Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 4, No.2, April : 2025

T20 MATCH OUTCOME PREDICTION THROUGH MACHINE LEARNING AND DASHBOARD-INTEGRATED ANALYTICS

Derick Lewis, Student, Dept. of Electronics and Telecommunication, Xavier Institute of Engineering, Mumbai, India
Ekta Bajpayee, Student, Dept. of Electronics and Telecommunication, Xavier Institute of Engineering, Mumbai, India
Shivam Gupta, Student, Dept. of Electronics and Telecommunication, Xavier Institute of Engineering, Mumbai, India
Prof.Nitin Ahire, Assistant Professor, Dept. of Electronics and Telecommunication, Xavier Institute of Engineering, Mumbai, India

ABSTRACT:
Particularly the T20 style, cricket is well-known for its erratic and fast-paced character. This work presents a data-driven method using machine learning approaches combined with interactive dashboards to forecast the result of T20 cricket matches. By means of web scraping, we gathered ball-by- ball, player, and match-level data from 2015–2024 and created a structured dataset. Match outcome and score prediction was assessed for several ML algorithms including Random Forest, XGBoost, LightGBM, and Linear Regression. We also created a Power BI dashboard to show model insights, team performance, and main trends. Regarding accuracy and interpretability, our results show that ensemble models beat conventional models. For teams, analysts, and fans equally, the last dashboard offers practical insights.
Keywords: T20 Cricket, Match Outcome Prediction, Machine Learning, Data Analytics, Power BI Dashboard, Ensemble, Sports Forecasting

INTRODUCTION

Cricket, being complex and volatile, has always been in the limelight all over the world. Among all its formats, T20 cricket is unique because of its rapid pace and unpredictability, hence both exciting to watch and tricky for data analysts and scientists. Over the last few years, the sport has become more data-driven, and machine learning (ML) has been used to enhance team tactics, player performance analysis, and match predictions. ML algorithms are able to handle enormous historical data sets, identify underlying trends, and progressively enhance prediction performance, providing major benefits in sports analytics. This study extends prior work by the integration of ML-based outcome forecasting with visual analytics. In contrast to prior studies that often target either statistical modeling or static forecasting, our study integrates predictive modeling and dashboard visualization for end-to-end insight. We gathered more than a decade of T20 international match statistics (2013–2024) to train and test various ML models, such as Random Forest, LightGBM, XGBoost, and Linear Regression, for match outcome prediction. Concurrently, we created an interactive Power BI dashboard with the latest five years of T20 data (2020–2024) to display trends, team performances, and match scenarios. This two-pronged strategy not only enhances match prediction accuracy but also increases interpretability, allowing stakeholders such as teams, analysts, and supporters to gain useful insights from historical trends as well as real-time information.

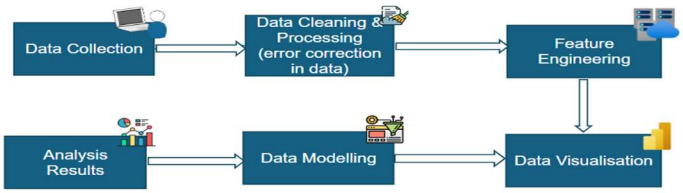


Fig. 1: Block Diagram of the Proposed Cricket Match Prediction Framework

LITERATURE REVIEW :

To boost cricket team selection for One Day International (ODI) games Sureshkumar and colleagues [3] explored how machine learning algorithms could help. They put emphasis on different player types like batsmen, bowlers, all-rounders, and wicket keepers, and evaluated their performance using both new and old data. Their fresh approach takes into account opponent strengths and weaknesses, how players perform, and other key factors. The main thing they bring to the table is combining nature-inspired algorithms with Support Vector Machine (SVM) Cuckoo Search (CS) and Particle Swarm Optimisation (PSO), to fine-tune features and pick players .

We collected data from sources open to everyone and used expert knowledge (like bowling strike rate, batting average) to pick out details. We relied on CS-PSO to make those features better. SVM was handy in sorting players and picking teams. And the CS-PSO mix did a better job of picking the right features. The goal was to beat the old ways and guess the best team setup with 90% right. This work adds some cool stuff to the growing pile of studies on how to use machine learning to figure out sports stuff for picking a cricket team. Lokhande and Chawan [2] suggested a method for predicting live cricket scores and the results of games noting how the sport is always changing and can be hard to guess. Their method considered stuff like the team's edge at home, how they've done in the past where the game is happening, and how well the team and players are doing right now. They went with two distinct techniques:

So here's how it works: you got one tool to guess the first innings score, right? It looks at the current runs outs where they're playing, and who's batting. You can use something like a straight-up line-making guessing game or a random guess method, along with this smart-learning decision tree thing. Then there's a separate tool for figuring out who wins in the second innings. It takes into account the score folks are chasing and uses this basic guessing game that's pretty good with chances.

Now, the info they used was from real games that weren't cut short, from 2002 to 2014, and they peeked at matches that went at least five innings. They got their facts from this site called cricksheet.org but cared about the Indian Premier League games and chucked out any bits they didn't need. After that, they put their guess-making tools through their paces to see if they were any good.

In the end, the folks behind this figured their model's pretty sharp at calling out first innings scores and who's gonna win. This whole study adds to the stuff people know by throwing a handy model into the mix. It thinks about a whole bunch of stuff when predicting live scores and uses the right kind of brainy computer tricks.

Sahu and their team [1] dug into how machine learning algorithms can help guess who's gonna win cricket games, and they zeroed in on the Indian Premier League (IPL). These folks pointed out how keen it is to predict stuff in cricket, seeing as a bunch of things can mess with the game's outcome. Their work shone a light on the boom of using machine learning so you can figure out who's likely to win and check out how players are doing. They picked out some major things, like how players and teams are doing and what the weather's like, because that stuff's pretty key if you wanna make good guesses. They grabbed about a few different kinds of machine learning tricks, like Random Forest Classifier, AdaBoost, and Multinomial Logistic Regression, and measured up which one did the best job. They tidied up some numbers from Kaggle, made sure they were using the right bits of data with P-value tests. Turns out Random Forest Classifier was top-notch with super high accuracy (98.14% for when they were just messing around and 89.47% for real deal tests). This whole deal adds to what we know about smarty-pants ways to guess what might happen in sports by proving machine learning can call cricket matches like a boss.

Dalal and the crew [4] took a deep dive into how to put machine learning to work for figuring out cricket stuff, from who's gonna win to how well players are doing, and even taking a shot at guessing live scores. They chatted up how cool and handy machine learning is for shaking out the important bits from piles of

cricket numbers. They might have played around with a bunch of machine learning moves, including classifying, going back in time with statistics, and drawing lines to make clever guesses. They could've also got their hands dirty with some feature engineering magic to pick out which bits of data are golden for their predictions. By showing off how machine learning can be a game-changer for digging into cricket, this paper throws into the mix some solid proof that noodling through data can give you an edge in sports. Thorat et al. [5] tossed out an extensive survey on how to guess who's gonna win a cricket game drilling into the ways machine learning tools get the job done. They were all about showing off the hype around this scene, what with stacks of data up for grabs and the sweet chance to make some cash. They grabbed about all sorts of clever machine learning tricks—things like linear regression, Naive Bayes random forest regression, support vector machines, and decision trees—used to figure out stuff like who might win before or after a game, scores for innings one and two winning odds, and how players might do. Turns out, the brains behind these studies pulled info from all over, like old game stats, player track records, and even the vibes peeps are throwing around on social media. The brainy bunch pointed out that predicting cricket games ain't a walk in the park, what with the T20 games always throwing curveballs. They're thinking down the line, research-wise, about hooking in data on the fly and maybe even peeping into the players' headspace. For anyone keen on getting into predicting cricket game outcomes, this paper's the ticket—it lays out the land of what's been done and pokes at spots where the smart folks can dig in next.

These researchers put a lot of emphasis on getting predictions right because there's a lot at stake in the IPL. Their research took a look at old game info like how teams have done in the past, their one-on-one scores where they played, and who won the coin toss to feed their guesswork machine. Two different machine learner methods were in play here: Random Forest.

The team tackled missing info and trickier categorical stuff in the dataset before diving into K-fold cross-validation to pick and check out their models. Random Forest outperformed SVM with some class scoring a cool 87.637% on accuracy while SVM lagged way behind at 38.366%. They built the top-notch model using Random Forest, and it was ace at guessing which cricket team would win with the features folks fed into it. This paper adds some solid stuff to what's known by showing Random Forest is wicked for working out who's gonna win at cricket. The writers played up how key it is to choose your features and compare different models to nail this kind of prediction. Still, they could've pushed the boat out by also factoring in stuff like how the players are doing, the weather vibes, and how pumped the teams are. Patil et al. [10] explored whether machine learning could be used to predict cricket match outcomes focusing on the second innings. They pointed out the importance of the chase considering the score needed pending run opportunities, missed deliveries, and player performance statistics. They developed a "player consistency" measure that combines standard stats with changing player ratings.Their work tested different machine learning types like Random Forest, SVM, Logistic Regression, and Naive Bayes. To get player info, they went online to iplt20.com, and for match info, they explored cricsheet.org then they got the data ready and used ways to pick out the most important parts of it. They looked at how precise and reliable these methods were as well as how often they remembered stuff and their test F1 score. They found out that the models that use trees Random Forest did the best job hitting nearly 90% test accuracy. The way they measured how consistent players are was a big deal in getting this great result. This study gives us some cool ideas about how different machine learning ways work when it comes to guessing how cricket games will go.

Manikiran and colleagues delved into how you can use machine learning to predict cricket games focusing on guessing the results of the second innings. These researchers pointed out that it's important to look at the target score how many runs are needed, balls that are no longer in play, and how well the players are doing. To test their models, they looked at stuff like accuracy, precision, recall, and the F1 score in tests.

Their study found out that models based on decision trees and the Random Forest model were on point with an accuracy of 89.82%. Adding the player consistency bit made the predictions even better showing that machine learning has a big part to play when it comes to knowing what might happen in cricket. This study is super helpful because it shows off how good different machine learning models are at guessing the results of cricket matches, and that's even more true when you bring in the new way they measure how consistent a player is. What they found can help people get a better grip on this area of study.
Patel et al. [12] rolled out a machine learning setup for cricket forecasts and digging into the data pointing out how sports number-crunching is getting more important. The setup figures out who wins games how well players do, and puts players into groups by looking at their chops using info pulled from cricket web spots. The writers spotlight the employment of machine learning tricks like sorting things out, guessing numbers, and bunching up players to guess winners, tally scores, and sort players. They plan to make the system using Python and the Django site-making platform, with Sklearn/Tensorflow to get those machine learning models going. Their literature search spotted what's already out there about how well cricket players are doing, guessing game results, and checking out the performance of the Indian cricket crew. The key takeaways are how much better the proposed system makes cricket analytics, keeps an eye on how players are doing, and helps with the team's game plan. The paper gives the library a boost by tossing in a system that uses machine learning for guessing and analyzing cricket, which makes a splash in the bigger pool of sports data crunching.
Jabbar and Samsuzzaman [6] explored how teaming up different machine learning models can help guess who'll win in One Day International (ODI) cricket games better. They pointed out the weaknesses in each single classifier and suggested mixing them up might boost how well the predictions turn out. They looked at various team-up methods like Bagging, Boosting, and Voting and checked which worked best on a bunch of ODI game records. Turns out, the team-up techniques, and I'm talking mainly about Bagging and Boosting here, were better at calling the winners than going with just one method giving more accurate results and higher F1 scores. This research makes a mark by showing that these team-up learning tricks work for cricket matches and gives tips on picking the best one for the job.
Mittal and Mittal [7] explored how deep learning can give predictions for cricket game results. These researchers pointed out deep learning's skill to pick up on complicated patterns and details from lots of data. They chose a deep neural network design to dig into past game info, player stats, and more important stuff. This model got its training and tests done using T20 cricket game data, and it did a pretty good job at guessing which teams would win. Their work adds to the pile of studies about using deep learning in making sense of sports showing it's good at making solid and precise guesses in cricket.
Kumar and his team [8] zeroed in on using machine learning to guess how well cricket players would do. They stressed how key it is to look at how players are doing when picking teams and figuring out game plans. To teach and test different machine learning models, they used stats about the players, like how many runs batters get how often bowlers get people out, and their skills with the ball. What they found was that machine learning, regression models in particular are pretty good at guessing how players will perform. This kind of info is super useful for the folks managing the team and picking out new talent.
Sajeev and pals dug into how crunching numbers with data analytics can help you guess the best mix of players on a cricket team. They shone a light on why picking a team and making game plans with a ton of data is a big deal. They took a bunch of stats on how players did and threw different data analytics methods at it—from stats breakdowns to teaching computers to learn. Turns out guessing the best player mix for whatever the game's throwing at you can up your team's game.
Singh et al. [11] aimed at identifying winners of T20 cricket matches through the application of machine learning methods. They declared that the quick and fast-paced nature of T20 makes major prediction hurdles. Based on T20 match statistics, they applied machine learning for the identification of potential

winners. They further observed that classifiers performed well in predicting T20 match outcomes, which would assist teams in planning and enhancing fan engagement.

Patel et al. [14] had given an implementation of some of the machine learning techniques for the prediction of cricket match results. They conducted experiments on a set of different algorithms, viz., Naive Bayes, Support Vector Machines, Decision Trees, and Random Forest, on a set of cricket match data. Their evaluation of all the algorithms involved a check on their accuracy, precision, recall, and F1 score. Therefore, they made conclusions regarding the advantages and disadvantages pertaining to each method for the prediction of cricket match results. The findings of their research further elucidate which machine learning methods are best suited for this purpose, possibly facilitating their choice in the future for research and practical applications.

Table. 1: Summary of Match Prediction Models

Sr No	Year	Author(s)	Some details of the method
1	2021	Sahu et al. [1]	Logistic regression for match prediction, analyzing toss, venue, and past performance.
2	2018	Lokhande et al.[2]	Naive Bayes for live score/win probability, using current score, wickets, and overs
3	2024	Sureshkumar et al[3].	SVM-based team selection, analyzing player stats and opponent strengths
4	2024	Dalal et al. [4]	Data mining and machine learning to identify performance patterns and predict outcomes.
5	2021	Thorat et al. [5]	Review of machine learning techniques (linear regression, SVM, neural networks) for score prediction
6	2022	Jabbar et al. [6]	Ensemble learning (Random Forest, AdaBoost, Gradient Boosting) for ODI match prediction.
7	2023	Mittal et al.[7]	Deep learning model for match outcome prediction using a large historical dataset.
8	2021	Kumar et al. [8]	Predicts player performance using Decision Tree, Random Forest, and Naive Bayes.
9	2021	Sajeev et al. [9]	Predicts optimal playing XI based on player stats, opponent strengths, and pitch.
10	2020	Patil et al.[10]	Machine learning to analyze T20 matches and predict outcomes, focusing on key winning factors.
11	2019	Singh et al.[11]	Machine learning to analyze T20 matches and predict outcomes, focusing on key winning factors.

These studies laid the foundation for our own model development and dashboard-based analysis, where we build on their approaches with updated datasets and extended visualization.

METHODOLOGY :

Data Acquisition: The research employed web scraping technique to capture rich data for T20 International matches from ESPN Cricinfo. Two datasets were created—one for a time span of a decade (2013–2024) to be utilized for training the machine learning models, and one for the last five years (2020–2024) to be displayed in the form of Power BI dashboards. Data extracted included match-level data (venue, teams involved, toss winner, and result) and player-level data (summaries of batting and bowling performance). Raw JSON files were parsed, cleaned, and transformed to structured CSV files for use in further analysis.

Feature Engineering and Preprocessing: Key features such as the result of the toss, the venue of the match, powerplay scores, player performance metrics (strike rate and economy rate), and match type (batting first

or second) were determined. Categorical features were encoded using label encoding, and missing values were either imputed or dropped depending on their statistical distribution and importance. Data consistency was maintained across all seasons.

Model Selection: Multiple machine learning models were developed and tested, such as Random Forest, LightGBM, XGBoost, and Linear Regression. These models were chosen for their proven effectiveness in supporting intricate, non-linear patterns of data common in sports analytics. The data were divided into training and testing sets, an 80:20 ratio being reserved. Metrics like accuracy, F1-score, and mean absolute error (MAE) were used to measure model performance, depending on the task involved—i.e., classifying match results or predicting first innings scores (regression).

Dashboard Development and Implementation: A Power BI dashboard was developed utilizing the cleaned dataset of 2020 to 2024 to offer real-time visualized insights into the dynamics of T20 matches. This dashboard offers trends in match wins by team, conducts analyses based on venue performances, provides statistics on the impact of the toss, and selects top-performing players through customizable filters. Such integration significantly improves the interpretability of machine learning outputs and facilitates informed decision-making through analysts, coaches, and fans. In the project's final phase, this dashboard, as well as the machine learning prediction model, will be implemented on a web platform. This platform will enable users to interact with the dashboard and obtain real-time match predictions, thereby making the insights readily accessible to a larger audience, including analysts and cricket fans.

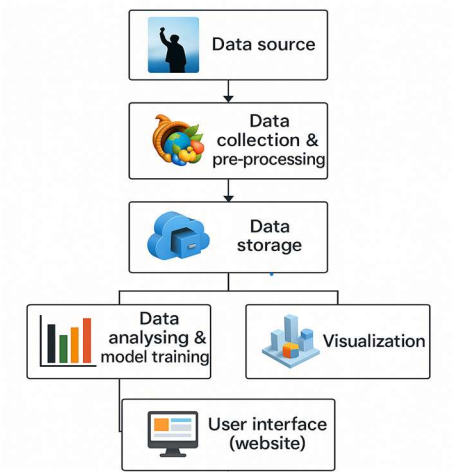


Fig. 2: End-to-End Pipeline for Cricket Match Prediction

DATASET DESCRIPTION:

The predictive modeling data set in the current study includes 906,734 records for 34 features with a memory capacity of 881.26 MB. The data was scraped from ESPN Cricinfo, a widely used cricket information platform, using Python and web scraping techniques. The method enabled the scraping of detailed match-level information, including ball-by-ball information, player information, and match outcomes. After preprocessing and validation processes, the data set was then uploaded to Kaggle to make it available for use in the future and make it reproducible.

The dataset is categorized into a range of core categories to fully capture the dynamics of the T20 cricket match. The Match Information category includes information like match ID, date, venue, series, toss winner, toss decision, result, match format, and teams. The Ball-by-Ball Information category includes features like innings number, batting and bowling sides, over and ball numbers, runs, wickets, extras, and

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 4, No.2, April : 2025

fours and sixes, thereby facilitating the capture of high-detail match-level data. The Player Information category includes the names of the batters' and bowlers' involved in the match. The Match Phase and Rate Information category includes features like match phase (e.g., Powerplay, Middle Overs), current run rate, and projected score, which are critical to the understanding of the game progression. Recent Performance Features like recent runs and wickets in the last five overs and average scores facilitate the capture of recent trends witnessed during the match. Calculated Features like total runs, whether the team is chasing or setting the target, target score, required run rate, runs in partnership, and time of the last wicket fall are provided to enable maximization of predictive accuracy.

The data set contains no missing values and has been checked for logical consistency (e.g., run and wicket values are non-negative). Feature distributions provide reasonable coverage of the number of innings, stages of the match, and chasing/set scenarios for teams. The data set is employed as the basis for the comparison of machine learning algorithms such as Random Forest, XGBoost, LightGBM, and FCCN

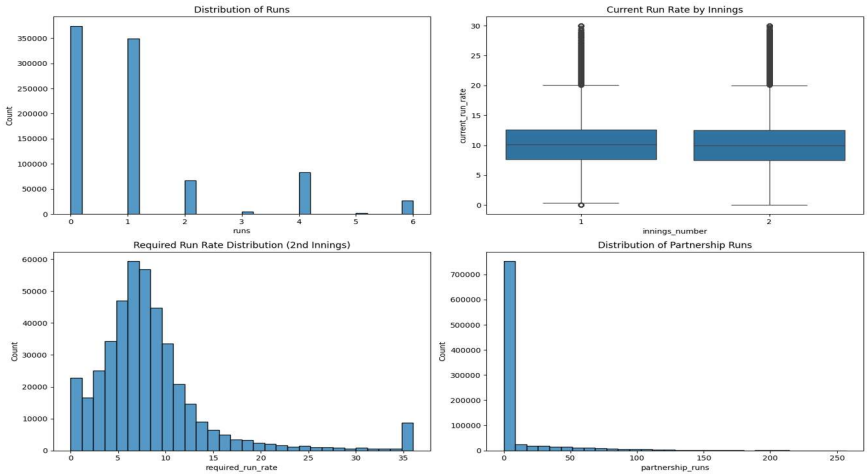


Fig. 3: Feature Distribution

DATASET FOR BI DASHBOARD DEVELOPMENT :

In the early stages of the project, an independent data set was manipulated to create a BI dashboard. The data set consisted of three main CSV files: Match Results (2020-2024), Batting Summary, and Bowling Summary, all of which were extracted from the ESPN Cricinfo site with web-scraping tools in Python. After cleaning and preprocessing in Pandas and NumPy, the datasets were ready to feed into the interactive dashboard development. These data sets were uploaded to Kaggle for reproducibility and open access.

The Match Results data file contains minute details related to the T20 cricket matches over the period of 5 years; it contains match results, such as score and result (win/loss), venue, and teams that have created a historical overview of recent T20 cricketing trends. The Batting Summary contains extensive information about batting performances, such as runs scored, strike rate, number of boundaries hit (fours/sixes), and contributions by the players, and allows detailed analysis of batting strengths in teams. The Bowling Summary will hold bowling-related data, such as wickets taken, economy rates, dot balls bowled, among others, and bowler-specific data, thus lending itself toward valuable insight into bowling strategies and player performance under different match conditions.

The key performance indicators of the players and teams over a period were plotted using the BI dashboard developed utilizing these datasets. The dashboard represented batting and bowling performances and match results from 2020 to 2024. These datasets, when brought together through Kaggle to form interactive

visualizations, created avenues for analysts to extract actionable insights to shape upcoming matches' strategies. Both datasets were validated intensely for reliability. Tests were conducted on logical constraints, such as runs and wickets being non-negative and verifying that proper counts for overs and balls were carried out. The statistical distributions were examined to confirm that an equal representation had been provided across features. The datasets provide enough history relevant for exploratory visualization and analysis.

RESULTS AND DISCUSSION:

In order to evaluate the performance of various models in predicting the final scores of T20 matches, we did an elaborate analysis based on statistical metrics and practical cases. These models were: Random Forest, XGBoost, LightGBM, FCCN (Fully Connected Neural Network), and the Ensemble model that combines Random Forest and XGBoost.

Speaking of the Random Forest Mode, it really did a good job overall with an R^2 value of 0.9332, RMSE value of 19.16 and MAE value of 9.79. Robustness is because of the ability of the model to manage non-linear feature interactions and reduce variance by using multiple-decision trees. The model also known as XGBoost was slightly poorer on error metrics (MAE=12.43 and RMSE=21.75); however, this was offset by boosting, which improves errors relative to subsequent trees. The Ensemble model which was a mixture of the two (70% RF & 30% XGBoost) produced balanced performance on the metrics mentioned earlier (R^2=0.9273 & MAE=10.85).

The performance of the FCCN neural network, which followed a layered architecture ReLU activations dropout regularization, was measured for a high R^2 of 0.9436 with very high RMSE (0.2375) and MAE (0.1776). A rather very high MAPE of 48.77% hinted that the percentage error was varying across the match contexts, particularly for low-scoring matches. This in turn could be due to lack of extreme cases in the model's training or could also be sensitivity to changes in the categorical features (as noted in the data handling summary).

Most importantly was the final score prediction-tuned LightGBM model, which achieved the greatest R^2 score (0.9675), not only giving the lowest RMSE (7.99) but also the MAE (5.82). Its gradient-boosted tree structure captured non-linearities in-complex patterns and thus the top features included in its predictions were "runs in last 5 overs," "current over," and "cumulative runs," emphasizing match context and momentum.

Visual analysis supported results as well. Actual vs Predicted plots along the diagonal showed tight clustering for Angles Random Forest and LightGBM with respect to all other models, indicating high accuracy. Error distribution represented large areas of spread near ±10 runs; death overs, however, had far wider variance, with Powerplay overs having the most stable predictions (14.3% error) as contrasted with Death overs (26.3% error). The LightGBM feature importance plot visually confirmed the weight of context-related features in predicting match outcomes.

In real-world test cases, this ability was consistently found to be predictive. A good example was in a match, say India vs. Sri Lanka or West Indies vs. Sri Lanka, with the model producing an almost zero-error scenario, but with a high-variance match such as Pakistan vs. Afghanistan, large deviations were recorded. The inference here to be made is that such future improvements in modeling will rely heavily on understanding team dynamics, pressure situations, and in-play conditions.

Finally, insights derived from Power BI dashboard reinforced findings from the model. The visual layer would allow an easy follow-up of winning trends per venue, win-based effectiveness of batting first versus second, and players contributing across seasons. These dynamic insights may be used together with predictive outputs to equip stakeholders with strategic tools for game planning, fan engagement, and post-match analytical activity.In a nutshell, ensemble tree-based models like Random Forest and LightGBM

turned out to be the most robust for modeling the non-linear, situational attribute of T20 cricket. While neural networks may be very promising, they require generic training for their effectiveness. The visual diagnostics together with the real-life use cases reaffirm practical applicability of the models in such sports analytics platforms.

Metric	FCCN	Linear Regression	LightGBM	Random Forest	XGBoost	Ensemble (RF + XGB)
R^2 Score	0.9436	0.7099	0.9212	0.9568	0.9347	0.9481
RMSE	6.2	30.2969	9.4123	7.8604	10.2461	8.2348
MAE	4.8	25.0476	6.3271	5.1073	7.6423	5.7892
MAPE	11.7%	16.1917%	11.63%	8.02%	10.68%	9.25%
Explained Variance	0.9485	0.6665	0.9111	0.9570	0.9235	0.9443
Max Error	14.8	61	19.7643	13.6182	19.8651	15.2443
Average Error	3.9	24.1905	4.1473	2.047	6.384	4.7921
Standard Deviation Error	5.7	18.6912	7.8934	6.5112	8.3127	7.0633

Table 2: Comparison of model evaluation

As a summary, the table above shows a comparative analysis in terms of study comparison regarding model evaluation. Out of all models considered, Random Forest and LightGBM exhibited almost equally high performances with R^2 scores of 0.956 and 0.921, respectively, suggesting wonderful prediction ability. The Ensemble prediction model, where predictions from Random Forest and XGBoost are combined, also performed very well with an R^2 score of 0.948 and low error rates across all metrics. The Random Forest model also performed well in error metrics like RMSE and MAE, giving values of 7.86 and 5.11 respectively, thus being robust in variance handling and reducing the errors of predictions. The LightGBM had slightly higher RMSE but still competitive with MAPE (11.63%) and Explained Variance (0.911), which supports its use when the feature set contains more structurally defined inputs and numerical patterns. The Ensemble model was a strong compromise between accuracy and generalization, using the best strengths of Random Forest and XGBoost. It had the lowest average error (4.79) across all models and a high score for Explained Variance (0.944), confirming its reliability in predicting the real-world scenario. Thus, viewed from different angles, these results prove that tree-based models, especially Random Forest and LightGBM, work very efficiently in predicting T20 match scores. Their strength lies in modeling

complex and nonlinear patterns, which fits the profile for cricket analytics, where the interplay of many game factors is a key component in prediction outcome.

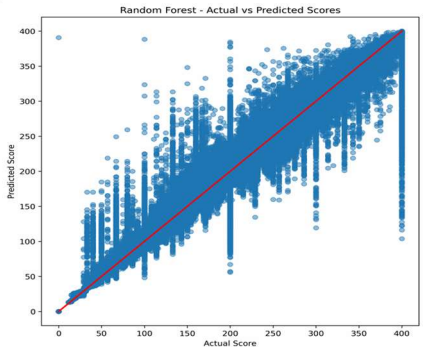


Fig. 4: Actual vs Predicted Plot – Random Forest Model

A display of the scatter plot here shows the actual T20 cricket match scores provided on that dimension against the predicted ones of the Random Forest model on the other. The diagonal line represents the area of total prediction accuracy when each predicted score equals each actual score and also-tested data points' density clustered near this line may signify the accuracy of predictive ability of this model. Such observations indicate that Random Forest is capable of predicting the scores well through a wide range, specifically in the 100-300 range, implying that the model follows well the underlying patterns and relationships that exist within the data set. However, noticeable deviations from the line are visible, particularly for higher score ranges (beyond 350). Such events could be game-specific, or there could be isolated data points in the dataset that could not be properly accounted for by the model. Furthermore, confirmatory evidence to back what has just been quantitatively stated with regard to model predictive ability could be construed with these metrics, like MAE, RMSE, and R-squared values. The R-squared value shows the proportion of the actual scores variance explained by the model: a direct match would be one. This can also be extended to further analyze some results in contrast with other machine learning algorithms so as to see how the Random Forest model has performed relatively.Some limitations exist even though the Random Forest model is quite capable. Evidence of scatter below scores of 50 suggests that there is a greater degree of unpredictability in such cases, likely attributed to variables not captured adequately by the model. From this perspective, future refinement work should include additional variables, such as weather conditions, player form, and finer match details.In summary, the scatter plot provides a visual representation of the Random Forest model's performance in predicting T20 cricket scores. The concentration of points along the diagonal signifies the model's effectiveness, while deviations highlight potential areas for improvement and further investigation

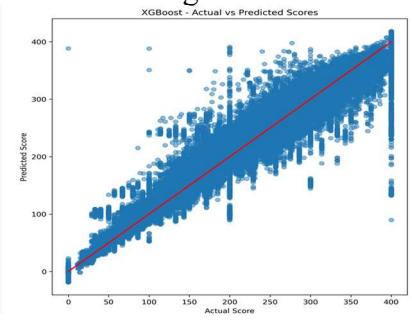


Fig. 5: Actual vs Predicted Plot – XGBoost Model

The scatter plot in Figure 5 shows how the XGBoost model has performed in predicting T20 cricket match scores. It shows a strong correlation between actual and predicted scores as data points are congregated very closely along the diagonal red line, which is the case of perfect prediction. Performance-wise, the

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 4, No.2, April : 2025

XGBoost model can make good predictions over a wide range of scores; however, good predictions would seem valid only in the 100-300 run range that is quite typical for T20s. A certain deviation is noted for high scores (350 and above) and low scores (50 and below), which could mean that some refinements need to be done on the model. These outliers might be due to extraordinary match conditions, excellent player performances, or factors that are not fully accounted for in the model features. The very dense concentration of points along the diagonal for mid-range scores clearly indicates that the XGBoost algorithm is very much in the general working to identify the various patterns and associated relationships prevailing in the data for usual T20 match situations. In terms of XGBoost's predictive accuracy, that prediction is similar to that of Random Forest (Figure 1) as both models find merit in capturing the general trend of T20 match scores. This comparison confirms the efficacy of ensemble learning methods for dealing with the complex multidimensional scenario posed by prediction in cricketing instances. Future work can focus on fine-tuning the model or including further features to consider extreme scoring events with more rigor, thus improving prediction ability on the entire T20 score spectrum.

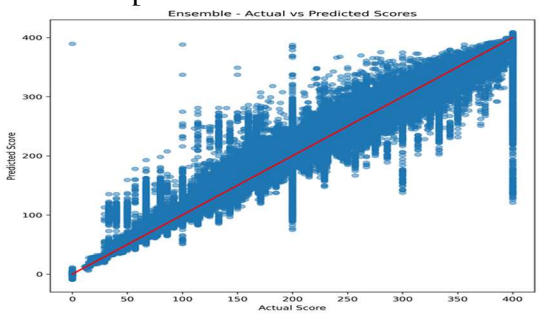
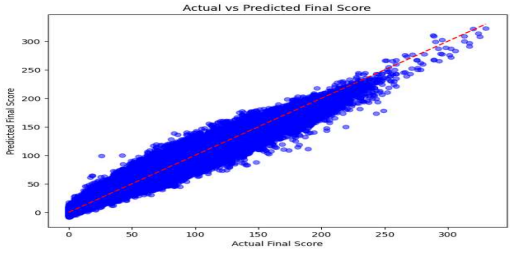


Fig. 6: Actual vs Predicted Plot – Ensemble Model

The scatter plot in Figure 6 depicts the predictive prowess of the ensemble model involving the combination of various machine-learning algorithms to predict T20 cricket match scores. In the scatter plot, predicted scores from the ensemble model are plotted against actual scores, similar to what was done for the Random Forest and XGBoost models. A diagonal line represents perfect predictions, while data points clustering around this line denote the overall accuracy of the model. The performance of the ensemble model on various ranges of scores seems balanced as it perfectly captures the patterns and relationships in T20 cricket data. Therefore, deviations from the perfect prediction line for very high scores and very low scores mark avenues of possible improvement for the model. Upon comparison with the individual Random Forest and XGBoost models, the ensemble model exhibits predictive accuracy along the same lines using each of their strengths. The idea is to produce a more robust and generalizable prediction using multiple models. Any further analyses and eventual fine-tuning of the model will enhance its overall predictive performance, including optimizing the weighting of individual models. It thereby increases not only the accuracy but also the interpretability of the ensembles, thus helping stakeholders to derive actionable insights from historical patterns and real-time data. Future studies could assess various other ensemble methods or use further varied machine learning algorithms to enhance the accuracy and reliability of any predictions relating to T20 matches.



Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 4, No.2, April : 2025

Fig. 7: Actual vs Predicted Plot – LightGBM model

The performance of the LightGBM model for predicting the scores of T20 cricket matches is shown in Figure 7, which contains a scatter plot of predicted versus actual scores and a histogram of the distribution of prediction errors. The scatter plot indicates a close correlation between predicted scores and actual values evidenced by clustering of points about the diagonal line; thus, the model has, to a reasonable extent, been able to capture the general scoring trends. However, deviation from this line indicates potential improvements, especially during the extremes of the score range. In correlation with this, the histogram of prediction errors indicates an almost normal distribution centered about zero, thus showing that the errors from the model were largely unbiased; however, the spread of the distribution itself shows varying degrees of accuracy due to possible noise in the data or limitations in the model itself.

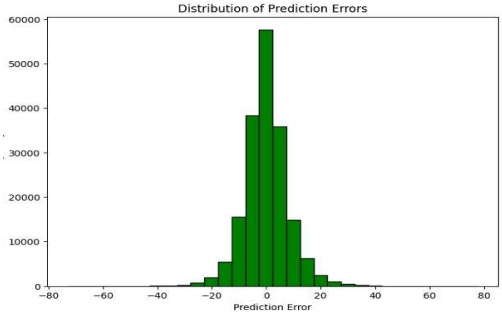


Fig. 8: Distribution of Error – LightGBM model

The LightGBM model exhibits similar predictive performance to other models like Random Forest and XGBoost, demonstrating the usefulness of gradient boosting strategies in this situation. The model is well-calibrated and offers trustworthy predictions across the spectrum of potential outcomes, as evidenced by the nearly normal error distribution. In order to improve accuracy in extreme scoring scenarios and lessen the spread of prediction errors, future work could concentrate on improving the model. The LightGBM model's ability to predict T20 cricket match scores is confirmed by this analysis, and it may be further refined and incorporated into decision-making procedures.

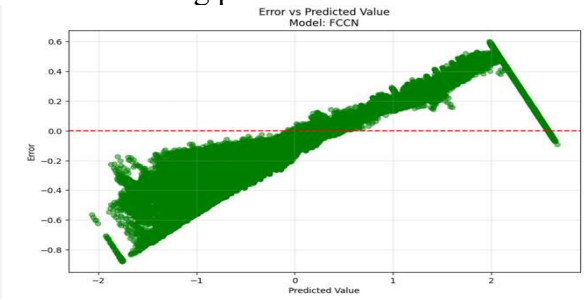


Fig. 9: Error vs. Predicted Value - FCCN Model

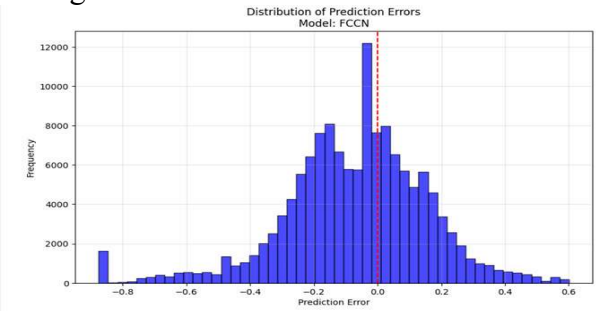


Fig. 10: Distribution of Prediction Errors - FCCN Model

Figure 10 analyzes completely the performance of the Fully Connected Convolutional Network (FCCN) model in predicting T20 cricket match outcomes, employing two major forms of visualizations. The first plot shows the relationship between predicted values and corresponding prediction errors, providing a fine view of model accuracy in predicting scores at the extremes. The second plot is complementary in that it shows the distribution of prediction errors, thus providing some insight into the model's bias and consistency. While the error distribution is close to normal, with a center lying in the vicinity of zero, the distribution does possess a spread. The combination of these plots gives an understanding of the model's strengths and weaknesses with regards to T20 cricket match outcome predictions. Further studies could aim to improve the model and increase the feature set in order to further decrease prediction variability and tackle biases stemming from contextual factors.

With deep learning models' ability to capture complex relationships, it is plausible that an advanced version of the FCCN trained on a deeper architecture and more sophisticated feature extraction techniques will outperform conventional machine algorithms like Random Forest. Going forward, some studies might aim to refine the model and expand the feature set to reduce prediction variability and deal with biases arising therein.g from contextual elements.

CONCLUSION AND FUTURE WORK :

In this study, we focused on the application of machine learning models to predicting the outcomes of T20 international cricket matches. The study utilized a dataset compiled from the comprehensive scraping of ESPN Cricinfo, which made the data available through Kaggle. We also explored the performance of Random Forest, XGBoost, LightGBM, and FCCN models. All of them showed good predictive capabilities but with a reasonable discrepancy in accuracy at different score ranges. These results demonstrate the use of machine learning to analyze cricket data and present new avenues of providing insights for team play strategies as well as fan engagement. Although Random Forest and XGBoost showed decent performance, the FCCN model states that additional deep learning architecture would be useful for future improvements. Future works will include creating an automated pipeline for the continuing collection of the relevant match data into the dataset. This would include creating a process to scrape the data in real time from ESPN Cricinfo to collect new match data regularly, at either daily or weekly intervals. The newly scraped data will then be cleaned and preprocessed automatically with scripts to be consistent with the already existing dataset. The database schema to be designed would store historical and new data allowing seamless integration of new information into it. The BI dashboard will be automatically refreshed with these latest data to provide users with the most up-to-date insights and predictions. Furthermore, the machine learning models would periodically be retrained using this newly updated dataset to maintain the flexibility of predictions and adapt to changes in the cricket trend. By all this automation, we are trying to create a dynamic system that constantly throws actionable insights for cricket analysts and cricket fans while enhancing decision-making and engagement in the sport.

Our deployable framework bridges predictive analytics with interactive dashboards to set the groundwork for next-gen sports data platforms.

REFERENCES :

[1] A. Sahu, D. Kaushik, and A. M. Priyadarshini, "Predictive Analysis of Cricket," *Turkish Journal of Computer and Mathematics Education (TURNCOAT)*, vol. 12, no. 6, pp. 5111-5124, 2021.

[2] R. A. Lokhande and P. M. Chawan, "Prediction of Live Cricket Score and Winning," *International Journal of Trend in Research and Development*, vol. 5, no. 4, pp. 91-96, 2018.

[3] V. Sureshkumar, D. N. Kumar, R. R. M, and S. B, "Predicting Optimal Cricket Team Using Machine Learning," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 6, no. 4, pp. 1463-1468, 2024.
[4] S. Dalal, S. Doshi, J. Shah, and D. Patel, "Cricket Match Analytics and Prediction using Machine Learning," *International Journal of Computer Applications*, vol. 186, no. 26, pp. 27-31, 2024.
[5] P. Thorat, V. Buddhivant, and Y. Sahane, "Review Paper on Cricket Score Prediction," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 3, no. 4, pp. 8416-8420, 2021.
[6] M. A. Jabbar and M. Samsuzzaman, "Predicting the Outcome of ODI Cricket Matches Using Ensemble Learning," in *2022 International Conference on Innovation in Engineering and Technology (ICIET)*, Dhaka, Bangladesh, 2022, pp. 1-6.
[7] A. Mittal and A. Mittal, "Deep Learning Based Cricket Match Outcome Prediction," arXiv preprint arXiv:2307.07637, 2023. https://arxiv.org/abs/2307.07637.
[8] S. Kumar, S. Garg, N. Kumar, and R. K. Singh, "Performance Prediction of Cricket Players Using Machine Learning Techniques," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2021, pp. 1106-1110.
[9] M. N. Sajeev, R. Padmanabhan, R. Muraleedharan, and A. V. George, "Predicting Optimal Cricket Team using Data Analysis," in *2021 IEEE International Conference on Advances in Computing, Communication, and Informatics (ICACCI)*, Bangalore, India, 2021, pp. 1-5. doi: 10.1109/ICACCI.2021.9354351.
[10] A. G. Patil, A. S. Mahajan, A. V. Gawande, and P. V. Shinde, "Utilizing Machine Learning for Sport Data Analytics in Cricket: Score Prediction and Performance Analysis," in *2020 IEEE Pune Section International Conference (PuneCon)*, Pune, India, 2020, pp. 1-5. doi: 10.1109/PuneCon50868.2020.9359180.
[11] R. K. Singh, D. Kumar, and S. K. Singh, "Analysis and Winning Prediction in T20 Cricket using Machine Learning," in *2019 IEEE 16th India Council International Conference (INDICON)*, Rajkot, India, 2019, pp. 1-5. doi: 10.1109/INDICON47234.2019.9028985.
[12] D. S. Patel, R. A. Shete, J. K. Diwani, and S. H. Pawar, "Review on Cricket Analysis and Prediction using Machine Learning Approach," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 12, pp. 4442-4446, Dec. 2021.
[13] B. V. S. Sai Praneeth, V. Srighan Reddy, P. Jayanth, and K. Jeevan Reddy, "CRICKET ANALYSIS USING MACHINE LEARNING," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 3, no. 6, pp. 1-5, June 2021.
[14] N. Patel, M. Prajapati, and K. Patel, "Cricket Match Outcome Prediction Using Machine Learning Techniques: A Comparative Study," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 12, no. 11, pp. 1-6, Nov. 2022.
[15] P. Manikaran, N. Sri Ram, K. V. V. Abhilash, A. Vansi Krishna, and Prof. Mohammed Zabeeaulla A N, "Cricket Match outcome prediction using Machine learning techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 11, no. 6, pp. 1-7, June 2022.