Journal of Theoretical and Applied Information Technology
15th May 2021. Vol.99. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

UNIQUE APPROACH FOR CRICKET MATCH OUTCOME PREDICTION USING XGBOOST ALGORITHMS

NIMMI WEERADDANA[1] , SAMINDA PREMARATNE
[2]Department of Information Technology, University of Moratuwa, Sri Lanka
E-mail: [1]nimmirashinika@gmail.com, [2]samindap@uom.lk

ABSTRACT

Context. Cricket is among the top five rated sports in the world based on a set of ranking criteria. It is highly popular in Asia, Australia, the United Kingdom, and many other countries. Several attempts have been made in prior research to predict the cricket match outcomes based on the current status of the match. *Goal*. ***Importance***. Insights to a cricket match such as what is the winning probability given the current status of the match? What could be the predicted score at the end of the next over? Who is the best batsman to join the game after a wicket? Who is the best bowler to break a partnership? When can another wicket is likely to occur in the future? are important to many stakeholders. These insights could be used by interested stakeholders such as team leads, coaches, sports professionals, etc. ***Novelty***. In this research, we present a novel approach to predict the winning team (winning probability) and score at the end of the next over using boosting algorithms. ***Results***. Our results yield 84.4 of accuracy for winning team classification and 1.41 of mean squared error for the score prediction. Besides, our results outperform the traditional data mining approaches. ***Scope***. We limit our research only to the Cricket matches between Sri Lanka and India to reduce the complexity of the scope of the problem. ***Conclusion***. We believe the research questions addressed through this research paper are useful to real-time analysis and decision making.

Keywords: *Data mining, XGBoost, Classification, Dummy variables, Regression*

1. INTRODUCTION

Cricket is a bat-and-ball game with eleven players. Among several sports, Cricket was ranking at the second place according to global criterion including global fan base, audience, popularity on the Internet, number of professional leagues in the world, the average salary of athletes in the top leagues, number of countries in which the sport is popular, gender equality, access to the general public, and etc [1]. Furthermore, the estimated fanbase is over 2.5 billion for Cricket [1]. Thus, reflecting the popularity of Cricket as a global sport. This implies, several stakeholders are interested in Cricket match outcomes worldwide; for example, sports professionals have intensively participated in activities that reward financial benefits by watching cricket tournaments and participating in the bidding. These professionals tend to seek consultation assistance from cricket experts, online tools in addition to using their own experiences. Moreover, consulting experts and using online tools are typically commercial options for predictions related to Cricket matches. The trustworthiness and underlying algorithms implemented in commercial options are not publicly available, thus it is a concern. In addition, experts' decisions might depend on their level of expertise and prone to personal biases [2]. As a result, there is a possibility that important patterns or information available in data could be omitted unseen by the experts occasionally [2]. The aforementioned limitations could be overcome by formulating mathematical models capable of predicting the outcomes of a Cricket match and providing insights about the future based on the match's current status. This particular field of study is known as *Sports Analytics* which is a quite challenging but emerging research domain in recent history [3].

Since sports professionals actively participate in rewarding financial activities related to Cricket, they have the desire to know interesting insights about the future forecasting of a live cricket match. They must be happy to know useful insights that require them to make decisions for activities such as bidding. The information such as the winning probability based on the current status of the match, the predicted score at the end of next over, the best batsman to join the match

after a wicket, the best bowler to break a partnership, when can another wicket would occur in future, etc are some useful insights about a Cricket match. Subasingha focused on One Day International (ODI) cricket matches and developed the CRIC-Win Analytic Engine, predicting the match outcome. The CRIC-Win analytic engine consisted of two sub-modules, one for predicting overall match outcome based on given pre-match data and the second module for predicting match outcome based on batting partnership both home team and opponent team. The CRIC-Win analytic engine was developed using the Naïve Bayes algorithm [4]. Our research is a continuation of Subasingha's prior work.

Pioneer research of Subasingha uses a Naïve Bayes algorithm to predict the Cricket match outcome. The results of his analysis shows that the results could be improved in terms of accuracy. However, in Subasingha's approach, only a few features were considered for the analysis. Furthermore, only the win or loss prediction was given based on the Country, Toss, First batting team. However, this does not account the current score to predict the win or loss status of the match. It is crucial to consider the current status of the match when predicting the match outcome. This is part of the research gap that we contributed in this research. Therefore, our research goals are encapsulated in the following two research questions.

- RQ1: How applicable the boosting algorithms to predict the winning probability based on the current status of the match?
- RQ2: How applicable are the boosting algorithms to predict the score at the end of next over?

We use a scraped dataset from espncricinfo [5] for this research. This dataset contained information about cricket matches such as: date, gender, winner, overs, player_of_match, team1, team2, toss_winner, venue, etc.

The rest of the paper is organized as follows. Section 2 explains the background information related to data mining techniques in brief. Section 3 describes the related work. The approach to the research questions is illustrated in Section 4. Section 5 discuses about the XGBoost algorithm. The implementation, results and evaluation for two research questions could be found in sections 6 and 7. Section 8 discuss the novelty of our research compared to prior work. Section 9 discusses the future directions and conclusions of the research.

2. BACKGROUND

We started by understanding the potentials of the existing data mining techniques that we believed useful in our research. This section presents the background information we documented that is related to our study.

2.1 SUPERVISED LEARNING

Since we are more focussed on predictions, our research required supervised learning, i.e., the method of training a model in the presence of both input and desired output labels. The two major approaches of supervised learning are classification and regression. In our research, labels are readily available for the two research questions that we defined; for the first research question: to predict the winning probability of a Cricket match, we planned to use the probability of the classification outcome, i.e., whether a Cricket match is about to win or lose by a particular Cricket team based on the current status of a cricket match; for the second research question, to predict the score at the end of next over, the labels of the score at the end of the next score is available for training. The aforementioned facts are the reasons for our approach being focussed on supervised learning.

2.2. CLASSIFICATION

In a supervised learning problem, if the required outcome label of a model is categorical and non-numerical, it is called a classification. The available data mining algorithms to model classification problems include logistic regression classifier, Naïve Bayes classifier, k-nearest neighbor classifier, support vector machines, decision trees, boosted trees, random forest, and artificial neural networks [16]. In this research, we will be using eXtreme Gradient Boosting classification (XGBoost), which we will be discussing in Section 6 to model the winning team of a Cricket match using the current status of a cricket match as the set of input features.

2.3 REGRESSION

In a supervised learning problem, if the expected output of a problem is known and is numerical, it is called a regression. Linear and polynomial regression, artificial neural networks, regression trees and random forests are some of the popular regression analysis methods in the

literature [17]. In our research, we will be using XGBoost regression for predicting the score at the end of next over. We will be discussing this in detail in Section 7.

2.4 Ensemble Learning

Ensemble learning algorithms are different from traditional data mining algorithms due to the architectural advancement; for example, the structure of an ensemble learning classification model composed of several weak classifiers that are combined in a way that they provide a collective result which is significantly higher than the individual weak classifiers. Furthermore, the rationale behind this advancement is that typical data mining algorithms output only a single hypothesis. As a result, traditional data mining algorithms suffer from searching through a space of hypothesis to find the best-approximated hypothesis which suits the dataset [18]. In contrast, the ensemble learning algorithms create a set of weak hypotheses that are capable of voting in some fashion to predict the label of new data. The boosting algorithms are examples for ensemble learning algorithms. We compare the results of our study with traditional data mining algorithms and discuss the importance of using ensemble learning in cases where one classifier alone cannot produce a result with high accuracy.

2.5 Boosting Algorithms

The word boosting denotes a means of building robust data mining in terms of prediction or classification capabilities [19]. A learning algorithm is converted to a base class of models with weak predictive capabilities in a boosting algorithm. Moreover, decision trees are used as the base class in many applications. This base class will form an ensemble model with a more resilient predictive capability. For example, a linear combination of weak predictive models' outputs will be the result in the case of regression [20]. Boosting methods such as Adaptive Boosting (AdaBoost) [21] and Gradient Boosting [22] are used in several practical examples due to their robustness.

2.6 AdaBoost Algorithm

The AdaBoost algorithm proposed by Freund and Schapire was the first practically feasible boosting algorithm, and is widely used in a variety of applications [23]. AdaBoost fits a sequence of weak learners to training examples (based on different weights). The training process begins by predicting the original data set and gives equivalent weight to every training example. If the prediction is incorrect using the first learner, then it gives higher weight to training examples which had been predicted incorrectly when creating the next learner in the sequence. Since this is an iterative process, it keeps adding new learners.

2.7 Gradient Boosting Algorithm

In the gradient boosting algorithm, a collection of regressors is used. It trains several models sequentially. Each model in the sequence continues to minimize the Mean Square Error (MSE) loss function (of $y = ax + b + \square$, where $\square$ is the error term) of the entire system by gradient descent optimization [24]. This learning technique fits new models in a sequential manner to provide a particularly more accurate estimation of the target. When base models are decision trees, such an ensemble learning scenario is called a Gradient Tree Boosting.

2.8 Extreme Gradient Boosting (XGBoost) Algorithm

Chen and Guestrin first introduced the XGBoost algorithm by improving upon the base gradient boosted decision trees by system optimization and enhancing the algorithm. This algorithm was designed to improve execution speed and model performance [25]. XGBoost algorithm works well with structured or tabular datasets on classification and regression predictions. Furthermore, this algorithm got a regularized objective for better generalization, and an additive solution for generic objective (cost) function [26].

3. RELATED WORK

Several attempts had been made by other studies related to the prediction of outcomes in various sports. Munir et al. implemented a real-time outcome prediction system for a T20 cricket match [3]. A decision tree approach was used in this research along with multivariate linear regression to evaluate the results. In this research, they divided a match into three segments and provided predictions at the end of each segment. Therefore, their model was not providing predictions for every next over.

Vignesh and Junaed implemented a prediction system that takes historical instantaneous state of a Cricket match, and

forecasts the Cricket match outcome whether it could result in a victory or a loss for a specific team [6]. They used the bagging (ensemble learning) method together with k-nearest neighbor clustering and ridge regression regularization to create a classifier for the Cricket match outcome prediction. The results of their research were as follows. Over 55% of the Cricket matches were having error margins less than or equal to 10 runs, in particular for non-home matches. For home matches, over 55% of the Cricket matches had the error margins less or equal to 20 runs. This research could be improved to further minimize the error margins.

Shah implemented a model that forecast the Cricket match outcome based on every ball played in live matches using Duckworth-Lewis formula [7]. The Duckworth–Lewis method is a mathematical formulae that was designed to calculate the target score for the second batting team in a limited-overs Cricket match which was interrupted by weather or other conditions [8].

Akhtar and Scarf forecasted the probability of Cricket match outcome whether it would result in a victory, draw, or loss at the beginning of each session using a sequence of logistic regression models in which the classification models were multinomial [9]. The limitation of their research could be observed if both teams lose wickets in the 1st day's match. Then the models were incapable of taking the number of wickets down into account for the second batting team. As suggested by Akhtar and Scarf, this limitation could be overcome by increasing the amount of data used for the research.

Singh and Singla came up with two separate models, i.e., a linear regression to predict the final score of the first innings , and a Naïve Bayes classifier to estimate the outcome of the match in the second innings for the ODI Cricket matches. These two predictive models were trained on data based on the past ODI matches [11].

Bailey and Clarke used a different approach, i.e., social network analysis to classify the winning team of a cricket match before the start of the match [10]. They used the number of tweets, positive and negative sentiments, and fan score predictions to classify the winning team using the support vector machine technique. However, this method does not consider the current status of the cricket match to predict the winning team.

In addition to cricket predictions, prior research exists related to the other sports. Iyer and Sharda explored the applicability of neural networks to rate team players and select specific team players for a competition [12]. Luckner et al. compared the forecasting accuracy of soccer prediction markets that are based on historical data related to the success of the soccer teams with a baseline which is a random predictor [13]. Dragan et al. used Naïve Bayes method to predict the match outcomes of basketball games in the National Basketball Association (NBA) league [14]. A statistical method to predict the winners of the international football tournaments including the Euro 2008 football tournament had been studied by Halicioglu [15].

The prior research of Subasingha implemented the CRIC-Win analytic engine as mentioned in Section 1. Subasingha's experiments yielded the highest accuracy using Naïve Bayes, whereas ensemble boosting algorithms yielded poor accuracy [4]. Moreover, they did not take the current status of the Cricket match also into account when developing the prediction models. With that, in this research, we are particularly interested in fitting our research to fill the following gaps.

1. Predicting the winning team based on the current status of a Cricket match.
2. Evaluating the potentials of boosting algorithms for further improvements in predicting Cricket match outcome.

Thus, we defined research questions of this study: RQ1 and RQ2 to address the aforementioned research gap. In addition, we compare our results with traditional data mining techniques.

4. RESEARCH METHOD

In this section, wel discuss the research setup and the approach for addressing the two research questions of this study. The overall methodology of this research is illustrated in Figure 1.

4.1 Data Collection and Preprocessing

We scraped data from espncricinfo [5] and obtained .yml files containing metadata about Cricket matches and details of the bowling records of the two innings for each match. In this research, we selected the Cricket matches between Sri Lanka and India in order to define the

scope of study. These .yml files were merged into a .csv file to create the dataset required to train the predictive models In the dataset we prepared, the associated data dimensions include date, gender, match_type, winner, won_by, overs, player_of_match, team1, team2, toss_winner, toss_decision, umpires, and venue per each cricket match. For each cricket match, every bowling detail is recorded including the data dimensions; ball_number, inning, batsman, non_striker, bowler, runs, extras, extras_reason, total, wicket, fielders, player_out, and over

non-linear method are examples for wrapper methods for feature selection.

Filter based attribute selection methods use statistical techniques to assess the relationship between each input variable and the target variable [28]. These methods are useful in computation time and robust to overfilling [27]. For example, the Pearson's correlation coefficient is a linear method and the Spearman's rank coefficient is a non-linear method and are filter based attribute selection methods.

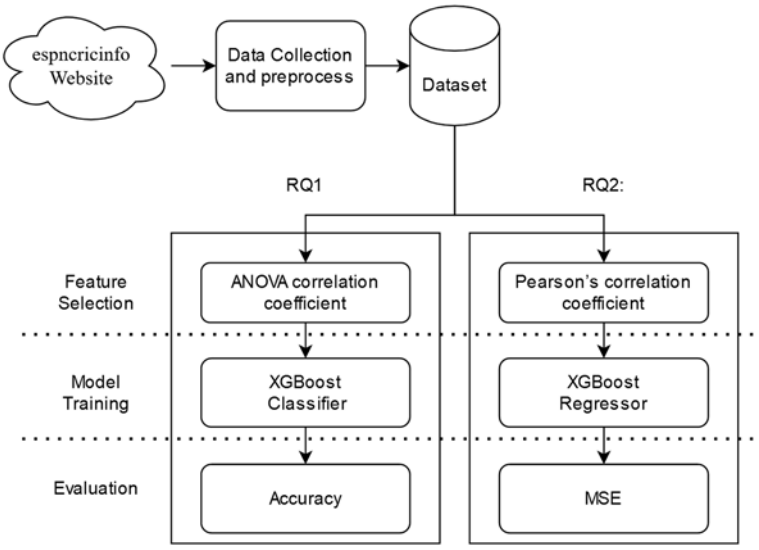

Figure 1: The Overall Research Method

4.2 Attribute Selection

Then, we performed the attribute section (a.k.a. feature selection). The attribute selection methods in literature are twofold based on how they combine the selection algorithm and the model building [27]: i.e., wrapper attribute selection methods and filter attribute selection methods.

Wrapper attribute selection methods evaluate subsets of variables to detect the potential interactions among variables. There is a tendency to overfit if the number of observations/samples is insufficient. This is the disadvantage of wrapper methods [27]. ANOVA (ANalysis Of VAriance) correlation coefficient: a linear method and Kendall's rank coefficient: a

In our research we have a lower number of training observations to each model. We used the ANOVA correlation coefficient to detect the best features to predict the winning probability associated with RQ1 and the Pearson's correlation coefficient to identify the best features to predict the score of the next over associated with RQ2.

4.3 Model Selection

As mentioned, we used the XGBoost algorithm for both RQ1 for the classification and RQ2 for the regression in this research. Moreover, XGBoost is the state-of-the-art algorithm to create supervised learning algorithms with a large number of attributes. The XGBoost algorithm's performance is incredible due to the usage of parallelization, tree pruning, and hardware optimization. In addition, the prior gradient boosting algorithm had been enhanced through both LASSO (L1) and Ridge (L2) regularization

in order to penalize more sophisticated models to prevent overfitting. XGBoost algorithm makes use of the distributed weighted quantile Sketch algorithm in order to find the optimal split points among weighted datasets more effectively [25] [29].

4.4 RQ1: How Applicable Are The Boosting Algorithms To Predict The Winning Probability Based On The Current Status Of The Match?

For RQ1, the variable of our interest is the winning probability given the current status of the match. We used the XGBoost classification algorithm to predict this.

We formulated this problem as a classification between win or loss prediction using the current status of the cricket match. The probability of the predicted outcome will be used to derive the winning probability.

Given the dataset D, which is a n x m matrix where m is the number of features and n is the number of bowling records, we extracted features related to winning probability. Each row represents a bowling record of a particular cricket match. The features representing a bowling record are as follows.

- ball_number
- over_number
- inning
- score
- acc_score
- is_wicket
- acc_wickets
- country

All the aforementioned attributes are numerical except for the country which is the hosting country that the Cricket match had been played. Let the number of countries in the dataset be c. We used one-hot vector encoding to encode the country into a set of dummy variables. Therefore the finalized dataset D' is a (m-1+c) x n matrix. Furthermore, we trained separate models for two teams; India and Sri Lanka.

4.5 RQ2: How Applicable Are The Boosting Algorithms To Predict The Score At The End Of Next Over?

For RQ2, the variable of interest is the predicted score at the end of the next over. We used the XGBoost regression algorithm to predict the score at the end of the next over using the accumulated score of the current over, and of adjacent time lags.

Given the dataset D, which is a n x m matrix where n is the number of features and m is the number of sample bowling records, we extracted features related to accumulated score. And grouped them by match over number and re-calculated the accumulated score for each over. Now each row represents a match over in a particular Cricket match. Let D' be the new dataset and n' be the updated number of samples in dataset D'. The features representing an over is as follows.

1. Over
2. Accumulated score of the orver
3. Accumulated score of 1st time lag
4. Accumulated score of 2nd time lag
5. Accumulated score of 3rd time lag
6. Accumulated score of 4th time lag
7. ...
8. Accumulated score of *l*th time lag

l is a hyperparameter depending on the dataset D' in which (l+2) x n' is the size.

5. XGBOOST ALGORITHM

This section briefly describes the mathematical interpretation of the XGBoost algorithm. Let D be a dataset where m is the number of training examples and i ranges from 1 to m where {xi,yi} represent the ith training example. The estimated label is represented as and is defined in equation (1) [25]

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F} \tag{1}$$

where, F is the space of decision trees which is also known as Classification And Regression Tree (CART). Each f_k corresponds to an independent tree structure.

For boosting tree algorithm, equation (2) is the regularized objective function minimized in which Ω(f) represents the L1 regularization. Besides, l is a differentiable convex loss function.

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{2}$$

For gradient boosting algorithms, equation (3) is the objective function. $\hat{y}_i^{(t)}$ is the estimation of the i[th] instance at the t[th] iteration.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t) \quad (3)$$

Equation (4) shows the second-order approximation which is used to optimize the objective falster.

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t) \quad (4)$$

In equation (4), $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the 1^{st} and 2^{nd} order statistics on the lost function [25].

6. RQ1: IMPLEMENTATION & EVALUATION

As mentioned in the introduction section, the implementation and evaluation of this research is done based on the ODI matches of two Cricket teams, Sri Lanka and India from the years from 2007 to 2016.

For RQ1: the boosting algorithm to predict the winning probability based on the current status of the Cricket match as mentioned in Section 4 was implemented using python, pandas and Sci-kit learn libraries [30][31].

6.1 ATTRIBUTE SELECTION

Let K' be the number of best attributes (features) for the winning probability. For this, the best features were selected using the ANOVA correlation coefficient, in particular, F-test in one-way ANOVA defined in equation (5) [32].

$$f = \frac{variationbetweensamplemean}{variationwithinthesamples} \quad (5)$$

The model's accuracy based on different numbers of features using the best set of features is shown in Figure 1. According to this graph it is clear that accuracy increases rapidly when the K' = 8 and starts decreasing the strength of the slope after K' = 14.

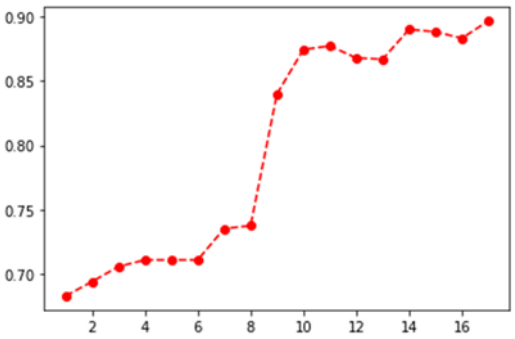

Figure 2: The x-axis is the number of best features. The y-axis is the accuracy of the XGBoost Classifier.

6.2 COST FUNCTION

To calculate the winning team's winning probability, an XGBoost Classifier was used with the objective function as a logistic function. The logistic function is defined in equation (6). The logistic function is graphically illustrated in Fig. 3.

$$\Phi(z) = \frac{1}{1+e^{-z}} \quad (6)$$

6.3 NUMBER OF ESTIMATORS

For this research, we analyzed the model behavior for different numbers of gradient boosted trees. The best number of estimators is 1000 for predicting the winning team of a match. See Figure 4.

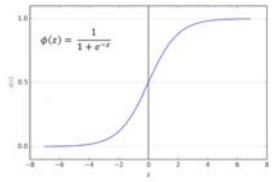

Figure 3: Logistic function

6.4 DEPTH OF DECISION TREES

The length of the longest path from a root to a leaf in a decision tree is known as the depth of that particular decision tree. The depth of the boosted decision trees is also another parameter that is required to be tweaked in order to obtain better prediction results. Figure 5 shows that best accuracy was obtained when the depth of the boosted decision trees was 12. It is also shown

that further increment of the depth does not yield reasonable improvements.

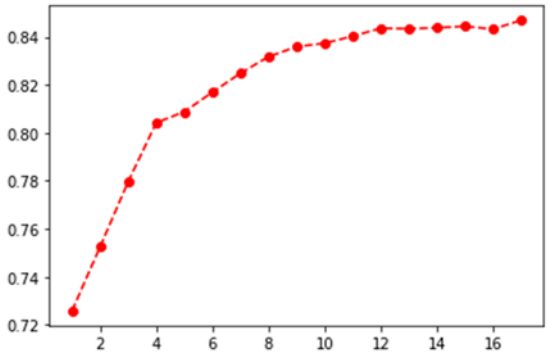

Figure 4: The x-axis is the number of estimators. The y-axis is the accuracy of the XGBoost Classifier.

6.5 ACCURACY

For RQ1, the classification outcome of the winning team is evaluated using the accuracy which is defined as the ratio between the summation of true positives and true negatives to the total number of examples. Furthermore, a training example is a bowling record, which is a point of time in the cricket match. Table 1. shows the confusion matrix yielded for the test data. Accuracy is defined in equation (7) where TP, TN, FP, and FN are represented by true positives, true negatives, false positives, and false negatives respectively.

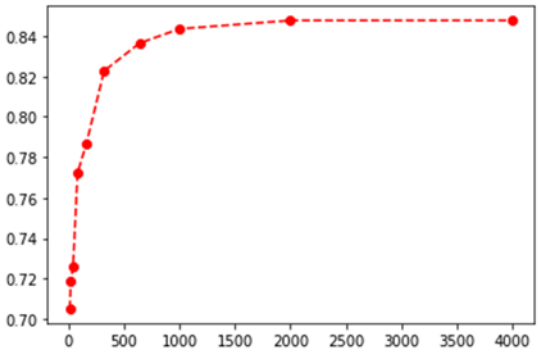

Figure 5: The x-axis is the depth of the decision tree. The y-axis is the Accuracy of the XGBoost Classifier.

$$(7)\ accuracy = \frac{TN+TP}{TN+Fn+TP+FP}$$

Table 1: Confusion matrix

		Actual Values	
		Positive	**Negative**
Predicted	**Positive**	3120(TP)	680 (FN)
Values	**Negative**	158 (FN)	925 (TN)

In our research, the classification accuracy was 0.844 (rounded over to the third decimal point) which also means that 84.4% of the predictions yielded correct.

The XGBoost algorithm resulted in promising outcomes for this research. We conducted a comparison of results of XGBoost Algorithm vs other data mining algorithms for the same dataset and the result for predicting the winning team is in table 2.

Table 2: Comparison of the Accuracy of the Winning probability

Data Mining Algorithm	**Accuracy**
Logistic Regression	0.6830
Nearest Neighbor	0.8180
Support Vector Machines	0.7111
XGBoost	0.8440

The error boundaries of the aforementioned models are shown in Table 3. Let n be the number of test examples. Let $\hat{R}(\hat{h})$and R) be the test and true error respectively. Then, with 95% probability, we have,

$$R) = \hat{R}(\hat{h}) \pm \frac{1.36}{\sqrt{\ }}$$

In our dataset, n = 4883.

Table 3: Error boundaries of winning probability prediction

ML Algorithm	**Upper boundary**	**Lower boundary**
Logistic Regression	0.2975	0.3365
Nearest Neighbor	0.1626	0.2015
Support Vector Machines (SVM)	0.2694	0.3084
XGBoost	0.1365	0.1755

According to the error analysis, XGBoost algorithms outperform the Logistic regression and SVM classifiers. The true error boundaries of the Nearest Neighbor classifier and the XGBoost are overlapping. Therefore, it is impossible to say whether XGBoost truly outperforms the Nearest Neighbor classifier.

7. RQ2: IMPLEMENTATION & EVALUATION

This section describes the implementation and evaluation of the results for RQ2: How applicable are the boosting algorithms to predict the score at the end of next over?

7.1 Attribute Selection

Let K'' be the number of best attributes (features) required to detect the score at the end of next over. For this, we used Pearson's correlation coefficient r which is defined in equation (8) where N is the number of training examples, x represents a training attribute value and y represents labels [33].

$$r=\frac{N\Sigma xy-\Sigma x\Sigma y}{\sqrt{[N\Sigma x^2-(\Sigma x)^2][N\Sigma y^2-(\Sigma y)^2]}} \quad (8)$$

The MSE is evaluated per each possible value for K and it is depicted in Figure 6. Accordingly, the MSE decreases as the K increases. The best value for K'' is 7 as per the graph.

$$SE = \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2 \quad (9)$$

7.2 Cost Function

In order to compute the accumulated score at the end of next over, an XGBoost Regressor was used for training the predictive model for computing the accumulated score at the end of each over. The objective function is the Squared Error (SE) function defined in equation (9) where n is the number of examples.

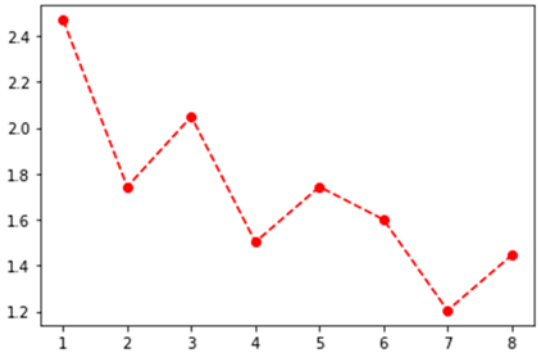

Figure 6: The x-axis is the number of best features. The y-axis is the MSE of the XGBoost Regressor.

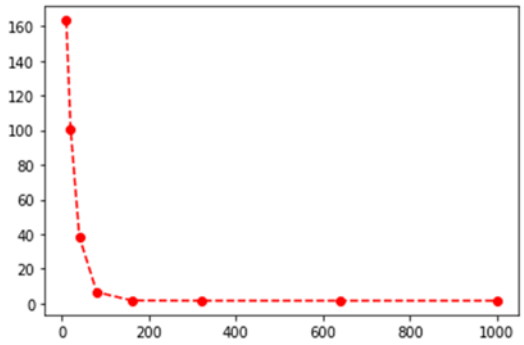

Figure 7: The x-axis is the number of estimators. The y-axis is the MSE of the XGBoost Regressor

7.3 Number of Estimators

The number of estimators is 160 to predict the winning team of a match. See Figure 7.

7.4 Depth of Decision Trees

According to the experiments the least MSE is found when the depth of the boosted trees is 3. The MSE remains nearly the same when the depth keeps on increasing further as in Figure 8.

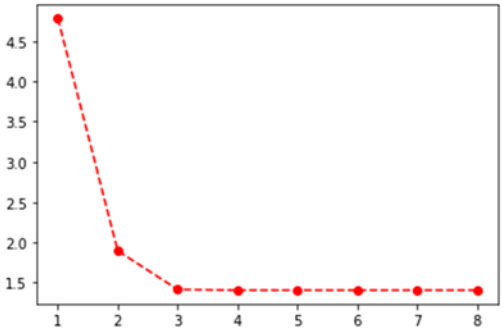

Figure 8: The x-axis is the depth of the boosted tree. The y-axis is the MSE of the XGBoost Regressor

7.5 Mean Squared Error (MSE)

Mean Squared Error (MSE) is a statistical measure that is calculated as the average of the squares of the error terms, i.e., the average squared difference between the estimated values and the actual values and is defined in equation (10). Let n be the number of samples in the test dataset. Y_i is the actual label and $\hat{Y_i}$ is the predicted value for the i^{th} test example. The formula is defined in (10).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (10)$$

Figure 9 shows the expected vs predicted scores at the end of next over. This was evaluated using the MSE of the test dataset.

The MSE of the test dataset is 1.41 (rounded over to the second decimal point) which means there is an error margin of +/- 1.41 score for each prediction.

A comparison of results yielded from XGBoost Regression vs other data mining algorithms for the same dataset is shown in Table 4.

This shows the robustness in extreme gradient boosting algorithms over the other data mining approaches for both RQ1 and RQ2 of this research.

Table 4: Predict accumulated score at the end of next over

Data Mining Algorithm	MSE
Linear Regression	2.666094960010279
Nearest Neighbor	3.1432467291003423
XGBoost	1.4142213882407835

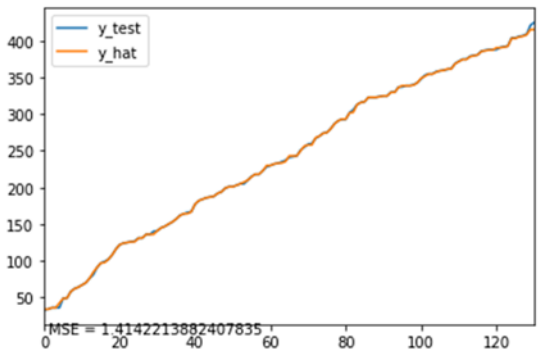

Figure 9: The expected vs predicted scores at the end of next over

8. COMPARISON WITH PRIOR WORK

Compared to prior work of Subasingha et al, our approach for predicting the winning probability of a team uses the current status of a Cricket match into consideration when calculating the score at the end of the next over. In addition, we used the current status of the Cricket match as well as the other factors such as country into the consideration when predicting the winning probability of a team. Our research tool a novel approach applying XGBoost ensemble learning algorithms to predicting the outcomes of a Cricket match to increase the accuracy.

.

9. CONCLUSION AND FUTURE WORK

The winning probability prediction/winning team classification and the predicted score at the end of next over are two main aspects of a cricket match based on the current status of the Cricket match were studied in this research. This research is a continuation of prior work of Subasingha et al. We analyzed the potentials of boosting algorithms to predict the outcomes of cricket matches. As per the evaluation, this research shows the improved boosting algorithms perform better than the prior research. Furthermore, extreme gradient boosting algorithms outperform the logistic regression and SVM, which are traditional data mining algorithms in the classification problem of predicting the winning team. The best accuracy of our winning team was 84.4%. For predicting the score at the end of the next over of a cricket match, the XGBoost regressor is outperforming the linear regression and KNN regression. The MSE of the predicted score at the end of next over was 1.41. The predictive models explored in this research could be directly integrated to the real-time dashboards which would help both internal and external stakeholders of an ODI cricket match for decision making.

This research is based on the data from two cricket teams. Therefore, this is not a generalized approach. Finding a generalized approach for predicting the outcome of a cricket match and predicting the score at the end of the next over could be a more challenging problem. Furthermore, the team players are not considered in this research, and only the bowling records were considered. The bowling records could have biases based on who the team players are participating in a particular cricket match. The aforementioned limitations should be addressed in future research. Additionally, we hope to address the remaining three research questions in future research, i.e., the research questions of "who is the best batsman to join the game after a wicket?" and "who is the best bowler to break the partnership?" could be addressed using associative rule mining techniques and the final

research question of "when can another wicket occur in future?" is a classification problem yet to be solved.

REFERENCES

[1] Show RS. Top 10 Most Popular Sports in The World [Updated 2020] [Internet]. sportsshow.net. 2020 [cited 2020Apr6]. Available from: https://sportsshow.net/top-10-most-popular-sports-in-the-world/

[2] V. Veppur Sankaranarayanan, "Towards a time-lapse prediction system for cricket matches," PhD Thesis, University of British Columbia, 2014.

[3] Munir F, Hasan M, Ahmed S. Predicting a T20 cricket match result while the match is in progress (Doctoral dissertation, Brac University).

[4] S.A.D.P Subasingha, S. C. Premaratne, K. L. Jayaratne and P. Sellappan, Novel Approach for Cricket Match Outcome Prediction Using Data Mining Techniques, International Journal of Engineering and Advanced Technology (IJEAT), ISSN 2249-8958.

[5] ESPN [Internet]. Cricinfo. [cited 2020Apr6]. Available from: https://stats.espncricinfo.com/ci/engine/stats/index.html

[6] V. V. Sankaranarayanan, J. Sattar, and L. V. Lakshmanan, "Auto-play: A data mining approach to ODI cricket simulation and prediction," in Proceedings of the 2014 SIAM International Conference on Data Mining, 2014, pp. 1064–1072.

[7] Shah P. Predicting Outcome of Live Cricket Match Using Duckworth-Lewis Par Score. International Journal of Systems Science and Applied Mathematics. 2017 Sep 28;2(5):83.

[8] Duckworth–Lewis–Stern method, Wikipedia [Available at: https://en.wikipedia.org/wiki/Duckworth%E2%80%93Lewis%E2%80%93Stern_method]

[9] Akhtar S, Scarf P. Forecasting test cricket match outcomes in play. International Journal of Forecasting. 2012 Jul 1;28(3):632-43.

[10] Bailey M, Clarke SR. Predicting the match outcome in one day international cricket matches, while the game is in progress. Journal of sports science & medicine. 2006 Dec;5(4):480.

[11] Singh T, Singla V, Bhatia P. Score and winning prediction in Cricket through data mining. In2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI) 2015 Oct 8 (pp. 60-66). IEEE.

[12] Iyer SR, Sharda R. Prediction of athletes performance using neural networks: An application in cricket team selection. Expert Systems with Applications. 2009 Apr 1;36(3):5510-22.

[13] Luckner S, Schröder J, Slamka C. On the forecast accuracy of sports prediction markets. InNegotiation, auctions, and market engineering 2008 (pp. 227-234). Springer, Berlin, Heidelberg.

[14] Miljković D, Gajić L, Kovačević A, Konjović Z. The use of data mining for basketball matches outcomes prediction. InIEEE 8th International Symposium on Intelligent Systems and Informatics 2010 Sep 10 (pp. 309-312). IEEE.

[15] Halicioglu F. Research on the Prediction of the likely Winners of the Euro 2008 Football Tournament. Journal of Quantitative Analysis in Sports. 2009 Jul 19;5(3).

[16] Sidana M. Types of classification algorithms in Machine Learning [Internet]. Medium. Medium; 2019 [cited 2020Apr6]. Available from: https://medium.com/@Mandysidana/machine-learning-types-of-classification-9497bd4f2e14

[17] Seif G. Selecting the best Machine Learning algorithm for your regression problem [Internet]. Medium. Towards Data Science; 2019 [cited 2020Apr6]. Available from: https://towardsdatascience.com/selecting-the-best-machine-learning-algorithm-for-your-regression-problem-20c330bad4ef

[18] Dietterich TG. Ensemble learning. The handbook of brain theory and neural networks. 2002 Mar;2:110-25. [Available at: https://courses.cs.washington.edu/courses/cse446/12wi/tgd-ensembles.pdf]

[19] Robert E. Schapire and Yoav Freund.Boosting: Foundations and Algorithms.MITPress, 2012.

[20] Beygelzimer A, Hazan E, Kale S, Luo H. Online gradient boosting. InAdvances in neural information processing systems 2015 (pp. 2458-2466).

[21] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-linelearning and an application to boosting.JCSS, 55(1):119–139, August 1997.

[22] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine.Annals of Statistics, 29(5), October 2001

[23] Schapire RE. Explaining adaboost. InEmpirical inference 2013 (pp. 37-52). Springer, Berlin, Heidelberg.

[24] Gradient Boosting Machines [Internet]. Gradient Boosting Machines · UC Business Analytics R Programming Guide. [cited 2020Apr6]. Available from: http://uc-r.github.io/gbm_regression

[25] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. InProceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining 2016 Aug 13 (pp. 785-

794).

[26] Brownlee J. A Gentle Introduction to XGBoost for Applied Machine Learning [Internet]. Machine Learning Mastery. 2019 [cited 2020Apr6]. Available from: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

[27] Feature selection [Available at: https://en.wikipedia.org/wiki/Feature_selection]

[28] Feature selection with real and categorical data [Available at: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/]

[29] Morde V. XGBoost Algorithm: Long May She Reign! [Internet]. Medium. Towards Data Science; 2019 [cited 2020Apr6]. Available from: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

[30] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[31] API design for machine learning software: experiences from the scikit-learn project, Buitinck et al., 2013.

[32] Frost J, Ronnel, Akanksha, Lyla, B G, K. SS, et al. How F-tests work in Analysis of Variance (ANOVA) [Internet]. Statistics By Jim. 2020 [cited 2020Apr6]. Available from: https://statisticsbyjim.com/anova/f-tests-anova/

[33] Study.com. Study.com; [cited 2020Apr6]. Available from: https://study.com/academy/lesson/pearson-correlation-coefficient-formula-example-significance.html