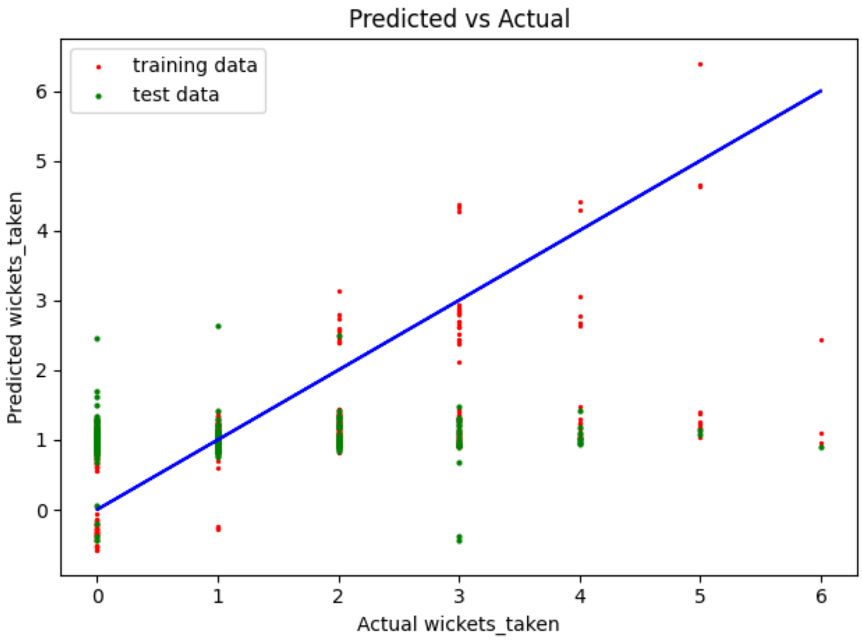


Figure 40: Predicted No. of Wickets Taken vs Actual No. of Wickets Taken

We also tried to predict the number of wickets taken by a bowler using a No. of Wickets prediction module. *Figure 40* shows the prediction performance of the model we built. It can be observed that the model does not perform well and has a significant bias error even after we have implemented a bias correction component for the module. Therefore, given the R^2 value of -0.03, which is way below the minimum recommended value of 0.10 as we have considered for other models, we decided not to use this model for the player rating system. *Table 11* summarizes the performance evaluation of the bowling performance prediction module.

Table 11: Evaluation Summary of Bowling Performance Prediction Module

Output Attribute	**RMSE**	**R^2**	**Selected for Player Rating**
Runs Conceded	16.16	0.28	Yes
No. of Deliveries Bowled	14.12	0.30	Yes
Wickets Taken	1.16	-0.03	No

5.2.3. Fielding Performance Prediction Model

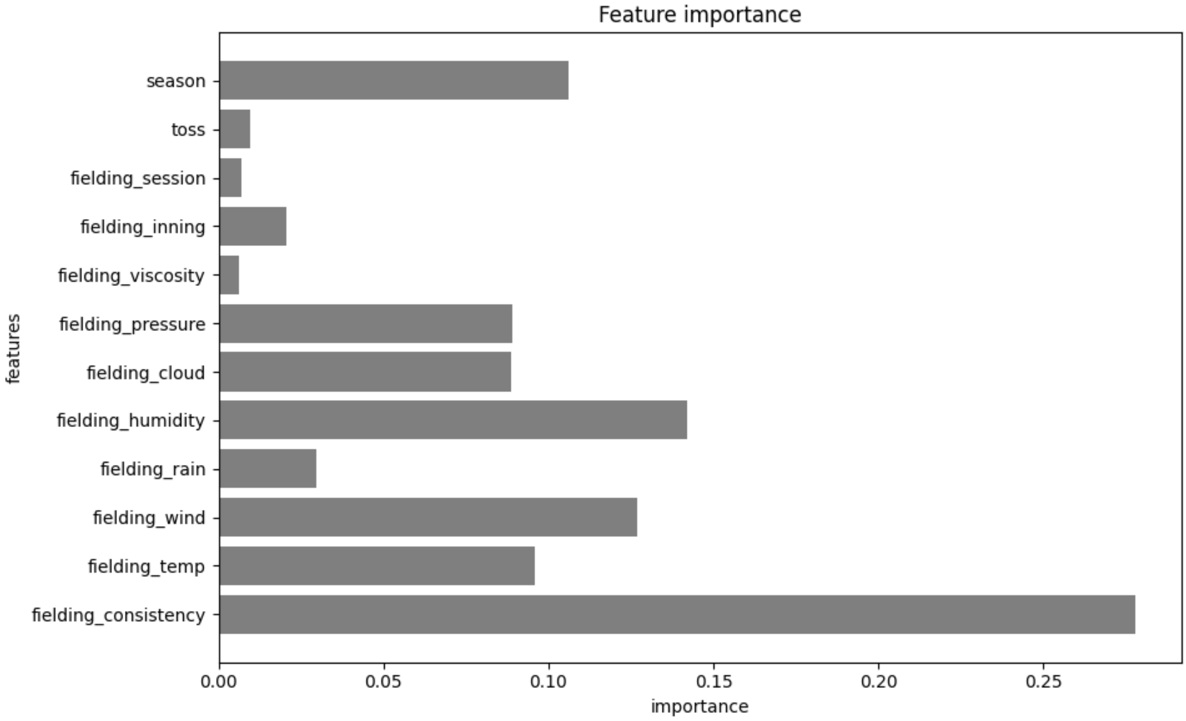


Figure 41: Feature Importance of Fielding Performance Prediction

We trained a fielding success rate prediction system to predict the fielding performance of players. *Figure 41* shows the feature importance of the considered input attributes. *Table 12* shows the selected features using the p-value calculation. But the fielding performance predictions were not accurate using the given set of features. Therefore, we concluded that the fielding performance of a player is not significantly impacted by weather or any other match conditions. But only on the player's fielding skills.

Table 12: Selected features for Predicting Fielding Performance

Output Attribute	**Selected Features**
Success Rate	Fielding consistency

Furthermore, based on the observations from Figure 42, our model could not predict the fielding performance of the players with adequate accuracy. The R^2 value was -0.12. Therefore, we discarded this model from the player rating system.

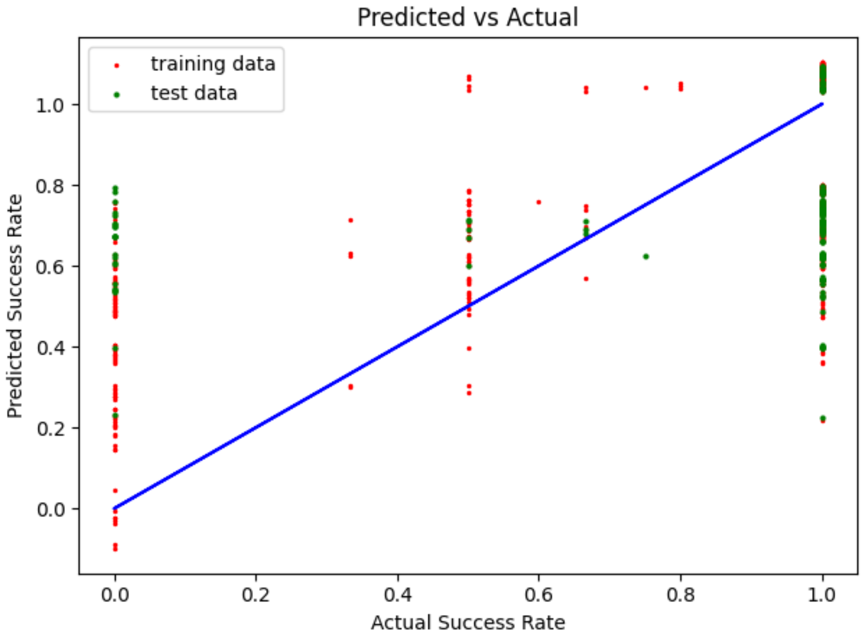


Figure 42: Predicted Fielding Success Rate vs Actual Fielding Success Rate

Table 13 summarizes the performance evaluation of the fielding performance prediction module.

Table 13: Evaluation Summary of Fielding Performance Prediction Module

Output Attribute	**RMSE**	**R^2**	**Selected for Player Rating**
Success Rate	4.89	-0.12	No

5.3.Player Rating Prediction

After evaluating the player performance prediction modules, we will evaluate the designed ANN-based system for rating the players based on the predicted performance. Our motive was to rate the players based on their contribution to the team towards winning. Rather than rating the players based on performance attributes such as runs scored, no. of wickets taken, runs conceded. Think of a system that can predict the result of a match based only on the performance of a single player. Our proposed system can identify players' performance contributions and rate the players with significant accuracy. In short, our system can predict the probability that the team can win a particular match, given the performance of a single player. We used the match result as the labelled data to train this supervised learning model. Initially, we considered 47 input variables, filtered out insignificant attributes using the p-value significance level, and narrowed down the feature list to 25 input attributes. We use the probability of winning predicted from the system as the rating of the players.

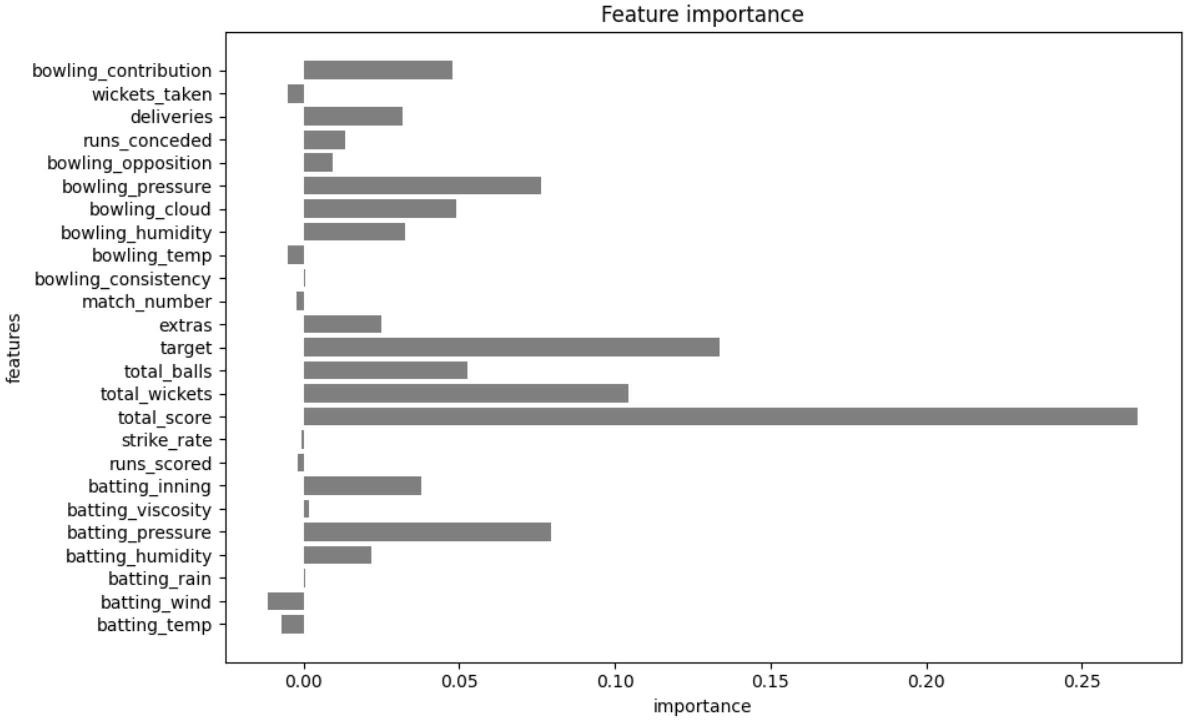


Figure 43: Selected 25 Input Attributes and Feature Importance for Player Rating

Figure 43 shows the feature importance of the features we used to train the ANN. As we expected, the total score of the team and the total runs conceded to the bowling team are significant factors, along with the total number of wickets that have fallen while the team is batting. *Table 14* shows the evaluation summary of our model with 10-fold cross-validation and test dataset. The confusion matrix of our player rating model's performance on the individual performance of players in 45 matches in the test dataset is shown in *Table 15*. The model performed with an accuracy of 85.39% on the test dataset. According to the results, our player rating model works well in predicting the result of the match based on the player performance.

Table 14: Player Rating ANN Evaluation Summary

10-Fold Cross-Validation Score (Average)	82.52%
Accuracy	85.39%
Classification Error	14.60%
Sensitivity	69.38%
Specificity	92.33%
False Positive Rate	7.66%
Precision	79.68%

In a team of 11 players, we cannot guarantee that all the players have performed well and contributed generously to the match's final result. That is why we have included the total score and target attributes for the system to get an overall idea of the team's performance when rating the player's performance. So, each players performance will be evaluated and rated relative to the overall performance of the team.

Table 15: Confusion Matrix of Player Rating Model

	Loss	Win
Loss	313	26
Win	45	102

Table 16 shows the cross-validation accuracy for 10 folds.

Table 16: 10-Fold Cross-Validation Results for Player Rating ANN

Fold	**Accuracy**
1	0.73540856
2	0.85992218
3	0.80933852
4	0.84824903
5	0.7890625
6	0.859375
7	0.8984375
8	0.87890625
9	0.73046875
10	0.84375

5.4.Team Performance Prediction and Optimum Team Selection

After the player performance prediction models and the player rating models are complete. We combined these two prediction models and came up with an algorithm to calculate the overall performance of the team based on the predictions made using the previous two modules; the player performance prediction module and player rating module. There are a few input attributes, such as total score and target in the player rating module; we calculated the average team performance and fed that as the input attributes for the player rating module. Then once we get the player ratings predicted, we recalculate the overall team performance for the 11 players selected based on the highest rating.

The list of input attributes to the neural network and their derivations for the experimental setup are shown in *Table 17*.

Table 17: Input Attributes and Their Source / Derivations for Experimental Player Rating Predictions

No.	Attribute	Source / Derivation
1	batting temp	Batting Weather Dataset
2	batting wind	Batting Weather Dataset
3	batting rain	Batting Weather Dataset
4	batting humidity	Batting Weather Dataset
5	batting pressure	Batting Weather Dataset
6	batting viscosity	Batting Weather Dataset
7	batting inning	Match Details Dataset
8	runs scored	Batting Performance Prediction
9	strike rate	Batting Performance Prediction
10	total score	Team performance calculation module
11	total wickets	Team performance calculation module
12	total balls	Team performance calculation module
13	target	Team performance calculation module
14	extras	Team performance calculation module
15	match number	Match Details Dataset
16	bowling consistency	4-2
17	bowling temp	4-4
18	bowling humidity	Bowling Weather Dataset
19	bowling cloud	Bowling Weather Dataset
20	bowling pressure	Bowling Weather Dataset
21	bowling opposition	4-6
22	runs conceded	Bowling Performance Prediction
23	deliveries	Bowling Performance Prediction
24	wickets taken	Bowling Performance Prediction
25	bowling contribution	Runs conceded / target

Our system predicted the performance of the 11 players who played in the 45 matches in the test dataset. At the same time, we selected the optimal 11 players for each match using our player rating system and predicted the optimal team's performance. As discussed in the bowling performance prediction module, we could not build a good prediction model for the number of wickets taken by a bowler. Therefore, we replace that value with each bowler's average number of wickets based on venue and opposition. *Figure 44* shows the predicted team total against the actual team total for the 45 matches in the test dataset. The team total prediction for the overall system had an RMSE value of 52.55 and an R^2 value of 0.42. This R^2 value is approximately identical to the R^2 value of the runs scored prediction module. Therefore, we can assume that by improving the performance of the batting performance prediction module, we can improve the accuracy of the team total prediction module.

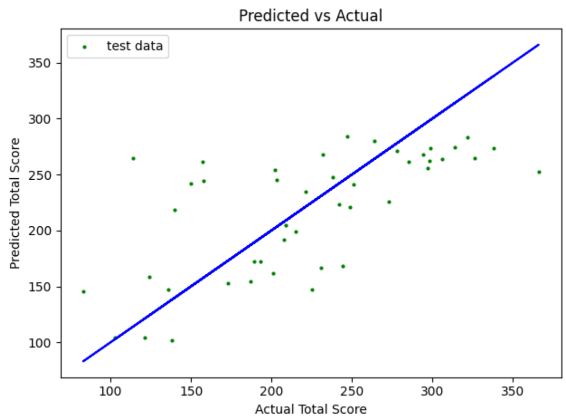


Figure 44: Predicted vs Actual Total Score for Test Dataset

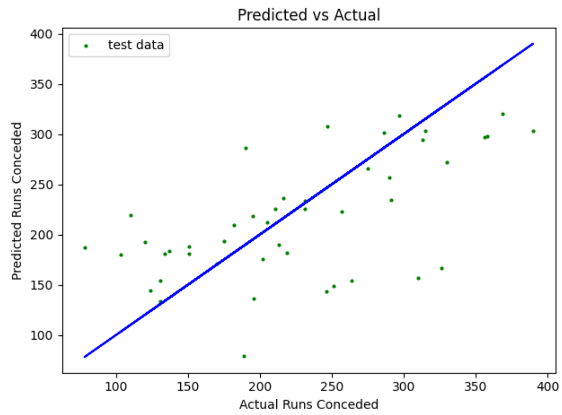


Figure 45: Predicted vs Actual Total Runs Conceded for Test Dataset

Figure 45 shows the plot diagram of predicted total runs conceded against actual total runs conceded by the bowler for the 45 matches in the test dataset. The total runs conceded prediction for the overall system had an RMSE value of 64.18 and an R^2 value of 0.33. This R^2 value is approximately identical to the R^2 value of the runs conceded prediction module. Therefore, we can assume that by improving the performance of the bowling performance prediction module, we can improve the accuracy of the total runs conceded prediction module. Overall, this will improve the accuracy of team performance prediction modules. Hence the accuracy of match result predictions from the system.

Figure 46 shows the actual runs scored by the team, the predicted score, and the score by the optimal team predicted using our system. We can observe that our system's optimal team generally tends to score more runs than the actual teams.

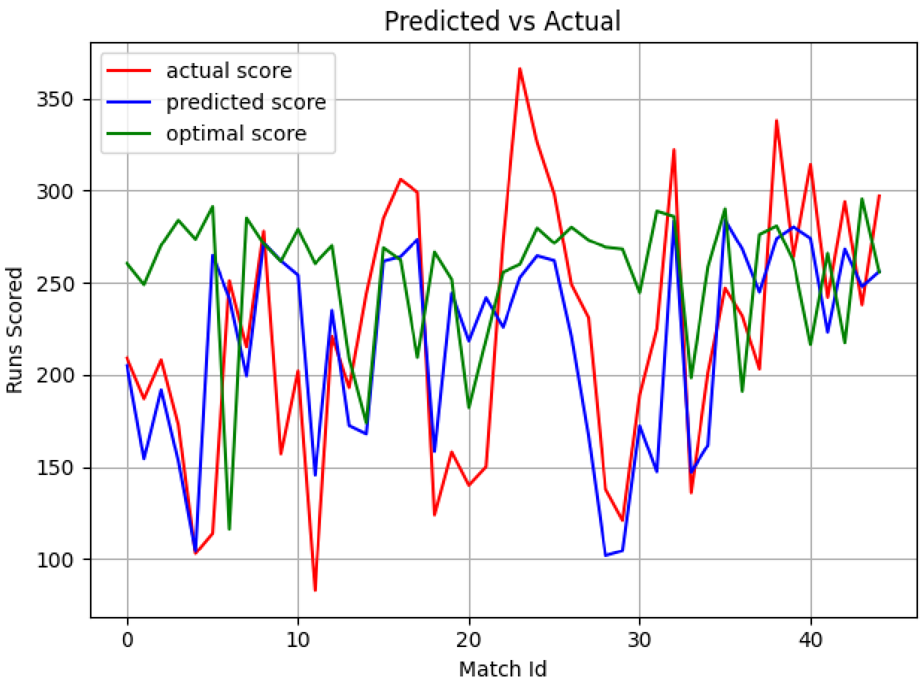


Figure 46: Predicted and Optimal Scores vs Actual Scores

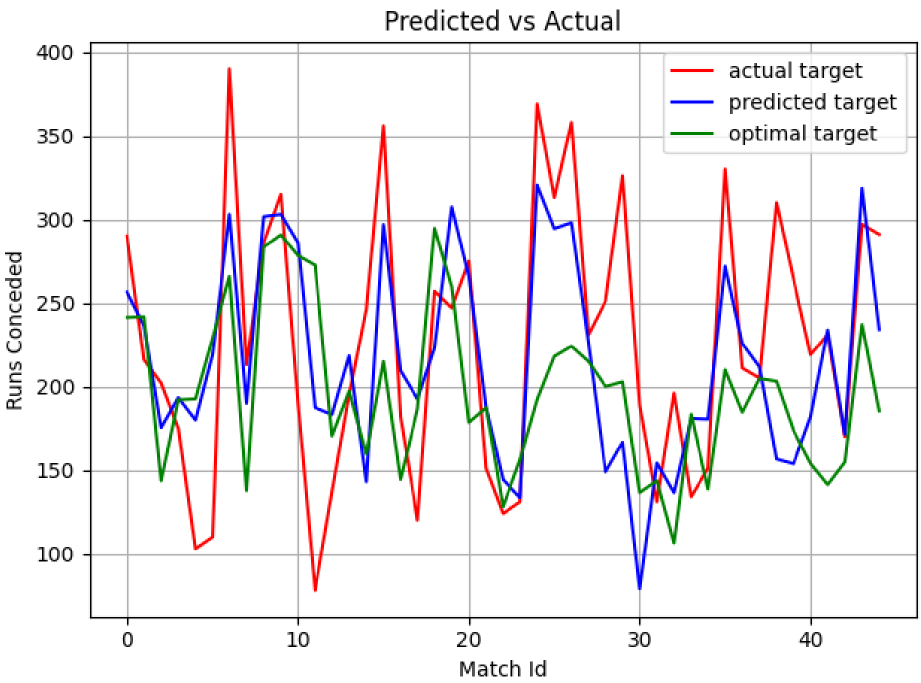


Figure 47: Predicted and Optimal Runs Conceded vs Actual Runs Conceded

Figure 47 shows the total runs conceded by the team to the opposition. From here, we can observe that the optimal team selected using our system tend to concede a lesser number of runs to the opposition team; scoring more runs while batting and conceding fewer runs while bowling should be the aim of every team in cricket for winning a match. Our optimal team selection system can deliver an optimal team to satisfy those requirements. *Figure 48* shows the plot of the winning margin for the actual and optimal team of 11 players. Predicted performance shows that the optimal team predicted using our system has a higher winning margin.

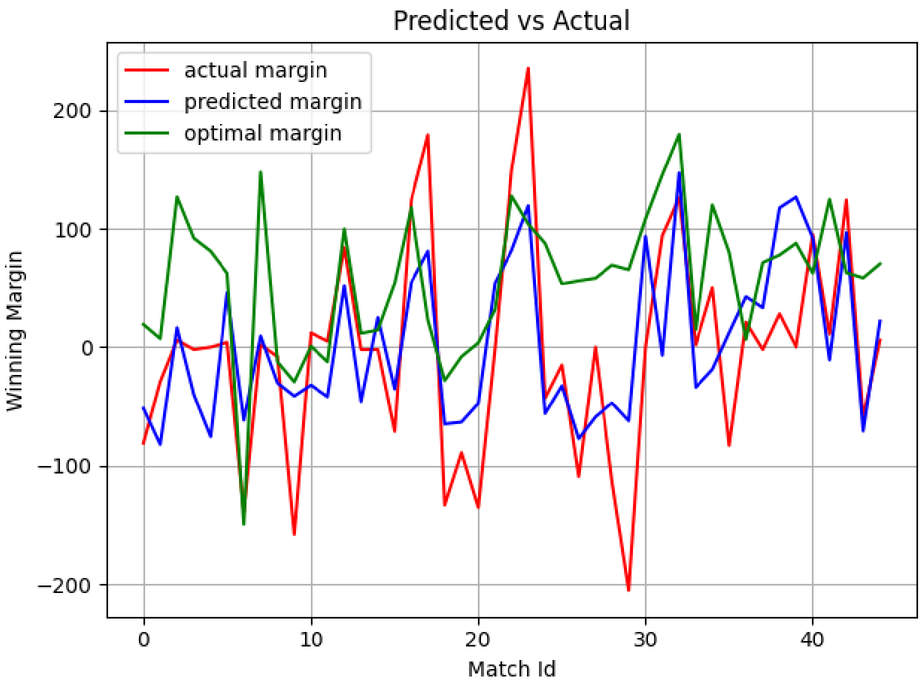


Figure 48: Predicted and Optimal Winning Margins vs Actual Winning Margins

Based on the winning margin, we predicted the result of the 45 matches in the test dataset. If the total runs scored is greater than the total number of runs conceded, we predict it as the team can win the match. Based on our prediction modules and calculations, we correctly predicted the match result of 34 matches out of 45 with an accuracy of 75.55%. According to the test dataset, Sri Lanka has won 14 matches out of 45 matches played. As predicted from our system, Sri Lanka should have won 17 matches out of the 45 matches. Based on our predicted optimal teams and predicted results, it seems that Sri Lanka could have won 35 matches if the 11 players were selected according to our proposed model. Based on the predicted results, the team's winning rate can be improved from 37.77% to 77.77% (105% improvement) if teams were selected using our proposed system.

Table 18 shows the comparison of actual and predicted results for the 45 matches in the test dataset.

Table 18: Test Dataset Match Results with Predicted and Optimal Team Results

No	Match Id	Result	Predicted Result	Optimal Result
0	1120286	Lost	Lose	Win
1	1120287	Lost	Lose	Win
2	1120288	Lost	Lose	Win
3	1120289	Lost	Lose	Win
4	1120290	Lost	Win	Win
5	1122726	Won	Lose	Lose
6	1122727	Lost	Lose	Win
7	1122728	Lost	Lose	Lose
8	1130738	Lost	Lose	Lose
9	1130739	Lost	Lose	Lose
10	1130740	Won	Lose	Lose
11	1130742	Won	Win	Win
12	1130743	Won	Lose	Win
13	1142584	Lost	Win	Lose
14	1142585	Lost	Lose	Win
15	1142586	Lost	Win	Win
16	1142587	Won	Win	Win
17	1142588	Won	Lose	Lose
18	1153243	Lost	Lose	Lose
19	1153245	Lost	Lose	Lose
20	1140380	Lost	Win	Win
21	1140381	Lost	Win	Win
22	1140382	Lost	Win	Win
23	1140383	Won	Lose	Win
24	1153840	Lost	Lose	Win
25	1153841	Lost	Lose	Win
26	1153842	Lost	Lose	Win
27	1144167	Lost	Lose	Win
28	1144168	Lost	Lose	Win
29	1144169	Lost	Win	Win
30	1144170	Lost	Lose	Win
31	1144171	Lost	Win	Win
32	1169332	Won	Lose	Win
33	1144485	Lost	Lose	Win
34	1144489	Won	Lose	Win
35	1144502	Lost	Win	Win
36	1144509	Won	Win	Win
37	1144517	Lost	Win	Win
38	1144521	Won	Win	Win
39	1144526	Lost	Win	Win
40	1193504	Won	Lose	Win
41	1193505	Won	Win	Win
42	1193506	Won	Lose	Win
43	1198487	Lost	Win	Win
44	1198488	Lost	Lose	Win

As we can observe from the above table, most of the predicted results are accurate and the optimal teams selected using the proposed method improve the probability of winning of the team. However, the match numbers 1130740 and 1142588 predicts incorrectly as the team loses when the actual result is a win. When we looked into the details, we were able to observe that the deviation was due to the fact that some players have performed significantly well when batting to score higher runs. Also, bowlers have conceded fewer runs and taken more wickets than predicted in the actual scenario.

5.5.Summary

This chapter discussed the experimental setup we designed for each module of the system and the evaluation and results of every module. In the end, we discussed the performance of the overall system. We compared the predicted results with actual results to evaluate the improvements made to team selection by using our proposed system.

CHAPTER 6

6. CONCLUSION AND FUTURE WORK

6.1. Introduction

This chapter will discuss the final conclusion that we can derive based on the research results. Then we will discuss the remarks and observations of the overall study. Later, we will verify how we achieved each research objective, the limitations of the research and future work that we can conduct to improve this team selection system further.

6.2. Overall Conclusion

In this study, we used Random Forest Regression to predict the player performance based on weather and other match conditions. Also, based on our evaluation, we identified humidity as a consistently significant factor affecting the performance of bowlers and batsmen. Then we used a Neural Network to select the optimum 11 players to win matches under given conditions. The results showed that we could improve the winning probability of the Sri Lankan team for ODI matches by an exceptional amount. These results can be further extended towards the prediction of optimum teams for Test Cricket and T20 Internationals.

While our team prediction Neural Network had an accuracy above 80%, the player performance prediction using Random Forest Regression had an accuracy level below 50%; Thereby limiting the overall performance of the team selection system. By improving the player performance prediction modules, we can perform more accurate team selection using this system. Future studies can be conducted to derive more attributes that would help predict player performance more accurately, thus improving the overall accuracy of the optimum team selection system.

6.3. Achievement of Objectives

We did a critical review of previous studies on analysing and predicting player performance based on match conditions. Then we did a critical review on research that proposed methods to select players to form a team maximizing the team performance in different sports. We developed web scrapers to collect match details and player performance from the website https://www.espncricinfo.com/. We manually collected weather data corresponding to each match venue from https://www.worldweatheronline.com/. Using relational database technologies and Python coding, we built relationships between the player performance data, match results and weather conditions to analyse how player performance gets affected by different match conditions. Using the past player performance data, we trained Random Forest Regression models to predict each player's batting, bowling, and fielding performance for different match conditions. Also, we trained a neural network to predict the team's winning probability by combining different players to select the optimum team. Then we proposed a method to combine the predicted player performance and predicted winning probability to select the optimal to play under given match conditions. Finally, we evaluated the accuracy of our prediction models and compared the predicted results with actual results and player performance for the matches played by the Sri Lanka team during 2017-2019.

6.4.Limitations and Future Work

We used the ODI matches played by the Sri Lankan cricket team during the 2010-2019 time period combined with weather data for each match for our study. While the web scrapers can automatically scrap match data from the website, we can extend the number of match data we can consider for training our models. However, the limitation of manually collecting the weather data for each match forced us to limit our dataset only to the above-mentioned time limit. Furthermore, the following limitations were identified in our study.

- **Runs saved by each fielder –** In our study, we could not collect data on how each player contributed to the team by saving runs while fielding. It would be an important factor in evaluating the player’s fielding performance. It would take a more in-depth study to analyse each match records to extract this level of data from the matches and would consume much time.
- **Match result prediction errors -** Our system could predict the match result with an approximate accuracy of 75% based on the test dataset. In predicting the match results, we have not considered the following factors, which we believe would improve the accuracy of the match prediction module if included in match prediction.
  - **Batting order –** In our study, we attempted to predict each team player with the most suitable batting position for each player. While predicting the match results for the actual 11 players, we predicted the result based on the batting position assigned to each player from the performance prediction module. This might affect the final prediction since the considered batting order of the team might not be the same for the actual team.
  - **Bowling order –** In evaluating and predicting the bowling performance of the players, we did not consider how the bowlers were combined/ should be combined to ball the 50 overs in the match. (Bowling at the start of the inning/

middle of the inning/ end of the inning). This data would help identify which bowlers perform better in the early of the inning/ middle of the inning and towards the end of the inning.

- **Fielding setup –** we did not consider the fielding position of players in predicting/ evaluating the fielding performance of players or predicting the outcome of the match. Analysing fielding performance by fielding position would help to evaluate the fielding performance in more depth.
- **Player injuries –** we did not collect data on players who had to leave the field in the middle of their performance due to injuries or other factors. Therefore, if a batsman or a bowler had to leave the field due to an injury, the system will identify the interrupted performance of the player as the complete performance for the match. This creates an error in evaluating and predicting the performance of players.
- **Players changes in the opposition team –** we did not consider the players of the opposition team and their performance under the given conditions in predicting the match outcome. By considering each player in the opposition team and their predicted performance under the match conditions, we will be able to predict the playing of 11 players of the opposition team. Based on the opposition playing 11, the team selection can be more refined to give accurate results. If both teams use a similar approach in team selection, it can be expected that the competitiveness of the match will also be improved since two optimum teams are playing against each other.
- **Byes, leg byes not considered –** we did not consider byes/ leg byes scored by the batting team when predicting the match outcome. Including byes and leg byes in predicting the match outcome would improve the overall accuracy of the match outcome prediction.

6.5. Summary

In this final chapter of the thesis, we concluded the findings of our study and then analyzed the limitations and how the study can be further improved to increase the accuracy of player performance prediction and match outcome prediction.

REFERENCES

[1] R. Al-Shboul, T. Syed, J. Memon, and F. Khan, “Automated Player Selection for a Sports Team using Competitive Neural Networks,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 8, p. 4, 2017.

[2] J. A. Scobie, S. G. Pickering, D. P. Almond, and G. D. Lock, “Fluid dynamics of cricket ball swing,” *Proceedings of the IMechE*, vol. 227, no. 3, pp. 196–208, Sep. 2013, doi: 10.1177/1754337112462320.

[3] D. James, D. C. MacDonald, and J. Hart, “The effect of atmospheric conditions on the swing of a cricket ball,” *Procedia Engineering*, vol. 34, pp. 188–193, 2012, doi: 10.1016/j.proeng.2012.04.033.

[4] D. Bhattacharjee and H. Saikia, “On Performance Measurement of Cricketers and Selecting an Optimum Balanced Team,” *International Journal of Performance Analysis in Sport*, vol. 14, no. 1, pp. 262–275, Apr. 2014, doi: 10.1080/24748668.2014.11868720.

[5] I. P. Wickramasinghe, “Predicting the performance of batsmen in test cricket,” *jhse*, vol. 9, no. 4, pp. 744–751, 2014, doi: 10.14198/jhse.2014.94.01.

[6] M. G. Jhanwar and V. Pudi, “Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach,” p. 10, 2016.

[7] K. Passi and N. Pandey, “Predicting Players’ Performance in One Day International Cricket Matches Using Machine Learning,” in *Computer Science & Information Technology*, Feb. 2018, pp. 111–126. doi: 10.5121/csit.2018.80310.

[8] A. S. Chetan Kapadiya and P. B. Kinjal Adhvaryu, “Intelligent Cricket Team Selection by Predicting Individual Players’ Performance using Efficient Machine Learning Technique,” *IJEAT*, vol. 9, no. 3, pp. 3406–3409, Feb. 2020, doi: 10.35940/ijeat.C6339.029320.

[9] A. Sinha, “Application of Machine Learning in Cricket and Predictive Analytics of IPL 2020,” other, preprint, Oct. 2020. doi: 10.20944/preprints202010.0436.v1.

[10] K. T. Özgünen *et al.*, “Effect of hot environmental conditions on physical activity patterns and temperature response of football players: Effect of hot weather in football,” *Scandinavian Journal of Medicine & Science in Sports*, vol. 20, pp. 140–147, Oct. 2010, doi: 10.1111/j.1600-0838.2010.01219.x.

[11] T. Allen, A. Bowley, P. Wood, E. Henrikson, E. Morales, and D. James, “Effect of temperature on golf ball dynamics,” *Procedia Engineering*, vol. 34, pp. 634–639, 2012, doi: 10.1016/j.proeng.2012.04.108.

[12] H. H. Lemmer, “AN ANALYSIS OF PLAYERS’ PERFORMANCES IN THE FIRST CRICKET TWENTY20 WORLD CUP SERIES,” p. 7, 2008.

[13] B. Zhao, “Web Scraping,” in *Encyclopedia of Big Data*, L. A. Schintler and C. L. McNeely, Eds. Cham: Springer International Publishing, 2017, pp. 1–3. doi: 10.1007/978-3-319-32001-4_483-1.

[14] “Supervised vs. Unsupervised Learning: What’s the Difference?,” Mar. 12, 2021. https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning (accessed Jul. 08, 2021).

[15] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory - COLT ’92*, Pittsburgh, Pennsylvania, United States, 1992, pp. 144–152. doi: 10.1145/130385.130401.

[16] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *MACHINE LEARNING IN PYTHON*, p. 6.

[17] O. Kramer, “Scikit-Learn,” in *Machine Learning for Evolution Strategies*, vol. 20, Cham: Springer International Publishing, 2016, pp. 45–53. doi: 10.1007/978-3-319-33383-0_5.

[18] G. Zhang and Y. Lu, “Bias-corrected random forests in regression,” *Journal of Applied Statistics*, vol. 39, no. 1, pp. 151–160, Jan. 2012, doi: 10.1080/02664763.2011.578621.
[19] R. Frank Falk and Nancy B. Miller, “A Primer for Soft Modeling.” University of Akron Press, 1992.

APPENDIX A – Web Scrapers

A.1. Batting Data Scraper

```
import pandas as pd
import re


def extract_batting_data(content, team):
    inning_index = -1

    headers = content.find_all("h5", {"class": "header-title label"})
    for i, header in enumerate(headers):
        if team + ' INNINGS' in header.get_text():
            inning_index = i

    batsmen_df = pd.DataFrame(
        columns=["Name", "Desc", "Runs", "Balls", "Minutes", "Fours",
"Sixes", "Strike_Rate", "Batting_Position"])

    if inning_index > -1:
        table_body = content.find_all('tbody')

        for i, table in enumerate(table_body[0:4:2]):
            if i == inning_index:
                rows = table.find_all('tr')
                batting_position = 0
                for row in rows[::2]:
                    cols = row.find_all('td')
                    cols = [x.text.strip() for x in cols]
                    if cols[0] == 'Extras':
                        continue

                    if len(cols) > 7:
                        batting_position += 1
                        batsmen_df = batsmen_df.append(pd.Series(
                            [re.sub(r"\W+", ' ',
cols[0].split("(c)")[0]).strip(), cols[1],
                             cols[2], cols[3], cols[4], cols[5], cols[6],
cols[7], batting_position],
                            index=batsmen_df.columns), ignore_index=True)

    return batsmen_df
```

A.2. Bowling Data Scraper

```
import pandas as pd


def extract_bowling_data(content, team):
    inning_index = -1

    headers = content.find_all("h5", {"class": "header-title label"})
    for i, header in enumerate(headers):
        if 'INNINGS' in header.get_text() and team not in
header.get_text():
            inning_index = i
    bowler_df = pd.DataFrame(columns=['Name', 'Overs', 'Maidens', 'Runs',
'Wickets',
                                      'Econ', 'Dots', '4s', '6s', 'Wd',
'Nb'])
    if inning_index > -1:
        table_body = content.find_all('tbody')

        for i, table in enumerate(table_body[1:4:2]):
            if i == inning_index:
                rows = table.find_all('tr')
                for row in rows:
                    cols = row.find_all('td')
                    cols = [x.text.strip() for x in cols]
                    bowler_df = bowler_df.append(pd.Series([cols[0],
cols[1], cols[2], cols[3], cols[4], cols[5],
                                                            cols[6],
cols[7], cols[8], cols[9], cols[10]],

index=bowler_df.columns), ignore_index=True)
    return bowler_df
```

A.3. Fielding Data Scraper

```
import pandas as pd


def extract_fielding_data(content, team):
    inning_index = -1

    headers = content.find_all("h5", {"class": "header-title label"})
    for i, header in enumerate(headers):
        if ' INNINGS' in header.get_text() and team + ' INNINGS' not in
header.get_text():
            inning_index = i

    fielding_dict = {}
    fielding_df = pd.DataFrame(
        columns=["Name", "Catches", "Run Outs"])

    if inning_index > -1:
        table_body = content.find_all('tbody')

        for i, table in enumerate(table_body[0:4:2]):
            if i == inning_index:
                rows = table.find_all('tr')
                for row in rows[::2]:
                    cols = row.find_all('td')
                    cols = [x.text.strip() for x in cols]
                    if cols[0] == 'Extras':
                        continue

                    if len(cols) > 7:
                        fielding_data = cols[1]
                        if fielding_data.startswith("c & b "):
                            value = fielding_data.split(" b ")[1].strip()
                            if value in fielding_dict.keys():
                                fielding_dict[value]["catches"] += 1
                            else:
                                fielding_dict[value] = {"catches": 1, "run
outs": 0}

                        elif fielding_data.startswith("c "):
                            value = fielding_data.split(" b
")[0].replace("c ", "", 1).strip()
                            if value in fielding_dict.keys():
                                fielding_dict[value]["catches"] += 1
                            else:
                                fielding_dict[value] = {"catches": 1, "run
outs": 0}

                        elif fielding_data.startswith("st "):
                            value = fielding_data.split(" b
")[0].replace("st ", "", 1).strip()
                            if value in fielding_dict.keys():
                                fielding_dict[value]["run outs"] += 1
                            else:
                                fielding_dict[value] = {"catches": 0, "run
outs": 1}

                        elif fielding_data.startswith("run out") or
fielding_data.startswith("st "):
                            values = fielding_data.replace("run out", "",
1).replace("(", "", 1).replace(")", "",
```

```
1).split("/")
                            for value in values:
                                value = value.strip()
                                if value in fielding_dict.keys():
                                    fielding_dict[value]["run outs"] += 1
                                else:
                                    fielding_dict[value] = {"catches": 0,
"run outs": 1}
    for key in fielding_dict.keys():
        fielding_df = fielding_df.append(pd.Series(
            [key, fielding_dict[key]["catches"], fielding_dict[key]["run
outs"]],
            index=fielding_df.columns), ignore_index=True)
    return fielding_df
```

APPENDIX B – Team Combination Algorithms

B.1. Team Performance Calulation Algorithm

```
from team_selection.create_final_dataset import
get_actual_players_who_played


def actual_team_players(pool_df, match_id):
    actual_player_df, wicket_keepers, bowlers =
get_actual_players_who_played(match_id)
    return
pool_df[pool_df['player_name'].isin(actual_player_df["player_name"].to_nump
y())]


def calculate_overall_performance(input_df, match_id):
    team_df = input_df.copy()

    magic_number = 11 / len(team_df)  # this is to compensate players
missing from actual 11
    extras = 14.26
    runs_scored = team_df["runs_scored"]
    balls_faced = team_df["balls_faced"]
    wickets_taken = team_df["wickets_taken"]
    runs_conceded = team_df["runs_conceded"]
    deliveries = team_df["deliveries"]

    # total_score = runs_scored.sum() * magic_number + extras
    # target = runs_conceded.sum() * magic_number
    # total_balls_faced = balls_faced.sum() * magic_number

    total_score = get_total_score(balls_faced, runs_scored, extras,
magic_number)
    target = get_total_conceded(deliveries, runs_conceded, wickets_taken) +
20 # compensate for byes/ leg byes
    total_balls_faced = calculate_total_balls_faced(balls_faced,
magic_number)

    if total_balls_faced < 300:
        wickets_fallen = 10
    else:
        wickets_fallen = 10 * 300 / total_balls_faced

    team_df["total_score"] = total_score
    team_df["total_wickets"] = wickets_fallen
    team_df["total_balls"] = total_balls_faced
    team_df["target"] = target
    team_df["extras"] = extras
    team_df["match_number"] = match_id

    def calculate_batting_contribution(row, key):
        return row[key] / total_score

    def calculate_bowling_contribution(row, key):
        return row[key] / total_score

    team_df.to_csv("final_team.csv")

    team_df["bowling_contribution"] = team_df.apply(
        lambda row: calculate_bowling_contribution(row, "runs_conceded"),
axis=1)
```

```
    team_df["batting_contribution"] = team_df.apply(
        lambda row: calculate_batting_contribution(row, "runs_scored"),
axis=1)

    # print(magic_number, runs_scored.sum(), balls_faced.sum(), extras)
    # print(magic_number, runs_conceded.sum(), deliveries.sum(),
wickets_taken.sum())
    # print("Total Score:", get_total_score(balls_faced, runs_scored,
extras, magic_number))
    # print("Runs given:", get_total_conceded(deliveries, runs_conceded,
wickets_taken))

    # TODO: for evaluation
    # SELECT * FROM `match_details` WHERE `wickets` < 10 AND `balls` < 300
AND `target` IS NOT NULL
    # where the team stopped batting, since they have chased the opposition
target before the 50 overs
    # need to compare predicted score with score for 50 overs. because the
predicted score will always be high

    return team_df, total_score, target


def get_total_score(balls_faced, runs_scored, extras, magic_number):
    if balls_faced.sum() > 300:
        return (runs_scored.sum() * 300 / balls_faced.sum()) + extras
    if magic_number < 1:
        return (runs_scored.sum() * magic_number) + extras
    return runs_scored.sum() + extras


def get_total_conceded(deliveries, runs_conceded, wickets_taken):
    if deliveries.sum() > 300:
        return runs_conceded.sum() * 300 / deliveries.sum()
    elif wickets_taken.sum() > 10:
        return runs_conceded.sum() * 10 / wickets_taken.sum()

    return runs_conceded.sum()


def calculate_total_balls_faced(balls_faced, magic_number):
    sum = balls_faced.sum() * magic_number
    if sum > 300:
        return 300
    return sum
```

B.2. Team Selection Algorithm

```
predicted_team = player_performance_predictions.sort_values(
        by="winning_probability", ascending=False)[:11]

    # COMBINATION ALGORITHM
    predicted_team = player_performance_predictions.copy()
    wicket_keeper =
predicted_team.copy().loc[predicted_team["is_wicket_keeper"] ==
1].sort_values(
        by=["winning_probability", "batting_contribution"],
ascending=[False, False])[:1]
    wicket_keeper_name = wicket_keeper.iloc[0]["player_name"]
    batsmen_df = predicted_team.loc[
        (predicted_team['bowling_consistency'] == 0) &
(predicted_team['player_name'] != wicket_keeper_name)]
    batsmen_df = batsmen_df.sort_values(by=["winning_probability",
"batting_contribution"], ascending=[False, False])[
                 :5]
    bowler_df = predicted_team.loc[predicted_team['bowling_consistency'] >
0]
    bowler_df = bowler_df.loc[bowler_df['deliveries'] > 0].sort_values(
        by=["winning_probability", "bowling_contribution"],
ascending=[False, True])[:5]

    predicted_team = pd.concat([batsmen_df, wicket_keeper,
bowler_df]).reset_index(drop=True)
    predicted_team, win_percent = predict_for_team(predicted_team)
    print("WIN % :", win_percent)

    team_df, total_score, target =
calculate_overall_performance(predicted_team, match_id)
```

APPENDIX C – Optimal Team Prediction Results

```
WIN % : 0.9417527978803296
Match_ID 1120286 Score: 260.41910681588166 Target: 261.3358809536207
             player_name  runs_scored  ...  wickets_taken  winning_probability
0      Lahiru Thirimanne    54.570277  ...       0.000000             0.948721
6   Danushka Gunathilaka    50.961551  ...       0.850211             0.964394
1          Upul Tharanga    32.476192  ...       0.000000             0.942071
2       Dinesh Chandimal    25.015158  ...       0.000000             0.934794
5     Niroshan Dickwella    23.255534  ...       0.000000             0.935895
8        Asela Gunaratne    23.262057  ...       0.887187             0.954413
9        Ashan Priyanjan    17.216755  ...       0.903566             0.933172
3       Avishka Fernando    13.503830  ...       0.000000             0.934052
4     Dimuth Karunaratne    11.798204  ...       0.000000             0.932958
7    Dhananjaya de Silva     0.000000  ...       2.378120             0.963158
10       Dhammika Prasad     0.000000  ...       0.882506             0.915654

[11 rows x 6 columns]
WIN % : 0.9817974006866901
Match_ID 1120287 Score: 249.53957108556037 Target: 224.58068876622326
             player_name  runs_scored  ...  wickets_taken  winning_probability
7   Danushka Gunathilaka    51.653047  ...       0.849364             0.984152
0          Upul Tharanga    44.305561  ...       0.000000             0.983002
3       Dinesh Chandimal    24.210603  ...       0.000000             0.981334
5           Kusal Perera    32.617341  ...       0.000000             0.982327
4      Lahiru Thirimanne    22.412888  ...       0.000000             0.981263
8        Asela Gunaratne    24.420582  ...       0.883212             0.983572
1       Avishka Fernando    13.151658  ...       0.000000             0.981860
2     Dimuth Karunaratne    11.349891  ...       0.000000             0.981608
10         Jeevan Mendis     0.638049  ...       2.724620             0.975751
6    Dhananjaya de Silva    10.519951  ...       1.200311             0.984289
9        Dhammika Prasad     0.000000  ...       0.877771             0.980613

[11 rows x 6 columns]
WIN % : 0.02622847233244988
Match_ID 1120288 Score: 270.20429134037397 Target: 163.5692438305874
             player_name  runs_scored  ...  wickets_taken  winning_probability
2          Upul Tharanga    46.193389  ...       0.000000             0.023453
4           Kusal Perera    54.633101  ...       0.000000             0.022639
7   Danushka Gunathilaka    51.378972  ...       0.851470             0.030590
5       Dinesh Chandimal    31.774938  ...       0.000000             0.022666
3      Lahiru Thirimanne    42.668443  ...       0.000000             0.022848
8        Asela Gunaratne    28.579489  ...       0.779207             0.026738
0       Avishka Fernando    14.082322  ...       0.000000             0.024395
1     Dimuth Karunaratne    12.229572  ...       0.000000             0.024010
9          Jeevan Mendis     0.603713  ...       2.357080             0.025583
6    Dhananjaya de Silva    10.891041  ...       1.163358             0.042883
10       Dhammika Prasad     0.000000  ...       0.785626             0.022707

[11 rows x 6 columns]
WIN % : 0.021962214277143424
Match_ID 1120289 Score: 283.89824344222797 Target: 212.04219366094196
             player_name  runs_scored  ...  wickets_taken  winning_probability
3           Kusal Perera    52.830144  ...       0.000000             0.019794
8          Dasun Shanaka    61.782553  ...       0.857143             0.023026
7   Danushka Gunathilaka    50.501563  ...       0.831112             0.024617
1      Lahiru Thirimanne    62.841861  ...       0.000000             0.019892
5           Kusal Mendis    31.032137  ...       0.000000             0.019941
9        Asela Gunaratne    23.541225  ...       0.790480             0.022492
```

```
6    Dhananjaya de Silva    16.702257  ...       0.791319             0.030166
10       Ashan Priyanjan    29.219204  ...       0.858570             0.022060
4     Niroshan Dickwella     9.349766  ...       0.000000             0.019605
0       Avishka Fernando    14.989792  ...       0.000000             0.020110
2     Chamara Kapugedera    12.032584  ...       0.000000             0.019881

[11 rows x 6 columns]
WIN % : 0.13792086768044082
Match_ID 1120290 Score: 273.5785013135841 Target: 212.57940225200437
             player_name  runs_scored  ...  wickets_taken  winning_probability
1           Kusal Perera    48.492729  ...       0.000000             0.043057
8          Dasun Shanaka    60.225026  ...       0.854962             0.190088
7   Danushka Gunathilaka    50.779192  ...       0.838931             0.291339
2      Lahiru Thirimanne    38.556940  ...       0.000000             0.041592
5           Kusal Mendis    33.224269  ...       0.000000             0.046951
9       Asela Gunaratne    23.675294  ...       0.789287             0.160676
10       Ashan Priyanjan    19.340824  ...       0.866390             0.105088
6    Dhananjaya de Silva    20.314231  ...       0.799443             0.515682
4     Niroshan Dickwella     6.997615  ...       0.000000             0.034671
0       Avishka Fernando    13.552420  ...       0.000000             0.046711
3     Chamara Kapugedera    10.404914  ...       0.000000             0.041276

[11 rows x 6 columns]
WIN % : 0.02896633270255455
Match_ID 1122726 Score: 290.97784251397667 Target: 249.06399063161987
            player_name  runs_scored  ...  wickets_taken  winning_probability
1         Upul Tharanga    53.657476  ...       0.000000             0.023334
9          Jeevan Mendis    44.728988  ...       0.868097             0.032160
6          Dasun Shanaka    44.730040  ...       0.949457             0.037469
7       Asela Gunaratne    43.621562  ...       0.922306             0.035690
0           Kusal Perera    80.905877  ...       0.000000             0.028791
2           Kusal Mendis    45.637632  ...       0.000000             0.023002
5       Dinesh Chandimal    96.137610  ...       0.000000             0.032307
3     Niroshan Dickwella    28.459216  ...       0.000000             0.021204
10  Dhananjaya de Silva    28.618193  ...       0.914977             0.031571
8       Ashan Priyanjan    15.331060  ...       0.979169             0.032442
4    Dimuth Karunaratne    14.905736  ...       0.000000             0.020658

[11 rows x 6 columns]
WIN % : 0.020860222133360817
Match_ID 1122727 Score: 127.83988403936381 Target: 294.7865283449341
           player_name  runs_scored  ...  wickets_taken  winning_probability
0        Upul Tharanga    0.000000  ...       0.000000             0.024037
4         Kusal Mendis   41.677212  ...       0.000000             0.020424
1    Lahiru Thirimanne   21.588453  ...       0.000000             0.021083
2     Avishka Fernando   17.452574  ...       0.000000             0.020842
5   Niroshan Dickwella   11.646025  ...       0.000000             0.020529
3   Chamara Kapugedera   11.538909  ...       0.000000             0.020508
9    Sachith Pathirana    0.000000  ...       0.895093             0.020222
10      Suranga Lakmal    4.347420  ...       0.896279             0.020186
7     Lakshan Sandakan    5.329291  ...       0.845580             0.020501
6      Dhammika Prasad    0.000000  ...       0.809198             0.020868
8         PWH de Silva    0.000000  ...       0.868113             0.020263

[11 rows x 6 columns]
WIN % : 0.018213400837219743
Match_ID 1122728 Score: 285.1386517898473 Target: 157.66539840151373
           player_name  runs_scored  ...  wickets_taken  winning_probability
1    Lahiru Thirimanne   92.012955  ...       0.000000             0.018184
0        Upul Tharanga   99.576906  ...       0.000000             0.018194
7       Angelo Mathews   18.243891  ...       1.031896             0.018266
```

```
5     Dinesh Chandimal    93.990905  ...       0.000000             0.018186
2         Kusal Perera    72.200695  ...       0.000000             0.018160
4   Niroshan Dickwella    32.935950  ...       0.000000             0.018136
3         Kusal Mendis    37.394224  ...       0.000000             0.018137
6       Lasith Malinga    33.244354  ...       0.971347             0.018358
8     Nuwan Kulasekara    32.814187  ...       0.882523             0.018257
9       Thisara Perera    10.624754  ...       0.989777             0.018256
10      Suranga Lakmal     1.808823  ...       1.078135             0.018213

[11 rows x 6 columns]
WIN % : 0.01862113157961083
Match_ID 1130738 Score: 268.4482007830638 Target: 303.435077461658
            player_name  runs_scored  ...  wickets_taken  winning_probability
7        Thisara Perera    54.814457  ...       0.961589             0.019625
9    Dhananjaya de Silva    51.660412  ...       0.831800             0.018617
3      Dinesh Chandimal    29.851680  ...       0.000000             0.018254
6        Angelo Mathews    85.567067  ...       0.921299             0.019695
5    Niroshan Dickwella    16.543704  ...       0.000000             0.018260
4      Avishka Fernando    29.127280  ...       0.000000             0.018253
0          Kusal Mendis    13.939311  ...       0.000000             0.018256
1    Dimuth Karunaratne    13.909205  ...       0.000000             0.018256
8        Lasith Malinga     2.030244  ...       1.027977             0.018877
10     Nuwan Kulasekara     3.346845  ...       0.929570             0.018485
2    Chamara Kapugedera    12.186741  ...       0.000000             0.018255

[11 rows x 6 columns]
WIN % : 0.01824106661709175
Match_ID 1130739 Score: 252.96016257797277 Target: 294.9568037201441
           player_name  runs_scored  ...  wickets_taken  winning_probability
7       Angelo Mathews    88.965205  ...       0.892781             0.018432
3         Kusal Mendis    35.775573  ...       0.000000             0.018168
2     Dinesh Chandimal    24.040107  ...       0.000000             0.018168
4   Chamara Kapugedera    44.709146  ...       0.000000             0.018167
5   Niroshan Dickwella    12.636513  ...       0.000000             0.018172
1     Avishka Fernando    23.178059  ...       0.000000             0.018169
0   Dimuth Karunaratne    19.019991  ...       0.000000             0.018169
6       Lasith Malinga     1.008844  ...       1.043534             0.018449
9     Nuwan Kulasekara     3.221455  ...       0.933002             0.018252
8    Seekkuge Prasanna     4.516523  ...       0.934040             0.018260
10     Dhammika Prasad     0.000000  ...       2.640181             0.018246

[11 rows x 6 columns]
WIN % : 0.018168829504431806
Match_ID 1130740 Score: 279.05843020886016 Target: 298.2724568673649
            player_name  runs_scored  ...  wickets_taken  winning_probability
0         Upul Tharanga    66.759088  ...       0.000000             0.018130
5          Kusal Perera    50.635269  ...       0.000000             0.018128
7        Thisara Perera    53.898967  ...       0.954545             0.018255
8   Dhananjaya de Silva    51.818583  ...       0.839684             0.018201
4      Dinesh Chandimal    29.754956  ...       0.000000             0.018127
6        Angelo Mathews    88.301155  ...       0.920776             0.018265
10         Dasun Shanaka    43.727353  ...       0.890288             0.018174
2     Lahiru Thirimanne    43.496872  ...       0.000000             0.018128
3      Avishka Fernando    29.956674  ...       0.000000             0.018127
9        Lasith Malinga     1.964349  ...       1.037883             0.018192
1    Chamara Kapugedera    13.138152  ...       0.000000             0.018128

[11 rows x 6 columns]
WIN % : 0.018760342300751964
Match_ID 1130742 Score: 251.35060584117642 Target: 292.51618334988683
           player_name  runs_scored  ...  wickets_taken  winning_probability
```

```
7       Angelo Mathews    93.303305  ...       0.893292             0.019052
1     Dinesh Chandimal    23.942827  ...       0.000000             0.018370
4         Kusal Mendis    33.913496  ...       0.000000             0.018366
0   Chamara Kapugedera    41.318364  ...       0.000000             0.018372
5   Niroshan Dickwella    11.158205  ...       0.000000             0.018373
3     Avishka Fernando    21.453127  ...       0.000000             0.018368
2   Dimuth Karunaratne    17.007491  ...       0.000000             0.018369
6       Lasith Malinga     1.014939  ...       1.053423             0.020413
9    Seekkuge Prasanna     4.087570  ...       0.936396             0.018861
10    Nuwan Kulasekara     3.199617  ...       0.933542             0.018819
8      Dhammika Prasad     0.000000  ...       2.547751             0.019002

[11 rows x 6 columns]
WIN % : 0.1695829703905207
Match_ID 1130743 Score: 269.99970148315026 Target: 190.41782711629554
           player_name  runs_scored  ...  wickets_taken  winning_probability
0        Upul Tharanga    64.519490  ...       0.000000             0.026111
5         Kusal Perera    52.597771  ...       0.000000             0.023767
1    Lahiru Thirimanne    50.965836  ...       0.000000             0.023567
9       Thisara Perera    47.058998  ...       0.913044             0.052418
6       Angelo Mathews    93.381607  ...       0.855949             0.770247
4     Dinesh Chandimal    25.005416  ...       0.000000             0.020821
3         Kusal Mendis    30.280953  ...       0.000000             0.021099
2   Chamara Kapugedera    35.196498  ...       0.000000             0.021655
7       Lasith Malinga     0.952546  ...       1.029655             0.731105
8     Nuwan Kulasekara     2.684464  ...       0.926006             0.122542
10       Nuwan Pradeep     0.000000  ...       1.148209             0.052082

[11 rows x 6 columns]
WIN % : 0.018521062585599657
Match_ID 1142584 Score: 224.76018732760576 Target: 210.64380026911593
           player_name  runs_scored  ...  wickets_taken  winning_probability
5   Niroshan Dickwella    43.645879  ...       0.000000             0.018360
3        Upul Tharanga    21.036112  ...       0.000000             0.018316
0     Dinesh Chandimal    41.141818  ...       0.000000             0.018354
2         Kusal Perera    24.309395  ...       0.000000             0.018320
6      Asela Gunaratne    23.543240  ...       0.855970             0.018845
8          Isuru Udana    14.308455  ...       0.730278             0.018715
9        Jeevan Mendis    14.218482  ...       0.727906             0.018684
7      Ashan Priyanjan    11.593449  ...       0.838430             0.018843
4   Chamara Kapugedera    13.580903  ...       0.000000             0.018303
1     Avishka Fernando     0.000000  ...       0.000000             0.018341
10     Shaminda Eranga     3.122455  ...       0.728998             0.018651

[11 rows x 6 columns]
WIN % : 0.03689170015851514
Match_ID 1142585 Score: 173.98667511950654 Target: 179.5979267889729
            player_name  runs_scored  ...  wickets_taken  winning_probability
2    Niroshan Dickwella    44.624908  ...       0.000000             0.023517
1         Upul Tharanga    21.002784  ...       0.000000             0.023988
3      Dinesh Chandimal    41.163011  ...       0.000000             0.023337
5          Kusal Perera    23.804189  ...       0.000000             0.023627
4     Lahiru Thirimanne    11.514911  ...       0.000000             0.022868
9         Dasun Shanaka     5.947465  ...       0.819540             0.037633
0      Avishka Fernando     0.000000  ...       0.000000             0.031656
10  Dushmantha Chameera    10.221239  ...       0.787601             0.032178
8     Seekkuge Prasanna     1.448168  ...       0.883530             0.041016
7       Dhammika Prasad     0.000000  ...       1.476492             0.050358
6         Nuwan Pradeep     0.000000  ...       0.970646             0.095632

[11 rows x 6 columns]
```

```
WIN % : 0.019093961134413757
Match_ID 1142586 Score: 269.0053423506299 Target: 219.41042878531118
             player_name  runs_scored  ...  wickets_taken  winning_probability
5           Kusal Perera    52.732069  ...       0.000000             0.019913
8         Thisara Perera    45.967758  ...       0.960054             0.018494
1      Niroshan Dickwella   43.688841  ...       0.000000             0.019555
3          Upul Tharanga    32.447942  ...       0.000000             0.019480
6          Dasun Shanaka    54.875378  ...       0.959316             0.018786
0       Dinesh Chandimal    23.577043  ...       0.000000             0.019644
2      Lahiru Thirimanne    39.080484  ...       0.000000             0.019504
4           Kusal Mendis     8.475612  ...       0.000000             0.019200
7        Dhammika Prasad     9.721147  ...       2.547460             0.018559
10  Dushmantha Chameera      7.730875  ...       0.953142             0.018405
9          Nuwan Pradeep     0.000000  ...       0.999567             0.018494

[11 rows x 6 columns]
WIN % : 0.5652564616801553
Match_ID 1142587 Score: 262.151394652547 Target: 164.34113544873125
             player_name  runs_scored  ...  wickets_taken  winning_probability
5           Kusal Perera    53.374489  ...       0.000000             0.426823
7         Thisara Perera    45.216887  ...       1.021335             0.916372
0     Niroshan Dickwella    44.396210  ...       0.000000             0.370137
3          Upul Tharanga    30.751381  ...       0.000000             0.313094
1       Dinesh Chandimal    22.315106  ...       0.000000             0.327015
2      Lahiru Thirimanne    36.997857  ...       0.000000             0.320721
4           Kusal Mendis     9.666326  ...       0.000000             0.215155
9       Nuwan Kulasekara     9.449112  ...       1.020734             0.820326
6         Lasith Malinga    10.631601  ...       1.139646             0.921009
8         Suranga Lakmal     1.595003  ...       0.969621             0.831708
10         Nuwan Pradeep     0.000000  ...       0.969394             0.755461

[11 rows x 6 columns]
WIN % : 0.019026846792916705
Match_ID 1142588 Score: 220.74107179151625 Target: 218.65566764627798
             player_name  runs_scored  ...  wickets_taken  winning_probability
3          Upul Tharanga    30.174281  ...       0.000000             0.018717
4      Lahiru Thirimanne    40.792131  ...       0.000000             0.018694
5           Kusal Mendis    40.422265  ...       0.000000             0.018698
9            Isuru Udana    11.090037  ...       0.884848             0.019238
6          Jeevan Mendis    28.921198  ...       0.900325             0.019576
2       Avishka Fernando    10.787556  ...       0.000000             0.018812
10     Sachith Pathirana    11.429006  ...       0.805688             0.019207
0     Dimuth Karunaratne    11.119922  ...       0.000000             0.018819
1     Chamara Kapugedera    10.370427  ...       0.000000             0.018815
7      Shehan Jayasuriya     9.836525  ...       0.914876             0.019362
8          Nuwan Pradeep     1.537723  ...       0.950108             0.019356

[11 rows x 6 columns]
WIN % : 0.605786692995115
Match_ID 1153243 Score: 261.2232855078502 Target: 283.4312544548007
             player_name  runs_scored  ...  wickets_taken  winning_probability
2          Upul Tharanga    64.225732  ...       0.000000             0.396379
3      Lahiru Thirimanne    51.064301  ...       0.000000             0.390986
9          Angelo Mathews   35.868822  ...       2.455161             0.831375
5     Niroshan Dickwella    28.427167  ...       0.000000             0.397852
4     Chamara Kapugedera    50.844355  ...       0.000000             0.350091
7       Nuwan Kulasekara    25.424219  ...       0.930643             0.852330
0       Avishka Fernando    14.891010  ...       0.000000             0.461341
1     Dimuth Karunaratne    11.584738  ...       0.000000             0.398289
6         Lasith Malinga    10.993331  ...       1.018680             0.952255
10  Dushmantha Chameera      9.616958  ...       0.893966             0.796893
```

```
8     Seekkuge Prasanna     5.144721  ...       0.933731             0.835863

[11 rows x 6 columns]
WIN % : 0.9416579273012193
Match_ID 1153245 Score: 251.7059650554206 Target: 279.6716598955012
           player_name  runs_scored  ...  wickets_taken  winning_probability
3        Upul Tharanga    49.912079  ...       0.000000             0.912650
5         Kusal Perera    62.333229  ...       0.000000             0.914745
8       Angelo Mathews    30.094268  ...       0.960598             0.975548
0     Avishka Fernando    34.425894  ...       0.000000             0.919373
1   Dimuth Karunaratne    12.062437  ...       0.000000             0.917136
4   Chamara Kapugedera    10.517765  ...       0.000000             0.899867
2    Lahiru Thirimanne    13.010275  ...       0.000000             0.917109
9          Isuru Udana    11.180425  ...       0.871956             0.974819
7        Nuwan Pradeep     6.159180  ...       0.833734             0.976231
6       Lasith Malinga     4.630338  ...       1.041991             0.979589
10     Akila Dananjaya     3.120075  ...       0.930102             0.971172

[11 rows x 6 columns]
WIN % : 0.022587971264019634
Match_ID 1140380 Score: 190.22148425480128 Target: 208.43999190445226
           player_name  runs_scored  ...  wickets_taken  winning_probability
5     Dinesh Chandimal    42.094864  ...       0.000000             0.019994
3   Niroshan Dickwella    41.728799  ...       0.000000             0.019967
1        Upul Tharanga    22.676923  ...       0.000000             0.020481
2    Lahiru Thirimanne    23.704484  ...       0.000000             0.020154
0     Avishka Fernando     0.000000  ...       0.000000             0.023223
4   Chamara Kapugedera    18.408128  ...       0.000000             0.019939
7       Thisara Perera    11.343975  ...       1.014580             0.025390
8       Lasith Malinga    12.048834  ...       1.085259             0.025163
10      Suranga Lakmal     3.591930  ...       0.999766             0.023931
9        Nuwan Pradeep     0.000000  ...       1.000988             0.024662
6      Dhammika Prasad     0.363548  ...       2.562093             0.025564

[11 rows x 6 columns]
WIN % : 0.29888022622592
Match_ID 1140381 Score: 261.63250592352676 Target: 207.32508907978905
             player_name  runs_scored  ...  wickets_taken  winning_probability
1          Upul Tharanga    49.278716  ...       0.000000             0.042423
5     Niroshan Dickwella    54.775762  ...       0.000000             0.043485
0      Lahiru Thirimanne    60.832934  ...       0.000000             0.044771
4       Dinesh Chandimal    24.629684  ...       0.000000             0.037662
2           Kusal Perera    33.984621  ...       0.000000             0.039488
8    Dhananjaya de Silva    23.274610  ...       0.916702             0.583798
3       Avishka Fernando    31.375181  ...       0.000000             0.038960
6        Akila Dananjaya    30.061334  ...       0.956277             0.760399
10           Isuru Udana    11.579228  ...       0.751547             0.459537
9    Dushmantha Chameera     8.915628  ...       0.840095             0.575756
7   Sachithra Senanayake     3.936652  ...       0.907400             0.661402

[11 rows x 6 columns]
WIN % : 0.021901840753210687
Match_ID 1140382 Score: 259.2935710298613 Target: 152.6519800757819
           player_name  runs_scored  ...  wickets_taken  winning_probability
1        Upul Tharanga    47.569607  ...       0.000000             0.021100
5   Niroshan Dickwella    53.580379  ...       0.000000             0.021285
0    Lahiru Thirimanne    53.313262  ...       0.000000             0.021286
2     Dinesh Chandimal    22.129539  ...       0.000000             0.020610
9       Thisara Perera    25.349424  ...       1.010067             0.022817
4         Kusal Perera    30.751024  ...       0.000000             0.020568
3     Avishka Fernando    29.606613  ...       0.000000             0.020580
```

```
6       Lasith Malinga    11.514146  ...       1.111425             0.023939
8     Nuwan Kulasekara     9.202465  ...       0.990384             0.022840
7       Suranga Lakmal     1.720966  ...       0.969586             0.023336
10       Nuwan Pradeep     0.000000  ...       0.923648             0.022559

[11 rows x 6 columns]
WIN % : 0.03917830863312456
Match_ID 1140383 Score: 262.8671302759668 Target: 160.6250804832245
           player_name  runs_scored  ...  wickets_taken  winning_probability
2        Upul Tharanga    47.293029  ...       0.000000             0.032203
3     Dinesh Chandimal    54.018416  ...       0.000000             0.032163
5   Niroshan Dickwella    52.860283  ...       0.000000             0.032277
1    Lahiru Thirimanne    54.198731  ...       0.000000             0.032214
4         Kusal Mendis    36.877068  ...       0.000000             0.032080
8      Akila Dananjaya    23.178402  ...       0.945177             0.046977
0     Avishka Fernando    24.032942  ...       0.000000             0.032331
9       Thisara Perera    11.515598  ...       0.995940             0.046144
10      Lasith Malinga     8.666866  ...       1.093556             0.045918
6       Suranga Lakmal     1.250151  ...       0.950588             0.050749
7      Dhammika Prasad     0.000000  ...       1.033453             0.047905

[11 rows x 6 columns]
WIN % : 0.058427664939656905
Match_ID 1153840 Score: 279.6806982341149 Target: 212.1226813507237
             player_name  runs_scored  ...  wickets_taken  winning_probability
5     Niroshan Dickwella    54.249643  ...       0.000000             0.069020
7         Angelo Mathews    85.279384  ...       0.815529             0.037602
1          Upul Tharanga    88.166582  ...       0.000000             0.113249
3           Kusal Perera    41.284234  ...       0.000000             0.050038
6   Danushka Gunathilaka    33.674101  ...       0.000000             0.038839
2       Dinesh Chandimal    18.610707  ...       0.000000             0.056790
9    Milinda Siriwardana    49.314859  ...       1.251411             0.025733
10   Dhananjaya de Silva    23.727686  ...       0.889376             0.025192
4       Avishka Fernando    22.445687  ...       0.000000             0.044806
0      Lahiru Thirimanne     0.000000  ...       0.000000             0.146529
8         Lasith Malinga    10.673766  ...       1.262378             0.034906

[11 rows x 6 columns]
WIN % : 0.7424067995050432
Match_ID 1153841 Score: 269.72150172094143 Target: 228.9844409133001
            player_name  runs_scored  ...  wickets_taken  winning_probability
5    Niroshan Dickwella    51.073087  ...       0.000000             0.641858
6        Angelo Mathews    87.804290  ...       0.869922             0.922448
0         Upul Tharanga    89.906544  ...       0.000000             0.790254
2          Kusal Perera    41.130636  ...       0.000000             0.599952
3      Dinesh Chandimal    23.799813  ...       0.000000             0.552210
9   Dhananjaya de Silva    23.738863  ...       0.940578             0.859459
4      Avishka Fernando    22.000696  ...       0.000000             0.531408
1     Lahiru Thirimanne     0.000000  ...       0.000000             0.634663
10      Shaminda Eranga     6.426942  ...       0.937116             0.818011
7       Dhammika Prasad     0.000000  ...       0.885157             0.913601
8     Jeffrey Vandersay     0.000000  ...       0.936491             0.902611

[11 rows x 6 columns]
WIN % : 0.6418453505119074
Match_ID 1153842 Score: 280.4898732945151 Target: 235.42262938704886
             player_name  runs_scored  ...  wickets_taken  winning_probability
7         Angelo Mathews     0.000000  ...       1.116553             0.840024
2      Lahiru Thirimanne    88.176464  ...       0.000000             0.658792
1          Upul Tharanga    94.720786  ...       0.000000             0.718656
8         Thisara Perera    41.749080  ...       1.070342             0.537533
```

```
10  Danushka Gunathilaka    44.740719  ...       0.868432             0.289530
5           Kusal Mendis     0.000000  ...       0.000000             0.850417
3       Dinesh Chandimal    22.654370  ...       0.000000             0.584490
9     Dhananjaya de Silva   29.243201  ...       1.399704             0.323687
4       Avishka Fernando    25.261353  ...       0.000000             0.536033
6         Lasith Malinga     0.000000  ...       1.434147             0.950960
0     Dimuth Karunaratne     0.999695  ...       0.000000             0.770179

[11 rows x 6 columns]
WIN % : 0.8449633416310907
Match_ID 1144167 Score: 280.6639938889239 Target: 234.9100193825154
             player_name  runs_scored  ...  wickets_taken  winning_probability
5     Niroshan Dickwella    54.195806  ...       0.000000             0.894524
6          Dasun Shanaka    51.800098  ...       0.872286             0.980541
1           Kusal Perera    40.572070  ...       0.000000             0.834782
0      Lahiru Thirimanne    56.687403  ...       0.000000             0.910473
7   Sachithra Senanayake    53.473114  ...       0.902468             0.974440
8          Jeevan Mendis    20.303435  ...       0.817352             0.974162
3       Avishka Fernando    21.509477  ...       0.000000             0.601178
2           Kusal Mendis    14.649068  ...       0.000000             0.605196
4     Dimuth Karunaratne    13.394360  ...       0.000000             0.572848
9          Nuwan Pradeep     3.265669  ...       0.834153             0.973330
10   Dushmantha Chameera    10.599031  ...       0.838505             0.973123

[11 rows x 6 columns]
WIN % : 0.0434189463979663
Match_ID 1144168 Score: 273.48907923029986 Target: 209.51358951751138
            player_name  runs_scored  ...  wickets_taken  winning_probability
4    Niroshan Dickwella    42.303361  ...       0.000000             0.028202
6         Dasun Shanaka    50.834196  ...       0.917477             0.061558
5      Dinesh Chandimal    88.727971  ...       0.000000             0.054477
0     Lahiru Thirimanne    76.864458  ...       0.000000             0.042917
1         Upul Tharanga     0.000000  ...       0.000000             0.039544
2          Kusal Perera     0.000000  ...       0.000000             0.039544
8        Thisara Perera    22.847955  ...       2.437243             0.044882
9         Jeevan Mendis    14.865958  ...       0.932302             0.042134
3          Kusal Mendis     0.717556  ...       0.000000             0.035427
7         Nuwan Pradeep     4.267391  ...       0.946572             0.048572
10  Dushmantha Chameera     9.861207  ...       0.915610             0.040351

[11 rows x 6 columns]
WIN % : 0.02456372761087139
Match_ID 1144169 Score: 268.08743845613037 Target: 222.75012387452227
           player_name  runs_scored  ...  wickets_taken  winning_probability
5         Kusal Perera    80.288927  ...       0.000000             0.023444
0    Lahiru Thirimanne    83.015053  ...       0.000000             0.023618
1        Upul Tharanga    23.646367  ...       0.000000             0.022453
4     Dinesh Chandimal    29.379701  ...       0.000000             0.021487
2         Kusal Mendis    23.249965  ...       0.000000             0.021817
6       Thisara Perera    10.200171  ...       1.322640             0.029260
3   Niroshan Dickwella    10.307268  ...       0.000000             0.021542
7       Lasith Malinga    22.321971  ...       1.107561             0.028327
8     Nuwan Kulasekara     9.309871  ...       1.004082             0.026520
10      Suranga Lakmal     1.547315  ...       1.368911             0.025847
9      Dhammika Prasad     0.000000  ...       0.942172             0.025885

[11 rows x 6 columns]
WIN % : 0.21743106744424748
Match_ID 1144170 Score: 283.5061912008594 Target: 139.8219482380956
           player_name  runs_scored  ...  wickets_taken  winning_probability
5         Kusal Perera    85.070547  ...       0.000000             0.039901
```

```
10      Angelo Mathews   89.864022  ...       0.786948             0.042599
0   Lahiru Thirimanne   84.458673  ...       0.000000             0.039164
1       Upul Tharanga    0.000000  ...       0.000000             0.030462
3    Dinesh Chandimal    6.757690  ...       0.000000             0.022987
7      Thisara Perera    7.474794  ...       2.572093             0.917000
2   Niroshan Dickwella    0.000000  ...       0.000000             0.030462
4        Kusal Mendis   30.178539  ...       0.000000             0.022292
9    Nuwan Kulasekara   11.697328  ...       1.135161             0.079001
6      Lasith Malinga    0.000000  ...       1.215895             0.937104
8      Suranga Lakmal    1.298780  ...       1.071107             0.230770

[11 rows x 6 columns]
WIN % : 0.5163773949197485
Match_ID 1144171 Score: 288.85220925551357 Target: 163.57077313945805
          player_name  runs_scored  ...  wickets_taken  winning_probability
2   Niroshan Dickwella   63.247010  ...       0.000000             0.264849
5        Kusal Perera   82.454850  ...       0.000000             0.331713
3       Upul Tharanga   49.934714  ...       0.000000             0.231065
0    Dinesh Chandimal   85.960995  ...       0.000000             0.321397
1   Lahiru Thirimanne   74.235503  ...       0.000000             0.293709
7      Thisara Perera    0.000000  ...       1.183201             0.978586
4        Kusal Mendis   30.842569  ...       0.000000             0.113987
6      Lasith Malinga    0.000000  ...       1.256849             0.980758
9      Suranga Lakmal   10.210127  ...       0.796455             0.882947
8    Nuwan Kulasekara    1.077775  ...       0.937458             0.937787
10     Dhammika Prasad    0.000000  ...       0.768855             0.343353

[11 rows x 6 columns]
WIN % : 0.15583472294278886
Match_ID 1169332 Score: 285.8913494005303 Target: 126.35651888652248
          player_name  runs_scored  ...  wickets_taken  winning_probability
1   Lahiru Thirimanne   54.575984  ...       0.000000             0.030642
2   Niroshan Dickwella   60.268309  ...       0.000000             0.029588
4    Avishka Fernando   88.539424  ...       0.000000             0.026863
3   Dimuth Karunaratne   63.162137  ...       0.000000             0.029285
6      Thisara Perera    0.000000  ...       1.112231             0.875393
5    Dinesh Chandimal   22.804238  ...       0.000000             0.038261
9      Lasith Malinga   45.230121  ...       3.971506             0.128645
7    Nuwan Kulasekara   40.322232  ...       0.891998             0.306299
0   Chamara Kapugedera   11.802392  ...       0.000000             0.036606
10   Seekkuge Prasanna   12.206702  ...       0.716383             0.071785
8     Dhammika Prasad    0.000000  ...       2.245266             0.140815

[11 rows x 6 columns]
WIN % : 0.6755825064795992
Match_ID 1144485 Score: 198.33656263232794 Target: 183.9852086847147
          player_name  runs_scored  ...  wickets_taken  winning_probability
0       Upul Tharanga    0.000000  ...       0.000000             0.790550
5   Niroshan Dickwella   80.080000  ...       0.000000             0.512955
6      Thisara Perera    0.000000  ...       1.006918             0.904580
1   Lahiru Thirimanne   15.843507  ...       0.000000             0.464694
3    Avishka Fernando   20.259476  ...       0.000000             0.378580
4   Dimuth Karunaratne   48.592631  ...       0.000000             0.354800
2    Dinesh Chandimal    6.771637  ...       0.000000             0.445973
10   Nuwan Kulasekara    0.000000  ...       0.986664             0.885783
9      Lasith Malinga   11.083850  ...       1.104814             0.892745
8         Isuru Udana    1.445461  ...       0.942369             0.896352
7     Shaminda Eranga    0.000000  ...       1.352033             0.904395

[11 rows x 6 columns]
WIN % : 0.9726492878316022
```

```
Match_ID 1144489 Score: 258.41874284911785 Target: 158.5547914732811
           player_name  runs_scored  ...  wickets_taken  winning_probability
5   Niroshan Dickwella    80.604643  ...       0.000000             0.970555
6       Angelo Mathews    16.457196  ...       3.218510             0.985577
4         Kusal Perera    14.441836  ...       0.000000             0.952224
0   Dimuth Karunaratne    54.152120  ...       0.000000             0.974308
1     Avishka Fernando    21.218462  ...       0.000000             0.961801
2   Chamara Kapugedera    10.927802  ...       0.000000             0.960474
3         Kusal Mendis    12.182618  ...       0.000000             0.957446
8       Suranga Lakmal    12.083909  ...       0.958944             0.984100
10       Jeevan Mendis     8.765548  ...       1.227950             0.983671
7       Lasith Malinga    11.338533  ...       1.062489             0.985108
9        Nuwan Pradeep     1.986077  ...       1.173244             0.983879

[11 rows x 6 columns]
WIN % : 0.06111149281491828
Match_ID 1144502 Score: 289.49006109625986 Target: 225.4219915486887
           player_name  runs_scored  ...  wickets_taken  winning_probability
2     Dinesh Chandimal    45.786505  ...       0.000000             0.038309
5         Kusal Mendis    53.040716  ...       0.000000             0.040047
1        Upul Tharanga    54.210783  ...       0.000000             0.039576
0   Dimuth Karunaratne   119.534398  ...       0.000000             0.088609
3         Kusal Perera    32.512125  ...       0.000000             0.036427
10      Thisara Perera    11.275461  ...       0.964864             0.048259
9     Sachith Pathirana   11.904182  ...       0.952034             0.049655
4     Avishka Fernando    14.170593  ...       0.000000             0.034186
7     Lakshan Sandakan     5.186652  ...       0.925301             0.106288
8      Dhammika Prasad     1.035922  ...       0.957349             0.069075
6       Suranga Lakmal     1.368254  ...       0.952033             0.121796

[11 rows x 6 columns]
WIN % : 0.018426787814769145
Match_ID 1144509 Score: 272.41458165571817 Target: 204.62510933287138
           player_name  runs_scored  ...  wickets_taken  winning_probability
0        Upul Tharanga   109.495436  ...       0.000000             0.018210
2    Lahiru Thirimanne    56.751849  ...       0.000000             0.018183
3   Niroshan Dickwella    54.922265  ...       0.000000             0.018183
5     Dinesh Chandimal     0.000000  ...       0.000000             0.018217
4     Avishka Fernando    46.008522  ...       0.000000             0.018183
7       Thisara Perera     0.000000  ...       0.882788             0.019002
9      Akila Dananjaya    21.298648  ...       0.785453             0.018395
8     Nuwan Kulasekara     9.870909  ...       0.853355             0.018464
1   Dimuth Karunaratne     0.995318  ...       0.000000             0.018199
6       Lasith Malinga     1.548476  ...       1.019396             0.019290
10     Dhammika Prasad     0.000000  ...       0.693483             0.018371

[11 rows x 6 columns]
WIN % : 0.019533671081547195
Match_ID 1144517 Score: 278.2407682454094 Target: 218.0481573642392
            player_name  runs_scored  ...  wickets_taken  winning_probability
0         Upul Tharanga   109.201490  ...       0.000000             0.018464
3    Niroshan Dickwella    57.033286  ...       0.000000             0.018404
5      Dinesh Chandimal    46.992747  ...       0.000000             0.018407
4          Kusal Perera    52.168453  ...       0.000000             0.018402
1     Lahiru Thirimanne    66.458541  ...       0.000000             0.018415
9   Milinda Siriwardana    35.072131  ...       0.698632             0.020733
6       Asela Gunaratne    32.557854  ...       0.689070             0.021195
8      Lakshan Sandakan     6.133135  ...       0.711982             0.020740
2    Dimuth Karunaratne     5.480930  ...       0.000000             0.018405
7           PWH de Silva    0.000000  ...       0.821203             0.021164
10    Jeffrey Vandersay     0.000000  ...       0.894317             0.020541
```

```
[11 rows x 6 columns]
WIN % : 0.018155668662698903
Match_ID 1144521 Score: 281.9192825285118 Target: 214.61913607138163
           player_name  runs_scored  ...  wickets_taken  winning_probability
0        Upul Tharanga   117.407384  ...       0.000000             0.018134
2         Kusal Perera    53.503738  ...       0.000000             0.018120
3    Lahiru Thirimanne    53.337107  ...       0.000000             0.018120
5   Niroshan Dickwella    54.250202  ...       0.000000             0.018120
4     Dinesh Chandimal    40.992846  ...       0.000000             0.018119
1     Avishka Fernando    98.193843  ...       0.000000             0.018128
8       Lasith Malinga    23.165564  ...       1.167170             0.018196
7    Seekkuge Prasanna    12.612866  ...       0.718282             0.018198
6      Akila Dananjaya    11.999326  ...       0.726556             0.018200
9     Lakshan Sandakan     5.398891  ...       0.680411             0.018192
10       Lahiru Kumara     0.060255  ...       0.773357             0.018184

[11 rows x 6 columns]
WIN % : 0.17497086064217343
Match_ID 1144526 Score: 256.54064701679107 Target: 197.14721034347983
           player_name  runs_scored  ...  wickets_taken  winning_probability
2    Lahiru Thirimanne    53.639070  ...       0.000000             0.040931
0        Upul Tharanga    98.384968  ...       0.000000             0.082027
3   Niroshan Dickwella    42.520125  ...       0.000000             0.033167
5     Dinesh Chandimal     0.000000  ...       0.000000             0.066833
7       Thisara Perera     0.000000  ...       0.955380             0.339805
4     Avishka Fernando    22.626221  ...       0.000000             0.030280
9     Nuwan Kulasekara    11.857314  ...       0.909354             0.242340
8       Suranga Lakmal    10.641334  ...       1.106110             0.250974
1   Dimuth Karunaratne     1.048026  ...       0.000000             0.046570
6       Lasith Malinga     1.563589  ...       1.110531             0.685407
10     Dhammika Prasad     0.000000  ...       0.690568             0.106348

[11 rows x 6 columns]
WIN % : 0.0303832883988305
Match_ID 1193504 Score: 216.4395676697358 Target: 173.893349942202
           player_name  runs_scored  ...  wickets_taken  winning_probability
4         Kusal Perera    48.700142  ...       0.000000             0.021963
3         Kusal Mendis    49.837692  ...       0.000000             0.021991
5   Niroshan Dickwella    25.168485  ...       0.000000             0.023514
2   Chamara Kapugedera    32.663731  ...       0.000000             0.022860
0   Dimuth Karunaratne    11.335177  ...       0.000000             0.024540
1     Avishka Fernando    11.335177  ...       0.000000             0.024540
6       Lasith Malinga     8.642841  ...       1.127446             0.049815
7     Nuwan Kulasekara    10.720657  ...       0.998816             0.039942
8    Seekkuge Prasanna     2.055141  ...       0.976227             0.037067
10        Nuwan Pradeep     1.720524  ...       0.945537             0.033332
9      Dhammika Prasad     0.000000  ...       1.105871             0.034651

[11 rows x 6 columns]
WIN % : 0.5128830637997213
Match_ID 1193505 Score: 265.85292835311384 Target: 161.22162027908962
            player_name  runs_scored  ...  wickets_taken  winning_probability
0     Lahiru Thirimanne    51.253839  ...       0.000000             0.664106
1         Upul Tharanga    50.184880  ...       0.000000             0.639611
2          Kusal Perera    48.944262  ...       0.000000             0.611080
3          Kusal Mendis    49.985817  ...       0.000000             0.602377
6        Angelo Mathews    53.749723  ...       0.865707             0.691502
5      Dinesh Chandimal    51.430490  ...       0.000000             0.633245
4    Chamara Kapugedera    33.353892  ...       0.000000             0.487545
7        Lasith Malinga     8.671777  ...       1.167761             0.658555
```

```
9      Nuwan Kulasekara    10.513118  ...       1.037705             0.225976
10  Dushmantha Chameera     9.601526  ...       0.000000             0.197727
8       Dhammika Prasad     0.000000  ...       1.243199             0.229991

[11 rows x 6 columns]
WIN % : 0.0282713142425686
Match_ID 1193506 Score: 229.5643287033664 Target: 159.9302350706621
           player_name  runs_scored  ...  wickets_taken  winning_probability
4         Kusal Perera    48.694108  ...       0.000000             0.021438
3         Kusal Mendis    50.001559  ...       0.000000             0.021454
5   Niroshan Dickwella    25.280739  ...       0.000000             0.022533
2   Chamara Kapugedera    34.144727  ...       0.000000             0.022036
10      Thisara Perera    13.798729  ...       1.007942             0.028861
0   Dimuth Karunaratne    11.221872  ...       0.000000             0.023106
1     Avishka Fernando    11.221872  ...       0.000000             0.023106
6       Lasith Malinga     8.617972  ...       1.132059             0.050831
7     Nuwan Kulasekara    10.228961  ...       0.998020             0.037535
8    Seekkuge Prasanna     2.093790  ...       0.974029             0.030984
9      Dhammika Prasad     0.000000  ...       1.104900             0.029100

[11 rows x 6 columns]
WIN % : 0.021877208123467738
Match_ID 1198487 Score: 295.49075373438257 Target: 257.0418085551721
            player_name  runs_scored  ...  wickets_taken  winning_probability
5          Kusal Perera    85.388553  ...       0.000000             0.023776
8        Angelo Mathews    97.842916  ...       0.792936             0.020200
0      Dinesh Chandimal    90.407333  ...       0.000000             0.023773
10       Thisara Perera    87.157641  ...       0.870055             0.019895
1          Upul Tharanga    90.723257  ...       0.000000             0.023758
6          Dasun Shanaka    82.855749  ...       1.040625             0.021439
2    Niroshan Dickwella    51.754160  ...       0.000000             0.022667
3          Kusal Mendis    44.868754  ...       0.000000             0.022086
9       Asela Gunaratne    32.188657  ...       0.885583             0.019911
7   Dhananjaya de Silva    25.396972  ...       0.863673             0.021257
4      Avishka Fernando     0.000000  ...       0.000000             0.021888

[11 rows x 6 columns]
WIN % : 0.01921641399859872
Match_ID 1198488 Score: 255.71898059712524 Target: 205.43823499348346
            player_name  runs_scored  ...  wickets_taken  winning_probability
3    Niroshan Dickwella    48.559021  ...       0.000000             0.018608
5          Kusal Mendis    44.516027  ...       0.000000             0.018658
2     Lahiru Thirimanne     9.840168  ...       0.000000             0.018677
10         Jeevan Mendis    30.929404  ...       0.712657             0.019466
8       Asela Gunaratne    32.161280  ...       0.745939             0.019743
6   Dhananjaya de Silva    25.562530  ...       0.758062             0.020146
7        Ashan Priyanjan    13.869683  ...       0.863890             0.020104
9           PWH de Silva    11.245387  ...       0.902129             0.019557
0    Dimuth Karunaratne    12.138665  ...       0.000000             0.018941
1    Chamara Kapugedera    10.928645  ...       0.000000             0.018908
4      Avishka Fernando     1.708172  ...       0.000000             0.018572
```