A Supervised Learning Based Approach for Predicting the Final Score in Limited Overs Cricket Matches

S.M.D.S.T. Sethunga

2019

A Supervised Learning Based Approach for Predicting the Final Score in Limited Overs Cricket Matches

A dissertation submitted for the Degree of Master of Science in Computer Science

S.M.D.S.T. Sethunga
University of Colombo School of Computing
2019

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute. To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: S.M.D.S.T. Sethunga
Registration Number: 2016/MCS/102
Index Number: 16441025

_____ _____

Signature Date

This is to certify that this thesis is based on the work of Mr. S.M.D.S.T. Sethunga under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:
Supervisor Name: Dr. A.R. Weerasinghe

_____ _____

Signature Date

Abstract

Cricket is a popular game in many countries across the world and the most popular sport in Sri Lanka. Knowing the outcome of a cricket match well before it happens has been valued by many parties involved with the game. In this study we propose a supervised learning based approach for making ball-by-ball predictions on the final score in fifty-over cricket matches. We propose an ensemble model of random forest regression to be applied for an inning segmented based on the resources remaining. Segment-wise modeling approach we followed made the overall model capable of adapting well to the different stages of an inning by locally optimizing its parameters. Our model out-performed the existing methodologies used in practice such as Duckworth-Lewis method and the run-rate method, which are also capable of making ball-by-ball predictions on the final score.

We also make ball-by-ball predictions on the winner of one-day international (ODI) matches during the second inning using a random-forest classifier. Incorporation of the prediction results from our first model as an input to the classifier proved beneficial in improving the winner prediction performance. Empirical results show that our model performs significantly better compared to the state of the art. Our model seemed unbiased regardless of the amount of target to be chased.

Acknowledgement

I would first like to thank my supervisor Dr. A.R. Weerasinghe of University of Colombo School of Computing, for supporting and backing me up throughout the research. He was always open to questions and was helpful whenever I came across any gray area. Thank you for sharing all the expertise and steering me in the correct direction.

I am grateful to my bellowed wife for providing me with unfailing support and continuous encouragement from the beginning. I am also thankful for all the domain expertise given and the inputs and the valuable comments provided. This accomplishment would not have been possible without her.

Finally, I must express my very profound gratitude to my loving parents for the continuous support provided throughout my years of study and through the process of researching and writing this thesis. Thank you.

Table of Contents

Page

1 Introduction 1
1.1 Context 1
1.2 Motivation 3
1.3 Supervised Learning 4
1.4 Problem Statement 4
1.5 Objectives 5
1.6 Scope 5
1.7 Synopsis 5

2 Literature Review 6
2.1 Predictions in Sports 6
2.2 Research on Cricket 6
2.2.1 Factors Affecting the Game 7
2.2.2 Modeling Ball-by-ball Outcome 8
2.2.3 Predicting Final Score 9
2.2.4 Predicting Winner 11
2.2.5 Resetting Targets 12
2.3 Ensemble Methods 14

3 Design and Methodology 16
3.1 Data Collection and Preprocessing 16
3.2 Terms and Notations 17
3.2.1 Current Score 17
3.2.2 End of Innings Score 17
3.2.3 Runs Scored in Remainder of the Match 17
3.2.4 Resources 18
3.3 Problem Formulation 18
3.3.1 Segmenting Criteria 18
3.3.2 Final Prediction 19
3.4 Response Variable 19
3.5 Historical Features 20

3.5.1 Average Inning Total . . . 20
3.5.2 Average Innings Total Conceded . . . 20
3.6 Instantaneous Features . . . 20
3.6.1 Batting Team . . . 20
3.6.2 Opposition . . . 21
3.6.3 Number of Balls Remaining . . . 21
3.6.4 Wickets Fallen . . . 21
3.6.5 Rate of Scoring . . . 21
3.6.6 Power-play . . . 22
3.6.7 Target . . . 22
3.6.8 Runs to Score . . . 22
3.6.9 Runs achievable from remaining resources (RARR) . . . 22
3.6.10 Runs achievable from remaining balls (RARB) . . . 23
3.7 Winner Prediction . . . 23
3.8 Training, Validation and Testing . . . 23

4 Results and Discussion . . . 24
4.1 Preliminary Analysis . . . 24
4.1.1 Impact of the inning . . . 24
4.1.2 Impact of Batting Team . . . 25
4.2 Final Score Prediction Performance . . . 26
4.3 Performance Comparison with Conventional Algorithms . . . 28
4.4 Performance Comparison with Methods Used in Practice . . . 33
4.4.1 Run-rate Method . . . 33
4.4.2 Duckworth-Lewis Method . . . 34
4.4.3 Comparison for all Matches . . . 34
4.4.4 Comparison of Individual Matches . . . 36
4.5 Performance Comparison with Previous Studies . . . 38
4.6 Winner Prediction Performance . . . 40

5 Conclusion . . . 46

6 Future Work . . . 48

References . . . 51

List of Figures

Page

1 Sample YAML data file of a single match, available at Cricksheet.org [1] . . . 16
2 Proposed model architecture . . . 19
3 Distribution of final scores for the two innings . . . 24
4 Difference in the average inning totals for the first and second innings, for different teams . . . 25
5 Error distribution in final score predictions of the proposed model, for the first inning (left) and the second inning (right) . . . 26
6 Distribution of errors in final score predictions at each stage of the match, for first inning (left) and second inning (right) . . . 27
7 Mean absolute error (MAE) in final score predictions for the two innings . . . 28
8 Distribution of the predicted scores and the actual scores for all stages of the matches, for first inning (left) and the second inning (right) . . . 29
9 Mean absolute error (MAE) for conventional algorithms, for the first innings across all the matches . . . 30
10 Mean absolute error (MAE) for conventional algorithms, for the second innings across all the matches . . . 31
11 Feature importance for the first five overs of the ensemble model during the first inning . . . 32
12 Feature importance for the overs from twenty five to thirty of the ensemble model during the first inning . . . 32
13 Feature importance for the last five overs of the ensemble model during the first inning . . . 33
14 Mean absolute error (MAE) for D/L method, run-rate method and our method, for the first innings across all matches . . . 35
15 Mean absolute error (MAE) for D/L method, run-rate method and our method, for the second innings across all matches . . . 35
16 Final score predictions during the first inning of a selected set of individual matches . . . 36

17 Final score prediction during the second inning of a selected set of individual matches 37
18 Mean absolute error (MAE) for Bailey et al. [2], Sankaranarayanan et al. [3] and our method during the first inning across all matches 39
19 Mean absolute error (MAE) for Bailey et al. [2], Sankaranarayanan et al. [3] and our method during the second inning across all matches 39
20 Accuracy of winner predictions after each ball bowled during the second inning 41
21 Variation of correct and incorrect winner predictions as the match progresses . 42
22 Precision, recall and F1-score for winner predictions across all matches 43
23 Winner prediction accuracy at the start of the second innings when chasing different target types 44
24 Winner prediction accuracy considering the entire second inning, when chasing different target types 44

List of Tables

		Page
1	Accuracy measures for final score predictions of the proposed model	26
2	Final score prediction accuracy for conventional algorithms during the first inning	29
3	Final score prediction accuracy for conventional algorithms during the second inning	30
4	Feature importance of the top ten features of the winner prediction model . . .	40
5	Winner prediction accuracy for different teams at the start of the second innings	43
6	Winner prediction accuracy for different teams throughout the second innings .	43

List of Abbreviations

ANOVA	Analysis of variance
CART	Classification and regression tree
D/L	Duckworth-Lewis
ICC	International cricket council
MAE	Mean absolute error
MOV	Margin of victory
MSE	Mean squared error
ODI	One-day international
RARB	Runs achievable from remaining balls
RARR	Runs achievable from remaining resources
RMSE	Root mean squared error
T20I	Twenty-twenty international
WASP	Winning and score predictor

Chapter 1: Introduction

Cricket is a popular game in many countries across the world and the most popular sport in Sri Lanka. As more teams are emerging to the game, the competitiveness has grown, making it harder for the teams to win matches more often than before. Thus teams tend to come up with new strategies to gain advantage over the opponent in the likes of knowing the game well ahead, and planning accordingly. With the vast popularity it has gained over the years, many fans and spectators all around the world are watching cricket and whenever a match is in progress, they also would like to stay ahead of the game and see how well their supporting teams are doing in the match. Because of this demand from the spectators for more intellect information, television broadcasters are also keen on providing comprehensive insights including predictions during the cricket matches. This shows the value of knowing the game well before for many parties involved in the game, hence this research is intended on predicting the final scores of an inning during the progression of a cricket match.

Prior to do the analysis in detail, it is important to have a clear view about the context of the research. In this chapter, we will provide a background to the field of study including a brief introduction to the game of cricket, how the game of cricket has been evolved, and how the factors affecting the game have also changed over time. Then we will look at some of the techniques used in this research. Finally we discuss the objectives and the need of such a study.

1.1 Context

Cricket belongs to the typical bat-and-ball game category where it involves two teams with eleven players in each. It is played in a cricket field – a round or an oval shaped ground, with a twenty meter long and ten feet wide rectangular playing surface at the center. Each team takes turns to bat, which is called an inning and try to score runs while the other team fields and bowls. The period in which the first team bats is called the first inning and the period in which the second team bats is called the second inning. A tossing of a coin decides which team gets to make the decision on whether to bat or bowl first. Team which scores the highest number of runs at the end of their inning is decided as the winner.

The game of cricket is played in three different formats at the international level: Test cricket, One-day international (ODI) cricket and Twenty-twenty international (T20I) cricket. In test cricket, each team gets a maximum of two innings to bat, but the match concludes after maximum

of five days, regardless of the number of innings played. There is no limitation of overs per inning, as an inning ends when the batting side gets all-out or when the batting team declares to end their inning. However, ninety overs per day is the standard acceptance if there are no interruptions occurred. In both ODI's and T20's each team is allowed to bat only once. The inning terminates when either all ten wickets have been fallen or the number of overs have been ended. The number of overs each side gets in ODIs and T20s are fifty and twenty respectively, given there have not been any interruptions to the match. However, this research will only focus on ODI cricket matches.

ODI matches are played as either day-matches or day-and-night matches. In day matches, both the innings are concluded within the day time. Day-and-night matches are started halfway through the daytime and continued under artificial lightings at night as the natural lighting fades. The first ever limited overs cricket match was played between Australia and England in 1971. A single inning was composed of forty overs with each having eight balls. Ever since the rules and regulations of the game have been evolved and the strategies and approaches of individual players as well as teams have also been changed accordingly. Nowadays ODI matches consist of fifty overs with six balls per over, given that there are no interruptions.

Introduced in 1992, a fielding restriction applies to the bowling side for a certain part of the match which limits the maximum number of fielders that can be positioned outside the thirty yard circle of the ground. At the time it was introduced, the rule allowed only two fielders to be positioned outside this thirty yard circle during the first fifteen overs of the inning, and up to five players were allowed for the rest of the inning. This restriction was modified in 2005 by dividing an inning into three phases known as powerplays. The first of the three powerplays applies for the first ten overs and the remaining two powerplays are of five overs each. Bowling team was in control over the timing of these two five-over powerplays. In 2008, batting team was granted the control of one of these five-over powerplays by an amendment to the rule. This modification lasted only for three years as the rule was changed once more in October, 2011. The second and the third powerplays were mandated to be taken between the sixteenth over and the forty-first over. In October 2012, the number of powerplays was reduced to two, keeping a mandatory ten-over powerplay at the start of the innings as before and a single five-over powerplay to be decided by the batting team. Along with that, the maximum number of players allowed to field outside the thirty-yard circle during the non-powerplay overs was also reduced to four. The final amendment to the rules was made in July 5, 2015. This made an inning a composition of three power plays again by removing the batting powerplay and making it a mandatory ten-over powerplay at the end of an inning.

These restrictions alongside the other rules of the game are decided by the International cricket council (ICC). Apart from those changes, instead of changing the ball after thirty six overs, use of two separate balls from the two bowling-ends was introduced. The prime aim of these changes was to increase the tempo of the game to gain more attraction and popularity among the fans as the T20 games are stealing the attention out from the fifty-over format. Though there is some criticism regarding these changes stating that those are favorable to the batting team and claiming that they made it further difficult for the bowling sides to restrict the runs. As the rules of the game change from time to time, the amount of runs scored by teams have vastly changed over time. Fifteen years ago, scoring three hundred in fifty overs considered being an excellent total, and more often was a winning score. But nowadays we can see that many teams surpass the three hundred mark very often, and three hundred is no longer considered as a winning score. Teams have been able to pass even the four-hundred mark, which would not have been even thought of as achievable few years back. These large totals have been further helped by the more batting-friendly conditions such as the negation of reverse-swing of fast bowlers due to the two separate balls used, making of flat-pitches where bowlers do not get any help from the pitch, grounds with smaller boundaries and the heavy bats which are way better than the earlier ones.

1.2 Motivation

Over the years, predictions in limited-over cricket have been done on many different aspects in practice. The most common type of prediction is the projected final score at any given stage of the inning, which solely depends on the run rate at that moment of the inning. But the drawback of this approach is that the final score depends not only on the current run rate, but also on the number of wickets left, batting team, the opposition and etc. Predictions are also made on the winner of the match using simple probabilities. For example, winner is predicted before a match commenced based on the results of the previous encounters of the two teams. Similarly, during the progression of a match, the winner is predicted based on simple statistics such as the number of times a team won chasing a particular target, number of times a team managed to defend a particular target and etc.

Winning and score predictor (WASP) [4] is a much more comprehensive prediction tool that has been used by the New Zealand's cricket broadcaster Sky Sports for some time. During the first inning of a limited overs match, WASP predicts the final score of that inning. In the second inning, it acts as a winning predictor which predicts the probability that the team chasing

would win. However, this tool did not gain much popularity among the other countries as its predictions seemed to fluctuate heavily as the match progresses. Therefore, this method was not adopted for international matches by the International Cricket Council (ICC). Duckworth-Lewis (D/L) method [5] of target resetting has been the most renowned and the only method that has been using in practice for international matches up to date since 2001. Despite being used for prediction, original intention of the D/L method was not to predict the scores or the winners. It was introduced to reset the targets for the team batting second by providing a cutoff mark for the runs and to decide the winner of interrupted cricket matches.

Similarly, as we will discuss in detail in the upcoming chapters, there are various drawbacks in these approaches, such as the lack of accuracy, unfairness, etc. Also the game of cricket has been evolved over time with many changes introduced to the rules of the game. Thus, this study will look at how to overcome these issues, by using a supervised learning based approach for making predictions for the final outcome of an inning, in terms of runs.

1.3 Supervised Learning

In computer science, machine learning is known as an area of using mathematical and statistical techniques for making the computer systems self-learn with the use of data. A typical machine learning system will take some data as inputs and it will learn to map each of these input data to an output. Machine learning approaches can be divided into three major categories based on the input data: supervised learning, unsupervised learning and semi-supervised learning. Supervised learning is the approach where a mapping from input data to the outputs is learned using the labeled data, whereas in unsupervised learning, the mapping of inputs and outputs is done using un-labeled data. Semi-supervised learning is used when some portion of the data has the labels and the remaining data does not have the labels. A label can be described as the expected output of that particular instance of the data. In this study we have used a supervised learning approach since the available data consists of labels, which are the final scores of each innings.

1.4 Problem Statement

The research work that has been done to predict the outcome of cricket in terms of final scores using machine learning is fairly less compared to the other sports [6] [7]. There are also a few existing probabilistic based approaches that are currently being used in practice for predictions. But most importantly, they all suffer from numerous drawbacks such as unavailability of ball

by ball predictions, lack of accuracy, unfairness etc. Thus, we can see that there is room for improvements on predictions made during a match.

1.5 Objectives

The objective of this research is to model the game of cricket to predict the final score of an inning using supervised learning to address the aforementioned problems. We will propose a method of making ball by ball predictions on the final score of an inning, with improved accuracy compared to the state of the art. We will also propose a method of using those predictions to decide the winner of interrupted cricket matches.

1.6 Scope

Our research will be focusing on making the final score predictions for ODI cricket matches. We will only be considering international cricket teams from Sri Lanka, India, Pakistan, Bangladesh, Australia, New Zealand, England, South Africa, Zimbabwe and Wes Indies out of the twelve countries that are considered as the full-members of the ICC. The two ICC full-members nations that we have omitted from the study are Ireland and Afghanistan. These two teams along with non-full-members were excluded from the study due to the lack of data as the number of matches played by those teams is low during the considered time period.

1.7 Synopsis

In this research, supervised learning based models are developed in order to predict the number of runs scored in an inning using the data of the matches played in the last ten years. In the upcoming chapters, we will discuss in detail about the process of developing such a model(s), difficulties arise in modeling this scenario as well as the adequacy of such models under different circumstances. Finally we will demonstrate some results and predictions using actual data for a set of most recent matches.

Chapter 2: Literature Review

2.1 Predictions in Sports

Analytics and predictions play a vital role in many sports and have applications in areas such as analyzing and improving team and player performances, strategic decision making before and during the games, match analysis by broadcasters as well as the betting industry. Therefore, numerous studies have been done on making predictions in various sports such as baseball [8][9][10], basketball [7][11][12], soccer and cricket.

Out of these sports, baseball belongs to the same bat-and-ball game category as cricket and has the most similarities to cricket. Markov chains have been extensively used over the years to model the run production in baseball [8][9][10]. Markov chains are considered well suited for baseball, since each play or the plate appearance can be considered a distinct state and transition probabilities from that state can be estimated given historical data. In such a study, Bukiet et al. [8] discuss about applying Markov chains on baseball to find the run distributions per half inning and per game, the optimal batting orders and the expected number of games a team should win. Nico et al. [9] model Major League Baseball (MLB) games as Markov Decision Processes (MDP) and uses Monte Carlo simulations to predict the final scores. Doing a similar study, Smith [10] predicts the scores and the winning team of MLB games using a Markov chain model.

Taking a different approach, Beneventano et al. [13] use two stepwise-multiple-regression models to predict the run-production and the run-prevention in baseball. For predicting the run-production, they used features such as number of home runs, On-base percentage, slugging percentage, number of stolen bases, batting average, etc. However, as we will discuss in the next sections, these features are specific to baseball and may not be directly applicable to a different sport, such as cricket.

2.2 Research on Cricket

Research work that has been done in cricket can be categorized into three major categories based on their main objectives: analyzing performance measures, finding an optimal strategy and predicting the outcome of a match. Many of the existing work have been done on analyzing the performance measures in cricket. Such studies are done with intentions of analyzing player

performances and ranking players [14][15], coming up with new measurements to assess player performances [15][16][17][18], etc. The second category of studies are done to acheive goals such as determining the optimal team to play in the next match [19], finding the optimal batting order [20], finding an optimal field placing for an opposition, etc.

However, given the objective of our study is to predict the final score, our interest lies in the third category out of the three mentioned above. In order to predict the outcome of a game of cricket, there have been various approaches taken by the previous studies. Some research such as [21] and [22] have predicted the winner of a match, while other research [2], [3] have been intended on predicting the final score of an inning. Some studies such as [23] have taken more granular details into account and have modeled the ball by ball outcome of the game. A noticeable characteristic of all these studies is that, despite the approach that has been taken, they all agree that there are some common factors that affect the game of the cricket, Therefore, crucial for any modeling effort.

2.2.1 Factors Affecting the Game

One of the major differences of the game of cricket from many other ball games is that, due to the nature of the sport, there are many external factors that affect the game other than solely player performances. As many other previous research show, apart from the player performances such as the average and strike rate of batsmen, average and economy of bowlers, current form, there are some external factors such as the ground, home advantage, the time of the day: whether it is morning, afternoon, or night session, the weather condition, result of the toss, etc., which also affect the game.

Jayalath [24] categorizes the factors affecting the outcome of cricket matches into two categories called controllable variables and uncontrollable variables. He identifies the factors such as playing combination, fielding strategies, aggressive playing behaviors as controllable variables, while venue, time of the day and toss result were categorized as uncontrollable variables. Author has used three models: a logistic regression model, a classification and regression tree (CART) model and a regression tree model to identify the most influential factors out of them. In the classification tree and the logistic regression approaches, he has used the win-loss factor as the binary response variable. For the regression tree approach, margin of victory which is the difference of runs by the two teams was used as the response variable.

In all three approaches they have used home-field advantage, day or night factor, coin-toss result and batting first or not as the predictor variables. For home field advantage, author has

used the geographical continent of each team instead of treating each country as a different venue. Thus, the teams were categorized into five geographical continents: Africa, America, Asia, Europe and Oceanic. This continent was then considered as a categorical variable in their model. As shown by the author, among these factors home advantage plays the most significant role in the outcome of a match for most of the teams including Sri Lanka, India, Pakistan, South Africa and New Zealand. South Africa has been the venue which favors the home team the most, whereas West Indies was proven to be the most neutral venue of all. These results were consistent in both the logistic regression model as well as CART model.

Bandulasiri [25] also used a logistic regression model to analyze the factors affecting the outcome of ODI matches. In his study, he has also used the binary variable of win or loss factor as the response variable. Opponent, winning the coin toss, day-night factor, batting first or not and whether the match was played at home or not were used as the predictor variables. He has included a second order interaction term between three variables: toss result, batting first or not and the day-night factor to the model. However, opponent was not included as part of the interaction variable to maintain the simplicity of the model. Author has used data gathered from a set of matches played during a 12 year time period from 1995 for his study. Justifying the results of Jayalath [24], Bandulasiri [25] also shows that the home advantage is the most significant factor of all, with having a winning probability four times higher than that of when playing at away venues. He further shows that the next closest factor which affects the outcome of a match after the home-advantage is the winning of the toss in day-night matches.

2.2.2 Modeling Ball-by-ball Outcome

Not many research work have been done on modeling the ball by ball outcome of a cricket match. In one such research, Swartz et al. [23] suggest a Markov Chain Monte Carlo based method to simulate the ball by ball outcome of matches with the use of a Bayesian Latent variable model. In their model they consider that there could be seven possible outcomes for a ball bowled, including falling a wicket, scoring zero, one, two, three, four and six runs. They have left out certain rare events such as scoring five runs, falling of a wicket and scoring runs in the same ball, etc. for the simplicity of the model. They also assumed that the outcome of a ball depends on the outcome of all the balls bowled up to that point from the beginning of the same inning. This would mean that the probability of outcome of each ball follows a conditional probability distribution.

They have first modeled the ball by ball outcome of first innings using data collected from 472

matches played between January 2001 and July 2006. In this attempt, while keeping the basic idea of the model as described earlier, they have also incorporated four other factors that affect the outcome of a ball including the batsmen, the bowler, wickets lost and the overs consumed into their model. With that, the probabilities could be computed for getting one of the outcomes for a particular batsman facing a particular bowler when a certain number of balls are bowled and a certain number of wickets have been fallen. However, the authors claim that calculating probabilities in such a granular level would lead to sever sparsity of data. As they elaborate, the reason behind this sparsity is that there are nearly one thousand different batsmen and bowlers and there would not be enough data to compute the probabilities for all combinations of batsmen and bowlers in different match situations. Therefore, they have categorized an inning into nine match situations depending the number of balls bowled and the number of wickets fallen. Then the actual probabilities were calculated for the different batsmen and bowlers for those different match situations to estimate the parameters of the probability distribution function.

These estimated parameters and the distribution function were then used to simulate the ball by ball outcome of an inning. As the authors highlight, one of the strong features of this model is that it can infer the behavior and simulate the outcome for various batsman-bowler combinations at different stages of a match, even if the same two players have not faced one another before. However, one of the major problems in that model is that they do not consider external factors like current score, home advantage etc., which were proven to be the most influential factors to the game by previous studies such as [24] and [25]. Other than that, since the outcome of a ball depends on the simulated outcome of previous balls that have been bowled up to that point, any error occurs at the early stage of the inning may propagate to the later part of the inning.

2.2.3 Predicting Final Score

Relatively more work has been done on predicting the final score of an inning, both before and during a match. In such a study, Bailey and Clarke [2] suggest a method for predicting the final score in ODI cricket, while the game is in progress using linear regression. First they have calculated the margin of victory (MOV) for all the matches they considered for their study. If a team batting first won the game, then the margin of victory would be the difference between the runs scored in the first inning and the runs scored in the second inning by the other team. If the team batting second won the match, then the margin of the victory would be the amount of resources left, which is identified by the number of balls left and the number of wickets left. However, since the MOV needs to be measured in a same scale for both of these scenarios, they

have used Duckworth-Lewis method of target resetting [5] to convert the remaining resources into runs in the second scenario. Then they have fitted a multivariate regression model to predict the final score of the inning. That model consists of six predictor variables including the average MOV between the two teams, average MOV against all oppositions, average past scores for the same inning, average past scores for the same inning at the same venue, average scores conceded for the same inning and the home country. Their model shows that the strongest predictor of the first innings total was the average of the past MOV between the two teams. The next strongest predictor was the average of past first innings scores by the same team.

However, this approach does not take current match situation into account, such as the number of balls bowled or the number of wickets that have fallen. Therefore, this prediction can be used only before a match is commenced. To overcome this limitation, authors have again used the D/L method to convert the number of resources available into runs at any given time of an inning. Then they combined it with the current score to compute the final score prediction, which they call the updated predicted total. They further compute a performance indicator by getting the difference of the prediction score that was calculated before the commencement of the inning and the updated prediction score that was calculated during the match. Authors claim that this indicator can be used to determine how well the batting team is performing at a given time of the match. However, the major drawback of their approach is that it is highly dependent on the Duckworth-Lewis method, which seems to be rendering a favorable decision towards one team under certain situations, according to [25]. Also, the factors they used are constant for the two teams for any match and the external factors such as the current match situation, venue, etc. are not considered.

Sankaranarayanan et al. [3] proposed a data mining approach to simulate and predict scores of one-day international cricket matches by segmenting each inning into five over segments. In each segment, they identify the runs scored using boundaries as home-runs and the runs scored by other means such as running between the wickets and extras as non-home-runs. Then they have used two separate models to predict the two types of runs. A set of attribute bagging ensemble classifiers [26] with nearest neighbor clustering were used to predict home-runs. A simple ridge regression model was used to predict non-home-runs. In the attribute bagging ensemble approach, a set of classifiers was trained using a random subset of features. The number of classifiers to be used has been determined through experimenting and the number of features per classifier was set to be the root of the total number of features. Once the classifiers were trained, authors have picked the top five neighbors based on the frequency count and the outputs of the five models were aggregated to get the final home-run prediction. For both of

these models, they have used historical features such as average number of runs scored by the team in an inning, average number of wickets lost in an inning, frequency of being all out, etc. Along with those, they have also used a set of instantaneous features such as home or away, powerplay, current score, wickets remaining, etc. for their models.

This approach has yielded an accuracy around sixty eight percent, which is better than the model by Bailey and Clarke [2]. However, the accuracy for the boundary-runs was fairly low. As the authors depict, sparsity of data could be a reason for not being able to achieve a much higher accuracy than what was achieved. Another drawback of this approach is that it can only predict the score only at every fifth over. Thus it cannot make predictions for an over in between and also cannot make any ball-by-ball prediction. Apart from that, as the prediction for a given segment depends on the prediction made on the previous segment, an error on one segment can propagate to the prediction of the rest of the segments.

2.2.4 Predicting Winner

Kampakis and Thomas [21] have suggested a machine learning based approach to predict the outcome of English county twenty over cricket matches. There they have evaluated several classification algorithms for predicting the win-loss outcome using two phases. In the first phase, they have used a set of features that reflects the overall skill level of the team such as win percentage, batting run rate of the team, bowling economy rate of the team, etc. They have also added interaction features using those primary features and a total of thirty one features were used. In the second phase they have used a different set of features that reflect individual players' skills. These include a set of base features such as batting average and strike rate for each individual batsmen, bowling average, economy and strike rate for each bowler, same metrics for different combination of batsmen, etc. Similar to the phase one, here also they have created new features by combining the base features as well as by combining different metrics. Altogether this has yielded more than five hundred features for the phase two. In both these phases some statistical techniques such as chi-square tests and Pearson correlation were used to identify the most important features out of them.

Then they have fitted four different algorithms: naive Bayes, logistic regression, random forests and gradient boosted trees using the above features. Out of all the algorithms used, highest accuracy of sixty three percent was achieved by a simple naïve Bayes learner which is around 62.5 percent and 64 percent in the two phases respectively. However, that accuracy is below what is achieved in other sports such as baseball and basketball [6], [7] according to

the authors, as well as below what was achieved by Sankaranarayanan et al. [3]. Other than that, these models do not take external factors such as venue, home team and toss result into account. As stated by the authors themselves, having those features would have yielded better performance in terms of accuracy.

As we discussed in the previous section, the study by Bandulasiri [25] was also focused on predicting the winner of ODI cricket matches. In his study he has evaluated the use of D/L method to predict the outcome of ODI matches and has compared it with the prediction made by the run-rate method. However, author has only considered scenarios where the first team bats for their entire quota of overs and the second team has to stop the play before they finish their full quota of overs. For that he has used a set of completed matches played between 1997 and 2007, and has assumed imaginary interruptions during the second inning. When such an imaginary interruption happens, the D/L method was used to find out the adjusted target score and compared it with the actual score at that point of time to find the winner. In the run-rate method, predictions are made by simply multiplying the remaining overs by the current run rate and finding out whether the team batting second will surpass the actual target or not.

As this study shows, the D/L method does a better job on predicting the winner compared to the run-rate approach. It also predicts the winner with an accuracy of around sixty seven percent at the thirty over mark and an accuracy of seventy five percent, at the forty over mark. However, as the author highlights, for a team batting first, the percentage of correctly predicting that the team loses the game, is only about fifty seven percent, which is quite low. On the other hand, if the team batting first is predicted to win the game, then it is accurate up to eighty percent. This means that the D/L method is biased towards the team batting first and has room for improvement.

2.2.5 Resetting Targets

The most renowned and the only method that has been using up to now for resetting the targets in interrupted limited-over cricket matches is the method introduced by the Duckworth and Lewis (D/L) [5]. It is based on a simple two-factor model, where the percentage of available resources is represented using the number of balls remaining and the number of wickets remaining. This relationship is considered to follow an exponential decay, where the amount of resources decreases at a rate that is proportional to the amount of resources available at that point of time. In a simpler terms, the amount of resources decreases slowly during the early stages of an inning when there are higher amount of balls left and with many wickets in hand. But the resources

start to decrease rapidly towards the later part of the inning, or when more wickets have fallen. The parameters for this exponential decay function were estimated using past match data played before 1997. However, authors have refrained from disclosing the mathematical definitions and the match information that they have used for parameter estimation due to the commercial confidentiality.

For the convenience of use, they have published a two-way table consisting of these available resources information they have calculated which is known as the Duckworth Lewis resource table. Thus when an interruption happens, the percentage of resources remaining is looked up from the table using the balls remaining and the wickets remaining. By inverting that percentage, it is possible to obtain the percentage of consumed resources and is considered to be same as the percentage of runs the team has scored so far. Using this information the final score for that particular inning can be calculated. When D/L method is used for target resetting, updated target which is also known as 'par score' for the team batting second is obtained by multiplying the first inning total by the proportion of resources consumed in the two innings. However, the drawback of their model is that it does not take other factors such as the team, venue, etc. into account. Also as shown by [25] and [22], D/L method statistically seems to be rendering a favorable decision towards the team who were batting when the interruption happens and further suggest that it could be improved.

Addressing the element of unfairness in the Duckworth-Lewis method, De Silva [22] suggests an alternative approach for resetting the targets in ODI and Twenty20 international matches. In his approach, an inning is considered as a set of segments, separated by interruptions. This was done as the tempo of the match drastically changes after an interruption, and even more if the overs were reduced. They also compute two metrics for each team called planned performance characteristic (P-characteristic) and the standard performance characteristic (S-characteristic). P-characteristic is the percentage of cumulative runs that the team plans to score as a function of the ball number. This P-characteristic is specific to each team and is a constant for all matches. The S-characteristic is the average runs that should be scored by a team in a shortened inning as a percentage of average runs scored by the same team in a full-length inning. This S-characteristic is developed using the performance of all one day international matches. Then the par-score is interpreted as a function of these two performance metrics along with other factors such as maximum number of balls in the shortened inning, number of balls allocated for the inning at the start, current ball number, final score of the first inning, etc. Using a set of ODI matches, authors show that their approach yields fairer results compared to D/L method when resetting the target in interrupted limited over matches. Also unlike in the D/L method, De Silva uses

factors related to batting teams, which makes the model yield specific results for separate teams. However, other than the two teams involved and the current match situation, his model does not take external factors such as venue, whether condition, day-night condition into account.

2.3 Ensemble Methods

An ensemble method in machine learning is the method of combining multiple learning algorithms or models to obtain better prediction performance than what could be obtained by a single algorithm or a model alone. According to [27] [28], ensembles tend to yield better results when there is a significant diversity among the models. They further claim that optimizing the weights of each base algorithm can yield substantially better generalization performance.

Ensemble methods belong to three high-level categories known as bagging [29], boosting [30] and stacking [31]. Bagging is a method of training separate learners on top of sample-sets created out of the training data set and aggregating the results of those separate models to get the final result. Boosting methods are used to improve the accuracy of a weak learning algorithm by assigning higher weights for the misclassified instances. This is done by reapplying the same algorithm several times and using weighted voting to aggregate the predictions of the resulting set of classifiers [30]. In stacking different learning algorithms are applied to a single dataset without performing any partitioning. Then the predictions of the different models are combined and using a meta-level-model to generate a final predictions [31]. Considering the fact that run production in cricket is known to have scoring patterns for different phases of an inning, a partitioning ensemble method such as bagging can be considered as the most suitable ensemble approach for our scenario.

Bagging, also known as bootstrap-aggregation, is generally done by creating k-random samples with replacements from the original dataset, producing k training sets with same size as the original data set [29]. Then separate models are trained on each of the sample and the results are aggregated based on a certain predefined criteria. This aggregation criterion can be a voting, averaging, weighted-averaging, etc. As per [29], bagging helps to reduce the variance of an algorithm and produce a more generalized result. However, because of the random sampling criteria that is used in bagging, if the dataset consists of naturally occurring groups such as run production in cricket, those will not be preserved in the resulting samples. Also, the intra-sample similarity is also not guaranteed, making the algorithm that learns on that sample unable to produce the optimal results. Attribute bagging ensemble [26] is another variation of bagging that was also used by [3], where the data is sampled by selecting random subsets of features of the

dataset. In attribute bagging models, apart from the features, data points can also be sampled similar to normal bagging methods. However, such ensembles will also have the same problem as the normal bagging method, as we discussed previously.

Chapter 3: Design and Methodology

This chapter describes the approach we followed in our study for modeling ODI cricket matches to predict the final score. It contains details the used data set, terms and notations used, and the problem formulation. Further we discuss in detail about the model architecture and the features used in the model.

3.1 Data Collection and Preprocessing

Ball by ball data is available as commentary logs in Cricinfo [32]website, for all international cricket matches. The raw data extracted from those commentary logs is available at Cricsheet.org [1], for all international matches played from 2006 to 2017 inclusive. These raw data includes features such as the over number and ball number, batsman, bowler, runs off the bat, extras, kind of wicket (if any), etc. Figure 1 shows a sample YAML data file obtained from Cricsheet.org [1].

```yaml
meta:
  data_version: 0.9
  created: 2017-07-10
  revision: 1
info:
  city: Hambantota
  dates:
    - 2017-07-10
  gender: male
  match_type: ODI
  outcome:
    by:
      wickets: 3
    winner: Zimbabwe
  overs: 50
  player_of_match:
    - Sikandar Raza
  teams:
    - Sri Lanka
    - Zimbabwe
  toss:
    decision: field
    winner: Zimbabwe
  umpires:
    - IJ Gould
    - RR Wimalasiri
  venue: 'Mahinda Rajapaksa
      International Cricket Stadium,
      Sooriyawewa'
innings:
  - 1st innings:
      team: Sri Lanka
      deliveries:
        - 0.1:
            batsman: N Dickwella
            bowler: Sikandar Raza
            non_striker: MD
                Gunathilaka
            runs:
              batsman: 2
              extras: 0
              total: 2
        - 0.2:
            batsman: N Dickwella
            bowler: Sikandar Raza
            non_striker: MD
                Gunathilaka
            runs:
              batsman: 0
              extras: 0
              total: 0
        - 0.3:
            batsman: N Dickwella
            bowler: Sikandar Raza
            non_striker: MD
                Gunathilaka
            runs:
              batsman: 0
              extras: 0
              total: 0
```

Figure 1: Sample YAML data file of a single match, available at Cricksheet.org [1]

As the ball by ball data is available, we have considered the data associated with each ball bowled in each match as a single instance of data. Thus, for a completed inning with fifty overs, there would be three hundred data points, if no extra balls were bowled. This would change if extra balls are bowled, if the match was interrupted or if the team was all out before the 50-over mark. We considered the matches played by international cricket teams from Sri Lanka, India, Pakistan, Bangladesh, Australia, New Zealand, England, South Africa, Zimbabwe and West Indies are considered. Other international teams such as Kenya, Ireland, etc. were excluded from the study as they have played only a very limited number of matches at the international level throughout the considered time period and therefore do not have enough data.

3.2 Terms and Notations

A set of important terms and notations used in our model are described below. Some of these terms are selected to be consistent with the previous studies [3] and [5] wherever applicable. Throughout the next chapters, the term R is used to denote the number of runs scored. Different subscripts of R denote different aspects of runs that we considered in the study.

3.2.1 Current Score

Current score is the amount of runs scored so far by a team, at any given stage of an inning. Given a match situation, the current score in the I^{th} inning is denoted by R_c^I. However since we are considering an inning at a time, the superscript I is dropped from the notation and R_c is used instead.

3.2.2 End of Innings Score

The amount of runs scored by a team at the end of their inning is known as the end of innings score and denoted by R_{eoi}. An inning may end when the batting team runs out of their full quota of resources. Additionally, for the team batting second the inning would conclude if they reach the target to be scored. The corresponding estimated value of R_{eoi} is denoted by $\hat{R_{eoi}}$.

3.2.3 Runs Scored in Remainder of the Match

At a given time of an inning, R_r denotes the amount of runs that will be scored by the batting team in the remainder of the match. This is equivalent to $R_{eoi} - R_c$. The corresponding estimated value is denoted by $\hat{R_r}$.

3.2.4 Resources

The combination of balls and wickets is identified as the resources. A team utilizes the balls and wickets to accumulate runs. In other terms, runs are scored by utilizing the resources. Amount of resources remaining when b balls left and w wickets fallen is denoted by $RS_{(w,b)}$. $RS_{(w,b)}$ decreases as the inning progresses. As we discussed in the section 2.5, these resource information against b and w can be found in the Duckworth Lewis [5] resource table. In a 50-over inning, the amount of resources remaining at the start of the inning $RS_{(0,300)}$ is 100%. At the end of the inning, the remaining resources could be either $RS_{(10,b)}$ or $RS_{(w,0)}$. Both of these are equivalent to zero. At any stage of a match, the amount of remaining resources $RS_{(w,b)}$ can be looked up from this table using the balls bowled and the wickets that have been fallen at that point of the game.

3.3 Problem Formulation

Predicting the final score can be broken down to two cases: (a) predicting the first inning score and (b) predicting the second inning score. At the first glance it may look as if both the cases look identical. However the prime difference in the two scenarios is the availability of the target score during the second inning. Also as we will discuss in detail in the section 4.1.1, the distribution of runs scored in the two innings show different characteristics altogether. Considering these into account, we trained separate models for the two innings. During an inning, the gameplay by the batting side changes as the inning progresses. For an example, the batting approach and the aggression of batsmen at the start of an inning are different to that of during the middle part of the inning, as well as during the last few overs. Similarly, if a team loses too many wickets at the start, they tend to play with more defensive approach thereafter. Our analysis shows that a single model for an entire inning does not adapt well to these changes at different stages of the inning. Therefore, to capture this variation of gameplay, we split an inning into ten segments, and trained separate random forest regression models for each of the segment in the suggested approach.

3.3.1 Segmenting Criteria

As discussed in the earlier section 3.3, the motivation behind segmenting an inning is to make the model capable of capturing the match situation more precisely. We identified that the batting approach and the aggression largely depends on the number of balls left and the number of wickets in hand. In other terms, it is dependent on the amount of resources left for the batting

team. Therefore, to consider both of these factors into account, we calculated the percentage of resources remaining $RS_{(w,b)}$ at the end of each ball bowled, using the DL resource table. Then each inning was segmented into ten equal length segments based on the $RS_{(w,b)}$.

3.3.2 Final Prediction

For a given instance of data, only one of the ten models will make a prediction and the remaining nine models will be idle. These ten models are fronted with a model selector, which directs the incoming data to the relevant model based on the segment that the data point belongs to. Following figure 2 shows the overall design of the proposed model.

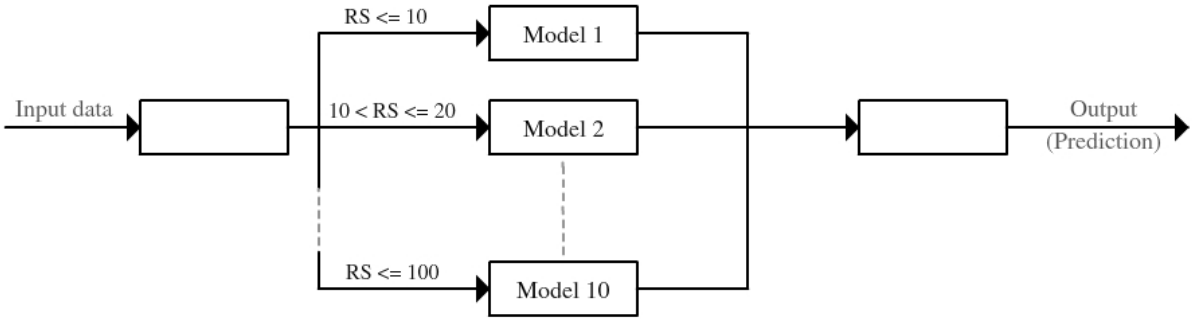

Figure 2: Proposed model architecture

3.4 Response Variable

As the objective of our study is to predict the final score of the inning, one of the below features could be considered as the dependent variable:

- Directly predict the final score - response variable would be $\hat{R_{eoi}}$.
- Predict the number of runs scored in the remainder of the inning ($\hat{R}_r$). Then combine it with the current score to get the final score prediction.

The key point to consider when selecting the response variable is, if the inning gets interrupted and the overs are reduced, then the final score should be predicted considering the reduced overs. However, considering the final score for the entire inning does not reflect the impact of reduced overs. Therefore, we followed the second approach by treating the amount of runs scored in the remainder of the inning at any given time as the response variable. Once $\hat{R}_r$ is obtained, $\hat{R_{eoi}}$ can be calculated using the following formula (1).

$$\hat{R_{eoi}} = R_c + \hat{R}_r \tag{1}$$

To model $\hat{R}_r$, we considered two types of independent variables known as historical features and instantaneous features.

3.5 Historical Features

Historical features include the statistics obtained from past matches between the two rival teams of the current match. Given a match, these features remain constant throughout the progression of the match.

3.5.1 Average Inning Total

Average inning total represents the average number of runs scored by a given team against the given opposition T, during the last 20 matches.

$$Avg.\,inning\,total = \frac{\sum_{i=1}^{20} inning\,total\,in\,match\,i\,against\,opposition\,T}{20} \tag{2}$$

This feature captures dominance of a given team over a given opposition. A moving window of most recent 20 matches are considered to capture the current batting form as a team, while making sure that there is enough data to eliminate the effect of any anomalies.

3.5.2 Average Innings Total Conceded

This represents the average number of runs conceded by the opposition team T, against any team during the last 20 matches. This feature captures bowling performance and the current form of the opponent team.

$$Avg.\,inning\,total\,conceded = \frac{\sum_{i=1}^{20} inning\,total\,conceded\,by\,opposition\,T\,in\,match\,i}{20} \tag{3}$$

3.6 Instantaneous Features

The instantaneous features consist of data that reflect the current match situation. Features described in sections 3.6.1 to 3.6.6 are applicable for both innings. Hence those were included in models used for both first inning and the second inning.

3.6.1 Batting Team

Batting team could be one of the ten countries discussed in section 3.1. It is one-hot encoded, resulting ten separate binary features.

3.6.2 Opposition

Opposition represents the team bowling at a given inning. This also takes the same set of values as the batting team, and hence results in ten separate binary features, after one-hot encoding.

3.6.3 Number of Balls Remaining

This represents the number of balls remaining in the match, which is the difference between the quota of balls allowed and the current number of balls bowled. In a typical match, the quota of balls allowed to be bowled is equivalent to 300, and hence can range from 0 to 300. In case the number of overs in the match were reduced due to some interruptions such as rain, etc., then the ball count of that reduced overs is considered as the total number of balls allowed. However, there can be situation where at the start, the match was planned to play with the full quota of 50 overs and the team was batting with the intention of batting out all the 50 overs. But in the middle of the inning it can get interrupted by rain and the overs might change from there onwards. In such scenarios, we still considered the reduced overs as the full quota of allowed balls to be bowled throughout the match.

3.6.4 Wickets Fallen

The number of wickets fallen in the inning so far can range from zero to ten. However, a batsman might retire due to an injury while batting and may never return to bat in the same inning. Then the number of wickets that are effectively remaining is one less than the actual number of wickets remaining. Since these are very rare events, such situations are not handled in our study.

3.6.5 Rate of Scoring

Rate of scoring is typically measured by the run-rate. Run-rate is calculated by dividing the number of runs scored during a time period by the number of overs bowled during that time period. A certain number of overs are treated as the time period to calculate the rate. Thus the run-rate during the last n overs when the total number of overs bowled is t, can be obtained using the below formula (4).

$$runrate = \frac{\sum_{i=t-n}^{t} Runs\ scored\ in\ i^{th}\ over}{n} \tag{4}$$

We calculated run-rate for three different time periods, in-order to capture the rate of scoring in different aspects.

- The overall run-rate – run-rate calculated considering all the overs bowled so far. Here $n = t$ in an uninterrupted match.
- Run-rate in the last over ($n = 1$).
- Run-rate over the last five overs ($n = 5$).

3.6.6 Power-play

This feature is a binary variable indicating whether the current over in operation belongs to the power-play or not. However, power-plays have been changing over the years very frequently, and only the first power-play has been fixed throughout. Therefore, we only took the first power-play into account, as our dataset did not have enough information to include the remaining power-play information.

Apart from the above features, there are some more instantaneous features that are only available for the second inning. Following are such features.

3.6.7 Target

Target is the number of runs to be scored to win the game. This is equivalent to one more than the total number of runs scored by the team batting first by the end of the first inning. This is given by formula (5).

$$target = R^1_{eoi} + 1 \tag{5}$$

3.6.8 Runs to Score

Runs to score is the number of runs left to be scored at any time of the match.

$$runs\_to\_score = target - R_c \tag{6}$$

3.6.9 RARR

This feature represents the amount of runs that can be achieved using the remaining resources. This is equivalent to the DŁ[5] estimation of runs that would be scored during the remainder of the inning, and is calculated using the equation (7).

$$RARR = \frac{target}{total\_resources} \times resources\_remaining \tag{7}$$

3.6.10 RARB

RARB is the amount of runs that can be achieved using the balls remaining. This is equivalent to the estimation of runs that would be scored during the remainder of the inning given by the run-rate method.

$$RARB = \frac{target}{total\_balls} \times balls\_remaining \tag{8}$$

3.7 Winner Prediction

Need for predicting the winner in cricket matches usually arise when one team has already batted and the other team has not completed their quota of overs or the wickets. Thus, we used a random forest classifier model to predict the winner during the progression of the second inning. Same set of features used for the final score prediction model during the second inning was used for this classifier as well. Apart from those, the predicted score from the final score prediction model was also included as an input feature to the winner prediction model. Similar to the final score prediction model proposed in section 3.3.2, this classifier also predicts the winner at the end of each ball bowled.

3.8 Training, Validation and Testing

Considering the evolution of the game, we picked the data of the matches played in the last five years out of all the available data. This includes 427 international matches played between January 2013 and June 2017. Matches that were abandoned due to various reasons such as rain, bad light, etc. were omitted from the study. Out of the selected matches, most recent 50 matches were filtered out as the holdout sample for testing and evaluating the results. During the model building, we used 10-fold cross validation for parameter tuning.

Chapter 4: Results and Discussion

4.1 Preliminary Analysis

Before start on training a predictive model, we carried out some preliminary data analysis on the data set. The object was to get a better understanding of the data, and to see what features would be important in predicting the final score.

4.1.1 Impact of the inning

Before start modeling the game, we explored the impact of the inning of the match to the final score. As in the Figure 3, the final scores are distributed differently for the two innings. The amount of runs scored in the first innings is slightly ahead of the runs score during the second innings. Averages of the final scores for the two innings also posses a clear difference. Sides batting first score around 253 on average, whereas teams batting second only score around 211 on average. To further validate whether this difference is significant enough, we carried out a student T-test, having the null hypothesis as "the average runs scored in the two innings are equal". Following are the results we obtained from the test.

$$T\ Statistic = 8.043$$

$$p - value = 5.314e - 14$$

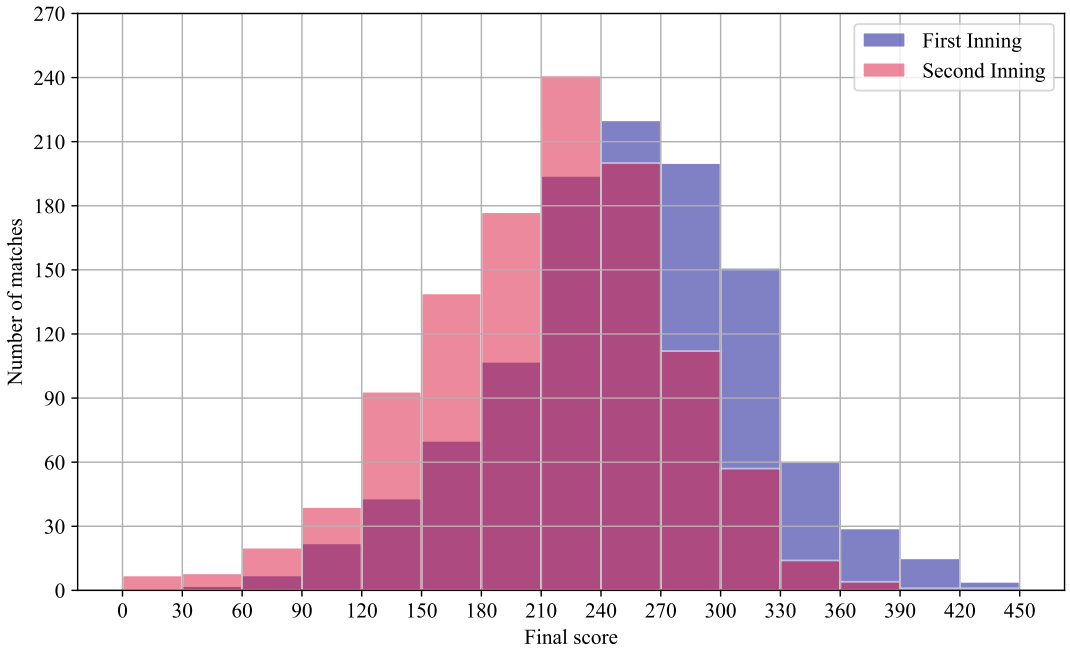

Figure 3: Distribution of final scores for the two innings

This p-value is well below 0.005 and indicates that there's enough evidence to reject the null hypothesis (with 99% confidence), which confirms that the average runs scored in the two innings are not the same. This justifies the use of separate models for the two innings in our approach.

4.1.2 Impact of Batting Team

DL method as well as run-rate method does not take the batting team into account. Treating all teams equally has been a strong factor for these two methods to gain the popularity and the public acceptance. However, our analysis shows that there is a difference in the performances by different teams. Figure 4 shows the average runs scored in an inning by each team separately for two innings. As it can be clearly seen in the graph, different teams perform differently in terms of the average inning score. Also as we observe previously, most of the teams do well in the first inning compared to the second inning. Interestingly, Zimbabwe seems to be the only team that does better in the second inning compared to the first inning. Further, to clarify whether this difference in performance is significant enough to be included to the model, we carried out a one-way Analysis of variance (ANOVA) test and the result was as below.

$$Statistic\,(F-ratio) = 2463.108$$

$$p-value = 0.0000$$

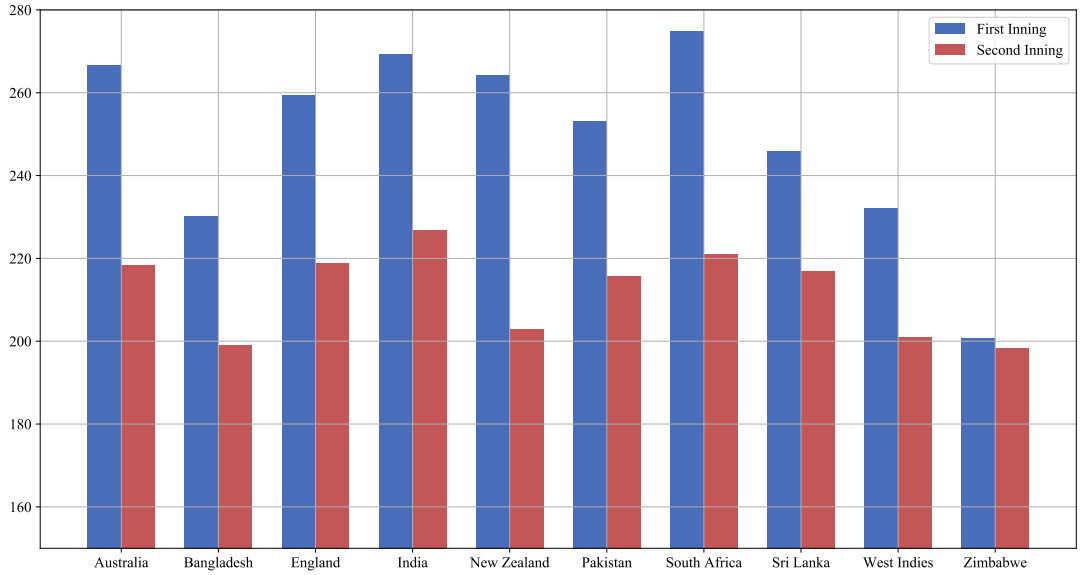

Figure 4: Difference in the average inning totals for the first and second innings, for different teams

Thus the p-value for the two-tailed test is well below 0.005, hence there's enough evidence (with 99% confidence) to conclude that the average runs scored by different teams is significantly different. This shows that treating all teams in the same manner is unfair, hence justifies our decision to include team specific information in our model.

4.2 Final Score Prediction Performance

Using the predictions made from our model, we calculated the overall Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) for all the matches and the results are available in Table 1. The prediction performance of the model during the second inning is slightly higher compared to the first inning. Figure 5 shows the distribution of mean errors per inning across all the matches. Mean error per inning was calculated by averaging out all the errors obtained by the predictions made after each ball bowled. These mean errors were calculated separately for the two innings, for all the matches. For first innings, only 18% of the matches have a mean absolute error which is less than 10 runs. For the second innings 46% of the matches have a mean absolute error less than 10 runs. This is a significant difference between the two innings, which suggests that the second inning predictions are more precise compared to the first innings.

Table 1: Accuracy measures for final score predictions of the proposed model

Inning	MSE	RMSE	MAE
First inning	1007.45	31.74	24.67
Second inning	835.48	28.90	18.77

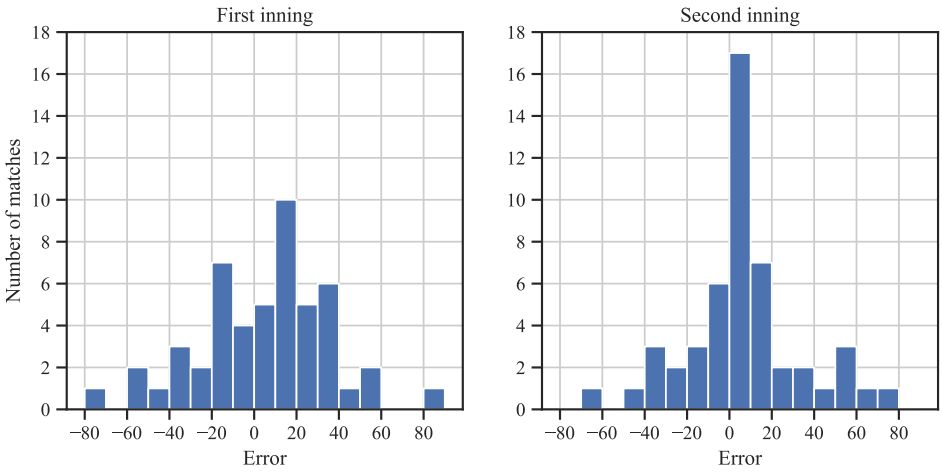

Figure 5: Error distribution in final score predictions of the proposed model, for the first inning (left) and the second inning (right)

Figure 6 shows the distribution of prediction errors, at each stage of the inning for all the matches. Despite the first inning having a lower spread, the errors are equally distributed in the range. Whereas in the second inning, even though there is a wider spread of errors at the start, a higher percentage of errors are cluttered around zero. This has resulted in a higher accuracy during the second inning. This is consistent with the error distribution in Figure 5, in which the percentage of errors around zero is much higher in the second innings. In both cases, errors at the beginning of the inning are widely spreaded. As the inning progresses, errors are getting lower and the spread become more concise. This is an expected behavior since as the match progresses, it gives a better idea on how the inning is progressing and hence the accuracy increases.

Figure 7 shows the mean of the absolute error values after each ball bowled. Second inning prediction errors converge to zero much faster compared to the first inning. The first inning take 30 overs to reach an error of 25 runs and 44 overs to reach an error of 15 runs. However, in the second innings, the error reaches 25 runs by the 17th over, and goes below 15 runs by the 39th over. This higher prediction accuracy could be due to the availability of the 'target' to be scored, as it gives the ceiling to the runs that could be scored. Also the teams batting second tend to maintain a constant scoring rate and try to keep up with the required rate. Thus the fluctuations in the run scoring can be said to be less compared to the first inning. On the other hand, teams batting first have no limit of runs to be scored. Therefore they tend to score as much as possible using the resources remaining. The amount risks taken by the batsmen to score are slightly higher during the first inning. This increases the unpredictability and thus a

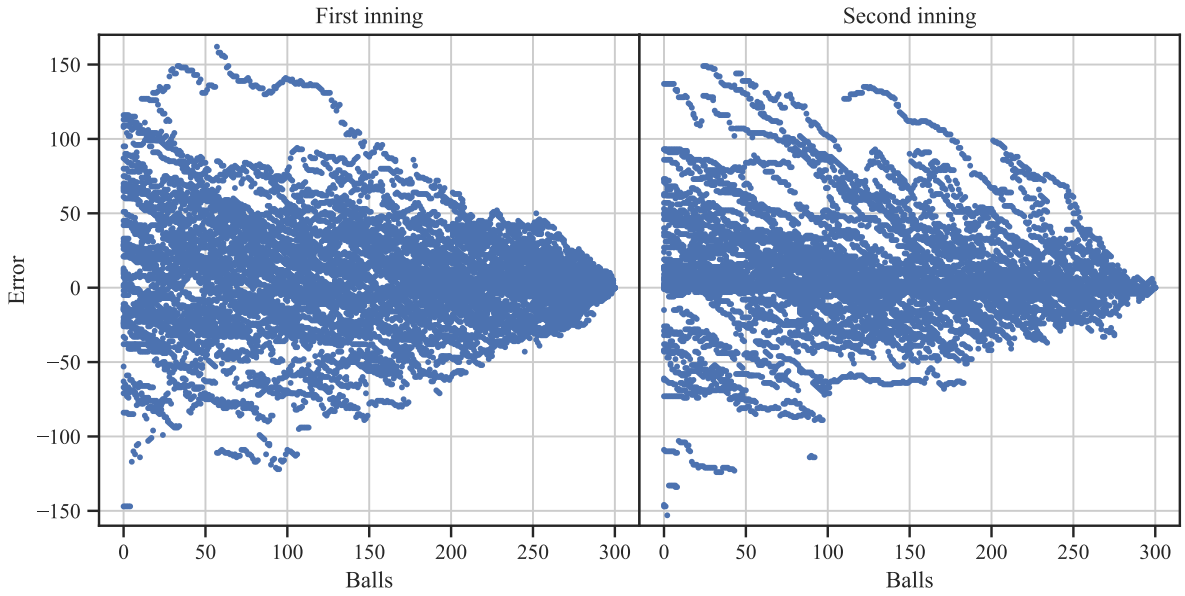

Figure 6: Distribution of errors in final score predictions at each stage of the match, for first inning (left) and second inning (right)

higher error rate was observed in the predictions.

Scatterplot in the Figure 8 shows the relationship between the predicted runs and the actual runs scored at all stages of each match, for the two innings. As in the figure, these values mostly lie along the diagonal which suggests that the predictions are in-lined with the actuals.

4.3 Performance Comparison with Conventional Algorithms

We trained several conventional regression models using linear regression, ridge regression, lasso regression, random forest regression and gradient boosted tree regression to predict the final score. The objective was to observe the behavior of conventional algorithms in predicting the final score. We also evaluated the use of deep neural networks for modeling the final score. A single model was trained from each of the algorithms for the two innings separately, as opposed to the segment wise model building we followed. Table 2 and 3 shows a comparison of accuracy meassures (MAE, MSE and RMSE) between the proposed model and the conventional algorithms. Figure 9 and 10 shows the comparison of the accuracy of the proposed model against the conventional algorithms for the two innings separately.

As we can see from the tables 2 and 3, segment-wise model training approach we followed has yielded the best results with the lowest error rates out of all. As it is noticeably from figures

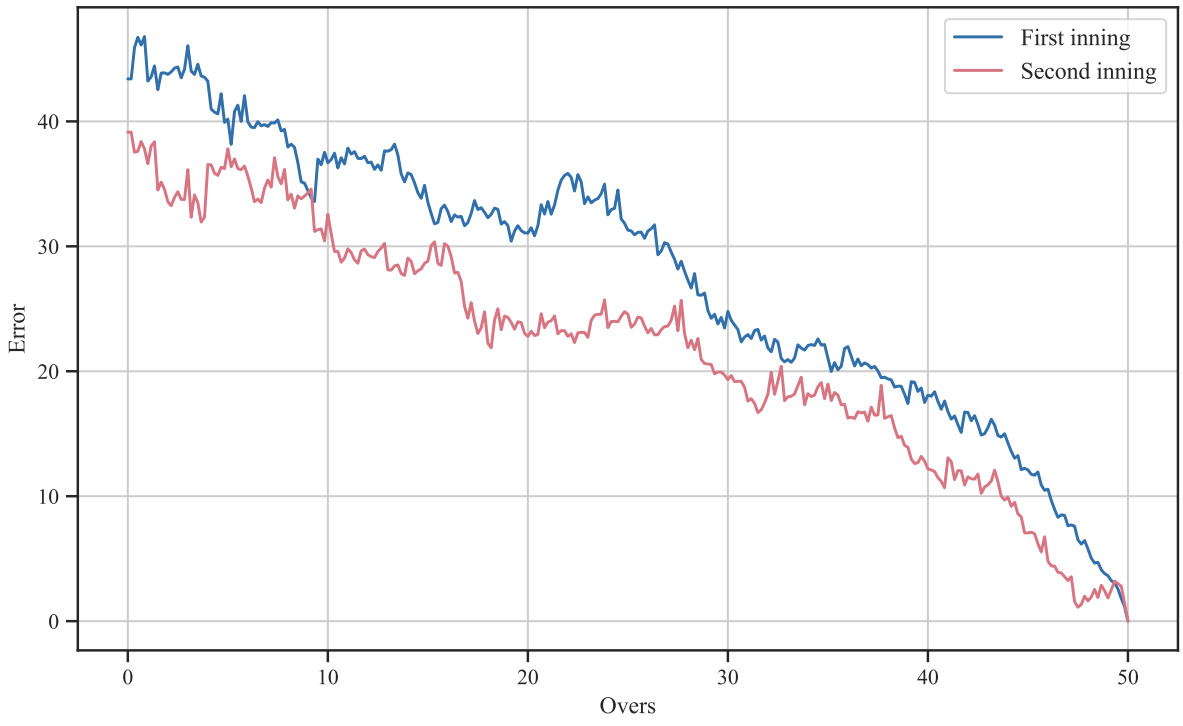

Figure 7: Mean absolute error (MAE) in final score predictions for the two innings

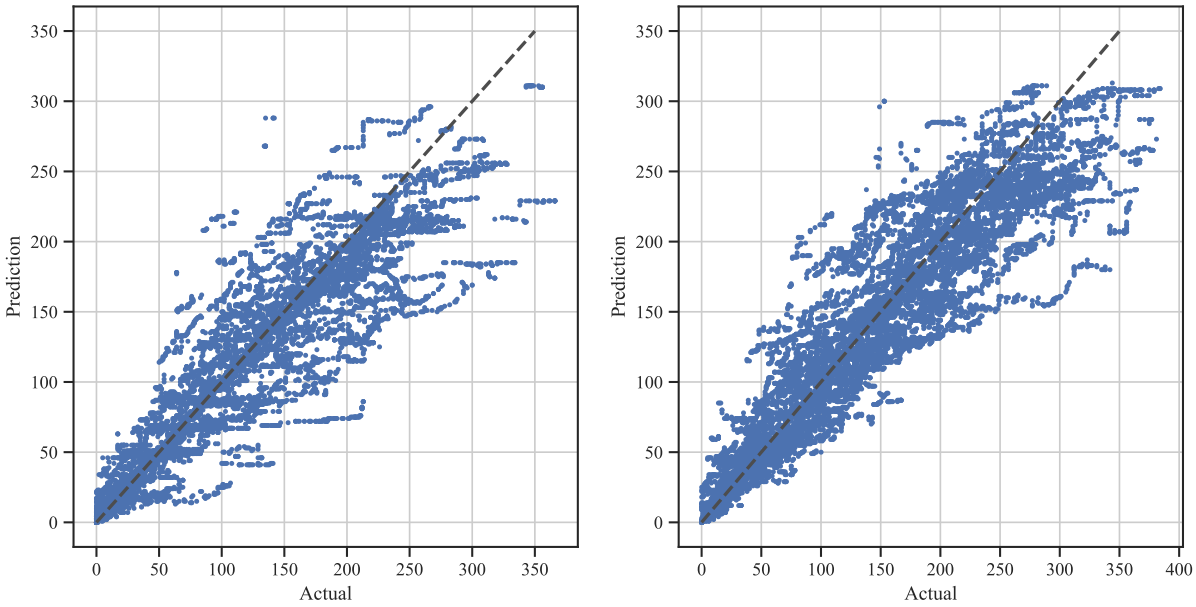

Figure 8: Distribution of the predicted scores and the actual scores for all stages of the matches, for first inning (left) and the second inning (right)

9 and 10, in the conventional algorithms the prediction error converges to zero as the match progresses, but suddenly increases after halfway through the inning. The error never converges to zero at the end of the 50 over mark. This means that the models predict that more runs would be scored even after the inning has concluded. This behavior exists in both the first and the second innings. The reason for this is, since a single model was trained for the entire inning, the weights of the input features of the model get generalized for the entire inning. However, these generalized weights would not be the optimal values for the different stages of the match. By segmenting an inning, our model has been able to locally optimize its features to suit the different stages of the inning.

Table 2: Final score prediction accuracy for conventional algorithms during the first inning

Algorithm	MAE	MSE	RMSE
Proposed model	**24.67**	**1007.45**	**31.74**
Linear regression	26.71	1174.21	34.27
Ridge regression	26.84	1211.67	34.81
Lasso regression	26.71	1183.86	34.41
Random Forest regression	25.97	1219.28	34.92
Gradient boosted tree regression	27.71	1295.23	35.99
Neural network (Deep learning)	31.12	1648.76	40.60

Table 3: Final score prediction accuracy for conventional algorithms during the second inning

Algorithm	MAE	MSE	RMSE
Proposed model	**18.77**	**835.48**	**28.90**
Linear regression	31.34	1747.44	41.80
Ridge regression	32.12	1805.24	42.49
Lasso regression	31.35	1750.27	41.84
Random Forest regression	32.72	1911.75	43.72
Gradient boosted tree regression	31.91	1877.33	43.33
Neural network (Deep learning)	25.45	1068.93	32.69

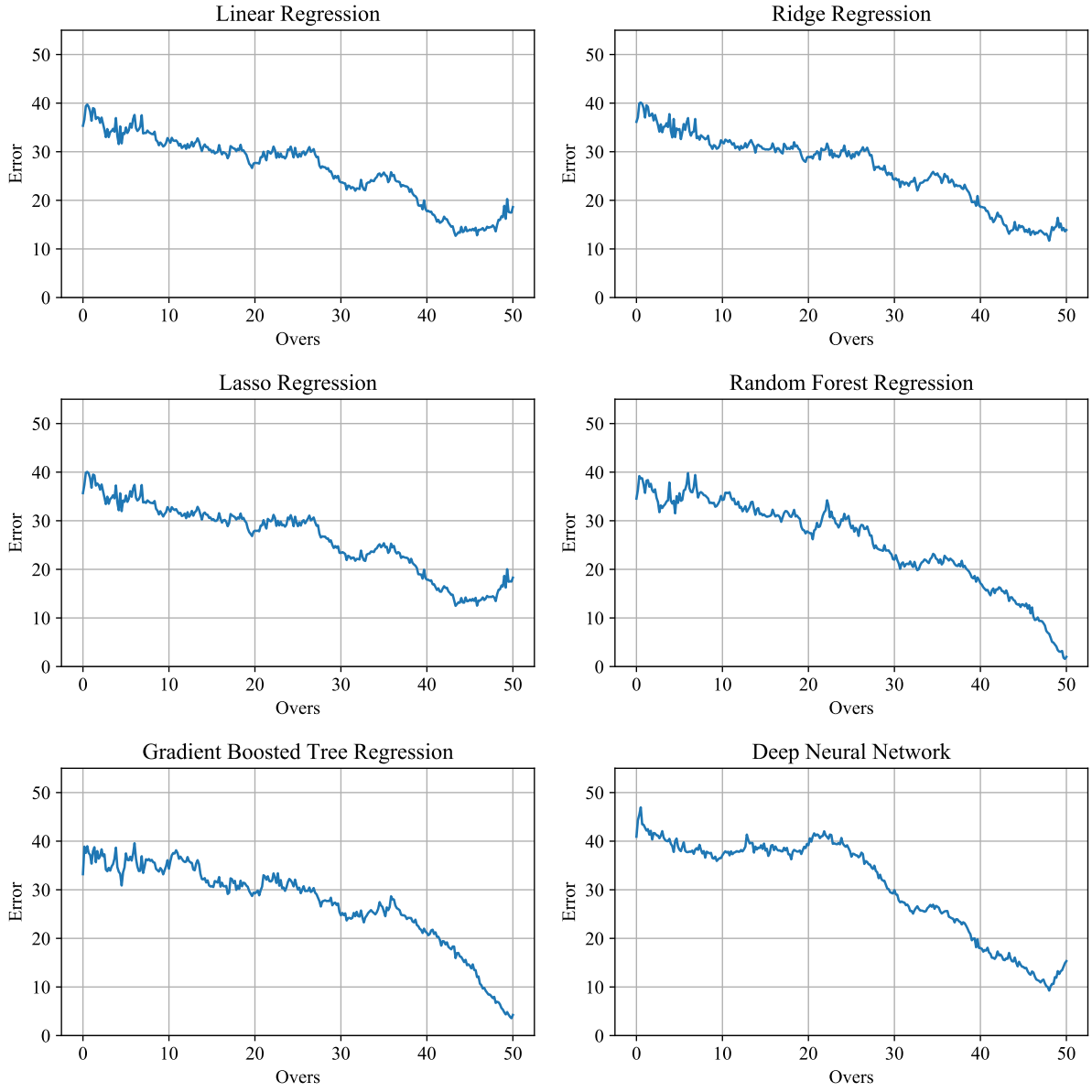

Figure 9: Mean absolute error (MAE) for conventional algorithms, for the first innings across all the matches

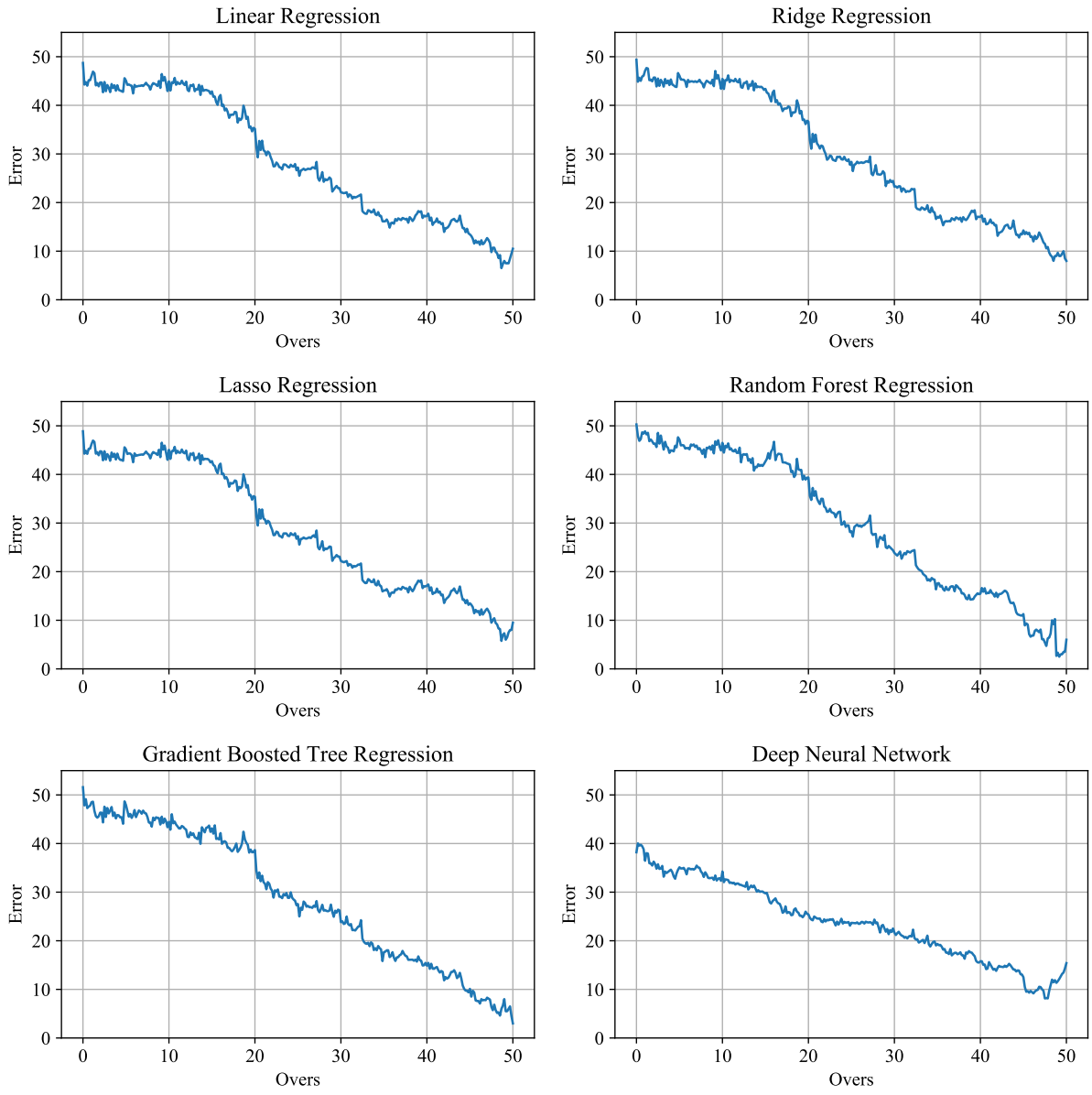

Figure 10: Mean absolute error (MAE) for conventional algorithms, for the second innings across all the matches

To further validate this, we analyzed the weights associated with each of the input feature of the proposed model at separate stages of the match. Figure 11 and Figure 12 show the feature importance of our model at three different stages of the first inning. For the first five overs avg_score (average amount of runs scored by the team, against the same opponent, during the last 20 matches) has the highest importance of all features. Other features have a very negligible amount of importance. During the overs between twenty-five and thirty, the importance of avg_score has gone down slightly and the $wickets\_remaining$ has gained a considerable amount of importance (figure 12). This is consistent with the fact that teams considering wickets as the most important factor during the middle part of the inning. Batting teams try not to

lose wickets at this stage and try to preserve the wickets for the later part of the inning, while the bowling team try to take wickets and break the momentum. This also shows that taking wickets during the middle overs is the best way to limit the batting team to a lower score. During the last five overs, $balls\_remaining$ has become the most important factor and rest of the features has become almost negligible according to the figure 13. Again, this is consistent with the known behavior in cricket, where teams try to get the maximum runs out of the remaining few overs, despite the number of wickets fallen. In overall, these results show that the segment wise model training approach we followed is more suitable than the conventional single model training approach for modeling the final scores.

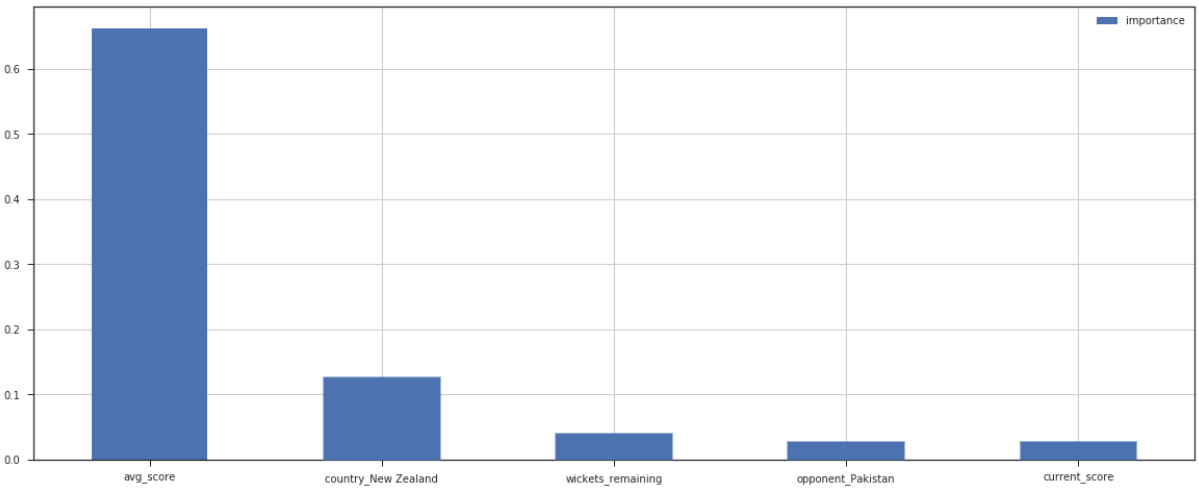

Figure 11: Feature importance for the first five overs of the ensemble model during the first inning

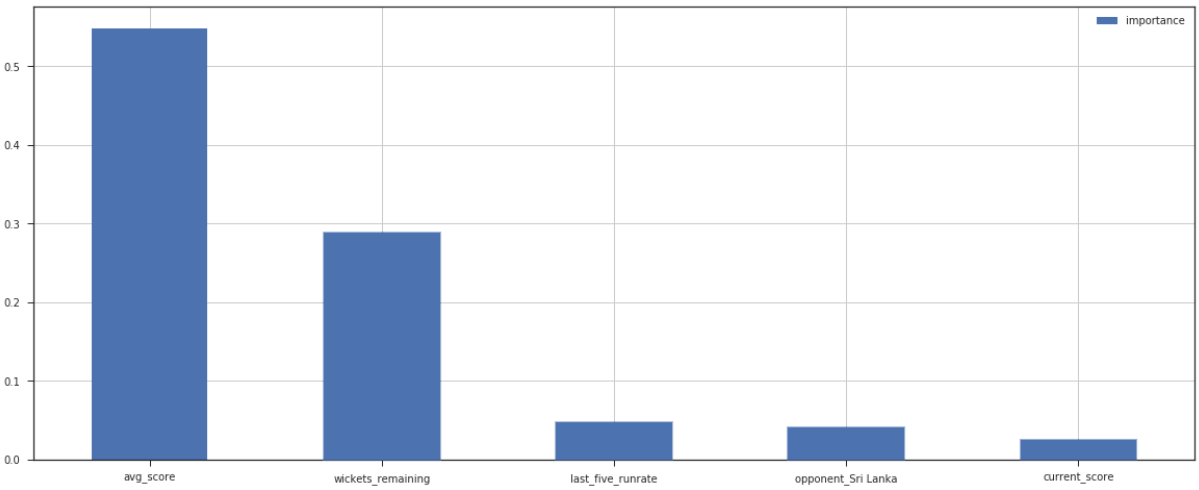

Figure 12: Feature importance for the overs from twenty five to thirty of the ensemble model during the first inning

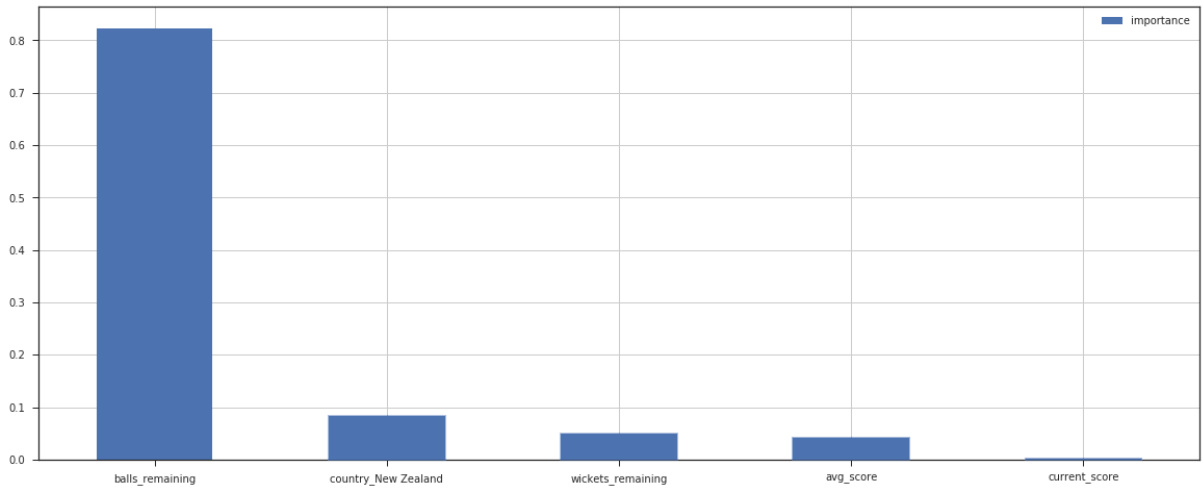

Figure 13: Feature importance for the last five overs of the ensemble model during the first inning

4.4 Performance Comparison with Methods Used in Practice

The Duckworth-Lewis method [5] and the run-rate method are the two methods that are currently being used in practice at the international level for setting targets. Both of these methods are capable of providing a projected final score at the end of each ball bowled. Here we will show that the machine learning based approach we followed yields better results compared to the conventional approaches used in practice.

4.4.1 Run-rate Method

Run-rate method is currently being used by all international matches to provide a projected score during the course of the play. This projected score is simply calculated by multiplying the current run-rate by the total number of overs allowed. This can be obtained using the following formula.

$$\hat{R_{eoi}} = \frac{R_c}{O_c} \times 50 \tag{9}$$

A slight variant of this equation (9) is to replace the current overs and the total number of allowed overs by the current legal balls bowled (b) and the total number of allowed legal balls (300), respectively. Then the equation becomes:

$$\hat{R_{eoi}} = \frac{R_c}{b} \times 300 \tag{10}$$

We used this variant of the equation (10) since our objective is to calculate final score projection at the end of each ball bowled. In both these cases, we have assumed that the total number

of overs is 50. However, this would change in rain affected games, or any shortened game due to other reasons.

4.4.2 Duckworth-Lewis Method

As discussed in section 3.2.4, the Duckworth-Lewis method [5] provides a way to calculate the amount of available resources $RS_{(w,b)}$, using the number of balls remaining and the number of wickets fallen. Using this information, it is possible to calculate R_{eoi} using the following formula.

$$\hat{R_{eoi}} = R_c + R_r \tag{11}$$

Since the ratio between R_r and R_c is equal to the ratio betwen the remaining resources and the comsumed resources, R_r in the equation (11) can be replaced by R_c to get equation (12):

$$\hat{R_{eoi}} = R_c + \frac{R_c}{(100 - RS_{(w,b)})} \times RS_{(w,b)} \tag{12}$$

Above equation (12) can be further simplified to the following equivalent form (13).

$$\hat{R_{eoi}} = \frac{R_c}{(100 - RS_{(w,b)})} \times 100 \tag{13}$$

4.4.3 Comparison for all Matches

We used equations (10) and (13) to calculate the final scores projected by the run-rate method and the D/L method respectively, to compare with the results of our model. Using the hold-out sample as the test set, we predicted the final score using the all three approaches: run-rate method, D/L method, and our proposed method. Then we calculated the mean absolute error (MAE) at each stage of the match for the three methods. Figures 14 and 15 shows these MAE values for the two innings separately. Our model has the lowest error out of the three methods, and have signifcantly outperformed the state of the art. Both D/L and run-rate methods have an extremely high error at the start of the inning, and the error reaches close to our model only towards the 40 over mark.

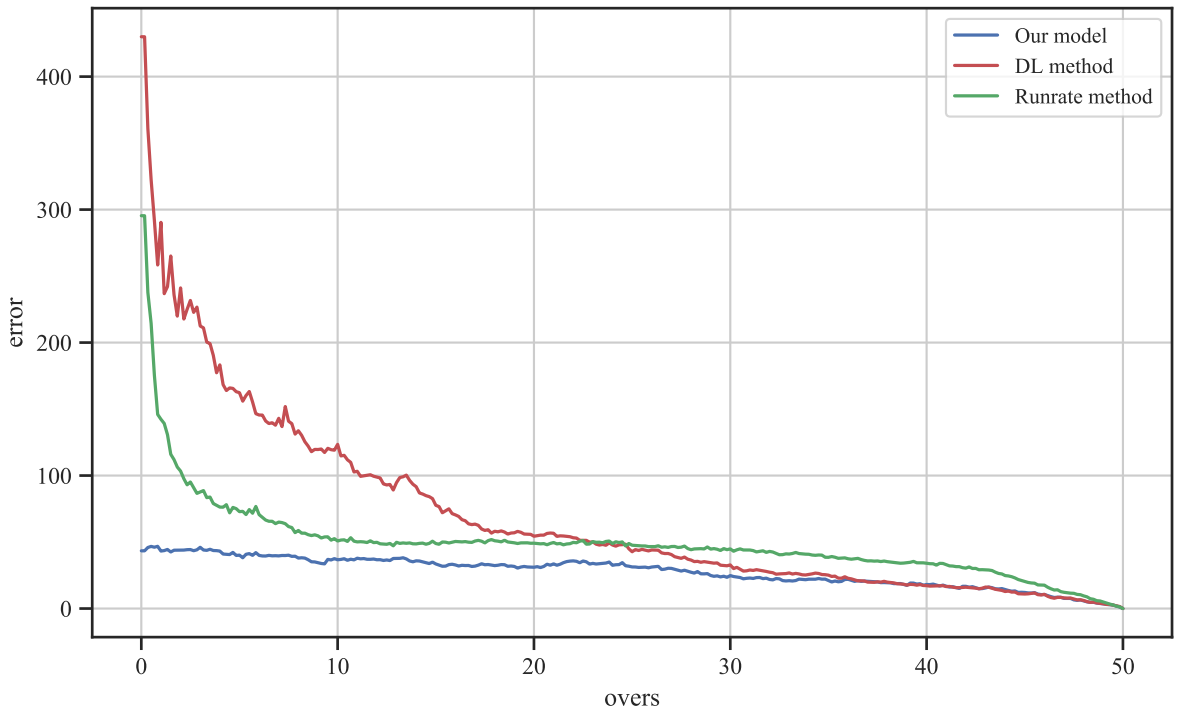

Figure 14: Mean absolute error (MAE) for D/L method, run-rate method and our method, for the first innings across all matches

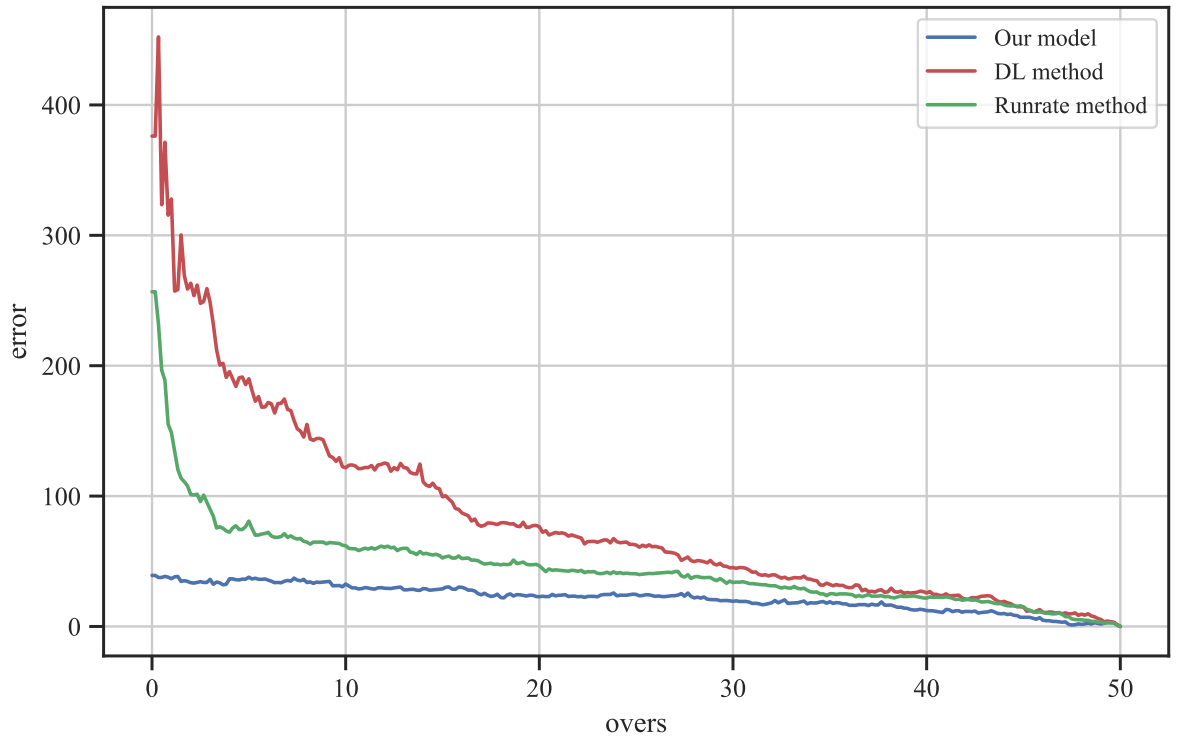

Figure 15: Mean absolute error (MAE) for D/L method, run-rate method and our method, for the second innings across all matches

4.4.4 Comparison of Individual Matches

An inning can be classified into four separate categories, depending on the amount of runs scored during the inning.

- High scoring inning – Final score is above 300 runs
- High-Mid scoring inning – Final score is between 250 to 300
- Mid-Low scoring inning – Final score is between 200 to 250
- Low scoring inning – Final score is below 200

These four categories are the general acceptance at the international level. Thus we picked some random matches, with one match per each from above category to observe how our model behaves under different circumstances. We also compared the results of our model for those matches against the two conventional methods. We used the raw predictions of the three methods for this purpose, since only one match is evaluated at a time. The predictions from all three methods were plotted against the number of balls bowled. The results can be found in the figures 16 and 17 for the two innings separately. Regardless of the number of runs scored during the first inning (Figure 16), final score prediction by our model stays very close to the actual score

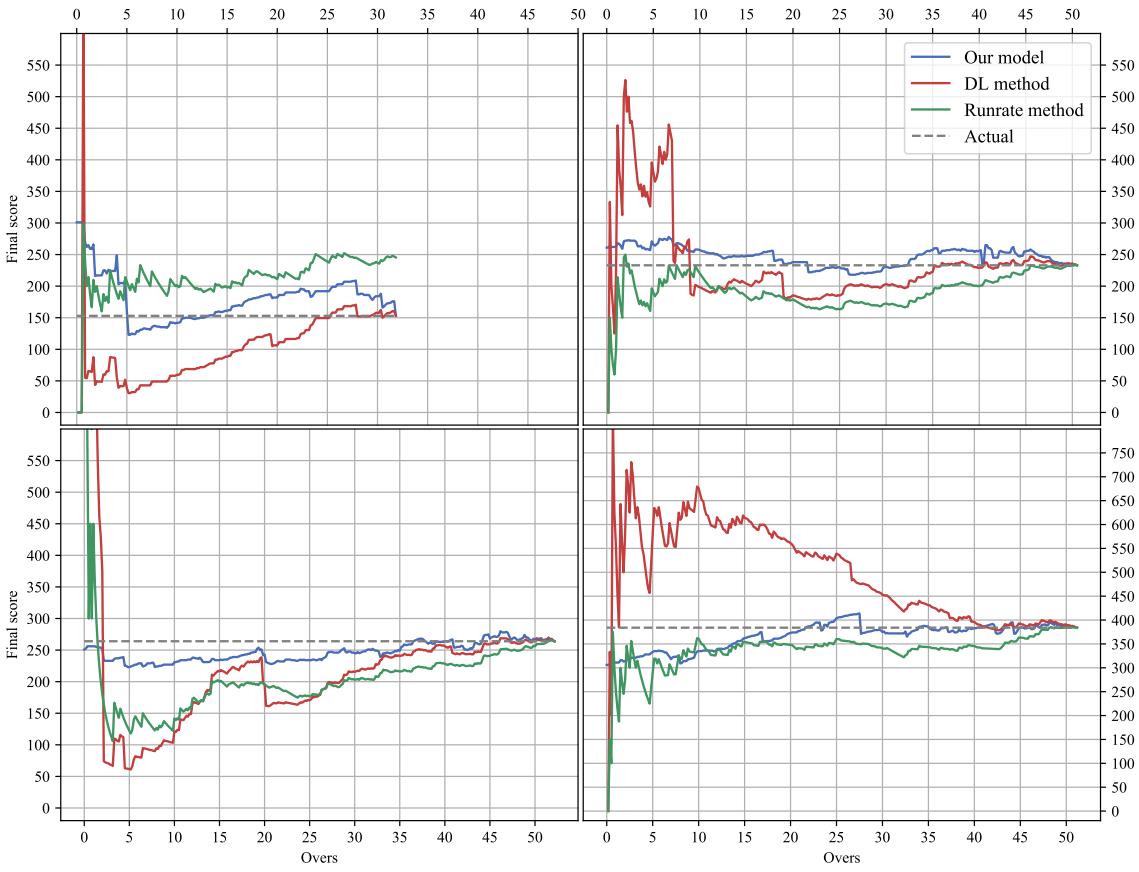

Figure 16: Final score predictions during the first inning of a selected set of individual matches

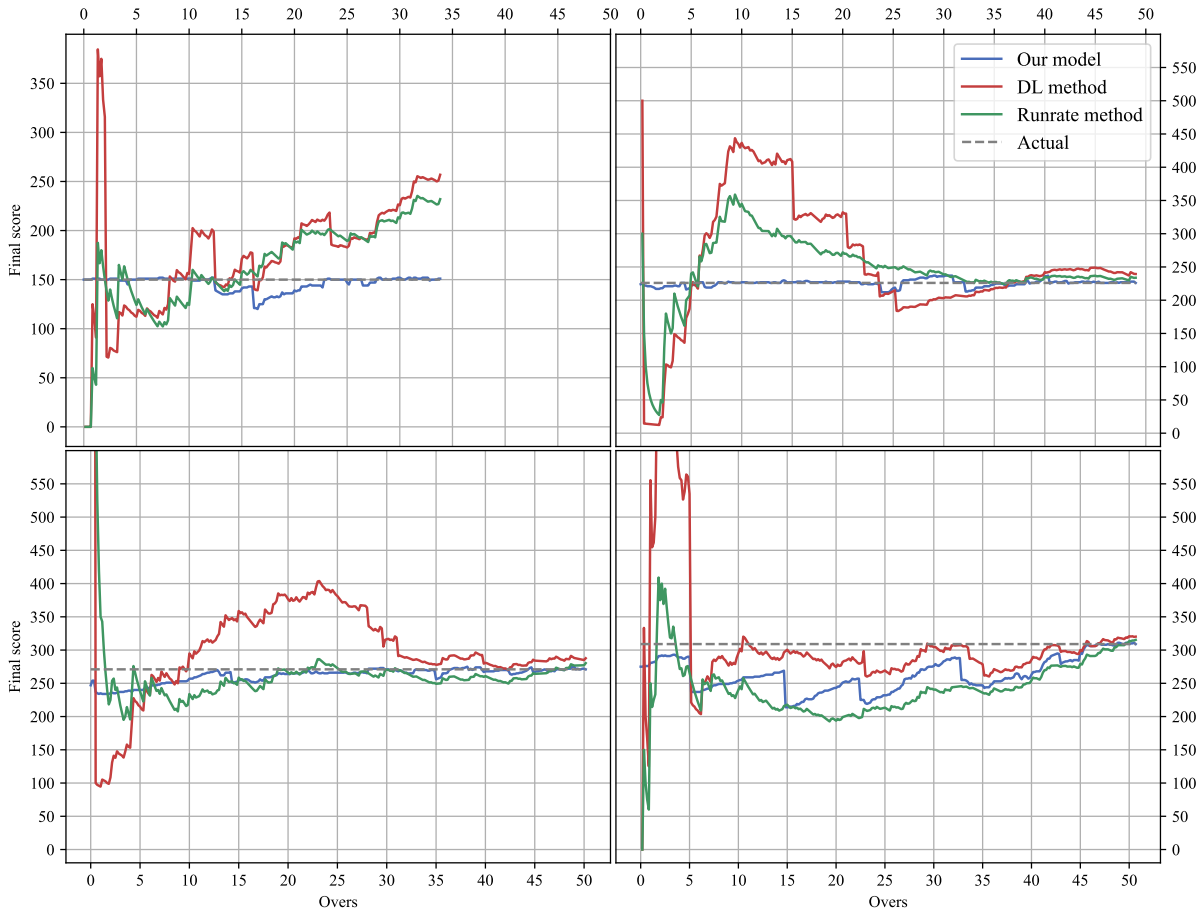

Figure 17: Final score prediction during the second inning of a selected set of individual matches

achieved by the team, from the begining to the end. The D/L and the run-rate method show a huge deviation at the start of the inning, and reach the actual score only towards the end of the inning. This also shows that, out of the two state of the art methods, D/L method has done a better job for certain innings, whereas run-rate method has made better predictions for certain other innings.

In the second innings (Figure 17), our model performs even better compared to the first inning and stays more closely to the actual score achieved with reduced fluctuations. The effect of our model specially highlights when the second inning ends before the fifty overs end and before the team ran out of wickets. Such situations occur when the team batting second wins the match while more resources are left to be consumed. In such situations, both D/L method [5] as well as the run-rate method predict that more runs will be scored. This is due to the fundamental limitation of those two models, which is that both of these models only use the wickets and the overs remaining into account. They do not consider the factors such as the target to be scored, etc. Hence these approaches are not capable of converging to the final score as the current score

reaches the target to be achieved. However, our proposed model considers these factors into account, hence have been able to give a more accurate prediction even if the inning terminates before all the resources end. In-fact, having the target to be scored as a factor in the model has yielded better results during the second inning.

Another characteristic that we can observe from Figure 16 and Figure 17 is, in all four situations we considered, the run-rate method seems to have performed better compared to the D/L method. This is evident from Figure 14 and Figure 15 as well. This supports the fact of using the run-rate method over the D/L method for providing a projected score during the international matches in practice. However, our results indicate that our proposed model is the most stable of the three, and outperforms the state of the art despite the type of the match or the inning. Our model also predicts the final score very closely to the actual target from the start to the end of the inning, whereas the D/L method [5] and the run-rate method fluctuate heavily under different circumstances.

4.5 Performance Comparison with Previous Studies

We evaluate our model against the two baseline models suggested by Bailey and Clarke [2] and Sankaranarayanan et al. [3]. The Bailey and Clarke's model [2] is capable of making over-by-over predictions on the final score, whereas the model suggested by the Sankaranarayanan et al. [3] capable of predicting the final score of the inning at the end of every fifth over. The most recent results of above two baseline models are available in the comparison done by Sankaranarayanan et al. [3] in their research. However, those results are based on the matches played in 2011 and 2012. Therefore we used the same set of matches, including the same train and test sets to train and evaluate our model, for a fair comparison. The prediction results are shown in Figure 18 and Figure 19.

During the first inning, the accuracy of our model goes head to head with the accuracy of the Sankaranarayanan's model, but with a slight drop throughout. Both of these models perform far better than the Bailey et al. model from the beginning of the inning. Our model shows random fluctuation in error as it makes the predictions at the end of every ball. The two base line models make predictions at the end of every over and at the end of every fifth over respectively, hence the accuracy has gotten smoothened. This behavior is same for the second inning as well, as in the Figure 19. During the second inning, the overall accuracy of our model as well as Bailey et al. model has gone down compared to the first inning, whereas Sankaranarayanan et al. model's accuracy has increased from the first inning. From 38th over onwards our model's accuracy

comes in line with the Sankaranarayanan et al. model accuracy.

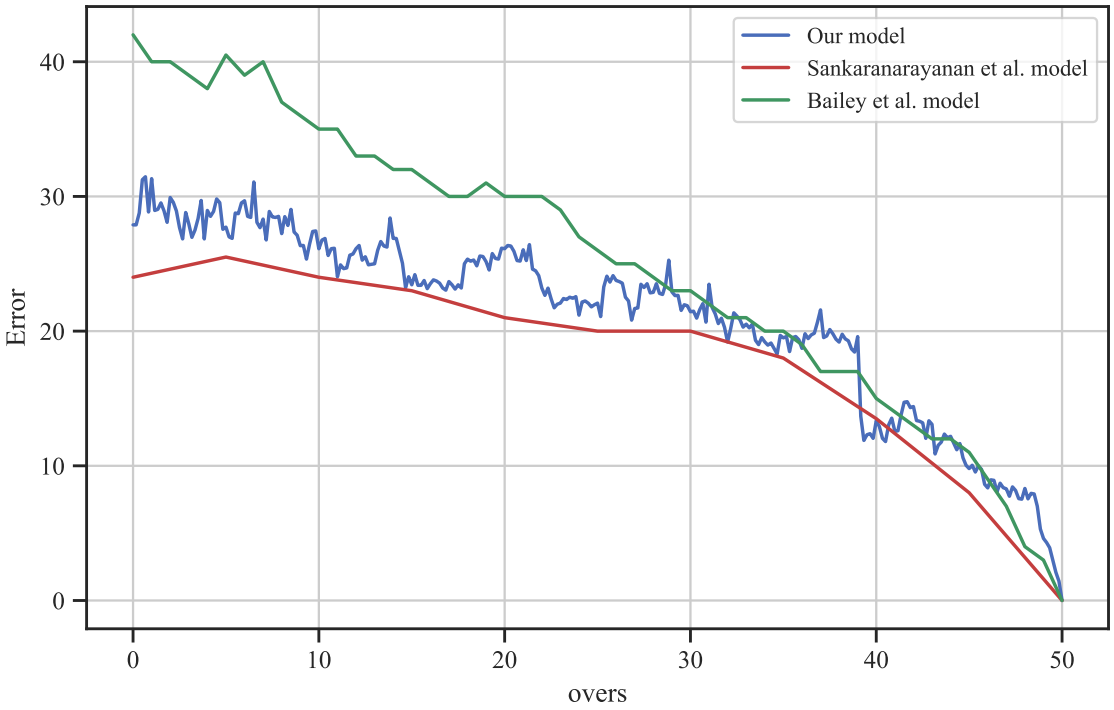

Figure 18: Mean absolute error (MAE) for Bailey et al. [2], Sankaranarayanan et al. [3] and our method during the first inning across all matches

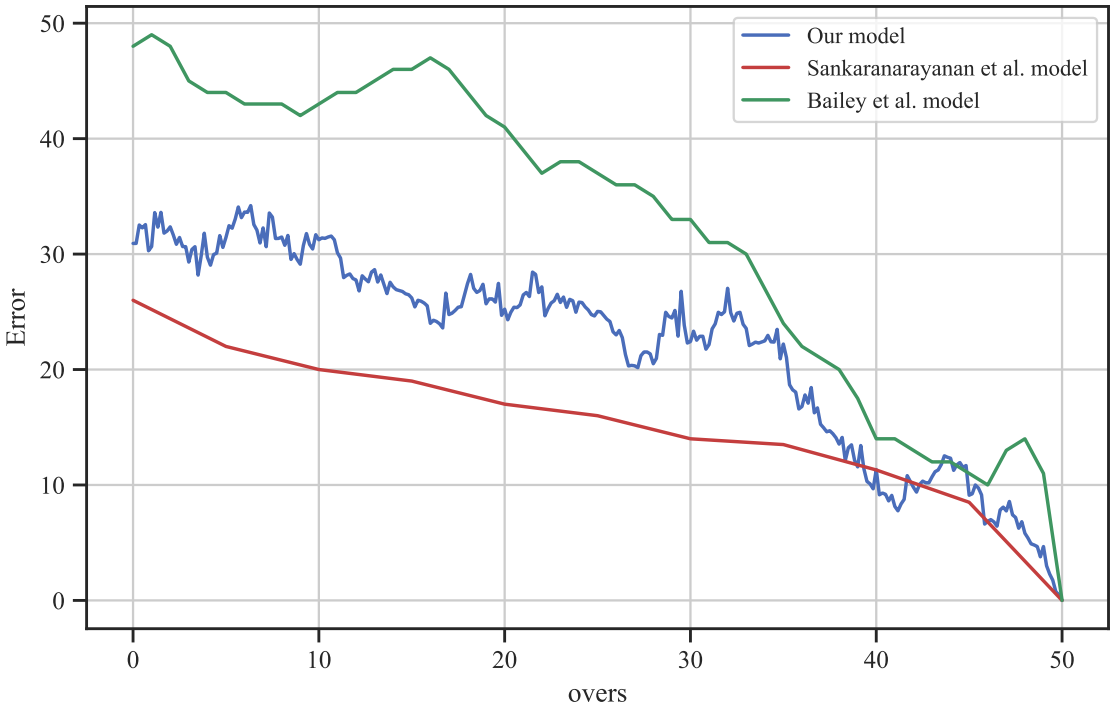

Figure 19: Mean absolute error (MAE) for Bailey et al. [2], Sankaranarayanan et al. [3] and our method during the second inning across all matches

4.6 Winner Prediction Performance

Using the prediction results of our final score prediction model, we predict the winner for the same data set. The winner is predicted during the second inning, before any ball is bowled and at the end of each ball bowled. Table 4 consists of the weights associated with the top ten features of the random forest classification model. Out of all the features included in the model, target score has the most impact of deciding the winner. The predicted final score has taken the fifth position in-terms of the importance. This suggests that combining the results of our previous model has a significant contribution towards predicting the winner during the second inning.

In order to compare these results with the state of the art, we used the winner prediction results of the D/L method and the run-rate method. In both the baseline models, the winner is obtained by comparing the predicted total by each model against the target score.

- predicted_score > target_score $\rightarrow$ team batting second wins
- predicted_score < target_score $\rightarrow$ team batting first wins
- predicted_score == target_score $\rightarrow$ match ties (no winner)

In the D/L method, above criteria are equivalent the criteria used by the ICC to decide the winner in interrupted cricket matches. In practice, the winner is decided by resetting the target depending on the current number of balls bowled and the wickets fallen. Then that reset target is compared against the current score to decide the winner. The criterion we used is the reverse calculation where we reset the score of the current batting team, instead of resetting the target. Figure 20 shows the prediction results for our model and the two baseline models. Predictions

Table 4: Feature importance of the top ten features of the winner prediction model

Feature	Weight
target	0.365524
runs_to_score	0.179508
wickets_remaining	0.088673
avg_score	0.079184
final_score_prediction	**0.060061**
opponent_Zimbabwe	0.040769
resources_remaining	0.032112
country_Zimbabwe	0.024267
opponent_South Africa	0.022737
current_score	0.014550
Others	0.092615

accuracy of our model clearly stands out from the two baseline approaches. At the start of the second inning, our model is capable of predicting the winner with an accuracy of 86.5%. At the same point, both the D/L method and the run-rate methods predict the winner with an accuracy of only 57.69%. This 28.81% difference of accuracy is a considerable improvement over the state of the art. The accuracy of the model reaches 90% by the 23rd over and becomes close to 95% by the 33rd over. The 100% accuracy was achieved towards the last two overs. The D/L method's accuracy gets closer to nineties around the 36th over and get along with our model towards the 38th over. Despite the D/L method reaches 100% accuracy before our model, it shows lot of drops after that, where as our model stays at 100% once it reaches there. The run-rate method seems to be slightly better in terms of accuracy compared to D/L method until the 35th over. It reaches the nineties at the 37th over and hovers around that mark until the last two overs. Overall, both D/L method and the run rate method seem to perform equally despite the slight variations.

Figure 21 shows the variation of the percentages of correct and incorrect predictions as the match progresses. Here a 'positive' means the team batting second is predicted to win the match, whereas 'negative' means the team batting second is predicted to lose the match. In these predictions the prefix 'true' refers to a correct prediction and a 'false' refers to an incorrect prediction. False positives are very close to zero throughout, whilst false negatives are slightly higher. Noticeably after the 31st over, number of false negatives starts to increase as true positives start to

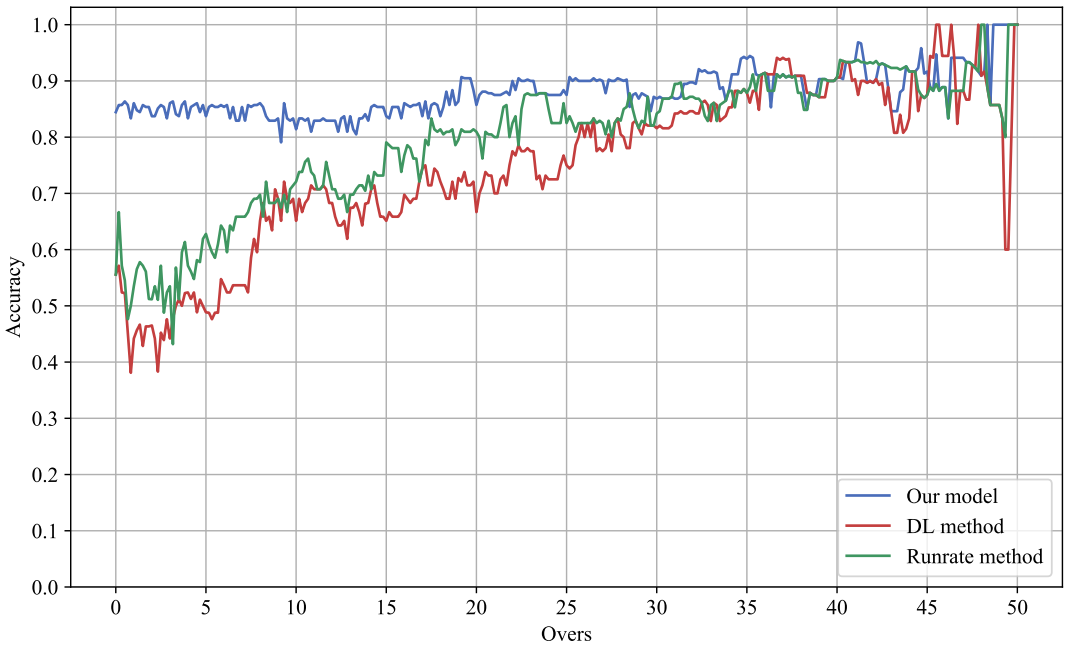

Figure 20: Accuracy of winner predictions after each ball bowled during the second inning

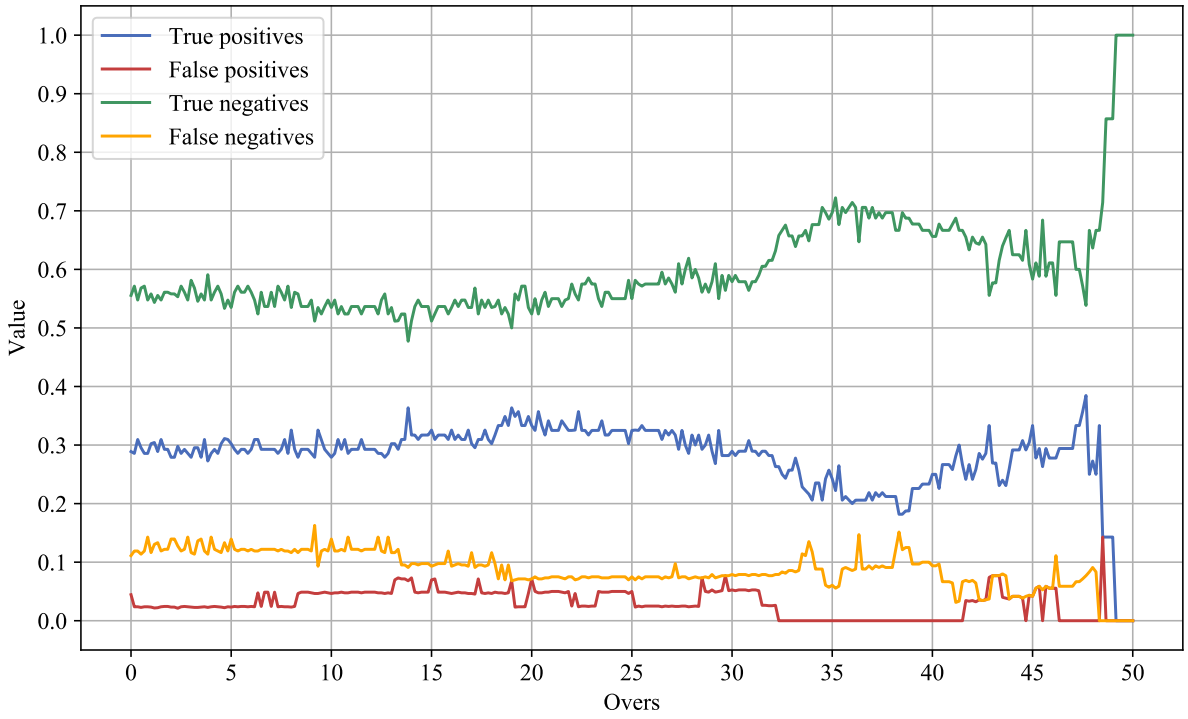

Figure 21: Variation of correct and incorrect winner predictions as the match progresses

decline slowly. As in Figure 22, the precision of our model starts around 90%, and varies between 80-95% until the 32nd over, and then reaches 100%. There onwards it stays there for the most parts with slight variations until the end of the inning. Overall, the precision is well above 85% throughout the inning and is above 90% for more than half of the time, which is the highest value recorded so far in winner predictions to our knowledge. This also means that if the model predicts that the team batting second will win the match, it is more like to get the same result as the actual outcome of the match. Recall and F1-Score also hovers around the precision up to 32nd over. There onwards, the recall has a slight descend while the F1-score remains on the same level throughout. Having a very high precision as well as recall suggests that our model predicts both the winners as well as the loser equally well, and is not biased towards any of the two.

Tables 5 and 6 shows the team wise break down of the prediction accuracies. Before the start of the inning, outcomes of the matches played by Australia, England, Ireland, South Africa, Sri Lanka, West Indies and Zimbabwe were predicted with 100% accuracy. Both India and Afghanistan have played only two matches each during the time period and out of them one for each were correctly predicted. Thus it is not fair to come to any conclusion given the low number of innings for those two teams. Table 6 contains the prediction accuracies for each team for all stages of the inning. Predictions made during the second innings played by England,

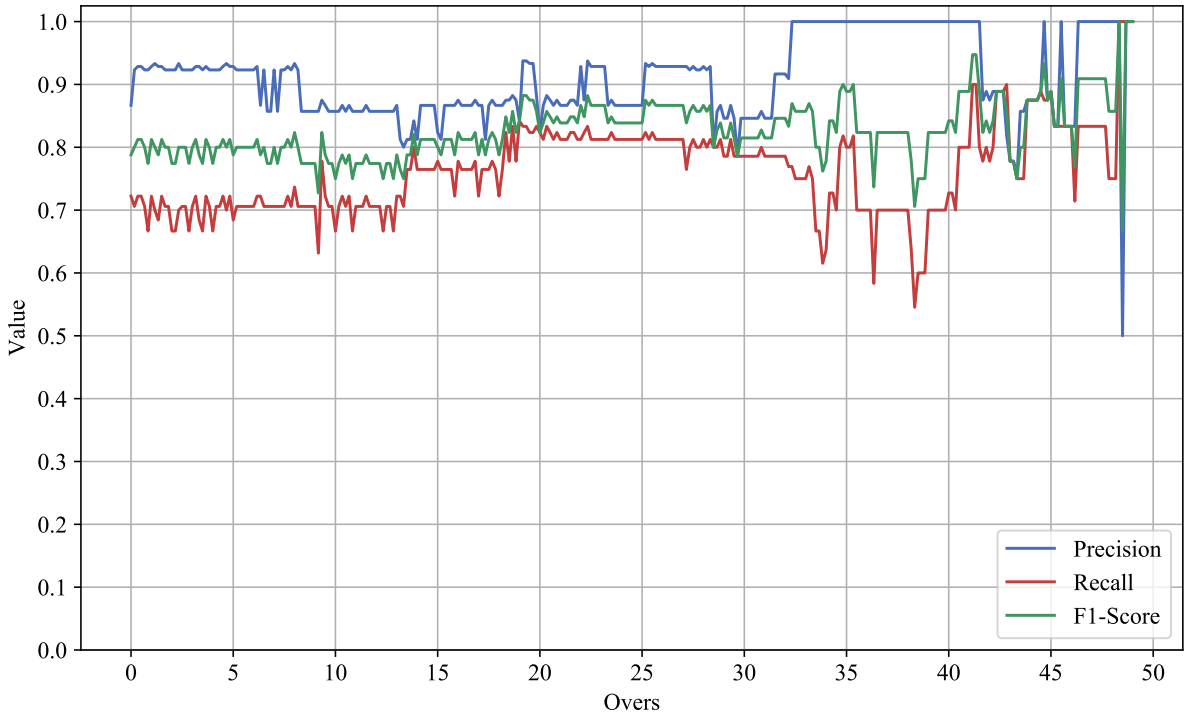

Figure 22: Precision, recall and F1-score for winner predictions across all matches

Table 5: Winner prediction accuracy for different teams at the start of the second innings

Team	Innings Count	Accuracy
Australia	3	100%
Bangladesh	5	60%
England	3	100%
India	2	50%
New Zealand	6	83%
Pakistan	5	80%
South Africa	7	100%
Sri Lanka	4	100%
West Indies	3	100%
Zimbabwe	1	100%

Table 6: Winner prediction accuracy for different teams throughout the second innings

Team	Accuracy
Australia	88.58%
Bangladesh	74.81%
England	100.00%
India	51.46%
New Zealand	84.42%
Pakistan	94.05%
South Africa	100.00%
Sri Lanka	100.00%
West Indies	100.00%
Zimbabwe	94.59%

South Africa, Sri Lanka and West Indies have an accuracy of 100%. This is consistent with the predictions made before the start of the innings in Table 5. Predictions for most of the countries have increased when the entire inning was considered. This is expected as the accuracy increases as the match progresses. However, India seems to be the only team whose prediction accuracy has not increased significantly over the innings.

Figure 23 and 24 shows the variation in the prediction accuracy for the different categories of

targets to be scored. At the start of the innings, there is no significant difference in the accuracy of our model for the four types of targets. However, both D/L method [5] and the run-rate method seem to be favoring towards matches with a high scoring targets. When the target is above 300 runs, both methods predict the outcome with 88.89% of accuracy and an accuracy of 70.59% for matches with a target between 250 and 300 runs. If the target is below 250 runs,

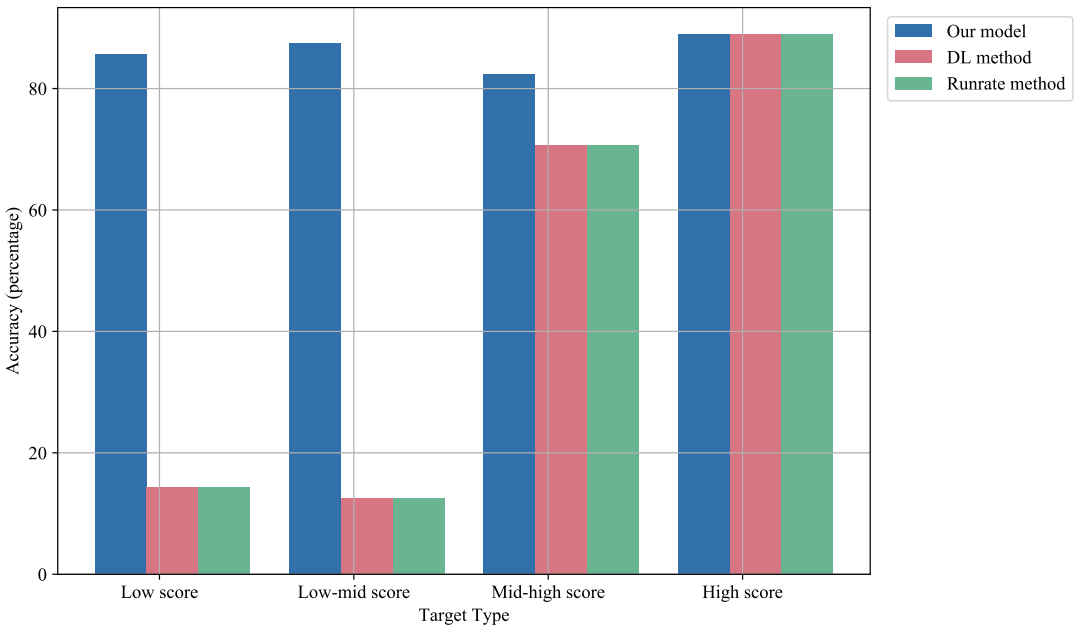

Figure 23: Winner prediction accuracy at the start of the second innings when chasing different target types

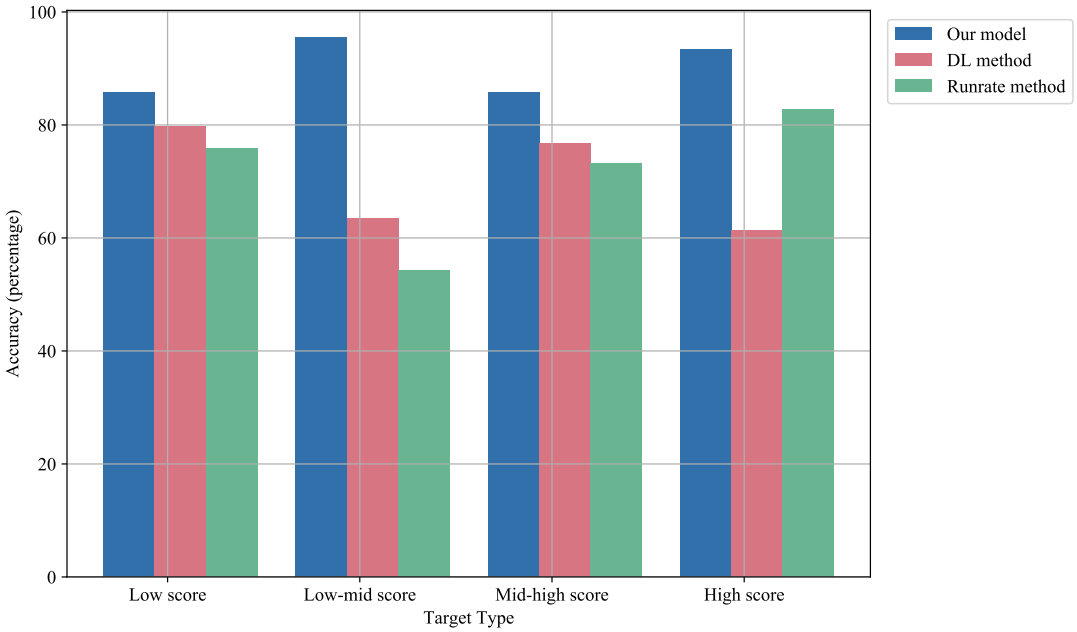

Figure 24: Winner prediction accuracy considering the entire second inning, when chasing different target types

then the accuracy drops significantly below 15%. There is more than 70% drop in the accuracy between the tragets above 300 and the target below 250, which is extremely high. The prediction accuracy of our model for the entire inning is also consistent with the predictions made at the start of the match, for the different types of target scores as in the Figure 24. D/L method and the run-rate methods have done slightly better when the entire inning is considered, compared prediction made at the start of the inning. However, a high variation in the accuracy for different target types still exists. D/L method [5] has a difference close to 18.38% between the highest and the lowest accuracies while the run-rate method having an even larger margin of 28.55% between the highest accuracy and the lowest accuracy. These results suggest that our model is consistent regardless of the target score, unlike the two state of the art methods. It also supports the fact that our model is consistent throughout the innings.

Chapter 5: Conclusion

The main objective of this study was to propose a supervised learning based approach for making ball-by-ball predictions on the final score of an inning in ODI cricket matches, while the match is in progress. We evaluated the use of existing conventional regression algorithms such as linear regression, ridge regression, lasso regression, ridge regression, random forest regression and gradient boosted tree regression to predict the final score. These algorithms gave decent results for most parts of the innings. However the accuracy started to drop heavily towards the end of the innings. Thus, to overcome this issue we proposed a segment-wise model building approach for predicting the final score.

In the proposed approach, each inning was segmented into ten equal length segments based on the resources remaining with the use of the Duckworth-Lewis resource table [5] and trained separate models for each segment. We evaluated several algorithms to use as the baseline algorithm and out of all, random forest regression gave the best results. This segment-wise modeling approach we followed was capable of adapting well to the different stages of the inning by locally optimizing its parameters. Results showed that this approach yields better results compared to the conventional algorithms, having the lowest MAE out of all with no deviations towards the end of the inning. Prediction made by our model during the second innings had higher accuracy and a precision than during the first inning. This could be due to the availability of the target to be scored, and the teams batting second tends to keep up with the required rate. Thus the fluctuations in the run scoring can be said to be less compared to the first inning.

Our model also out-performed the D/L method [5] and the run-rate method when predicting the final score in terms of accuracy, fairness and the stability. Both the D/L [5] method and run-rate method have extremely high errors at the start of the match, where as our model had a MAE close to thirty at the start of the inning. Even though the accuracy of D/L [5] and run-rate method increases as the match progresses, they never surpass the accuracy of our model at any stage. This behavior has been the same for all the matches despite the amount of runs scored during the match.

We compared our approach with two previous studies done by Bailey and Clarke [2] and Sankaranarayanan et al. [3] on predicting the final score in ODI matches. The results showed that our approach out-performed the over-by-over prediction method suggested by [2]. Bailey and Clarke [2] method consistently have a higher error rate than ours throughout the innings.

However, the predictions made at the end of every fifth over by the Sankaranarayanan et al. [3] model had higher accuracy than the proposed model. It starts with a lower MAE than our model at the start of the innings and converges to zero much faster. This higher accuracy could be due to the use of two separate models for predicting the boundary-runs and non-boundary-runs in their model. Thus we see that there is room for further improvement in accuracy of our proposed model.

Finally we proposed a method of making ball-by-ball predictions on the winner of one-day international (ODI) matches during the second innings using a random forest classifier. Providing the prediction results from our final score prediction model as an input feature to the classifier increased the accuracy of the winner prediction. Results indicated that our model performs significantly better compared to the state of the art. At the start of the inning, our model predicts the winner with an accuracy of 86.5%, whereas D/L [5] method and the run-rate method had an accuracy below sixty percent. Our model also had a precision above 85% and a recall above 70% for most parts of the match, which suggested that the predictions made by our model are unbiased. This accuracy and the precision achieved by our model are the highest recorded in winner prediction in ODI matches to our knowledge.

The proposed model was also capable of predicting the winner equally well for different levels of targets to be chased. During the low scoring matches, accuracy at the start of the innings for D/L [5] method and the run-rate method drops down below 20%. However, our model maintained the same accuracy for all matches despite the match being a low-scoring one or a high scoring match.

Chapter 6: Future Work

In this study we proposed a method of making ball by ball predictions on the final score of an inning in ODI cricket matches while the match is in progress. The result showed that despite our model being capable of making ball by ball predictions, the accuracy is below of what was achieved by Sankaranarayanan et al. [3]. This could be due to the use of separate models for the prediction of boundary-runs and non-boundary-runs in the approach proposed by Sankaranarayanan et al. [3]. It is possible to apply the same concept on top of our proposed approach to check whether that will increase the accuracy.

During this study, we used historical and instantaneous features related to the teams and external factors related to a given match to model the final score and the winner. However we did not use batsmen specific or bowler specific information. The model is not aware of facts like whether a proper batsman is batting or a tail-ender is batting. Similar for bowlers, the model would treat in the same manner when a front-line bowler is bowling as well as a part-time bowler is bowling. Therefore, our model can be further improved by bringing in batsman and bowler information such as batting statistics of the current batsmen, batsmen to follow, bowling statistics of the current bowler, overs left by each bowler, etc. as input features.

Further, our study was focused only on the ODI matches. However the methodology we followed could be extended to be used to predict the final score and the winner in T20I matches as well. The input features would be identical for both the formats, but the number of segments to partition the data will be only four. Thus the model can be seamlessly retrained on top of data collected over a set of T20I matches, and can be evaluated to see the results and the applicability of this approach for T20I as well.

References

[1] Cricsheet.org, “Cricsheet”, https://cricsheet.org, 2018. Accessed: 22-Apr-2018.

[2] M. Bailey and S. R. Clarke, “Predicting the match outcome in one day international cricket matches, while the game is in progress”, *Journal of Sports Science and Medicine*, vol. 5, no. 4, pp. 480–487, 2006.

[3] J. S. V. V. Sankaranarayanan and L. V. S. Lakshmanan, “Auto-play: A data mining approach to odi cricket simulation and prediction”, 2014.

[4] S. Brooker and S. Hogan, “A method for inferring batting conditions in odi cricket from historical data”, 2011.

[5] F. Duckworth and A. Lewis, “A fair method for resetting the target in interrupted one-day cricket matches”, *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 220–227, 1998.

[6] M. Purucker, “Neural network quarterbacking”, *IEEE Potentials*, vol. 15, no. 3, pp. 9–15, 1996.

[7] S. M. Z. Shi and A. Zimmermann, “Predicting ncaab match outcomes using ml techniques – some results and lessons learned”, (Prague, Czech Republic), 1996.

[8] B. Bukiet, E. R. Harold, and J. L. Palacios, “A markov chain approach to baseball”, *Operations Research*, vol. 45, no. 1, pp. 14–23, 1997.

[9] N. Cserepy, R. Ostrow, and B. Weems, “Predicting the final score of major league baseball games”, 2015.

[10] Z. Smith, “A markov chain model for predicting major league baseball”, 2016.

[11] A. Zimmermann, “Basketball predictions in the ncaab and nba: Similarities and differences: Basketball predictions in the ncaab and nba”, *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 06 2016.

[12] B. Loeffelholz, E. Bednar, and K. Bauer, “Predicting nba games using neural networks”, *Journal of Quantitative Analysis in Sports*, vol. 5, no. 1, pp. 1–17, 2009.

[13] P. Beneventano, P. Berger, and B. Weinberg, “Predicting run production and run prevention in baseball: The impact of sabermetrics”, 06 2019.

[14] D. Beaudoin and T. Swartz, “General the best batsmen and bowlers in one-day cricket”, vol. 37, pp. 203–222, 01 2003.

[15] H. Lemmer, “An analysis of playersṕerformances in the first cricket twenty20 world cup series”, *South African Journal for Research in Sport, Physical Education and Recreation*, vol. 30, 09 2008.

[16] S. Das, “On generalized geometric distributions: Application to modeling scores in cricket and improved estimation of batting average in light of notout innings”, *SSRN Electronic Journal*, 12 2011.

[17] H. Lemmer, “The single match approach to strike rate adjustments in batting performance measures in cricket”, *Journal of sports science & medicine*, vol. 10, pp. 630–4, 12 2011.

[18] A. Lewis, “Towards fairer measures of player performance in one-day cricket”, *Journal of the Operational Research Society*, vol. 56, pp. 804–815, 07 2005.

[19] S. Perera, D. Attygalle, and A. Sunethra, “A study on selecting the best batsmen for the next one-day international cricket match: In sri lankan context”, 12 2014.

[20] M. Ovens and B. Bukiet, “A mathematical modelling approach to one-day cricket batting orders”, *Journal of Sports Science and Medicine*, vol. 5, pp. 495–502, 07 2006.

[21] S. Kampakis and W. Thomas, “Using machine learning to predict the outcome of english county twenty over cricket matches”, 2015.

[22] R. de Silva, “A fair target score calculation method for reduced-over one day and t20 international cricket matches”, *Journal of Mathematical Sciences & Mathematics Education*, vol. 8, no. 2, pp. 6–19, 2013.

[23] P. S. G. T. B. Swartz and S. Muthukumarana, “Modeling and simulation for one-day cricket”, *The Canadian Journal of Statistics*, vol. 37, no. 2, p. 143–160, 2009.

[24] K. P. Jayalath, “A machine learning approach to analyze odi cricket predictors”, *Journal of Sports Analytics*, vol. 4, pp. 73–84, 2018.

[25] A. Bandulasiri, “Predicting the winner in one day international cricket”, *Journal of Mathematical Sciences & Mathematics Education*, vol. 3, no. 1, pp. 6–17, 2008.

[26] R. Bryll, R. Gutierrez-Osuna, and F. Quek, “Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets”, *Pattern Recognition*, vol. 36, pp. 1291–1302, 06 2003.

[27] P. Sollich and A. Krogh, “Learning with ensembles: How overfitting can be useful”, vol. 8, pp. 190–196, 01 1995.

[28] L. Kuncheva and C. Whitaker, “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy”, *Machine Learning*, vol. 51, pp. 181–207, 05 2003.

[29] L. L.Breiman, “Bagging predictors”, *Machine Learning*, vol. 24, pp. 123–140, Aug 1996.

[30] R. Quinlan, “Bagging, boosting, and c4.5.”, pp. 725–730, 01 1996.

[31] D. Wolpert, “Stacked generalization”, *Neural Networks*, vol. 5, pp. 241–259, 12 1992.

[32] ESPNcricinfo, “Espncricinfo - cricket teams, scores, stats, news, fixtures, results, tables”, http://www.espncricinfo.com, 2018. Accessed: 22-Apr-2018.