Cricket Score Prediction in One Day Internationals Using Machine Learning Approach

Kavan Vadodariya
Computer Engineering
Marwadi University
Rajkot, Gujarat, India
kavan.vadodariya120040@marwadiuniversity.ac.in

Jenish Ambaliya
Computer Engineering
Marwadi University
Rajkot, Gujarat, India
jenish.ambaliya118291@marwadiuniversity.ac.in

Het Dhingani
Computer Engineering
Marwadi University
Rajkot, Gujarat, India
het.dhingani120384@marwadiuniversity.ac.in

Urvi Dhamecha
Computer Engineering
Marwadi University
Rajkot, Gujarat, India
urvi.dhamecha@marwadieducation.edu.in

Abstract- Cricket, especially in its One Day International (ODI) format, produces substantial data, facilitating the use of predictive analytics for score estimation and match outcome prediction. Precise score projection is a big challenge owing to vary match factors, that includes surface characteristics, wicket losses, and momentum changes. This study is for evaluating the efficacy of three machine learning algorithms like Linear Regression, Random Forest, and XGBoost which is used in forecasting first-innings scores in ODIs with historical ball-by-ball data. The dataset, obtained from cricket repositories, covers the period from 2003 to 2017 and comprises essential features including completed overs, wickets lost, venue, and runs scored in the final five overs. Subsequent to preprocessing and feature encoding, models were trained on 70% of the dataset and evaluated on 30%, utilizing R², RMSE, MAE, and MSE metrics for assessment. The results demonstrate that Linear Regression was ineffective (R² = 0.6191), Random Forest exhibited significant enhancement (R² = 0.8230), and XGBoost surpassed both (R² = 0.8615), illustrating its proficiency in managing non-linear connections and intricate feature interactions. These result highlights how efficiently the ensemble methods and boosting methods works in real-time, give sensitive predictions in sports analytics and also offering key insights to teams, stakeholders and broadcasters.

Keywords— Cricket prediction, One Day International (ODI), Machine learning, Real-time score prediction, Linear Regression, Random Forest, XGBoost, Sports analytics.

I. INTRODUCTION

Cricket is the most popular sport in India and is played all around the world. Cricket, in its One Day International format has gained popularity among fans, with fans closely associating themselves with the scores of each game and performances of each player. Over the time it has become a data-rich sport where analysis plays a vital role in strategy making. Nowadays, most teams are utilizing predictive analysis to get the best possible denouement. Various aspects, including the final scores can now be predicted by certain methods. By using machine learning techniques, we can use predictive analysis to make deductions on player's performance, match outcomes or other strenuous inferences like momentum shifts. One Day Internationals (ODIs) is one of the ideal format for predictions and analysis. By getting access to historical datasets, it has become easier to find meaningful patterns and trends that can help in predicting match outcomes as well as total score.

Predicting the final score is still a challenging task, especially in ODIs where each team plays 50 overs, predicting the final score solely based on run rate is quite challenging. These approaches lack to adapt to match conditions like wickets falling, pitch behavior, etc. As a result, these predictions are mostly inaccurate and are not according to the match context. Therefore, there is a need for a solution that uses historical data and match specific variables to provide accurate score according to real-time conditions. The principal objective of this paper is to compare various machine learning algorithms that can be used in order to formulate score predictions with critical features associated with ODI matches.

The algorithms that are predominantly being used in today's cricket world include Lasso Regression, Random Forest Regression, Gradient Boosting (XGBoost), ARIMA / LSTM (Long Short-Term Memory), KNNs and RNNs. Of these, the methods that are widely being used are Linear Regression, XGBoost and Random Forest method. These models are trained using various match features such as overs completed, wickets lost, venue, wickets lost in last five overs, runs scored in last five overs and more.[1] Each model has been provided with the accurate past match details, helping the system to learn the patterns of variables of overs, runs, wickets and ball-by-ball update in real time, etc. By comparing the performances of these algorithms, it determines the most effective model for accurate and context-aware score prediction. These models are not only used in the field of sports analytics, but also demonstrate how machine learning models can be utilized for real time quick decisional environment as diverse as ODI cricket match. The comparative implementation and use of the models mentioned above further highlights the strengths and limitations, providing exceptional insights for future models to be developed. "Runs prediction and win prediction in cricket is a useful tool for teams, broadcasters, and other stakeholders in the game to make educated decisions and increase their engagement with the sport. This work used machine learning models to predict the runs scored by a team in the first innings and to predict the winner of a cricket match on a real-time basis."[2]

The proposed approach contributes to the field of sports analytics by rule-based prediction systems and data-driven frameworks. The study chiefly centers on computing the impact of key predictors amalgamating it into the system. Also, this research highlights the use of machine learning models in delivering the real-time predictions which are accurate and meaningful and can support in strategy making in modern cricket.

II. LITERATURE REVIEW

This review highlights the use of algorithms for analysis, results achieved from the models, key-findings and the limitations faced.

Some recent researches have boosted these findings, Sherilyn et al. (2023) implemented linear regression and it achieved low accuracy which shows that regression cannot perform with complex datasets and gradient boosting or ensemble methods can perform much better and provide accurate predictions.[1]

Sudhini et al. (2025) implemented models like Random Forest and XGBoost to predict the scores in different formats like ODIs and T20s. XGBoost achieved better accuracy than Random Forest. But the research lacked the involvement of some features like pitch conditions and momentum shift.[3]

Similarly, Chandru and Prasath (2024) implemented models like weighted K-means for Normalization. With the help of clustering, it got good predictions and generated good accuracy by using normalization.[4]

Some comparative studies have also got attention. Thinakaran (2023) used Random Forest and XGBoost for comparative analysis. Surprisingly in this analysis Random Forest achieved higher accuracy than XGBoost, showing the power of ensemble learning models .[5]

Lamsal and Choudhary (2018) implemented algorithms like Logistic Regression and Naïve Bayes and through that they achieved good predictions in IPL Their models achieved an accuracy of 71%.[6]

Ahmad et al. (2024) implemented algorithms like SVM to achieve good accuracy and a strong performance in ODIs. This analysis achieved an accuracy of 85.7%.[7]

Param et al (2024) Implemented multiple algorithms like Random Forest, Support Vector Machine (SVM) and Naïve Bayes, along with player performance to achieve an accuracy of 89.92%.[8]

Similarly, Mozar et al. (2022) implemented algorithms like Ensemble Learning and multivariate Regression and it showed good efficiency in predicting win probability. Hence it demonstrated good and stable results and deployment through good GUI systems. [9]

Table1: Literature Review Table

Author(s) & year	Methodology used	Key Findings	Limitations
Kevin et al. (2023)	Linear Regression	Achieved satisfying accuracy.	Limited to just regression and small dataset
Satwani et al. (2024)	Random Forest, Gradient Boosting	Provided real-time predictions like run-rate, overs, etc.	Performance drops if there is a change in match condition.
Sudhini et al. (2025)	Random Forest, XGBoost	Achieved high accuracy of 93%.	Didn't mention some features like pitch conditions
Chandru and Prasath (2024)	Weighted K-means for Normalization	Improved prediction by using normalization	Used on Normalization which is not sufficient
Thinakaran (2023)	Random Forest, XGBoost	Random Forest achieved higher accuracy compared XGBoost	Focused only on T20s
Lamsal and Choudhray (2018)	Logistic Regression, Naïve Bayes	Achieved good predictions in IPL	Lacked models like gradient boosting
Ahmad et al. (2024)	SVM	Achieved high accuracy and a strong performance	Complexity in models takes so much computation time
Param et al. (2024)	Random Forest, SVM, Naïve Bayes	Random Forest achieved a good accuracy of 89%	Lacked features like pitch conditions
Mozar et al. (2022)	Random Forest, Lasso Regression	Showed good efficiency in predicting win probability	Not enough for real-time ingestion

Overall, this literature review shows the vital role machine learning plays in sports analytics. While recent advancements in machine learning models have significantly improved the prediction. Also, it has made advancements in ensemble learning and gradient boosting to make good and advanced predictions. For making more advanced models deep learning and real-time streaming data are used.

III. ALGORITHMS

A. Linear Regression

Linear Regression is the simple and easy to implement algorithm in machine learning and also one of the most used algorithm. It predicts the target by working on linear relationships. It finds the best straight fit line between independent variable and dependent variable. It can serve as a strong baseline model and provide fast predictions.

In cricket score prediction, Linear Regression uses variables such as overs completed, runs scored, wickets lost, venue, runs scored in last five overs and wickets lost in last five overs to predict the final score.[1] Because of its simplicity it is easy to implement. Through this analyst, broadcasters and team coach can quickly understand the factors and easily get the estimated final score.

However, Linear Regression assumes that relation between its target and features is purely linear, which might not capture the complexity in cricket matches, where there can be sudden batting collapse or changes in pitch behavior. Overall, the efficiency, low cost and simplicity make Linear Regression a valuable starting point in score prediction.

B. Random Forest

Random Forest is an ensemble learning algorithm, it constructs many decision trees during training time and outputs the average prediction (for regression) from all trees or majority vote (pick the class which is most trees occurs) for classification. This is achieved using two separate randomness processes: first, each tree is trained on a random selection of features and second, it is trained on a bootstrapped subsample of the data. This technique helps in controlling overfitting and improves generalization.

Random Forest can take into consideration several match-level variables at the same time like number of overs completed, wickets lost, rate of runs scored in last 5 overs and various venue-specific statistics for predicting cricket scores. Each tree may capture distinct things about the game, and when combined they result in a strong, stable prediction.

Because Random Forest can learn in live match conditions, it was the methodology of choice to be used for real-time, contextual score estimates. But it can suffer slowdowns due to more complex models, and computational cost may become substantial on very large datasets.

C. XGBoost

XGBoost is a powerful machine learning algorithm on the Gradient boosting framework. It sequentially trains trees, and each new tree traces the errors made by previous ones. Its speed, scalability and accuracy come to a large extent from some of the advanced regularization techniques used in XGBoost.

For example, if trying to predict cricket scores XGBoost would be able to capture complex non-linear relationships between features (e.g. current run-rate, remaining overs, wickets fallen, momentum shift, condition of pitch) efficiently. Unlike Linear Regression, it does not assume a straight-line relationship, allowing it to adapt to sudden momentum changes in matches. The algorithm minimises a loss function over multiple trees created following the residual errors left by the prior trees. In addition, XGBoost is one of the best algorithm because it can handle large and complex datasets and able to detect tiny patterns while adjusting to match-by-match conditions. XGBoost is also flexible in handling many data formats as well as missing data. This makes XGBoost one of the best performing algorithm in sports analytics.

IV. DATASET DESCRIPTION

A. Source of Data:

The dataset is collected from Kaggle. And it has ball-by-ball records for One Day Internationals (ODIs) matches.

B. Dataset Size and Time Span:

The Dataset consists of **ball-by-ball records** from approximately **2003 to 2017**. It also includes many features, such as batting, bowling, match details, etc.

1) No. of Rows: 350,898
2) No. of Columns: 15
3) Time Span: 3 Jan 2003 to 10 Jul 2017

C. Key Features

Table2: Dataset key attributes

Feature	Description	Example
Mid	Unique match ID	5
Date	Date of match	2005-03-08
Venue	Stadium	Lords
Bat_team	Current batting team	Ireland
Bowl_team	Current bowling team	England
Batsman	Striker facing the ball	Sterling
Bowler	Bowler delivering the ball	Henderson
Runs	Runs scored so far	5
Wickets	Wickets scored so far	0
Overs	Current over (with balls)	2.1
Runs_last_5	Runs scored in last 5 overs	35
Wickets_last_5	Wickets lost in last 5 overs	2
Striker	Runs scored by striker	25
Non-striker	Runs scored by non-striker	12
Total	Total team score	315

V. METHODOLOGY

The methodology is divided into six different steps: Data Preparation, Feature Encoding, Train-Test Split, Model Development, Model Evaluation, and Comparative Analysis. Each step plays an important role in building a good framework for predicting ODIs scores. While including mathematical formulas showed clarity on how models were implemented.

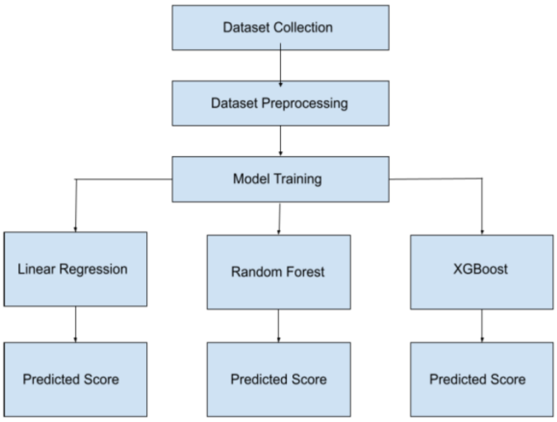

Fig1: Data flow diagram

A. Data Preparation

The dataset is important for this study, consisting of ODI match records with features such as runs scored, wickets lost, overs completed, venue, batting team, bowling team, and statistics from the last five overs. The target variable is the final total score of the innings.

Formally, the dataset can be represented as:

$$D= \{(X1, y1), (X2, y2), \ldots, (Xn, yn)\}$$

Where, Xi represents feature vector and Yi represents corresponding total score.

Data preprocessing involved deleting duplicate rows or empty rows and removing columns of no use in evaluation. Many features influencing the score were used, such as runs, wickets, run rate, etc. This ensures a clean and structured dataset. By refining the data, the methodology created a reliable foundation for accurate model training and testing.

B. Feature Encoding

Numerical features like Runs scored, Wickets lost (0–10 wickets), and overs are linear in nature, which can be directly fed to machine learning model but Categorical Features such as Venue, Batting Team and Bowling Team needs some sort of transformation (one-hot encoding). They are pivotal because context is provided as a result of these attributes; one can be assured that certain venues do have batting paradises and some teams tend to play better (or worse) based on opposition.

C. Train-Test Split

Cross-Validation is one of the most important steps in any predictive modeling task where we train and build our model on one subset of the dataset and test it on yet another independent subset. In a general sense, this is called overfitting where model works very well in training data but fails miserably on unseen cases. The dataset was randomly partitioned into training (70%) and testing (30%) sets in this study.

Formally, the dataset is split as:

$$D=Dtrain \cup Dtest, Dtrain \cap Dtest= \emptyset$$

with proportions:

$$|Dtrain|=0.7n, |Dtest|=0.3n$$

where n represents the total number of match instances.

Here, Dtrain is used to learn patterns and train the model. While, Dtest evaluates how well the models works on unseen matches. The choice of having 70:30 is because it provides sufficiently large training set to capture diverse match scenario while keeping enough test data to reliably assess model performance. This split ensures that the model works perfectly and through testing it can give accurate results.

D. Model Development

The core of this methodology lies training three machine learning models- **Linear Regression**, **Random Forest** and **XGBoost** on the prepared dataset:

1) Linear Regression

Linear Regression works on linear relationship between independent variable and the target:

$$\hat{y}=\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p$$

where ŷ is the predicted score, xi are input features (runs, wickets, etc.), and βi are coefficients estimated using least squares. The model minimizes the **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} \quad \backslash(yi-\hat{y}i\backslash)2$$

2) Random Forest

Random Forest builds an ensemble of decision trees using different samples, where each tree makes a prediction and average is taken for the output:

$$\hat{y} = \sum_{j=1}^{k} \text{Tj } (X)$$

Where, Tj(x) is prediction for jth tree.

3) XGBoost

XGBoost adds multiple trees in a sequence to correct previous errors. At stage t:

$$\hat{Y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i)$$

Where, ft(xi) is new tree and n is learning rate.

By training these models with same dataset, it ensures a fair comparison among all models and showing the predictive modelling in ODIs and predicting the score in real-time.

E. Model Evaluation

Model Evaluation is very important to predict the results accurately and getting good efficiency and reliability from models. In this study we used four metrics: **R^2 score, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Squared Error (MSE)**

R^2 Score:

$$R^2\ 1 - \frac{\sum_{i=1}^{n} \backslash(yi\text{-}\hat{y}i\backslash)2}{\sum_{i=1}^{n} \backslash(yi\text{-}\bar{y}i\backslash)2}$$

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \backslash(yi\text{-}\hat{y}i\backslash)2}$$

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |yi\text{-}\hat{y}i|$$

Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \backslash(yi\text{-}\hat{y}i\backslash)2$$

Here R^2 captures overall explanatory power and RMSE provides sense of error margin.

R^2 ranges from 0 to 1, where 1 means all the variations are perfectly explained and 0 means nothing. It shows the accuracy of the model which 0 to 1 is 0% to 100%.

RMSE is calculated by taking the average of the errors between predicted value and actual value. A perfect model will have value closer to zero. A lower RMSE shows how fit the model is.

MAE is calculated by taking the difference between actual values and predicted values. It shows the average absolute error. Lower MAE means high and better accuracy. It indicates the distance between predicted and actual values.

MSE is calculated by taking the average of squared differences between predicted value and the actual values for each data point. The lower MSE shows how better and fit the model is. A perfect model has value closer to zero.

VI. RESULTS

A. Model Performance Evaluation

Among all the models, Linear Regression was the poorest due to its inability to work with complex datasets and it failed to capture non-linear patterns, hence it resulted in low accuracy and high error values. Random Forest performed better than Linear Regression and showed improvement in accuracy and also lowered the error values. XGBoost was best among all algorithms, achieving highest accuracy as well as lower error values. Showing how well gradient boosting works with real-world complex data.

Table2 Model Evaluation

Model	R^2 Score	RMSE	MAE	MSE
Linear Regression	0.6192	38.52	29.12	1479.16
Random Forest	0.8231	26.27	17.25	687.6
XGBoost	0.8616	23.23	16.55	540.36

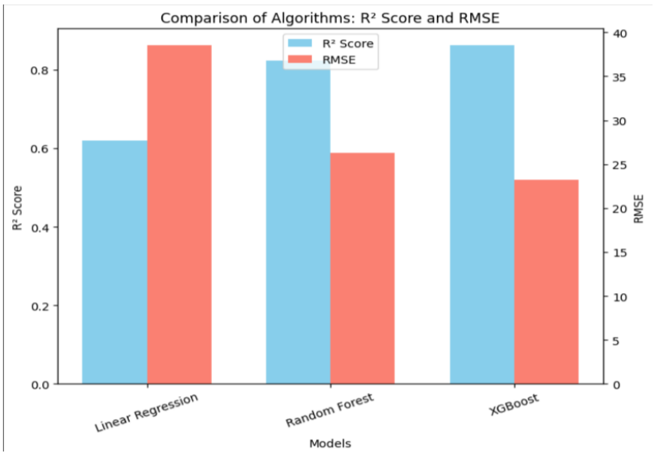

Fig2: Comparison of Algorithms- R^2 score and RMSE

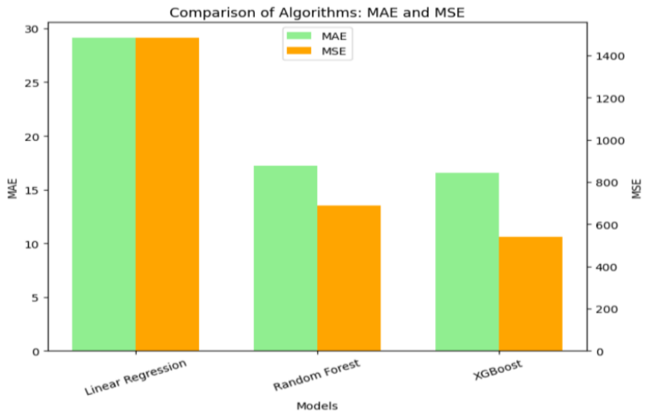

Fig3: Comparison of Algorithms- MAE and MSE

B. Analysis of Results

1) Linear Regression:

Linear Regression generated a poor accuracy of just 61% and also had high error margins. It also performed poor compared to other algorithms, showing its limitations in noticing the complex patterns present in the data. It shows the inadaptability of Linear Regression while performing with real-world complex data and it showed overreliance on finding linear patterns.

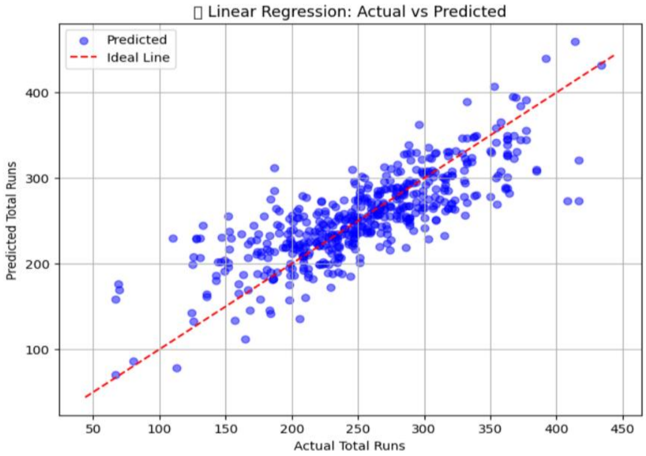

Fig4: Graphical analysis of Linear Regression (Actual vs Predicted Score)

2) Random Forest:

Random Forest generated a good accuracy of 82% showing a good performance than Linear Regression. It showed subsequently lower error margins which enabled the effectiveness in non-linearities within the dataset. These highlights the Random Forest's adaptability to complex datasets while generating good accuracies compared to classic regression techniques.

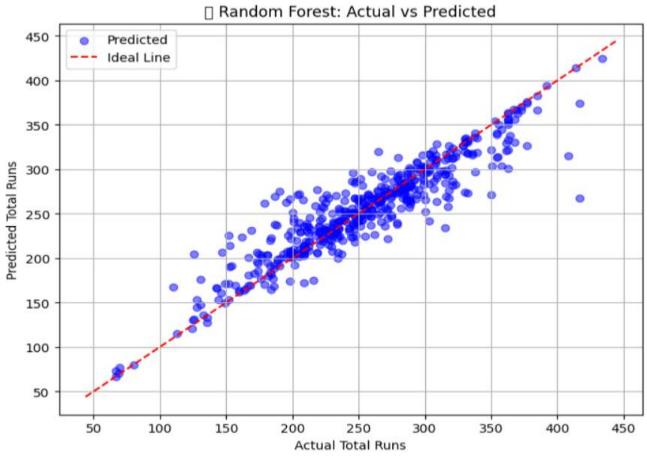

Fig5: Graphical analysis of Random Forest (Actual vs Predicted Score)

3) XGBoost:

XGBoost gave the most accurate predictions among all other models, as it achieved the highest accuracy and low error margins. It showed the power of gradient boosting algorithms while handling large and complex datasets. These comparison analysis showed the power and efficiency of XGBoost and its capability to work with complex real-world data for predictive modelling.

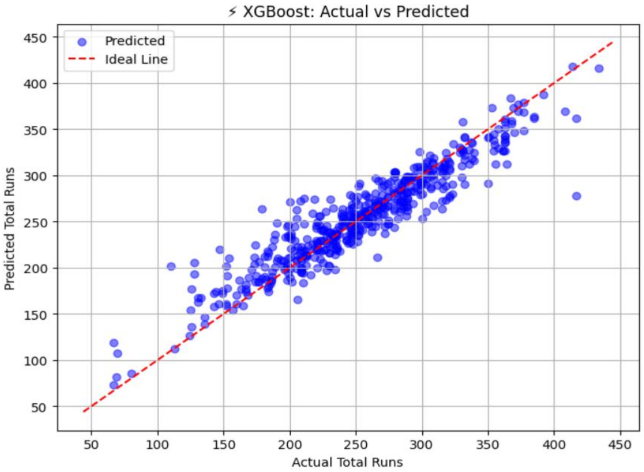

Fig6: Graphical analysis of XGBoost (Actual vs Predicted Score)

VII. CONCLUSION

This comparison analysis showed the importance of selecting best models for predictive analysis when addressing large datasets with complex and non-linear relationships. Linear Regression widely regarded for its simplicity, was found to be inadequate int this context, as it failed to capture deeper patterns in data and it also failed when there was sudden change in momentum or sudden batting collapse. Its limitation reaffirms the challenges of relying solely on regression techniques for real world applications where non linearity is inherent.

In contrast, Random Forest showed notable improvement compared to Linear Regression in predictive accuracy, reflecting the strength of ensemble learning by using the aggregation of multiple decision trees. Its capacity to generalize than linear model highlights its suitability for more complex datasets. Despite forming performing this well Random Forest was outperformed by a more advanced boosting technique.

XGBoost emerged as the most accurate and effective model in this comparison, achieving superior predictive accuracy and also maintaining lower error margins. The results showed why XGBoost is best performing algorithm than others while maintaining its reputation as a state-of-the-art algorithm for predictive modeling.

From this evaluation, it is evident that while simpler models provide interpretability, advanced ensemble approaches such as XGBoost offer significantly greater reliability and accuracy. For real-world predictive tasks, particularly those involving complex and dynamic data, gradient boosting frameworks stand out as the most practical and impactful choice.

VIII. REFERENCES

[1] S. Kevin, B. Yadav, A. K. Pandey, and G. Rajbhar, “T20 Cricket Score Prediction Using Machine Learning,” *IRE Journals*, vol. 7, no. 6, pp. 49–53, Dec. 2023.

[2] A. Satwani, K. Coutinho, N. John, and J. A. Arul Jothi, “Live Cricket Predictions for Runs and Win Using Machine Learning,” in *Proc. IEEE Int. Symp.*, 2024.

[3] T. R. Sudhini, et al., “Analysis of Cricket Score Predictor using Machine Learning,” *SSRN*, 2025.

[4] M. Chandru and S. Prasath, “Dataset Normalization in Cricket Score Prediction Using Weighted K-Means Clustering,” *Int. J. of Innovative Studies in Applied Engineering (IJISAE)*, vol. 12, no. 21s, 2024.

[5] K. Thinakaran, “Predicting T20 Match Innings Score Using Novel Random Forest Compared With XGBoost,” *Saveetha School of Engineering*, 2023.

[6] R. Lamsal and A. Choudhary, “Predicting Outcome of IPL Matches Using Machine Learning,” *arXiv preprint*, 2018.

[7] A. Fayyaz, A. Zafar, M. Wasim, S. S. Abbas, A. Ahad, A. A. N. Godinho, P. J. Coelho, and I. M. Pires, “Cricket Score Prediction using Machine Learning Techniques,” 2024.

[8] P. Dalal, H. Shah, T. Kanariya, and D. Joshi, “Cricket Match Analytics and Prediction using Machine Learning,” 2024.

[9] O. Mozar, S. More, S. Nagare, and N. Pathak, “Cricket Score and Winning Prediction,” 2022.