Cricket match outcome prediction using AI Techniques

Jerryl Davis[a], Amit Bhatia[b], Rajesh Goel[c], Pulkit Malhotra[d], Harsh Bhardwaj[e], Vikas Hooda[f], Dr. Narayana[g]

[a]Bain Capability Center, [b]EXL Services India Private Limited, [c]Telus International Limited, [d]Nagarro Software pvt. Ltd.,
[e]Conduent Technologies, [f]Hewlett Packard Enterprise Ltd., [g]Great Lakes

Abstract

The Indian Premier League (IPL) is a professional Twenty20 cricket league in India contested every year by teams representing different cities in India. The brand value of the IPL in 2019 was ₹475 billion (US$6.7 billion) tells you how big it has grown over the years. There have been twelve seasons of the IPL tournament so far starting back in 2008. This paper investigates machine learning technology to deal with the problem of predicting cricket match results based on historical match data of the IPL. We have used 5 seasons of IPL match data (2008-2013) as our dataset during this investigation.

Machine learning techniques including Logistic Regression, Naive Bayes, K-Nearest Neighbor (KNN), Support Vector Machine, Decision Tree, Random Forest, Boosting, Bagging, Gradient Boosting, LSTM and GRU have been adopted to generate predictive models from distinctive feature sets. In this study, we focus on predicting the match outcome, ball by ball when the second team starts batting; our dataset includes features around each ball bowled during a match and the runs scored by individual batsman on the balls they faced, making the data, sequential in nature. Our experimental tests showed Logistic Regression showing better results as compared to other techniques with an accuracy of 78%. We implemented LSTM and GRU due to the sequential nature of data and the results have been very encouraging. This experiment also helps the chasing team in understanding the point where the team started losing due to predicting the outcome on every ball and also helps teams to chase better by formulating a better strategy.

Introduction

Twenty20 cricket or Twenty-20 (often abbreviated to T20), is a shortened format of cricket. At the professional level, it was originally introduced by the England and Wales Cricket Board (ECB) in 2003 for the inter-county competition. In a Twenty20 game the two teams have a single innings each, which is restricted to a maximum of 20 overs. Together with first-class and List A cricket, Twenty20 is one of the three current forms of cricket recognized by the International Cricket Council (ICC) as being at the highest international or domestic level. A typical Twenty20 game is completed in about three hours, with each innings lasting around 90 minutes and an official 10 minute break between the innings. This is much shorter than previous forms of the game, and is closer to the timespan of other popular team sports. It was introduced to create a fast-paced game that would be attractive to spectators at the ground and viewers on television.

The Indian Premier League (IPL) is a professional Twenty20 cricket league in India contested during March or April and May of every year by eight teams representing eight different cities in India. The league was founded by the Board of Control for Cricket in India (BCCI) in 2008. The IPL has an exclusive window in ICC Future Tours Programme.

The IPL is the most-attended cricket league in the world and in 2014 ranked sixth by average attendance among all sports leagues. In 2010, the IPL became the first sporting event in the world to be broadcast live on YouTube. The brand value of the IPL in 2019 was ₹475 billion (US$6.7 billion), according to Duff & Phelps. According to BCCI, the 2015 IPL season contributed ₹11.5 billion (US$160 million) to the GDP of the Indian economy.

Sports analytics are a collection of relevant, historical, statistics that when properly applied can provide a competitive advantage to a team or individual. Through the collection and analyzation of these data, sports analytics in form players, coaches and other staff in order to facilitate decision making both during and prior to sporting events. As technology has advanced over the last number of year's data collection has become more in-depth and can be conducted with relative ease. Advancements in data collection have allowed for sports analytics to grow as well, leading to the development of advanced statistics as well sport specific technologies that allow for things like game simulations to be conducted by teams prior to play, improve fan acquisition and marketing strategies, and even understand the impact of sponsorship on each team as well as its fans.

The prediction of the outcome of the match while the match is in progress proves very beneficial to team members, team coaches and also bettors. For example, games tactics can be developed by club managers based on the outcome of previous matches or statistics related to certain teams. IPL being a very dynamic league, bettors and bookies are incentivized to bet on the match results before or during a game.

The betting market in India is worth $150 billion, a year. That includes $200 million bet on every one-day international played by the Indian cricket team, according to the Doha-based International Centre for Sports Security, which promotes integrity and security in sports.

One of the primary techniques used in sports analytics research is Machine Learning. These techniques are used to predict the outcome of a match by developing classification models based on certain independent features. The model is then trained using these independent features based on previous matches. Its effectiveness is then calculated using metrics such as predictive accuracy and error rate among others.

Since cricket matches are recorded using multiple independent variables within a historical dataset and one dependent variable, (the outcome of the match) this problem can be dealt with using predictive analytics (classification methods) within machine learning. A classification algorithm will process the input dataset to construct a classification model based on the available historical matches to predict the outcome of future matches as accurately as possible.

Most of the analytical techniques used, try to predict the outcome of a match even before it has started. In this paper, different techniques have been used to predict the outcome of a match while it is in progress, this means that a total of 120 predictions are made for every ball bowled when the second team starts batting.

We used 5 years' data collected from the IPL-T20 tournaments. More details on the data and empirical results are discussed in Sections 3 and 4 respectively. Another important aim of this paper is to seek influential features in a cricket match that might have influence on the outcome of a match.

Most of the researchers have tried to explore One-Day Internationals (ODI) and Test match format of cricket but as T20 is new and dynamic, it will be intriguing to investigate. This study can benefit cricket club managers, sport data analysts and scholars interested in sport analytics, among others.

The paper is organized as follows: Section 2 includes a brief overview about the game of cricket, previous work related to sports analytics and the application of machine learning to predict match outcomes. Section 3 discusses the framework for the prediction model and methods applied. Section 4 is dedicated to the description of the dataset and experimental results. Finally, conclusions and future work are presented in Section 5.

2. Literature Review

In the fast pace world where things are evolving so fast, sports are not left untouched specially cricket. Cricket has grown leaps and bounce in last few years especially after the introduction of Twenty 20 format of it. With its introduction craze and its popularity has grown exponentially. With the increased revenue and price money the importance of predicting the game has become more important than before and hence analyzing the data around it has become a popular exercise. With evolving technology especially AI, numerous player performance related statistics are available thesedays.

In cricket, to predict an outcome of a match, the primary task is to extract out the essentials factors (features) which affect result of a match. Interesting works have been done in the field of predicting outcome in cricket.

Bandulasirihas analyzed the factors like home field advantage, winning the toss, game plan (first batting or first fielding) and the effect of Duckworth Lewis method for one-day cricket format. Furthermore, Bailey and Clarke mention in their work that in one-day cricket format, home ground advantage, past performances, venue, performance against the specific opposition, current form are statistically significant in predicting total runs and predicting the outcome of a match. Similarly V. V. Sankaranarayanan, J. Sattar, and L. V. Lakshmanan, in their article, discuss around modeling home-runs and non-home runs prediction algorithms and considers taking runs, wickets, frequency of being all-out as historical features into their prediction model. But, they have not leveraged bowler's features and have given more emphasis to batsmen. Kaluarachchi and Aparna have proposed a tool that predicts match apriori, but player performance has not been considered into their model.

Nimmagadda et al. applied statistical techniques to predict a T20 match result while the match is in progress. The authors have designed a model using a statistical approach to achieve the optimum outcome. Firstly, a multiple regression model is tested to develop a prediction model. Using runs scored per over in the first inning and second inning, algorithms such as Logistic Regression with multi-variable linear regression and Random Forest are used to predict the final result.

The main result obtained was based on the impact of toss winner and resultant match winner. The predictive model considered the innings score at regular intervals and the final scores to predict the match result. The model predicted score and run rate projected score were quite near to the final score, in particular the score predicted by the model was more accurate to the actual score. When no feature selection was applied to the dataset the model's accuracy was not satisfactory, i.e. slightly above 50%. Pathak &Wadhwa investigated the prediction of the result for cricket matches using data mining techniques. They experimented on predicting the outcome for ODI (One Day International) match format based on various factors such as home ground, toss decision, innings, fitness of team players and other dynamic strategies. A Support Vector Machine (SVM) method was used to predict the result. Evaluating the accuracy of these techniques, they developed a tool COP (Cricket Outcome Predictor), which gives the probability for winning an ODI match. The data under study was the international cricket match data from 2001 to 2015 for ODI format and scraped from cricinfo website. Results obtained clearly showed that the classifiers derived by the SVM method outperformed those of Naïve Bayes and Random Forests methods. SVM produced 62% accuracy, whereas the accuracy rates of the other methods were around 60%. The COP tool developed in R software enabled a user to select the features to predict the match outcome, and the user could change between the classifiers to make multiple predictions. A notable result was observed when COP system was applied on the India vs. Australia series in which Naïve Bayes derived more competitive classifiers in terms of predicting the match outcome. Jhanwar & Pudi conducted an experimental study to predict the outcome for ODI cricket matches using data mining techniques. The authors investigated the match result using team players' performance individually in batting and bowling aspects. Initially the potential of 22 players was studied using their career statistics and KNN, Support Vector Machine (SVM), Random Forests, Logistic Regression and Decision Trees techniques were applied. To predict the outcome of the match, the relative strength of each team is studied, along with the venue of the match and toss result. The data considered under the study was cricket matches from 2010 to 2014 for 9 country teams in international One-Day format. The accuracy of the KNN model was higher than the other models in predicting the relative strength of the team players giving almost 71% accuracy for the ODI match. There was no feature selection involved in this study. Kampakis& Thomas conducted a study to predict the outcome of cricket matches in twenty over format. The competition under study was the English Cricket Cup and the model was tested on seasons 2009 to 2014, based on the data from previous matches. A model was developed on simple prediction and then further investigation was carried out on complex features for in-depth analysis. Initially the team data was used and then player data was analyzed. Feature selection methods utilized were Chi-square testing, mutual information and Pearson correlation. The authors utilized Naive Bayes, Logistic Regression, Random Forests and Gradient Decision Trees on the selected features from the data. By applying these methods to predict the match outcome, it was found that the model derived by Naïve Bayes offered around 64% prediction accuracy on the dataset used. At the same time comparing the accuracy of different techniques, Naïve Bayes produced the highest level of accuracy, the lowest was Gradient Decision Trees. Munir et al. experimented with twenty over format cricket matches to predict the outcome using various data mining techniques. The main aim of the study was to combine pre-game and in-game data to predict the outcome. They considered the T20 International match data along with IPL data till 2015 as the training data set.

In depth analysis was conducted by segmenting the data on the basis of venue, one team against all other teams, batting first and so on. Decision Tree was applied to predict the match outcome, and produced models with around 78% accuracy for the team that bats first and 75% when it bats second. IG technique was used for feature selection.

K. Kapadia, H. Abdel-Jaber, F. Thabtah et al., in their research paper, come up with intelligent models to predict a match outcome based on the impact of home ground and toss winner respectively. The team that wins the toss contemplates factors such as weather, pitch and outfield to decide whether to bat or field first, with the intention of securing a strategic advantage. Two models are formulated in the paper, one depicting the impact of home ground and the other considering the effect of toss decision. The machine learning algorithms that have been implemented to derive the predictive models are Naïve Bayes, Random Forest, K-nearest neighbor and Model Decision Tree.Naive Bayes is the most accurate model to predict the winner derived by the considered machine learning techniques against the Home Team features. Similarly, KNN is the most accurate model when it comes to predicting the match results using the toss winning feature. The accuracy they get through these model lie in between 50-65%.

3. Methodology

This research paper attempts to predict the outcome of a T20 match while the match is in progress. This is different from the earlier techniques that have been implemented in the papers. Most of the research around predicting the match outcome takes into account a lot of different features and only makes one prediction before the match begins.

We have tried to follow a different approach where we make predictions on the outcome of a match when the second team starts batting. This prediction is in regards to the second batting team and if the team will win or lose based on the match circumstances. The features that we have used to train the model are different from any other set of features that have been used before. The feature set as shown in Table 1, comprises of every ball bowled to the team batting second and every run scored on every ball, also taking into account, the extras and the over, these being our independent variables and the result or the outcome being the dependent feature. As you can see from the table below, these are the final set of feature that are fed to the model. The data is then split into training and testing with a ratio of 70:30. Here another important note to take care of, is that the whole match is fed to the model for training so If there are a total of 100 matches, 70 wholes matches which consist of 120 balls each are fed to the model for training, similarly, a total of 30 whole matches are fed for testing, this means that there are a total of 120*30 test predictions by the model. This is way different than any other approach used before.

When it comes to predicting the outcome of a match, we not only use traditional Machine learning model to predict the outcome but also implement deep learning models like LSTM and GRU, which are part of a special section of deep learning models called Sequence models. These models are used to access the sequential nature of the data, thus showing better results during prediction. Traditional Machine learning models have shown varying accuracy over this data. The accuracy tends to vary with the increasing amount of data and hence, the use of deep learning models like LSTM and GRU make it more preferable. These models come from the family of neural networks, thus, feature engineering is taken care of, by the model and unlike other deep learning models, and they consist of a memory based architecture which helps in retaining the various aspects of the match such that it is able to predict the outcome at different situations. The accuracy of the model tends to increase steadily as more data is fed to the model.Our machine learning pipeline follows the traditional methodology of preparing the unstructured data, then splitting the data and feeding it to our suite of machine learning models. We then evaluate the model using metrics like accuracy, recall, precision and F1 score.

Table 1	
The dataset variables description	
Match_id	The match Id for each match in the IPL
Ball	The current order of the ball bowled
Over	The current over of the match
Balls Played	Balls left to be bowled
Bman1	Runs scored by the first batsman on each ball
Bman2	Runs scored by the second batsman on each ball
Bman3	Runs scored by the third batsman on each ball
Bman4	Runs scored by the fourth batsman on each ball
Bman5	Runs scored by the fifth batsman on each ball
Bman6	Runs scored by the sixth batsman on each ball
Bman7	Runs scored by the seventh batsman on each ball
Bman8	Runs scored by the 8th batsman on each ball
Bman9	Runs scored by the 9th batsman on each ball
Bman10	Runs scored by the 10th batsman on each ball
Bman11	Runs scored by the 11th batsman on each ball
Bowl1	Runs scored on every ball bowled by bowler 1
Bowl2	Runs scored on every ball bowled by bowler 2
Bowl3	Runs scored on every ball bowled by bowler 3
Bowl4	Runs scored on every ball bowled by bowler 4
Bowl5	Runs scored on every ball bowled by bowler 5
Bowl6	Runs scored on every ball bowled by bowler 6
Bowl7	Runs scored on every ball bowled by bowler 7
Bowl8	Runs scored on every ball bowled by bowler 8
Bowl9	Runs scored on every ball bowled by bowler 9
Bowl10	Runs scored on every ball bowled by bowler 10
Extra_runs	Extras given by bowlers calculated for each ball
Batsman_runs	Runs scored by batsman on each ball
Remaining Balls	The total no of balls left
Wickets Left	The total no of wickets left
Present Score	Current score
Target	Total runs scored by the team batting first
Result	This tells if the team batting second won or lost

Table showing features fed to the model

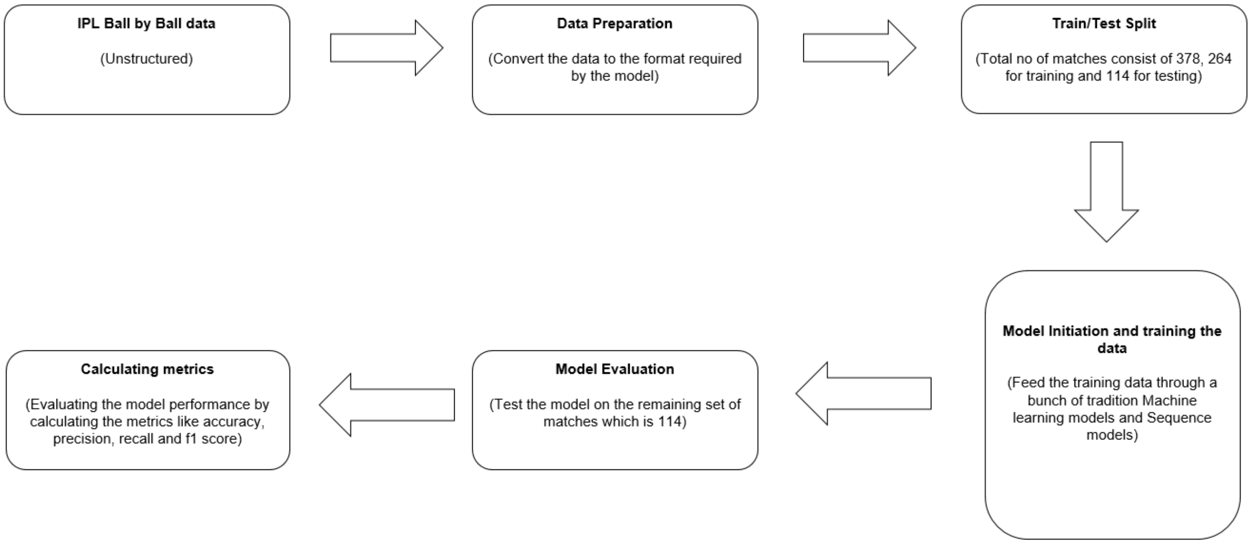

IPL Ball by Ball data
(Unstructured)
Data Preparation
(Convert the data to the format required by the model)
Train/Test Split
(Total no of matches consist of 378, 264 for training and 114 for testing)
Model Initiation and training the data
(Feed the training data through a bunch of tradition Machine learning models and Sequence models)
Model Evaluation
(Test the model on the remaining set of matches which is 114)
Calculating metrics
(Evaluating the model performance by calculating the metrics like accuracy, precision, recall and f1 score)

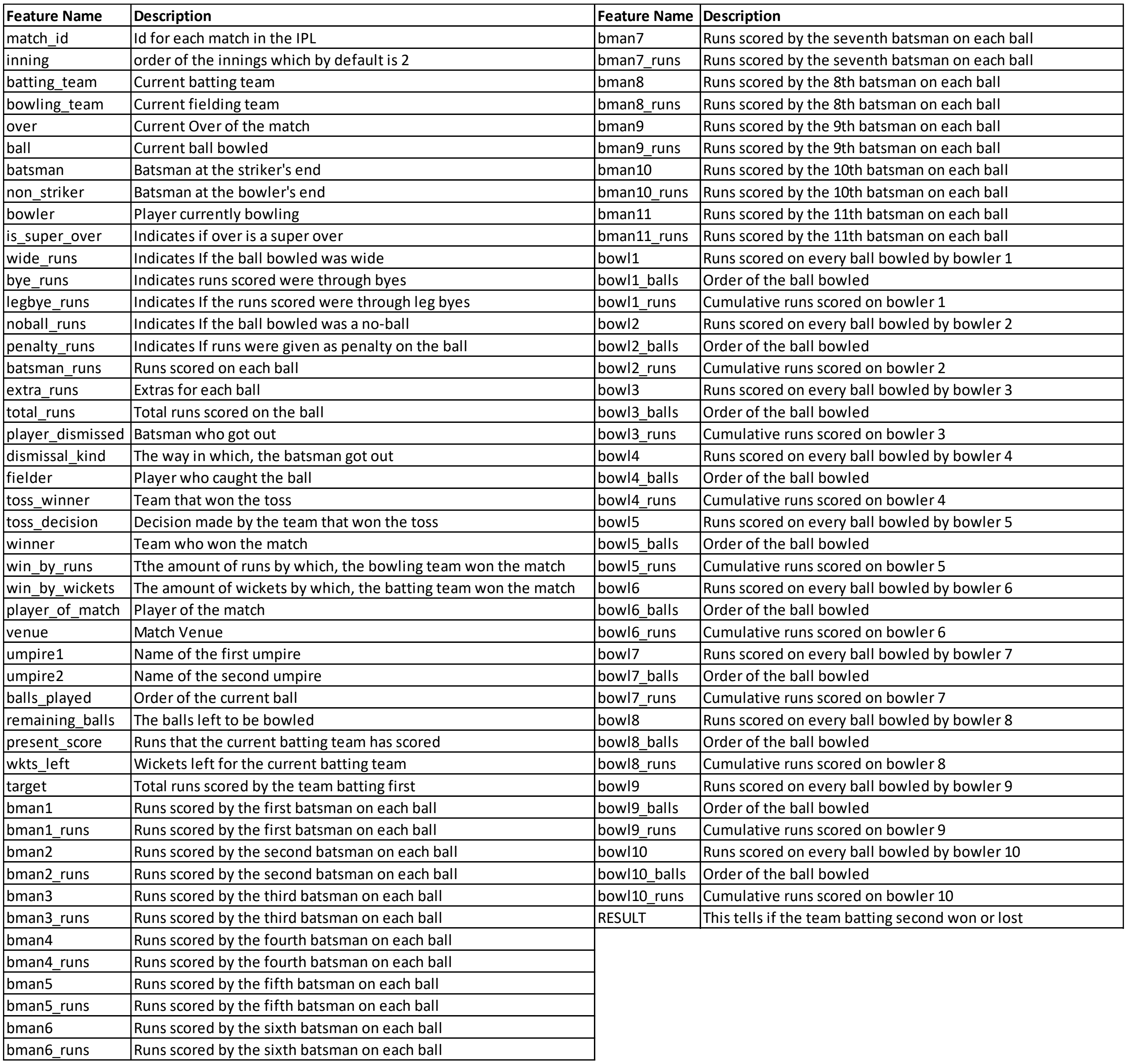
Figure 1 Approach Methodology

4. Data and Result Analysis

4.1 Data and feature Description

Historical data of Indian Premiere League (IPL-T20) tournaments is captured to perform prediction analysis. We consider IPL Cricket matches for 5 years (2008 to 2013) and store them in a dataset. The dataset used consists of 90 columns and 43492 rows. This dataset captures every ball bowled and every run scored and also takes into account certain other features. There are 378 matches which have been recorded in the dataset through the Cricinfo and IPL website. 'Match ID' is unique to every match which we use to split the data for training and testing.In order to train the model, matches as a whole needs to be provided.
The raw data collected needs to be modified and transformed to the way, necessary for the model to gain unique insights about every match circumstance.

This would then help the model to predict, taking into account, and the different situations. Even though, there are 90 columns, through feature selection process, we have come up with 31 features which provides better accuracy, this has been mentioned in great detail in the next section. The initial features have been described in Table 2.



Feature Name	Description	Feature Name	Description
match_id	Id for each match in the IPL	bman7	Runs scored by the seventh batsman on each ball
inning	order of the innings which by default is 2	bman7_runs	Runs scored by the seventh batsman on each ball
batting_team	Current batting team	bman8	Runs scored by the 8th batsman on each ball
bowling_team	Current fielding team	bman8_runs	Runs scored by the 8th batsman on each ball
over	Current Over of the match	bman9	Runs scored by the 9th batsman on each ball
ball	Current ball bowled	bman9_runs	Runs scored by the 9th batsman on each ball
batsman	Batsman at the striker's end	bman10	Runs scored by the 10th batsman on each ball
non_striker	Batsman at the bowler's end	bman10_runs	Runs scored by the 10th batsman on each ball
bowler	Player currently bowling	bman11	Runs scored by the 11th batsman on each ball
is_super_over	Indicates if over is a super over	bman11_runs	Runs scored by the 11th batsman on each ball
wide_runs	Indicates If the ball bowled was wide	bowl1	Runs scored on every ball bowled by bowler 1
bye_runs	Indicates runs scored were through byes	bowl1_balls	Order of the ball bowled
legbye_runs	Indicates If the runs scored were through leg byes	bowl1_runs	Cumulative runs scored on bowler 1
noball_runs	Indicates If the ball bowled was a no-ball	bowl2	Runs scored on every ball bowled by bowler 2
penalty_runs	Indicates If runs were given as penalty on the ball	bowl2_balls	Order of the ball bowled
batsman_runs	Runs scored on each ball	bowl2_runs	Cumulative runs scored on bowler 2
extra_runs	Extras for each ball	bowl3	Runs scored on every ball bowled by bowler 3
total_runs	Total runs scored on the ball	bowl3_balls	Order of the ball bowled
player_dismissed	Batsman who got out	bowl3_runs	Cumulative runs scored on bowler 3
dismissal_kind	The way in which, the batsman got out	bowl4	Runs scored on every ball bowled by bowler 4
fielder	Player who caught the ball	bowl4_balls	Order of the ball bowled
toss_winner	Team that won the toss	bowl4_runs	Cumulative runs scored on bowler 4
toss_decision	Decision made by the team that won the toss	bowl5	Runs scored on every ball bowled by bowler 5
winner	Team who won the match	bowl5_balls	Order of the ball bowled
win_by_runs	Tthe amount of runs by which, the bowling team won the match	bowl5_runs	Cumulative runs scored on bowler 5
win_by_wickets	The amount of wickets by which, the batting team won the match	bowl6	Runs scored on every ball bowled by bowler 6
player_of_match	Player of the match	bowl6_balls	Order of the ball bowled
venue	Match Venue	bowl6_runs	Cumulative runs scored on bowler 6
umpire1	Name of the first umpire	bowl7	Runs scored on every ball bowled by bowler 7
umpire2	Name of the second umpire	bowl7_balls	Order of the ball bowled
balls_played	Order of the current ball	bowl7_runs	Cumulative runs scored on bowler 7
remaining_balls	The balls left to be bowled	bowl8	Runs scored on every ball bowled by bowler 8
present_score	Runs that the current batting team has scored	bowl8_balls	Order of the ball bowled
wkts_left	Wickets left for the current batting team	bowl8_runs	Cumulative runs scored on bowler 8
target	Total runs scored by the team batting first	bowl9	Runs scored on every ball bowled by bowler 9
bman1	Runs scored by the first batsman on each ball	bowl9_balls	Order of the ball bowled
bman1_runs	Runs scored by the first batsman on each ball	bowl9_runs	Cumulative runs scored on bowler 9
bman2	Runs scored by the second batsman on each ball	bowl10	Runs scored on every ball bowled by bowler 10
bman2_runs	Runs scored by the second batsman on each ball	bowl10_balls	Order of the ball bowled
bman3	Runs scored by the third batsman on each ball	bowl10_runs	Cumulative runs scored on bowler 10
bman3_runs	Runs scored by the third batsman on each ball	RESULT	This tells if the team batting second won or lost
bman4	Runs scored by the fourth batsman on each ball		
bman4_runs	Runs scored by the fourth batsman on each ball		
bman5	Runs scored by the fifth batsman on each ball		
bman5_runs	Runs scored by the fifth batsman on each ball		
bman6	Runs scored by the sixth batsman on each ball		
bman6_runs	Runs scored by the sixth batsman on each ball		

Table showing features of a dataset

4.2 Feature Selection

Feature selection methods are intended to reduce the number of input variables to those that are believed to be most useful to a model in order to predict the target variable.Some predictive modeling problems have a large number of variables that can slow the development and training of models and require a large amount of system memory. Additionally, the performance of some models can degrade when including input variables that are not relevant to the target variable.

The method that we have used for feature selection is wrapper method. Wrapper feature selection methods create many models with different subsets of input features and select those features that result in the best performing model according to a performance metric. These methods are unconcerned with the variable types, although they can be computationally expensive.

The features in our dataset mostly consist of a combination of numerical and categorical values. The features that were selected were meant to be sequential in nature such that all the features combined would provide the match summary to the model at the time of training. Based on Table 2, a total of 90 columns were selected, the final set of features fed to the model were based on the process of constant training and model evaluation. For the wrapper method, we used classification algorithms like Logistic Regression, Gaussian Naïve Bayes, KNN, SVC, Decision Tree Classifier, ensemble techniques like AdaBoost and Bagging Classifier, Gradient Boosting Classifier and Bernoulli Naïve Bayes. From our testing, we came to know that out of the 88 features, only 32 were required to be fed to our traditional Machine Learning models. In case of deep learning models, we again feed the same amount of features to it.

4.3 Experimental Setting and Evaluation Measures

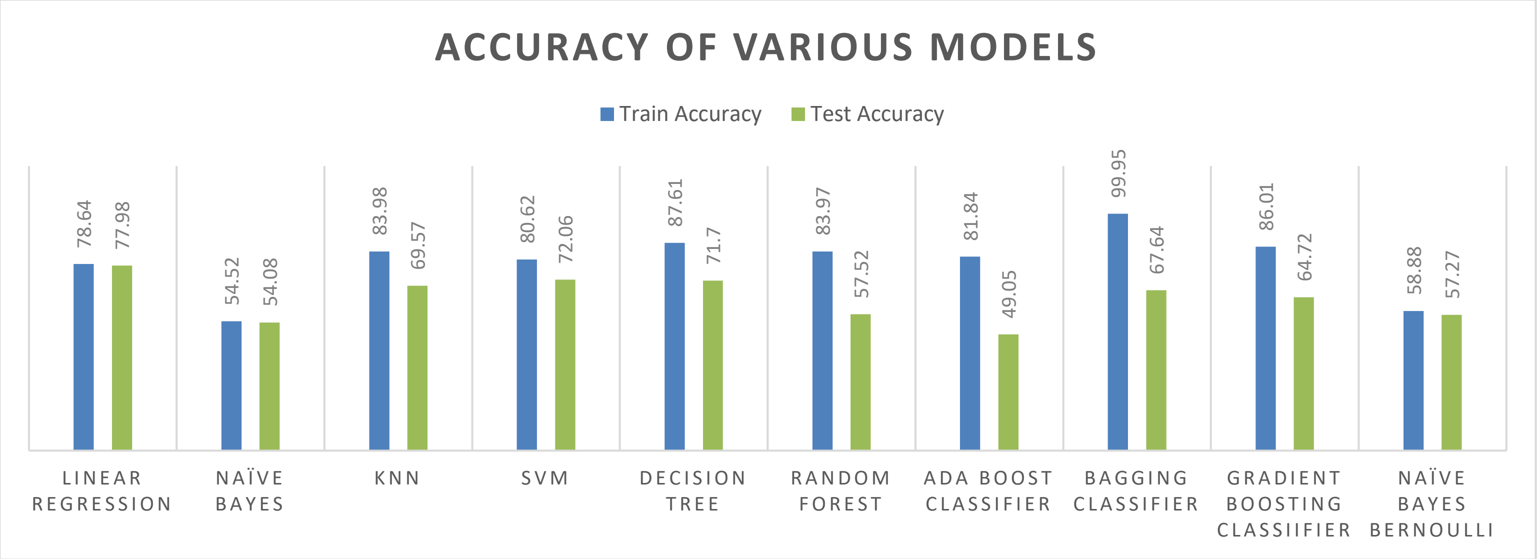
For this research, all the experiments were conducted on Google Colab, a tool that allows you to write and execute Python in your browser.Colabnotebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. To visually explore the data, we used the in-built function like Matplot-lib and plotly. The processing machine used to conduct this experiment is an Intel i7 processor with 16GB of memory on Windows 10, 64 bit operating system. Different algorithms are adopted to deal with the research problem.The predictive models derived by the machine learning algorithms have been evaluated using various metrics including classification Accuracy, Precision and Recall.

4.4 Result Analysis

4.4.1 Case 1: Traditional Machine Learning Models on IPL Data

Traditional Machine learning models have been used on the dataset to predict the result of the match for every ball bowled. We use a total of 378 matches, out of which, 264 matches have been used for training and 114 matches have been used for testing. Since our aim is to predict the outcome of a match when the second team starts batting, we plan to train the model taking into account the whole second innings, also taking into account, the sequential nature of the data. The attribute ‘Result’, will be the target class used by the model for predicting the outcome of a match. Since this is a ball by ball prediction, for every match, we make 120 predictions (increases if you include extras) so since, we have 114 matches, we make a total of 120*114, 13680 predictions. We then check the accuracy of the model and on our train and test data. We also calculate the precision, recall and f1 score for the best performing model.

The classification results (accuracy) derived by the model for the train and test data have been shown in the table below. You can see that most of the traditional machine learning models tend to over fit when you use them on sequential data, the only model that seems to have been performing efficiently are Linear Regression, Naïve Bayes and Naïve Bayes Bernoulli while all the other models are clearly overfitting. KNN shows a training accuracy of 83% which is similar for Random Forest but they show very poor test accuracy. Model like Decision Tree, Bagging and Gradient Boosting Classifier also show huge training accuracy but fail miserably when it comes to test data. Since, Linear Regression is the least over fit; we calculate metrics like Precision, Recall and F1 score for the model which will be shown later as we compare the accuracy with Deep Learning Models.



Graph showing train and test accuracies for various algorithms

4.4.2 Case 2: Deep Learning Models on IPL Data

The performances of traditional machine learning models tend to be over fit or their accuracy tend to decrease with increasing the training samples. In this regards, we prefer deep learning models. There are various factors because of which deep learning models are in heavy use at the moment. They are as follows:

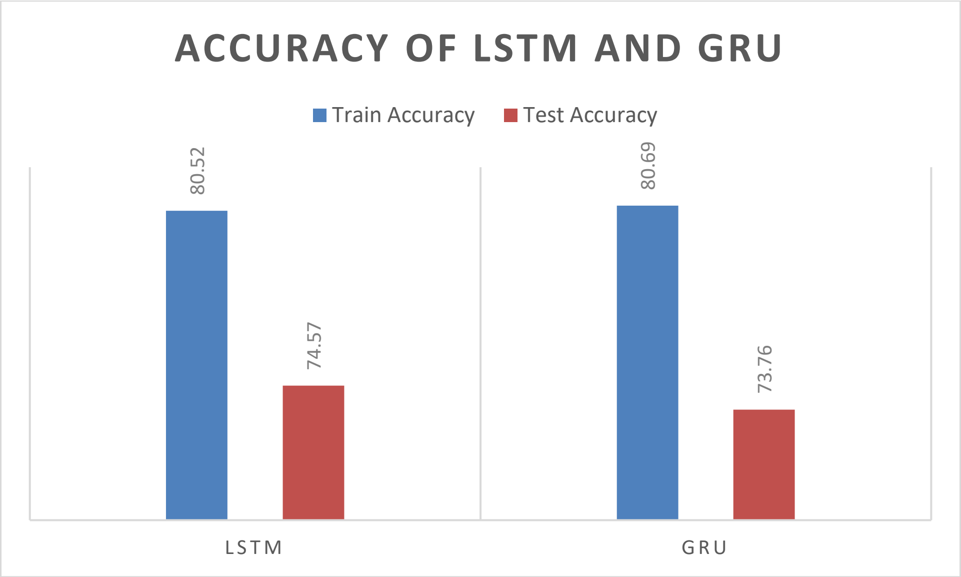
1. Easy-to-extract features
2. Good at extracting patterns easily
3. Easy to predict from training data
4. No assumptions made on the data

There are various deep learning algorithms that are in use but since our data is sequential in nature, we require a model which has a memory state and this lead us to implementing RNN models. We use two basic RNN models in LSTM and GRU. A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

LSTM and GRU

These networks are capable of remembering long-term dependencies. They are designed to remember information for long periods of time without having to deal with the vanishing gradient problem. They have internal mechanisms called gates that can regulate the flow of information. These gates can learn which data in a sequence is important to keep or throw away. Thus, it can pass relevant information down the long chain of sequences to make predictions.

We used LSTM and GRU on the IPL dataset. For training, we again used a total of 264 matches and for test, we used a total of 114 matches. The train and test accuracies for both the models have been done mentioned in the table below. Both the models tend to be less over fit and maintain very similar accuracies although LSTM tends to be a tad bit higher.



Graph showing the train and test accuracy

I

4.4.3 Case 3: Evaluating Models based on different metrics

In order to understand the performance of the model, we evaluate it based on different characteristics. The approach was to calculate the accuracy, precision, recall and F1 score of LSTM and GRU, we also include Linear Regression as well since the model showed a good test accuracy on the IPL dataset. We use the models to predict test data for different range of overs, 1-5, 6-10, 11-15 and 16-20. We then calculate the accuracy, precision, recall and F1 score for the various range of overs. A total of 13680 predictions are made by the three models individually. If we compare the three models, we can see that Linear Regression has much better accuracy across the different range of overs and it has the highest accuracy overall. The accuracy of Linear Regression model is 77.89 overall and it is 72.85 and 70.83 for LSTM and GRU respectively. We can see that the accuracy of all the models are the highest for the overs 11-15 which is 78.72, 79.13 and 81.12 for LSTM, GRU and Linear Regression respectively. If we also take into account other metrics like Precision, Recall and F1 score, we see that Linear Regression again has the highest metrics overall. For precision, we see that the highest value depicted is for the overs, 11-15 and the overall value again is that for Linear Regression. For Recall, the highest is shown By GRU overall, followed by LSTM and then Linear Regression, the F1 score is again, highest for Linear Regression as compared to the other three models.

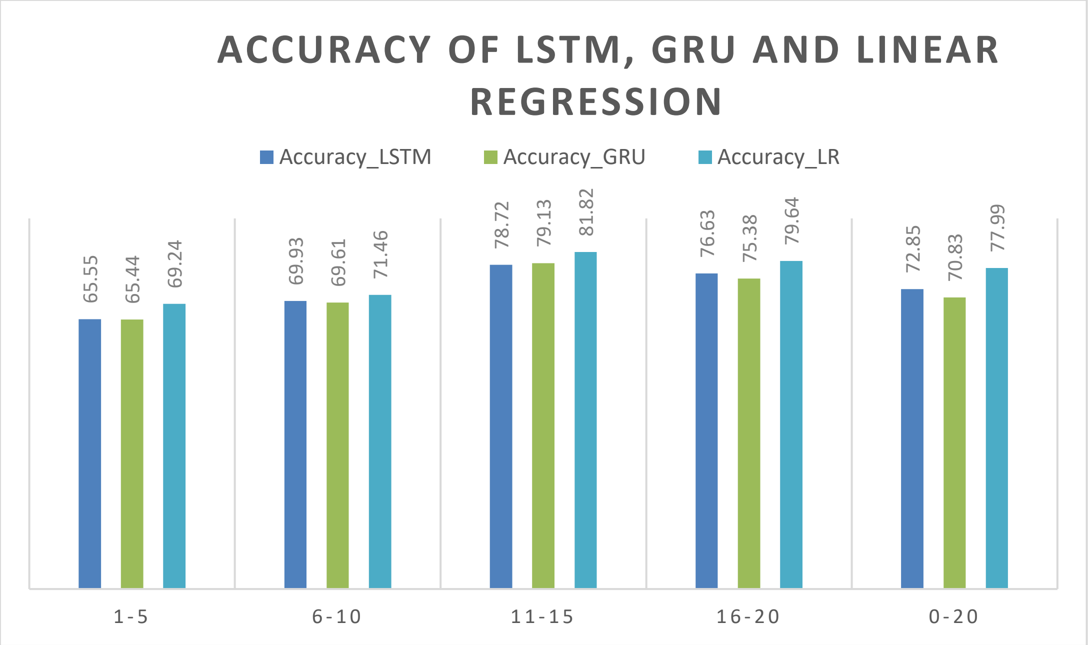


Fig 1: Accuracy for LSTM, GRU and Linear Regression for different range of overs

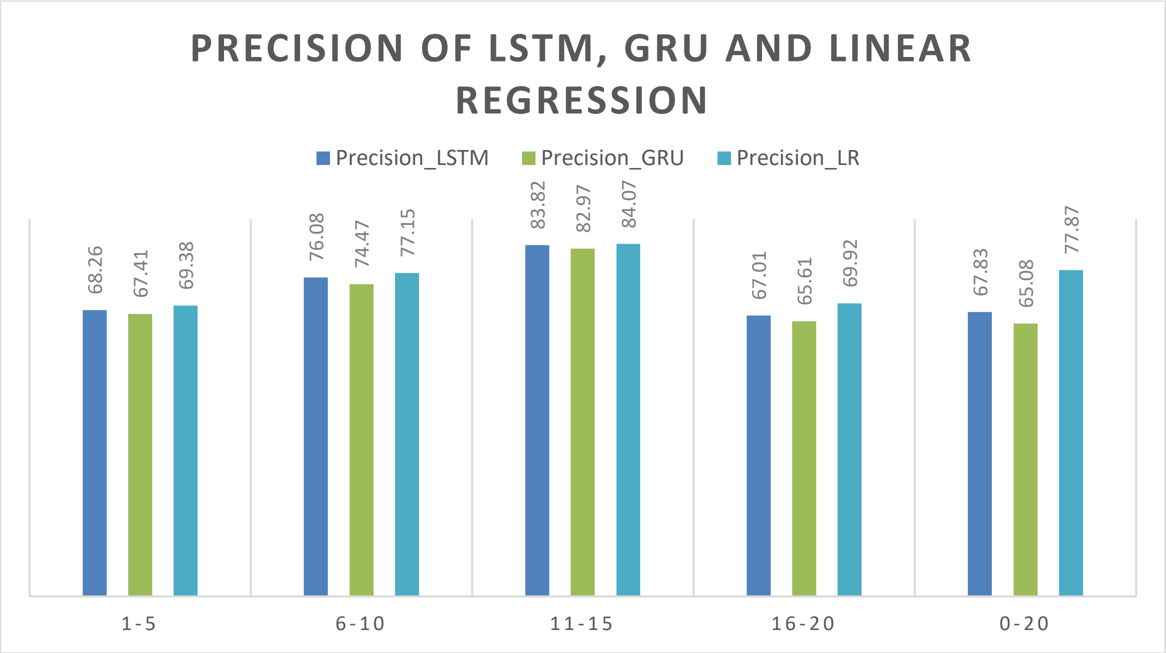


Fig 2: Precision for LSTM, GRU and Linear Regression for different range of overs

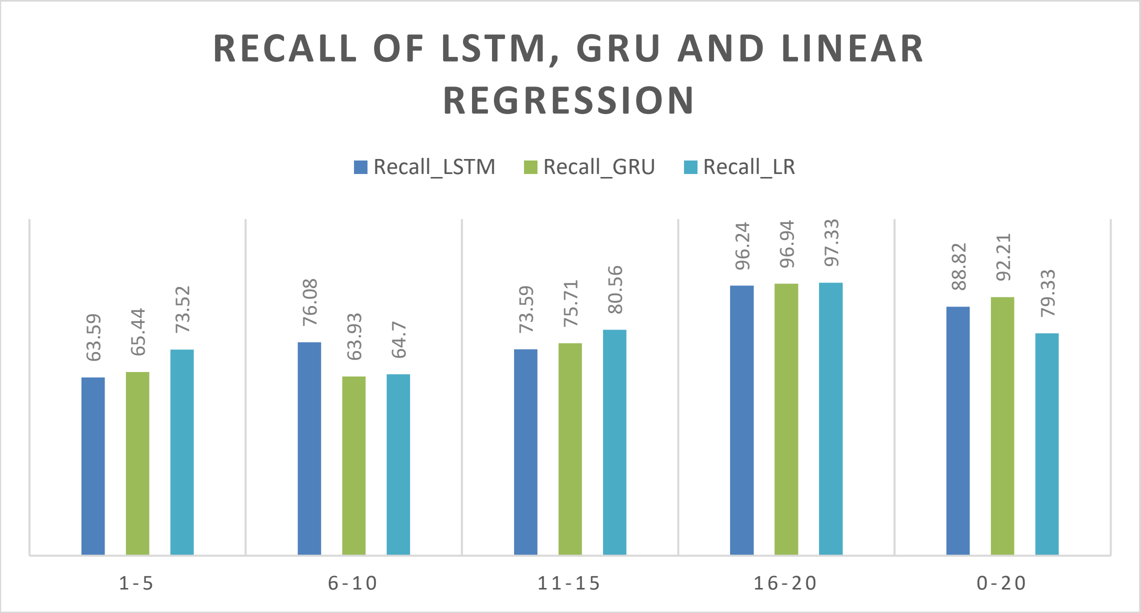


Fig 3: Recall for LSTM, GRU and Linear Regression for different range of overs

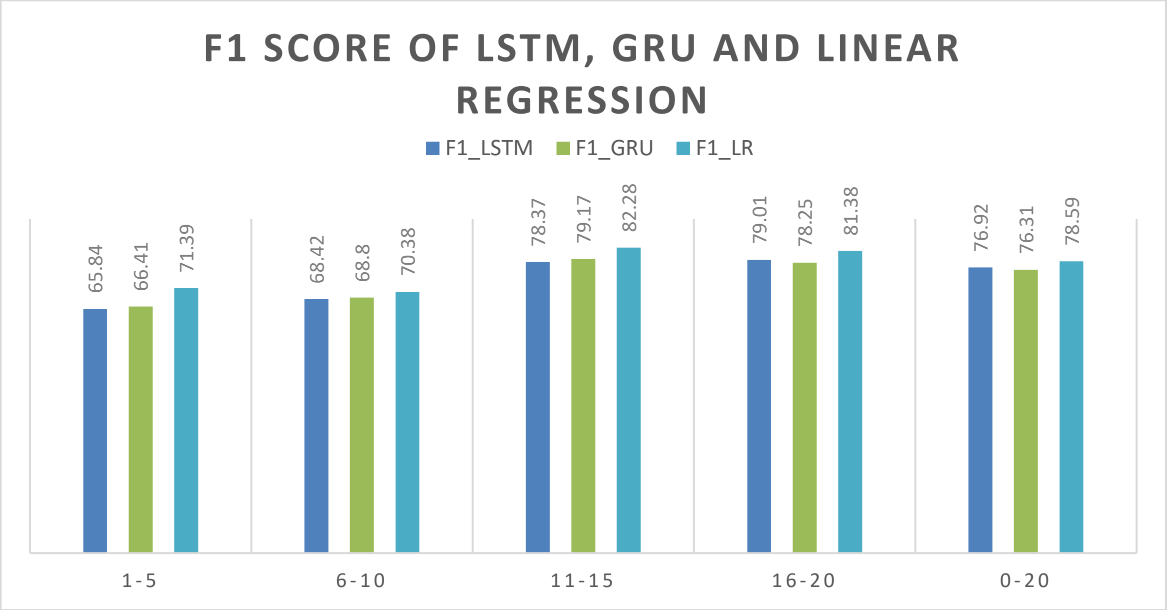


Fig 4: F1 score for LSTM, GRU and Linear Regression for different range of overs

Conclusion and Future Work

Applying machine learning for analyzing cricket sports by considering historical game data, players’ performance, natural parameters, pre-game conditions and other features is beneficial for multiple stakeholders. In a dynamic format like T20, where the situation in a game changes on every ball, it becomes challenging to predict the match outcome. For predicting the final outcome of a T20 cricket match, we have investigated machine learning technology for the possibility of improving the prediction rate of the results of matches.

In our dataset, we can see that most of the matches were won by Chennai Super Kings (15.1%) and they batted first for most of the time (14.9%) whereas Delhi Daredevils batted second most of the time (13.1%) and only won (10.4 %) of the total matches. This model can used to analyze chases such that chasing a score, by a T-20 team could be improved substantially as the model provides a ball by ball description.

You can check the outcome at every stage to understand if the team is winning or losing while chasing a score and it gives a better perspective with the run rate also taken into account. This experiments helps in understanding when the team started losing the match while chasing since it predicts the outcome on every ball. It helps the teams to formulate a better strategy while batting second in a match.

Currently we have built our model considering there are 11 non named (generic) players per team and didn’t take any individual players statistics into consideration. Once we have the numbers from first innings and the target score, our model does the ball by ball prediction for the second innings. During this entire process we considered data from 6 IPL seasons (2008-2013). Going forward we would like to use data from all IPL seasons and expect our model to perform even better. In future we would like to include features having details around individual players like past performances against the opponent, venue details, rankings, country where match is being played, etc. Currently we are also considering if we can build a UI on top of our model so that we can feed our model with live match stats and see it perform in real time. And finally, in future we also intend to build a recommendation model which can suggest during the second innings as to which player is best suited to play in a particular situation during the match. This can help the team management to analyze the situation in a better way and take actions accordingly. We would also need to run the model on more seasons and also take into account, various flavors of LSTM and GRU.

References

[1] SandeshBanankiJayantha,b,∗ , A team recommendation system and outcome prediction for the game of cricket, pp.263-264

[2] Kumash Kapadia, Sport analytics for cricket game results using machine learning: An experimental study, Applied Computing and Informatics, (2019)

[3] RabindraLamsal, Ayesha Choudhary, Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning, arXiv:1809.09813v4 [stat.AP] 25 Jun 2019

[3] Analytics, C., 2017. Anaconda software distribution. Computer software Vers,2-2.

[4] Wikipedia on the Game of Cricket, website [Online]http://en.wikipedia.org/wiki/Cricket

[5] CricInfo, Website for cricket data, [online]http://www.cricinfo.com

[6] A. Nimmagadda, N.V. Kalyan, M. Venkatesh, N.N.S. Teja, C.G. Raju, Cricket score

[7] and winning prediction using data mining, Int. J. Adv. Res. Development 3 (3)

[8] (2018) 299–302.

[9] N. Pathak, H. Wadhwa, Applications of modern classification techniques to

[10] Predict the outcome of ODI cricket, Procedia Comput. Sci. 87 (2016) 55–60.

[11] Quinlan, J.R., 2014. C4. 5: programs for machine learning. Elsevier.

[12] Foundation for Statistical Computing. Retrieved January 04, 2019 from http://

[13] Reddy, S., 2018. IPL 2017. Retrieved December 05, 2018 from https://www.

[14] kaggle.com/somashekharareddy/ipl-2017#DIM_MATCH.xlsx.

[15] Stolte, C., Hanrahan, P., Chabot, C., n.d. Tableau: Business Intelligence and

[16] Analytics Software. Retrieved December 15, 2018 from https://www.

[17] tableau.com/.

[18] Thabtah F., 2006. Pruning techniques in associative classification: survey and

[19] Comparison. J. Digital Information Manage., Volume 4:202-205. Digital

[20] Information Research Foundation.

[21] Thabtah, F., Cowling, P., Peng, Y., 2005. A study of Predictive Accuracy for Four

[22] Associative Classifiers. J. Digital Information Manage., Volume 3:202–205.

[23] Digital Information Research Foundation.

[24] I.H. Witten, E. Frank, M.A. Hall, C.J. Pal, Data Mining: Practical Machine

[25] Learning Tools and Techniques, Morgan Kaufmann, 2016.

[26] K. Kapadia et al. / Applied Computing and Informatics xxx (xxxx) xxx