Real-Time Cricket Intelligence Platform

Md Asraf Ali
Department of Computer Applications
PES University
Bengaluru, India
asrafalidhaka123@gmail.com

Tamal Dey
Department of Computer Applications
PES University
Bengaluru, India
tamal_dey@pes.edu

***Abstract*—This paper aims at improving the analytical approach towards T20 cricket by providing real-time information. In particular, the implementation result published via this research work will display live scores of all the games in action together with animated statistical charts targeting not only fans but also industry players. Analysis is also provided before the match and suggestions given on the call to bat or bowl first based on points discussed with respect to historical performance of the team in a particular ground and environmental setup. Further, a machine learning model will also be created to predict, in real time, the likelihood that a team is going to ace the pursuit in the second innings and these estimates will be updated after each delivery.**

***Index Terms*—Keywords: Real-Time Analytics, T20 Cricket Insights, Machine Learning Prediction, Venue-Specific Trends, Dynamic Statistics.**

I. Introduction

With the early development of a new level of data analytics and machine learning, the field of sports has been transformed significantly, and cricket offers a good example, where the use of real-time analysis and other strategies makes a decisive impact on the course of a game. The fast and unfavorable evolvement of T20 cricket makes it a quite profitable environment that can be used to enhance decision-making through data-driven practices and stimulate interest on the part of fans. Based on it, this paper strives to use the possibilities by creating a comprehensive platform that suits the T20 event. The platform consists of a live updating of the score, moving performance measures, and context-relevant analytical content, thus enhancing the experience of the spectators and analysts. Leaning on historical data (particularly on the specific venue) as well as on environmental factors (e.g., weather, pitch conditions, etc.), the needs of the initiative are pre-match recommendations on the issue of taking the toss. At the center of the exercise is a machine-learning model that is kept running after every ball and predicts the likelihood that a team chose to pursue a target in the second innings. In this way, it aims to complete the data analysis with real-life practical implication in T20 cricket by using Python, ReactJS, Node.js, and MongoDB distributed database, and generalizing the empirical findings of the current cricket prediction research. Finally, the goal of the paper is to contribute to the field of sports analytics that conceptually focuses on strategic decision-making and providing discipline-specific data to fortify the accuracy of the made decisions with the help of evidence obtained during the game in real-time and empirically based notions of toss strategies.

This paper tries to solve ongoing issues faced in the twenty-over cricket, especially when it comes to the aspect of toss-making and time-to-time outcome prediction. The system uses ground-specific conditions and historical vast data sets, so it provides evidence-based suggestions on the toss methods. Also, an algorithm based on random forest working under the guidance of efficient studies constantly checks the availability of wickets and rate at which one can score runs to further calculate the chances of victory. The interface of the platform is built on the technology ReactJS, and its back-end is constructed on the technologies Node.js and MongoDB, which guarantees its convenience and scalability at the same time. Some of the interesting innovations are having player consistency measures in the model of prediction that shows accuracies of 98 percent.

This program will help create a basis of future cricket analytics by gathering data of ESPNcricinfo in training the model via web-scraping libraries like Beautiful Soup. In comparison to 96.31 percent of correctly predicted T20 matches, it stands out of such websites as CricViz by virtue of containing the toss support and live prediction features. Considering that it achieved 90 percent in live score completion, it is taking machine learning in sports to the next level by associating the data understanding with fans of the sport, cricket.

The basic aim of the present study is to innovate the strategic and analytical approach of T20 cricket to include the use of powerful technology and new features. This paper is format specific and complete product; that will solve the shortcomings of the existing systems, their most significant shortcoming is the absence of a complete and exhaustive value of toss selection. The focus of the platform on the real-time, actual-ball estimations also reflects the blistering change of the game, as well as provides partners with instant, practical knowledge. Further refinement of predictive modeling based on the paper plan shows that it will be able to revolutionize the team strategy and extend the fan interaction with the T20 cricket, thus, making a notable contribution to the sports analytics field at large.

II. Related Work

Machine learning has evolved into one critical analytical tool in the presence of cricket, and it aids in accurate forecasting of outcomes and determines a strategy to be used. A landmark study, Predicting IPL victories using an Ensemble modelling procedure, top-selling here, and done by the researchers Singh et al. that was supported in a second International Conference on Artificial Intelligence and ML Applications, understands the use of ensemble machine learning to predict the matches in the Indian Premier League (IPL). Decision tree and random forest classifiers are used to perform the research to evaluate the player performance, the team statistics, the venue condition and the weather variables. We found that the ensemble construction could give better predictive accuracy as compared to when independent classifiers, including KNN and logistic regression are implemented. The results provide wisdom into the need of data collection rigor, preprocessing, and hyperparameters optimization to generate credible forecasts. The research promises to aid the stakeholders such as team strategists, bookies, and players of fantasy cricket by presenting data-driven information. At the same time, this study forms the foundation of future research novelty of cricket analytics, in particular, the implementation of the real-time modeling and the deep-learning in cricket analytics. Being used as an ensemble method, and explicitly focusing on situational aspects such as weather and player data, the study is most relevant to this paper, which aims at making the real-time T20 victory prediction algorithms more accurate.[1]

Machine learning has proven to make significant improvements with its level of predictive analytics and the provision of strategy in modern cricket analysis. It is a paper by Singh et al. with the title Predicting IPL Victories: an Ensemble Modeling Approach using Comprehensive Dataset Analysis, which is presented in the 2 nd International Conference on Artificial Intelligence and Machine Learning Applications, which reports the Decision Tree and Random Forest methodologies of predicting the Indian Premier League victory; however, it did look at the performance of players, team-wise statistics, the features about the venue and the weather conditions. Severe data preprocessing and repetitive hyperparameters are discussed as the study has reported high accuracy performance than other predictive models like KNN and logistic regression. As a result, the research is specific to the stakeholders such as fantasy players and team managers, and it indicates future improvements in crickets analytics especially in using deep learning. Further, it is connected with this paper that further optimizes the T20 victory predictions through combining the ensemble approach and clear contextual appraisals. [2].

III. Methodology

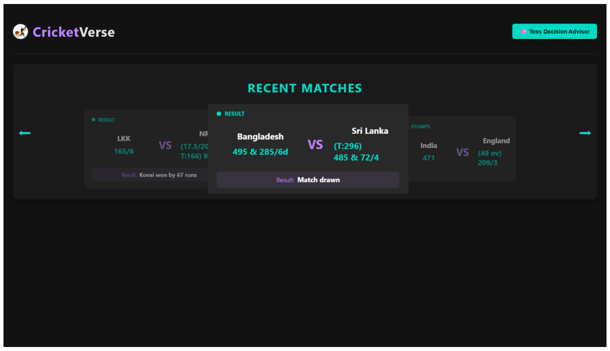
The given paper proposes an iterative approach that will be used to meet the functional demands of this paper application in terms of four fundamental capabilities; the real-time score provider, continuously updating statistics, toss prediction, and ball-by-ball victory estimation of T20 matches. The system aims to increase user participation and strategic decision-making due to use of advanced analytics and current web technologies. In this respect, a systematic framework is described, which encourages efficacy with user-centric design. As a first pass, the implementation of key features to track live score information, insights into venues specific staff and the win probabilities in real-time were identified by user need analysis, match system considerations and dynamic accuracy of the product. The system architecture was built in such a way that it is scalable with interactive front-end interface based on ReactJS and the back end using Node.js, Express.js, and MongoDB as the means to efficiently handle and retain data. The application of machine learning features contributed to high prediction accuracy, especially Random Forest in Python, with such libraries as Scikit-learn, Pandas, and NumPy being added to investigate player results, previous games, and playing conditions. To complement the same, web-scraping software packages were implemented to access real-time information in web-based sources like the ESPNcricinfo that allowed dynamic update and extensive model training.

A. System Design & Architecture

System Design Architecture from this point-of-view, this paper is architect as a layered system to ensure scalability, modularity, and the ability to make real-time T20 cricket analytics in an efficient manner. It includes these four main layers: Presentation Layer, Business Layer, Service Layer and Database Layer, each of which is dedicated to perform the certain functions and to meet the needs of the paper. The Presentation Layer, which is carried out in ReactJS, provides an interactive and a user-friendly interface both to the fans and the analysts and portrays real-time score, dynamic statistics, toss-decision suggestions, and real-time win projections. The Business Layer is responsible to control the core logic, such as consumption of real-time match data, modelling of the win probability based on a machine learning model (Random Forest), and advancement of venue-specific information in regards to throwing decision. Service Layer acts as an intermediary, enables communication between front end and back end through APIs and is implemented in front of working with Node.js and Express.js). Using MongoDB to serve as The Database Layer, structured match data, player statistics, and venue-based trends are stored in document model that captures individual valuable attributes of team scores, match results and timestamps so that an efficient retrieval could be achieved. Real-time module through Beautiful Soup and Requests can be implemented in the Service Layer to retrieve current data on websites like ESPNcricinfo hence making the system responsive in real matches.

B. Live Score and Statistics Display:

This research paper will provide the wielder of this product with immediate feedback and comprehensive body of knowledge in the form of intuitive live score and statistics user interface custom-made to Twenty20 matches. The people using it get important performance statistics like the current run rate, run rate a chasing team needed to set, and an approximate number of runs that a team is likely to finish with based on on-the-fly game paths. Supplementary ball-by-ball information which includes deliveries bowled, runs scored, and wickets taken are also spread continuously. In order to achieve this data integrity, the system applies web scraping techniques, specifically Beautiful Soup API and Requests API to scrape current information on reputable sources, specifically, the ESPNcricinfo. These updates are dynamically displayed in a ReactJS front end thus providing a smooth interactive interaction between the fans as well as the analysts. Moreover, the portal also provides micro-data, such as team and individual player performance, team affiliation and past pitch trends that helps the user to quiz the match mechanics and come up with educated forecast, besides enhancing user interaction through graphically simulated interface.



RESULT
LKK 165/6 VS NR (17.5/20 T:166) 9
Result: Kovai won by 67 runs

RESULT
Bangladesh 495 & 285/6d VS Sri Lanka (T:296) 485 & 72/4
Result: Match drawn

STUMPS
India 471 VS England (49 ov) 209/3

Fig. 1: Live Score and Statistics Display

C. Second Inning Win Prediction:

Chances of a chasing team winning their target score is re-evaluated by the powerful second-innings win prediction engine in this paper that has been tailor-designed in respect to two-twenty cricket. Contemplating match outcomes in the real time, ball-by-ball, the engine weighs the probabilities in terms of wickets in hand, run requirements, and in terms of the individual player performance. While Python machine learning framework Scikit-learn and Pandas were used in the development of a Random Forest model, the model deals with historical data in the interaction with the contemporary factors to come up with the correct estimates. The data are collected as real-time data through the web-scraping tool Resources like the ESPNcricinfo, Beautiful Soup, and Requests to make sure that the model inputs include the actual data. The derived predictions are conveniently incorporated with the ReactJS front-end and presented as user-friendly and interactive visualisation, and thus, enable the analysts to have a strategic analysis and promote the fan engagement during the live games.

D. Toss Decision Recommendation:

It is a total package that incorporates the system of generating recommendations that can be used by the Twenty-20 team in order to make evidence-based recommendations once they have won the toss. Using the analysis of historical match data and the specific ground conditions i.e. Venue-Specific trends, pitch characteristics and weather conditions like overcast conditions which play an advantage to swing bowling, the system suggests suitable selection of batting first or bowling first. A pandas data analysis is supported with NumPy libraries to access historical data that has been previously stored into the MongoDB database in order to obtain patterns based on which decision would have been of the greatest benefit at each of the venues. The data obtained in regard to the environmental factors are acquired in real-time when using scraping tools such as Beautiful Soup and Requests on sites known to be responsible and thus the data would be accurate and relevant. The resulting recommendation will be shown on the ReactJS front-end, and will provide teams with the clear, data driven information whether they should bat or bowl first hence improving strategy decision and the possible outcomes of the match.

E. User-Centric Design & Accessibility

It is a digital media platform, in which priority is given to a user interface that has been specifically designed to be innovative and highly intuitive, where the navigational structures are understandable and contextually designed. It is universally designed and accessible, so design and user-focussed with keyboard accessibility, high-contrast and support of screen-readers are integrated into the system to support inclusivity in the user experience and adhere to current UX-UI design standards.

IV. DESIGN AND MODELING

A. High Level Diagram

The architecture shown in diagram below is the one usually associated with Petopia application. It includes four layers which are vertical in nature they are the Presentation Layer, Business Layer, Service Layer and the Data Service Layer.

- **Presentation Layer**: The user interface forms an interface which is between the management and the users of the system. In this regard, all screens of the presentation layer are defined as the UI screens.
- **Business Layer**: The business layer forms the central decision making engine of the application that wraps all business logics and domain rules. It receives commands given by users and makes decisions as well as performs corresponding calculations. Some of its most important activities include a strict developer of user inputs.
- **Service Layer**: The Service layer plays the role of mediator between Business layer and Data service layer. Its main task is to act as an intermediary between these layers: starts pet-related operations as it sends pet data and pictures to Data Service layer where they are stored and then retrieves the same pet-related data and pictures to the Data Service and returns to the Business layer.
- **Data Service Layer**:Data Service layer is used as a custodian of data and it invokes SQL commands in the database to support CRUD operations. It is in charge of

such but not limited to storage of the pet information, preservation of the user information, management of the related images thorough and comprehensive functions within the scope of Data Service layer.

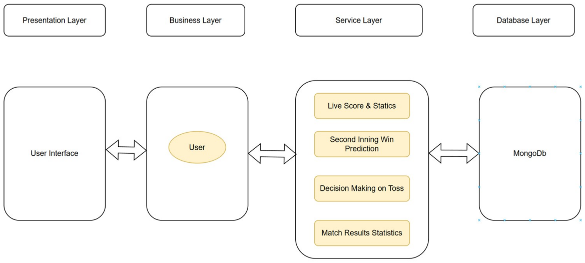


Fig. 2: Architecture Diagram

B. Use Case Diagram

According to the definitions of Unified Modeling Language (UML) a use case diagram represents visually the relationship between the actors and the system. Actors according to the UML standard are all the entities that interact with the system except the system.

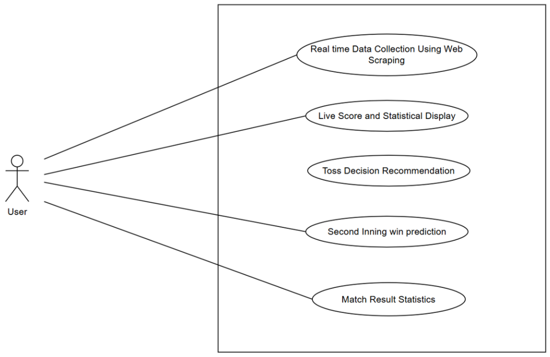

Real time Data Collection Using Web Scraping
Live Score and Statistical Display
Toss Decision Recommendation
Second Inning win prediction
Match Result Statistics
User

Fig. 3: Use Case Diagram

1.) User - A user who is using the application can meet several use cases: (1) registration, (2) log in/log out, (3) updating his/her personal profile, (4) posting the advertisement on the sale and adoption of pets, (5) viewing both advertisements on sale of pet and adoption, (6) contacting potential owners via system, (7) booking veterinarian and pet-walking services. All the scenarios will be presented as individual tasks that the user should complete.

C. Process Flow Diagram

The chart explaining the step-by-step process through which designed and evaluated its Machine Learning model can be seen below.
2.) Start - The procedure begins with data collecting.
1.) Data Collection — Information pertaining to cricket games is gathered through various channels which may include real time game values, past records and individual reports on player performance.
3.) Data Cleaning – Raw data is subjected to processes like missing value imputation, correction of inconsistency as well as transformation of available variables into new ones during the process of pre-processing. This is required so that there is uniformity of data and it will be feasible to analyse it later.
4.) Feature Extraction – To create the accuracy of the predictive model, key performance indicators; batting performance, bowling effect, and run rates were made use of to generate the conclusion.
5.) Train/Test Split – The dataset is divided into training and testing subsets to ensure the model is trained effectively and evaluated properly.
6.) Training of the model using ensemble techniques - The model is trained using an ensemble approach, which incorporates multiple algorithms to improve predictive accuracy.
7.) Testing — The trained model is tested with unseen data to measure its performance.
8.) Model Evaluation – Key evaluation metrics such as accuracy, precision, recall, and F1-score are analysed to determine the model's effectiveness.
9.) Testing on Real Data — The model is further verified using real-world match scenarios to assess its predicted dependability.
10.) End – The procedure closes, guaranteeing that the model is ready for deployment.

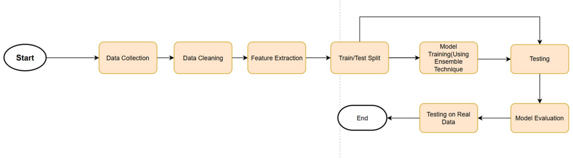


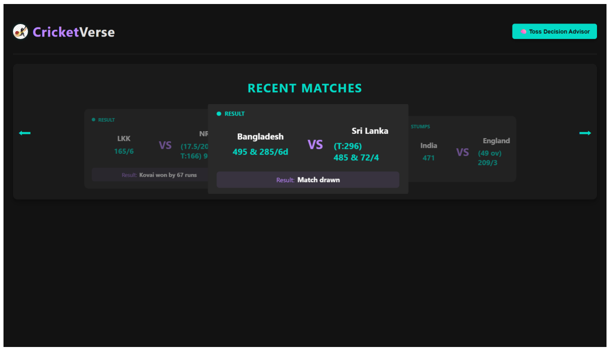
Fig. 4: Process Flow Diagram

V. Implementation

It is a complete analytics app of T20 cricket with attributes as invaluable features like live cricket score and stats, which is automatically updated in real-time, prediction of the second inning, and timer suggestion on the toss. The acquisition of the match data itself can be considered an objective-oriented task performed by utilizing Beautiful Soup, as well as Requests, with the help of which the desired data can be taken directly out of the databases of reliable sources, namely ESPNcricinfo. The resulting data is then run through python libraries like Pandas and NumPy allowing the ball to ball updates, in-depth player statistics as well as specific to venue data to be shown through a live ReactJS view. Such machine-learning models as Random Forest forecast results in the second inning, combining current game circumstances with the past trends. Furthermore, the toss advice system is based on the data about a venue, pitch, and weather in order to enable the strategic decisions. An intuitive user interface and the use of accessibility features like keyboard navigation and screen-reader support makes it easy to use and interesting even to user with disabilities.

A. *Live Score and Statistics Display:*

It is a complete real-time service that offers real time updates of the Twenty20 (T20) matches through live-scoring and statistics delivery in a deliberately heavy duty implementation. Runs, wickets, overs and other significant data points (like run rates and run rates) are retrieved and continuously delivered to the user on ball by ball basis. With this aim, a set of web-scraping tools, with Beautifu Soup and Requests being the most prominent, aims to collect useful data of good quality the sources of which can be trusted, in particular, ESPNcricinfo, and then display it within a dynamically functional front-end interface implemented using ReactJS. The system also provides a visualization of the individual player performance, full team overviews, and those trends relating to the venue so that the researchers, analysts, and spectators could discuss the dynamics of the match and stay active with the help of a convenient interactive interface.



RESULT
LKK 165/6 VS NR (17.5/20 T:166) 9
Result: Kovai won by 67 runs

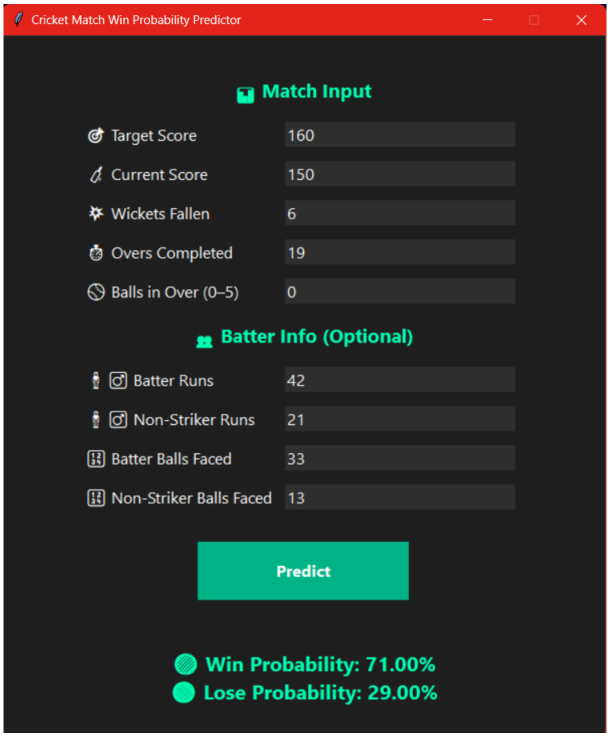
RESULT
Bangladesh 495 & 285/6d VS Sri Lanka (T:296) 485 & 72/4
Result: Match drawn

STUMPS
India 471 VS England (49 ov) 209/3

Fig. 5: Live Score and Statistics Display

B. *Second Inning Win Prediction:*

Second-Innings win Prediction Engine is a tool that utilizes ball by ball data and analyses the likelihood of the chasing team winning a T20 match, as the match proceeds in real-time. A number of wickets in hand, necessary run rate and individual performance of bowlers are fed to come up with the probabilities of winning. Using a Random Forest classification model programmed in Python using Scikit-learn and Pandas, the model combines past and current data collected by using the ESPNcricinfo web pages. Intuitive, interactive graphics have been represented at the front end based on ReactJS, which increases the engagement of spectators and assists in the process of strategic decision-making.



Match Input

Target Score: 160
Current Score: 150
Wickets Fallen: 6
Overs Completed: 19
Balls in Over (0–5): 0

Batter Info (Optional)

Batter Runs: 42
Non-Striker Runs: 21
Batter Balls Faced: 33
Non-Striker Balls Faced: 13

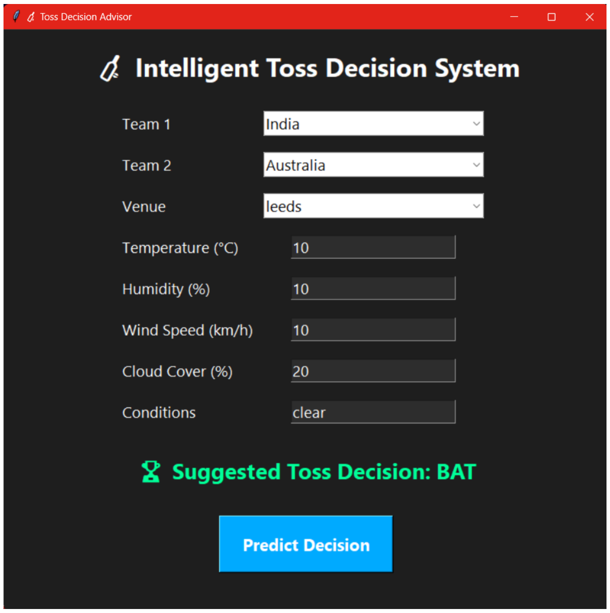
Win Probability: 71.00%
Lose Probability: 29.00%

Fig. 6: Second Inning Win Prediction

C. *Toss Decision Recommendation:*

The Toss Decision Recommendation System is an application that aims at helping to make the best toss calls in the T20 cricket environment based on the dynamic and constantly changing match and playing environments. Using supervised machine-learning processes the system determines whether a winning team ought to choose to bat or always to bowl. In order to train, the Random forest Classifier is applied on Python with Scikit-learn and Pandas, using a curated dataset, consisting of all previous T20 matches.

The models take into under consideration the venue-specific features, team constellation, real-time meteorological parameters, temperature, humidity, the speed of the wind, the presence of clouds, and the overall match conditions (clear or overcast). The weather information is not just taken manually but is pulled programatically via third party APIs (e.g. visual crossing) and adjusted to correlate with performance trend so as to be contextually relevant.



Intelligent Toss Decision System

Team 1: India

Team 2: Australia

Venue: leeds

Temperature (°C): 10

Humidity (%): 10

Wind Speed (km/h): 10

Cloud Cover (%): 20

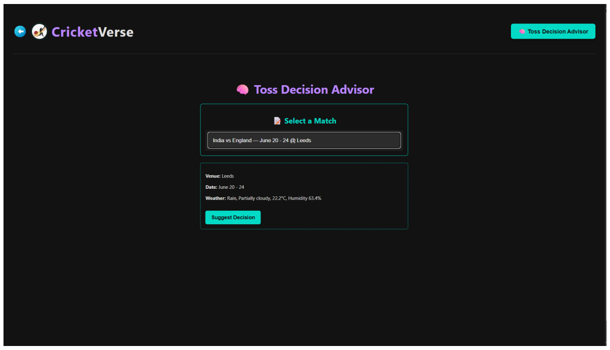
Conditions: clear

Suggested Toss Decision: BAT

Fig. 7: Toss Decision Recommendation

D. Toss Decision Recommendation UI:

The current paper proposes the development of a user interface, which is built on behalf of ReactJS and receiving match-related contextual information through dropdown lists and text boxes. These inputs are then analyzed to come up with a real-time advisory recommendation on the decision to be taken when it comes to the toss. This kind of functionality is demonstrated to enhance the reactive capabilities of coaching personnel, analysts, and spectators, hence the enhancement of tactical involvement and the experience of the entire fan base.



Toss Decision Advisor

Select a Match

India vs England — June 20 - 24 @ Leeds

Venue: Leeds

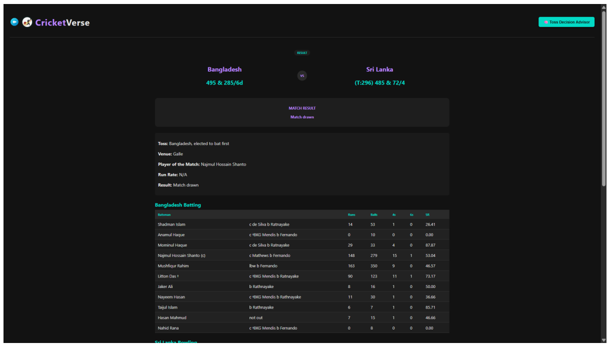
Date: June 20 - 24

Weather: Rain, Partially cloudy, 22.2°C, Humidity 63.4%

Fig. 8: Toss Decision Recommendation UI

E. Generate Match Result Statistics:

It is an all inclusive database of finished T20 games, capturing match margins, player and team performance and total team scores at each ground. Records of past data are registered in MongoDB database and further analyses are performed with the help of Python libraries like Pandas to identify patterns and trends[1]. These data are presented on an interactive ReactJS interface which provides the results in a structured, understandable form, thus enabling a user to learn the results of matches and the results of the performance of teams.



Bangladesh: 495 & 285/6d

Sri Lanka: (T:296) 485 & 72/4

MATCH RESULT
Match drawn

Toss: Bangladesh, elected to bat first

Venue: Galle

Player of the Match: Najmul Hossain Shanto

Run Rate: N/A

Result: Match drawn

Bangladesh Batting

Batsman		Runs	Balls	4s	6s	SR
Shadman Islam	c de Silva b Ratnayake	14	53	1	0	26.41
Anamul Haque	c †BKG Mendis b Fernando	0	10	0	0	0.00
Mominul Haque	c de Silva b Ratnayake	29	33	4	0	87.87
Najmul Hossain Shanto (c)	c Mathews b Fernando	148	279	15	1	53.04
Mushfiqur Rahim	lbw b Fernando	163	350	9	0	46.57
Litton Das †	c †BKG Mendis b Ratnayake	90	123	11	1	73.17
Jaker Ali	b Rathnayake	8	16	1	0	50.00
Nayeem Hasan	c †BKG Mendis b Rathnayake	11	30	1	0	36.66
Taijul Islam	b Rathnayake	6	7	1	0	85.71
Hasan Mahmud	not out	7	15	1	0	46.66
Nahid Rana	c †BKG Mendis b Fernando	0	8	0	0	0.00

Sri Lanka Bowling

Fig. 9: Generate Match Result Statistics

VI. Conclusion

It is a data-centric evolution of the T20 day-match and plays with advantages of real-time analytics, predictive modelling and user-friendly interface design, all complementing each other, seamlessly. By the use of Python, machine-learning algorithms, MongoDB, web scraping, and reactJS front-end, the system can provide actionable information, such as: in-progress Tracking of live matches; estimation of winning probability; and recommendations on strategic tosses. This functionality goes beyond reinforcing the captivation of a fan but also provide them with useful tools so that analysts and cricket-governing authorities can make meaningful decisions. Focusing on accessibility and usability, it becomes one of the platforms of smart, participatory, and all-embracing cricket analysis of the new digital generation.

VII. Future Enhancements

A. Player Impact Index Performance Forecasting:

In the future, it would be good to include the component of a Player Impact Index and Performance Forecasting. A machine-learning system would calculate a Player Impact Score using both current data, e.g. during the game, and historic data, e.g. past matches, form, opposition, pitch conditions, thus providing a complete indication of the individual contribution. Moreover, the system would help come up with an informed prediction of how a certain player is likely to perform in the ongoing match or innings and this would provide important information on not only the players who are likely to make or break the game but also the game changers.

B. Fan Engagement Personalization Features:

Another developmental opportunity entails the integration of the Fan Engagement and Personalization Features. By utilizing all these features, users had an opportunity to subscribe to the athletes or clubs and get notifications customized to individual needs as well as consult personal statistics profiles. Live poll, as well as in-stream chat, are considered to be interactive features that can also improve the interaction of spectators during the competitions.

References

[1] A. Yadav, D. Kumar, and S. Verma, “Winner Prediction in One-Day International Cricket Matches Using Machine Learning Framework: An Ensemble Approach,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2021.

[2] A. D. Bhatt and K. P. Patil, “Prediction of IPL Match Outcome Using Machine Learning Techniques,” *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 5, no. 6, pp. 1-5, 2021.

[3] R. Sharma and A. Tiwari, “Prediction of IPL Match Outcome Using Machine Learning Techniques,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 7, no. 6, 2020.

[4] R. Jain and M. Rathi, “Prediction of IPL Match Outcome Using Machine Learning Techniques,” *International Journal of Computer Applications*, vol. 178, no. 7, pp. 12–16, 2019.

[5] Corporate Finance Institute, “Ensemble Methods – A Machine Learning Technique,” [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods. [Accessed: May 2025].