Dynamic Winner Prediction in Twenty20 Cricket: Based on Relative Team Strengths

Sasank Viswanadha[1], Kaustubh SSS[1], Madan Gopal Jhawar[2], and Vikram Pudi[2]

[1] Mahindra École Centrale, Hyderabad, India,
sasank14168@mechyd.ac.in, kaustubh14141@mechyd.ac.in
[2] International Institute of Information Technology, Hyderabad, India

Abstract. Cricket as a sport has evolved over years into various formats the latest addition being Twenty20(T20) cricket which is the shortest and the most dynamic of all forms of the sport. T20 cricket has gained in popularity over the last decade since its introduction in 2007. Due to the short nature of a T20 game, the match dynamics can change unexpectedly, making the problem of predicting the winner quite challenging and interesting. In this article, we address the problem of predicting the outcome of an IPL match, as the game progresses using supervised learning approach based on the relative team strengths. The problem of computing relative team strength simplifies to modeling individual players' batting and bowling strengths. We use the career statistics as well as the form (performances in recent games) to compute the batting and bowling strengths of a player. ...specify the end result; best classifier and so...

Keywords: Winner Prediction, Sports Analytics, Supervised Learning, Player Modeling, Random Forest

1 Introduction

The popularity of data driven decision-making in sports has been on the rise since the advent of statistical modeling in sports analytics. Basketball is one of the best examples of how analytics have changed the way sports are played and player performance is measured. Although cricket is one of the most popular sports on the planet, it falls behind other major sports like basketball and baseball when it comes to using of sports analytics for on-field decision making and prediction. Data-driven decision making and analysis of players in cricket can help teams make better choices regarding player selections and also make the sport more exciting for its viewers.

Today, cricket is played in three formats namely; Test cricket, the longest form of the game, One Day International and T20 format (20 overs per team), which is the shortest. T20 cricket is the most exciting format of cricket. Its popularity and viewer-ship have increased widely since its inception. The Indian Premier League (IPL), founded by Board of Control for Cricket in India (BCCI)

is a professional Twenty20 cricket league in India contested during April and May of every year by teams representing Indian cities. The IPL is the most-attended cricket league in the world and ranks sixth among all sports leagues. Our research in this article is focused on Twenty20 cricket, specifically the Indian Premier League but the methodology could be easily extended to all forms of cricket.

To predict the outcome of the Twenty20 cricket matches, we consider our analysis from the beginning of the second innings. The second innings is divided into segments of equal length (measured by number of balls bowled). We propose a method where we estimate the batting and bowling strengths of each of the 22 players. These player strengths are then used to calculate the relative team strength of the team batting second. This relative strength along with features such as runs remaining, balls remaining and wickets remaining are used as features in supervised learning to predict the winner of the game, after each segment.

The key contributions of this paper are :

- We propose a dynamic(first of its kind) approach to predict the winner of Twenty20 cricket matches in the second innings of the match, as the game progresses.
- We use a team composition based approach which largely depends on the strengths of individual players.
- We also discuss some dynamic features which prove to be important while trying to analyse a match which is in progress.

2 Related Work

In recent times, the application of statistical analysis techniques has been quite extensive for the sport of cricket particularly in the context of player rating and squad optimization. [1] defines and discusses the manner in which, targets can be modified in rain interrupted games. A more visual comparison of player strengths are put forth in [4] and [2]. Further, these resources discussed in [1] were adopted in studying player performances in [3]. The approach of player modeling considering the strength of opposition, venue of the match along with other factors was in discussed in [5].There have been very few efforts in addressing the problem of winner prediction while a cricket match is in progress. The application of supervised learning techniques; Support Vector Machines (SVM) and Naive-Bayes Classification towards predictive analysis considering various factors such as toss, competing teams, home venue etc., for the winner prediction in an ODI match was presented in [6].

[7] considers both the historical data along with instantaneous match state for One Day Internationals (ODI),so as to predict the match winner making use of nearest-neighbor clustering and linear regression algorithms. [7] also introduces the idea of using segments to break down an innings and makes predictions for each segment. We have incorporated this idea into our study. [8] puts forth a novel approach based on team composition, computing relative team strengths

based on the team's cumulative batting and bowling strengths in order to predict the winner of an ODI match. In addition, [8] also showed that the team composition keeps changing and that it is important to consider the players playing in every game instead of taking only the statistics of the team as a whole like [6], [7] and [9] did. This idea has been incorporated in our study and forms an important part of our approach. We used the concepts presented in [10] in our study to develop the formulae for calculating the *batsman_score* and the *bowler_score*. [11] presents an approach of winner prediction for the ninth season of IPL, by modeling the individual player strengths into cumulative batting and bowling scores. In this paper, we aim to address the problem of dynamic winner prediction in a Twenty20 cricket match, a study first-of-its-kind in Twenty20 cricket.

3 Problem Formulation and Notation

3.1 Overview of Twenty20 Cricket : Rules

In the T20 format of cricket each of the two playing teams bats for a maximum of 120 deliveries and bowls for a maximum of 120 deliveries. The team that scores the maximum amount of runs in the 120 deliveries of before they lose their 10 wickets wins the match

Over A sequence of six balls bowled by a bowler from one end of the pitch is called an *over* in cricket terminology. So, in the T20 format a team can bat and bowl for a maximum of 20 *overs*.

Innings An *innings* is one of the divisions of a cricket match during which one team takes its turn to bat. There are two *innings* in a game of cricket. In this paper, we restrict our study to the second *innings* of a match.

State In our study, we define *state* to represent the different stages in the match at which we make the predictions using our model. We consider 21 *states* for each match; 1 at the beginning of the second *innings* and 20 at the end of each *over* of the second *innings*. It is to be noted here that the number of *states* considered to make predictions can be changed.

3.2 Notation

In this section, we introduce the notation to be used throughout this paper. We use *match* to represent a match, $innings_1$ and $innings_2$ to denote the first and second innings respectively. We use $Team_A$ to represent the team batting in $innings_1$ and $Team_B$ to represent the team batting in $innings_2$. $Score_A$ denotes the *runs* scored by $Team_A$ in $innings_1$. $Target$ denotes the number of *runs* that $Team_B$ needs to score to win the match, $Target = Score_A + 1$. S_i, $0 \leq i \leq 20$ represents the *states* in a *match*. S_0 corresponds to the *state* at the

end of $innings_1$ and the remaining *states* $1 \leq i \leq 20$ each correspond to the *state* at the end of *over* i in $innings_2$. $Players(match, Team_A)$ denotes the set of 11 players in $Team_A$ playing in *match* and $Players(match, Team_B)$ denotes the set of 11 players in $Team_B$ playing in *match*.

Table 1. Career Statistics

Notation	Description
$C_{matches\_played}$	# Matches Played by the player
$C_{batting\_innings}$	# Matches in which the player has batted
C_{runs_scored}	# Runs Scored by the player
C_{overs_batted}	# Overs in which the player has batted
C_{not_outs}	# The player remained $not - out$
$C_{batting\_average}$	# Average Runs scored by the player before getting *out*
$C_{batting\_strike\_rate}$	# Average runs scored by the player per 100 balls
$C_{bowling\_innings}$	# Matches in which the player has bowled
$C_{wickets\_taken}$	# Wickets taken by the player
$C_{runs\_conceded}$	# Runs conceded by the player
C_{overs_bowled}	# Overs in which the player has bowled
$C_{bowling\_average}$	# Runs conceded by the player per wicket taken
$C_{bowling\_strike\_rate}$	# Balls bowled by the player per $wicket\_taken$

$C(p)$ denotes the set of career statistics of a player p and $F(p)$ denotes the set of recent statistics (recent 4 games) or form of a player p. The career statistics and recent statistics used in this study are shown in Table 1 and Table 2 respectively.

At each state, there are 3 parameters along with the relative team scores that we use in our model to make predictions.

- $R^i_{runs\_remaining}$ denotes the number of runs $Team_B$ needs to get to win the *match* at *state* i. $R^i_{runs\_remaining} = Target - runs\ scored\ by\ Team_B\ at\ state\ i$
- $R^i_{wickets\_remaining}$ denotes the number of wickets $Team_B$ has in hand at *state* i. $R^i_{wickets\_remaining} = 10 - wickets\ lost\ by\ Team_B\ at\ state\ i$
- $R^i_{ball\_remaining}$ denotes the number of balls $Team_B$ yet to play at *state* i. $R^i_{balls\_remaining} = 120 - balls\ played\ by\ Team_B\ at\ state\ i$

4 Methodology

4.1 Batsman Rating

Calculation of Batting Average: Batting Average is defined as the average number of *runs* scored by the batsman before he gets *out*. Batting average for the career statistics is calculated in the following way

$$C_{batting\_average} = \frac{C_{runs\_scored}}{C_{batting\_innings} - C_{not\_outs}}, \tag{1}$$

Table 2. Recent Statistics: Previous 4 games

Notation	Description
$F_{batting\_innings}$	# Matches in which the player has batted
F_{runs_scored}	# Runs Scored by the player
F_{overs_batted}	# Overs in which the player has batted
F_{not_outs}	# The player remained $not - out$
$F_{batting\_average}$	# Average Runs scored by the player before getting out
$F_{batting\_strike\_rate}$	# Average runs scored by the player in 100 balls faced
$F_{bowling\_innings}$	# Matches in which the player has bowled
$F_{wickets\_taken}$	# Wickets taken by the player
$F_{runs\_conceded}$	# Runs conceded by the player
F_{overs_bowled}	# Overs in which the player has bowled
$F_{bowling\_average}$	# Runs conceded by the player per wicket taken
$F_{bowling\_strike\_rate}$	# Balls bowled by the player per $wicket\_taken$

Calculation of Batting Strike Rate: Batting Strike Rate is defined as the average number of *runs* scored by the batsman before per 100 balls faced. Batting strike rate for the career statistics is calculated in the following way

$$C_{batting\_strike\_rate} = \frac{C_{runs\_scored}}{(C_{overs\_batted} * 6)} * 100 \tag{2}$$

The batting average and strike rate for the recent statistics is calculated similar to equations 1 and 2.

Calculation of Batsman Score The quality of the batsmen a team possesses can greatly affect the outcome of a game. Consistency and fast run-scoring ability are two traits common to all the good batsmen. Batting average and is a measure of the consistency of the batsman and batting strike rate is a measure of his fast run-scoring ability. We calculate the batsman score for all the players in a *match* using these two statistics as shown in algorithm 4.1.

The parameter used in line 2 is a measure of how consistently a player gets to bat. It is used to differentiate between a seasoned batsman and a part-time or tail ending batsman. Line 3 calculates the *career_score* of a player. Line 4 calculates the *recent_score* of a player. Lines 6 - 9 normalize the *career_score* and *recent_score* of all the players in a *match*. Line 9 calculates the *batting_score*, which is a weighted sum of *career_score* and *recent_score*.

4.2 Bowler Rating

Calculation of Bowling Average: Bowling Average is defined as the average number of *runs* conceded by the bowler per *wicket* he takes. Bowler average for

Algorithm 1 Modeling Batsmen

Input: $Players(match, Team_A)$, $Players(match, Team_B)$
Output: $\phi^p_{batting\_score}$ for all $p \in (Players(match, Team_A) \cup Players(match, Team_B))$

1: **for** $p \in (Players(match, Team_A) \cup Players(match, Team_B))$ **do**
2: $u \leftarrow \sqrt{\frac{C_{batting\_innings}}{C_{matches\_played}}}$
3: $\phi^p_{career\_batting\_score} \leftarrow u * C_{batting\_average} * C_{batting\_strike\_rate}$
4: $\phi^p_{recent\_batting\_score} \leftarrow u * F_{batting\_average} * F_{batting\_strike\_rate}$
5: **end for**
6: **for** $p \in (Players(match, Team_A) \cup Players(match, Team_B))$ **do**
7: $\phi^p_{career\_batting\_score} \leftarrow \frac{\phi^p_{career\_batting\_score}}{max(\phi^p_{career\_batting\_score})}$
8: $\phi^p_{recent\_batting\_score} \leftarrow \frac{\phi^p_{recent\_batting\_score}}{max(\phi^p_{recent\_batting\_score})}$
9: $\phi^p_{batting\_score} = 0.8 * \phi^p_{career\_batting\_score} + 0.2 * \phi^p_{recent\_batting\_score}$
10: **end for**

the career statistics is calculated in the following way

$$C_{bowling\_average} = \frac{C_{runs\_conceded}}{C_{wickets\_taken}} \tag{3}$$

Calculation of Bowling Strike Rate: Bowling Strike Rate is defined as the average number of *balls* bowled by the bowler per wicket taken. Bowling strike rate for the career statistics is calculated in the following way

$$C_{bowling\_strike\_rate} = \frac{(C_{overs\_bowled} * 6)}{C_{wickets\_taken}} \tag{4}$$

The bowling average and strike rate for the recent statistics is calculated similar to equations 3 and 4.

Calculation of Bowler Score The quality of the bowlers a team possesses also has significant impact on the game's outcome. Economical bowling and high wicket-taking ability are two traits common to all the good bowlers. Bowling average and bowling strike rate is a measure of the economical bowling while bowling strike rate is a measure of the bowler's high wicket-taking ability. We calculate the bowler score for all the players in a *match* using these two statistics as shown in algorithm 4.2.

The parameter used in line 2 is a measure of how consistently a player gets to bowl. It is used to differentiate between a seasoned bowler and a part-time bowler. Line 3 calculates the career bowling score of a player. Line 4 calculates the recent batting score of a player. Lines 6 - 9 normalize the career and recent bowling scores of all the players in a *match*. Line 10 calculates the cumulative *bowling_scores* which is a weighted sum of *careerscores* and *recent_scores* of *player p*.

Algorithm 2 Modeling Bowlers

Input: $Players(match, Team_A)$, $Players(match, Team_B)$
Output: $\phi^p_{bowling\_score}$ for all $p \in (Players(match, Team_A) \cup Players(match, Team_B))$

1: **for** $p \in (Players(match, Team_A) \cup Players(match, Team_B))$ **do**
2: $u \leftarrow \sqrt{\frac{C_{bowling\_innings}}{C_{matches\_played}}}$
3: $\phi^p_{career\_bowling\_score} \leftarrow u * (\frac{1}{(C_{bowling\_average} * C_{bowling\_strike\_rate})})$
4: $\phi^p_{recent\_bowling\_score} \leftarrow u * (\frac{1}{(F_{bowling\_average} * F_{bowling\_strike\_rate})})$
5: **end for**
6: **for** $p \in (Players(match, Team_A) \cup Players(match, Team_B))$ **do**
7: $\phi^p_{career\_bowling\_score} \leftarrow \frac{\phi^p_{career\_bowling\_score}}{max(\phi^p_{career\_bowling\_score})}$
8: $\phi^p_{recent\_bowling\_score} \leftarrow \frac{\phi^p_{recent\_bowling\_score}}{max(\phi^p_{recent\_bowling\_score})}$
9: **end for**
10: $\phi^p_{bowling\_score} = 0.8 * \phi^p_{career\_bowlting\_score} + 0.2 * \phi^p_{recent\_batting\_score}$

4.3 Calculation of Relative Team Strength : Static Feature

A team's batting and bowling strength will be a consolidated measure of the batting and bowling strengths of the 11 players playing in that match. Algorithm 4.3 illustrates the computation of $Relative\_strength_{Team_B/Team_A}$. Lines 1- 4 normalize the $\phi^p_{batting\_score}$ and $\phi^p_{bowling\_score}$ for all the players. Lines 5- 6 sum up the batting and bowling scores of all the players in $Team_A$ to compute $\phi^{Team_A}_{batting\_score}$ and $\phi^{Team_A}_{bowling\_score}$. Line 9 computes $Relative\_strength_{Team_B/Team_A}$. We only calculate $Relative\_strength$ for $Team_B$ because $Team_B$ bats in $innings_2$ according to our notation and we make predictions only for $innings_2$ in our model. $Team_A$ bowling score has a negative impact on $Team_B$ batting score and vice-versa in the formula in line 9.

4.4 Dynamic Features

To predict the outcome of an ongoing T20 (IPL) match we first split the $innings_2$ into 21 states. S_0 at the end of $innings_1$ and $S_i (1 \leq i \leq 20)$ at the end of i overs. At a state S_i we use the following three dynamic features to make the prediction:

- Runs remaining to be scored to win the match $R^i_{runs\_remaining}$
- Wickets that $Team_B$ still has in hand $R^i_{wickets\_remaining}$
- Balls remaining to be played by $Team_B$ in the innings $R^i_{ball\_remaining}$

These features capture the state of the match at any given instance while the match is in progress. These features change as the match progresses. These three features along with the $Relative\_strength_{Team_B/Team_A}$ (static feature: remains constant throughout a match) are given to a classifier along with the label (1 if $Team_B$ wins, 0 otherwise) to forecast predictions.

Algorithm 3 Modeling Team

Input: $Players(match, Team_A)$, $Players(match, Team_B)$,$\phi^p_{batting\_score}$, $\phi^p_{bowling\_score}$ $\forall p \in$ $(Players(match, Team_A),\ Players(match, Team_B))$

Output: $Relative\_strength_{Team_B/Team_A}$

1: **for** $p \in (Players(match, Team_A) \cup Players(match, Team_B))$ **do**

2: $\quad \phi^p_{batting\_score} \leftarrow \frac{\phi^p_{batting\_score}}{max(\phi^p_{batting\_score})}$

3: $\quad \phi^p_{bowling\_score} \leftarrow \frac{\phi^p_{bowling\_score}}{max(\phi^p_{bowling\_score})}$

4: **end for**

5: $\phi^{Team_A}_{batting\_score} = \sum_{p \in (Players(match, Team_A)} \phi^p_{batting\_score}$

6: $\phi^{Team_A}_{bowling\_score} = \sum_{p \in (Players(match, Team_A)} \phi^p_{bowling\_score}$

7: $\phi^{Team_B}_{batting\_score} = \sum_{p \in (Players(match, Team_B)} \phi^p_{batting\_score}$

8: $\phi^{Team_B}_{bowling\_score} = \sum_{p \in (Players(match, Team_B)} \phi^p_{bowling\_score}$

9: $Relative\_strength_{Team_B/Team_A} = \frac{\phi^{Team_B}_{batting\_score}}{\phi^{Team_A}_{bowling\_score}} - \frac{\phi^{Team_A}_{batting\_score}}{\phi^{Team_B}_{bowling\_score}}$

5 Experiments and Results

5.1 Datasets

The data used for this study can be categorized into: *historicaldata* pertaining to the career statistics and *ball_by_balldata* pertaining to various states during a match. The dataset for *careerstatistics* was scraped from the $cricinfo$ website [12] for seasons 3-10 of IPL. Similarly, for the *ball_by_ball* data for each match in seasons 3-10 of IPL, the dataset was scraped from the *cricsheet* website [13]. The dataset constitutes the match dynamics recorded after each *ball* such as; $runsscored$, $wicket\,lost$, $current\,batsmen$, $current\,bowler$ along with other useful information like $winnerofthematch$, using which training data is labeled, $date\,of\,fixture$ and $toss\,outcome$. We combined data from both these sources to make predictions using our model. IPL seasons 3-7 were used for training our model, season 8 for validation of parameters and seasons 9 and 10 were used for testing the accuracy of our model.

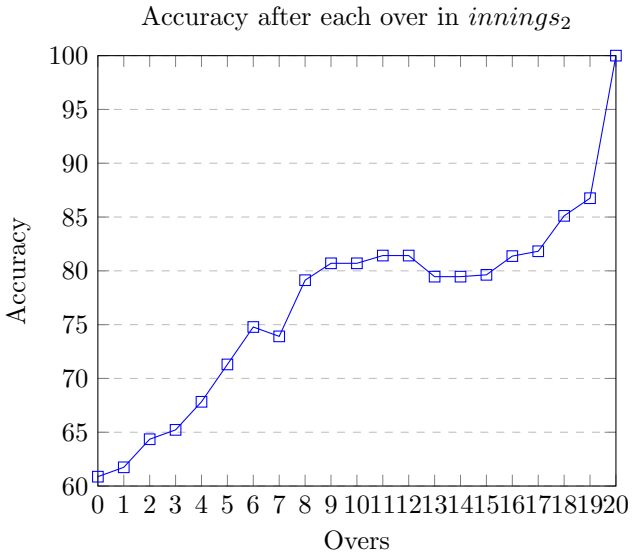
5.2 Weights

To select the weights used in line 9 of Algorithms 4.1 and 4.2, and the number of previous matches to be considered for the form (recent statistics) of a player, we used the data from seasons 3-8. Using exhaustive experimentation, we compared the estimated rankings of the batsman and bowlers before the start of a match (using the selected values of the two parameters) with the actual player rankings in that match. This process was repeated for all the matches in seasons 3-8. The minimum difference between the predicted and actual rankings was achieved when the previous 4 matches of a player were considered and the weight used in line 9 of Algorithms 4.1 and 4.2 equaled 0.8.

5.3 Results

Using the features listed in the previous sections along with the match outcome as label, we assessed various binary classifiers such as SVM, Random Forests, kNN and Decision Trees through their scikit-learn [14] implementations. The *ParameterGrid* mechanism has been used to evaluate all possible combinations of parameters for all the above listed algorithms. The Random Forest algorithm with parameters: $max\_features = 2$, $n\_estimators = 24$, $max\_depth = 9$ has yielded highest accuracy among the best models for all other classifiers. These results are illustrated in 5.3. We also compared our model with another baseline model; considering only the dynamic features for prediction obtaining an accuracy of 54.87% at the beginning of the *secondinnings* as against 60.87% recorded by our model. This shows the significance of $Relative\_strength_{Team_B/Team_A}$ as a reasonably good feature for making predictions. From the plot in 5.3, we observe an increasing trend in prediction accuracies after each over as the match progresses until completion. This proves the ability of our classifier to predict the winner with increasing confidence after each over. This also agrees with common intuition, that as the game nears its end, it is easier to predict the winner based on a given match state. While we examine the increasing trend of prediction accuracies in 5.3, some fluctuations are observed around the 7^{th} and 13^{th} overs which could be due to the inherent unpredictability of the sport.

The overall prediction accuracy obtained regardless of the number of defined states is 76.17%. To the best our knowledge, this is the highest recorded accuracy reported for Twenty20 cricket. [11] reported an accuracy of 69.64 for IPL Season-9. Although we cannot directly compare , because of the differences in underlying approaches (static and dynamic), it is notable that our model started off at an accuracy 60.87% at the beginning of the second innings increased to 86.75% at the end of the 19th over.



6 Conclusion and Future Work

The problem of dynamic winner prediction in a Twenty20 cricket match has been successfully addressed in this paper. A combination of dynamic features aforementioned along with the $Relative\_strength_{Team_B/Team_A}$ has furnished promising results. Furthermore, in order to further reinforce the prediction model, we intend to compute the $Relative\_strength_{Team_B/Team_A}$ dynamically during the match to achieve a better prediction accuracy.

References

1. Duckworth, Frank C., and Anthony J. Lewis. A fair method for resetting the target in interrupted one-day cricket matches. Journal of the Operational Research Society 49.3 (1998): 220-227.
2. Kimber, Alan. A graphical display for comparing bowlers in cricket. Teaching Statistics 15.3 (1993): 84-86.
3. Beaudoin, David, and Tim B. Swartz. The best batsmen and bowlers in one-day cricket. South African Statistical Journal 37.2 (2003): 203.
4. Van Staden, Paul Jacobus. Comparison of cricketers bowling and batting performances using graphical displays. (2009).
5. Lemmer, Hermanus H. THE ALLOCATION OF WEIGHTS IN THE CALCULATION OF BATTING AND BOWLING PERFORMANCE MEASURES. South African Journal for Research in Sport, Physical Education and Recreation (SAJR SPER) 29.2 (2007).
6. Khan, Mehvish, and Riddhi Shah. Role of External Factors on Outcome of a One Day International Cricket (ODI) Match and Predictive Analysis.
7. Sankaranarayanan, Vignesh Veppur, Junaed Sattar, and Laks VS Lakshmanan. Auto-play: A Data Mining Approach to ODI Cricket Simulation and Prediction. SDM. 2014.
8. Madan Gopal Jhawar, Vikram Pudi. "Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach." European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2016 2016), September 2016 , Conference Center, Riva del Garda. Report no: IIIT/TR/2016/32
9. Kaluarachchi, Amal, and S. Varde Aparna. CricAI: A classification based tool to predict the outcome in ODI cricket. 2010 Fifth International Conference on Information and Automation for Sustainability. IEEE, 2010.
10. A Criterion for Comparing and Selecting Batsmen in Limited Overs Cricket. GDI Barr Dept of Statistics & Economics, BS Kantor Dept of Economics, University of Cape Town. (2011)
11. Deep C Prakash, C Patvardhan and Vasantha C Lakshmi. "Data Analytics based Deep Mayo Predictor for IPL-9". International Journal of Computer Applications 152(6):6-11, October 2016.
12. ESPN Cricinfo, http://www.espncricinfo.com
13. IPL data, http://cricsheet.org
14. Pedregosa, Fabian, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12.Oct (2011): 2825-2830.