iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75686

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

Decentralised Cricket Strategy Analysis System

Dr. Kavita A. Sultanpure[1], Nidhi Karkera[2], Prajakta Kumbhare[3], Mandar Mahapure[4], Jay Kumawat[5], Ameya Pathak[6]
Department of Information Technology, Vishwakarma Institute of Technology, Pune, 411037, Maharashtra, India

Abstract: ***The integration of data science methodologies, which provide deep insights into player performance, match dynamics, and predictive analytics, has reshaped the cricket strategy landscape. At the same time, blockchain embodies a paradigm shift toward secure, transparent data management in many sectors. This paper discusses possible synergies between these two domains in the context of cricket strategy management. Concretely, it describes the development of a Python-based application that combines data-driven generation of cricket strategies from historical match data with their secure storage and sharing using a simulated multi-node blockchain. After touching upon the architecture of the application regarding role-based access for coaches and players, data analysis towards strategizing, and persistence through blockchain, the results also highlight the potential of this integrated approach to enhance security, transparency, and accessibility of strategic information in cricket, therefore opening further avenues in the domain of sports analytics and technology.***

Keywords: Cricket Strategy, Data Science, Blockchain Technology, Decentralized Ledger Technology, Sports Analytics, Python-based Application, Multi-Node System, User Authentication, Data Analysis

I. INTRODUCTION

The landscape of competitive sports has dramatically changed, and especially so in cricket, where every decision is increasingly based on empirical evidence rather than intuition. While data science may be used to identify the best tactics and player matchups, there remain gaps in securely and transparently managing these derived strategies within the ecosystem of the team. Traditional storage often compromises integrity and controlled accessibility for this sensitive proprietary information.

The proposed system herein describes a development in which strategy generation, through a data analytics engine implemented in Python, is integrated with a simulated blockchain ledger for secure storage and verifiable sharing.

The main contributions of this work are:

1) The system will be a new development that will harness data science techniques in rendering context-aware cricket strategies using historical data of different matches.
2) Investigation and implementation of a simulated blockchain to create a secure, transparent, and auditable record for storing and disseminating these strategies.
3) An analysis of the possible benefits and limitations of this integrated approach within the wider context of sports analytics and information management.
4) The integration of data-driven strategy creation and DLT provides a solution that meets the identified need within cricket for secure and trusted management of strategy, and is extendable to other domains that manage sensitive strategic information.

The rest of the paper is organized as follows: Section 2 reviews relevant literature on data science in cricket and blockchain technology. Section 3 outlines the technologies used. Section 4 describes the methodology, including data processing, strategy generation, and blockchain implementation. Section 5 presents the results and features of the application. Section 6 concludes the paper.

II. LITERATURE REVIEW

[1] This paper titled "CRICKET SCORE DATA ANALYSIS" by Baji et al. (2023) statistically controls charts and graphical representations that can be used to analyze the performance of the ICC players, to obtain better tactical decisions according to their strengths and weaknesses. This study applies statistical quality control charts to statistically analyze the batting scores earned by ICC players in individual control charts and moving range control charts. These findings highlight the importance of using data analysis in gaining a competitive advantage in cricket.

[2] This paper titled "Cricket Match Analytics and Prediction using Machine Learning" by Dalal et al. (2024) applies different types of machine learning models to predict the outcomes of a cricket match based on the score that needs to be chased and other player-related features. To predict the second innings scores, the models used include Random Forest, Support Vector Machine, Logistic Regression and Naive Bayes Classifier. The results also show that tree-based modeling, particularly Random Forest, has a greatly higher classification accuracy for cricket.

[3] This paper titled "Artificial Intelligence and Data Analytics in Cricket" by Nazeer et al. (2022) examines the application of artificial intelligence and data analytics in the sport of cricket in player selection, tactical strategies, and performance prediction. This tutorial examines the application of artificial intelligence and data science techniques in cricket to analyze player performance, devise strategies for the game, and predict player and team performance based on past data and contextual information for the betterment of team performance.

[4] This paper titled "How Data Science is Transforming Cricket" by Shah and Saeed (2024) highlights the increasing impact of data science on cricket, including video analysis, decision-making processes, and player evaluation, which can enhance the game. The paper discusses how data science is being used to analyze player performance, understand team strengths and weaknesses, and make data-driven decisions to improve gameplay and fan experience.

[5] This paper titled "Predicting Cricket Outcomes: A Comparative Analysis of Machine Learning Models for Optimal Accuracy" (2025) discusses the construction of a cricket match outcome prediction model using machine learning and match data, a comparison of prediction performance of various machine learning models, and the development of optimal algorithms for predicting the outcome of cricket matches. The research seeks to provide predictive perceptions that can inform planned decisions in cricket match situations.

titled "Shot-Net: A convolutional neural network for classifying different cricket shots" by Foysal et al. (2018) proposes a new neural network architecture for recognizing cricket shots within video footage. The model's primary objective is to assist in post-match analysis by automatically classifying different types of cricket shots within video frames. Utilizing a convolutional neural network (CNN), the model has the potential to provide perceptions into player techniques and game strategies.

[7] This paper titled "Cricket Match Analytics Using the Big Data Approach" (2021) presents the latest advancement in predicting the score and winner of the game of cricket using big data analytics and machine learning models. Due to the availability of huge amounts of data today, we investigate the use of machine learning linear regression models with and without the big data framework Spark ML to predict cricket team scores.

[8] This paper titled "Cricket Data Analysis and Visualization using Python and Power BI" (2024) demonstrates how Python and Power BI can be used to create dashboards that collect, process, and display data with the goal of gaining perceptions into strategies for playing cricket. The paper also discusses web scraping of data , Python, Pandas, and Power BI, to analyze T20 World Cup to provide data on player selection and strategy formulation.

[9] This paper titled "Predict the Match Outcome in Cricket Matches Using Machine Learning" by Rehman et al. (2022) investigates the use of several machine learning classifiers to predict the outcomes of ODI cricket matches, finding that tree-based models perform well. The study applies machine learning algorithms such as Naïve Bayes, SVM, KNN, Random Forest, and Decision Tree to predict ODI match outcomes, concluding that Random Forest and Decision Tree perform well in terms of accuracy, precision, and recall.

[10] This paper titled "AI-Driven Analysis of Cricket Match Trends under Varying Environmental Conditions" (2025) examines the influence of the weather conditions and historical data on match outcomes using deep learning algorithms. This study proposes the use of CNN and LSTM models to determine the effect of weather on the outcome of cricket matches, team performance, and individual performance, and validates that weather data improves prediction accuracy.

[11] This paper titled "Cricket Score Prediction Using Machine Learning" by Preetham HK et al. (2023), This research builds a machine learning model to predict the cricket score based on the previous cricket match and other factors. The aim of this study is to create a prediction system that takes data of matches played in the past and makes predictions of future match events such as final scores and results using machine learning algorithms.

[12] This paper titled "Applications of Machine Learning in cricket: A systematic review" (2022) presents a systematic review of research work carried out in the last two decades, on the applications of machine learning in cricket across a range of topics including outcome prediction, player performance, analytics, injury prediction and detection, and optimization. The work, covering 2001-2021, discusses the expanding interest in cricket data analytics driven by the large amount of data and the evolution of ML technology.

[13] This paper titled "Predicting Players' Performance in One Day International Cricket Matches Using Machine Learning" (2018) attempts to predict individual player performance in ODIs using supervised machine learning techniques. Predicting the number of runs scored by a batsman and wickets taken by a bowler in an ODI match is done by analyzing the characteristics and statistics of the players using various supervised machine learning techniques.

[14] This paper titled "Relative Analysis and Performance of Machine Learning Approaches in Sports" by Ishwarya et al. (2021) compares the various machine learning techniques being applied within the context of sports data. The study compares various machine learning algorithms and their performance in the sport data analysis domain.

[15] This paper titled "Data Science and AI in Cricket: Revolutionizing Performance Analysis and Decision-Making" (2023) highlights the use of data science and artificial intelligence in cricket ranges from analyzing the performance of players to optimizing strategies. Their research shows how Data Science and AI benefit cricket teams in areas such as player performance analysis, game strategy optimization, and injury prevention.

III. METHODOLOGY

The development of this cricket strategy management application wasn't just coding—it followed a focused, four-step methodology: data processing, strategy generation, simulated blockchain integration, and user interface design.

1) Data Acquisition and Preprocessing: Our starting point was the cleaned_match_data.csv file, the backbone of our historical analysis. This step was crucial: the raw data needed to be absolutely ready.

We used specific helper functions to guarantee that every numerical metric—from Strike Rate to Runs scored—was correctly parsed and formatted for analysis, eliminating any inconsistencies or non-numeric characters. Once the data was reliable, the system dynamically filtered this historical record based on the user's selection: the specific batsman, bowler, and venue. This immediate focus allowed us to calculate highly relevant statistics from the subset, including:

- Head-to-Head Performance (the ultimate matchup statistics).
- Venue Performance (how each player typically fares at that specific ground).
- Overall Form (general match statistics for a baseline assessment).

This precision filtering ensured the strategies were always grounded in the most relevant historical context.

Match Context: Match_no, Date, Venue

Batsman Performance/Style: Batsman_Name, battingStyle_batsman, Batting_Position, Runs_batsman, Balls, Strike_Rate, Dismissal, description_batsman, Winner_batsman

Bowler Performance/Style: Bowler_Name, bowlingStyle_bowler, playingRole_bowler, Overs, Runs_bowler, Wickets, Economy, description_bowler

Data Cleaning for Numeric Integrity: The final step of data preprocessing involved the cleaning of all performance metrics containing numeric values (eg. Strike_Rate, Runs_batsman). The raw data usually contained non-numeric characters (eg. symbols, trailing spaces), so they could not be converted to float types. The next helper function, clean_numeric_column, could be added to clean out any extraneous characters that are not digits or decimals, allowing for a subsequent summation.

```
def clean_numeric_column(value):
        """Cleans numeric columns by removing non-numeric characters and converting to float."""
        if pd.isna(value):
            return None
        try:
            # Filter to allow only digits, decimal point, and negative sign
            cleaned_value = ''.join([ch for ch in str(value) if ch.isdigit() or ch == '.' or ch == '-'])
            if cleaned_value == '-' or cleaned_value == '':
                return None
            return float(cleaned_value) if cleaned_value else None
        except:
            return None
```

2) Strategy Generation: The core of the strategy generation process is contained within the generate_detailed_strategy function, which takes as input the loaded and filtered DataFrame, the selected batsman, bowler, and venue. The function analyzes the relevant historical data with the pre-existing insights in order to compile a detailed cricket strategy. This strategy will be compartmentalized into sections such as Head-to-Head Analysis, Venue Performance, Style Matchup, Tactical Advice, Form Analysis, and Famous Player Insights. Finally, the function returns the sections compiled into a dictionary, which will later be displayed to the user through the Streamlit user interface.

3) Data-Driven Insight Generation: The key analysis capability of the system can be thought of as the generate_detailed_strategy function, using rich columnar match data to perform dynamic multi-layered statistical analysis. This function processes information across four primary dimensions - H2H, Venue-specific records, Player Styles, and overall Form.

Dynamic Data Filtering and Aggregation : It utilizes Pandas DataFrame filtering and aggregation capabilities for deriving important matchup metrics dynamically.

The Head-to-Head (H2H) performance is isolated by filtering records where both the Batsman_Name and Bowler_Name match the current selection.

The snippet below demonstrates the critical calculations for H2H runs, balls faced, and the resulting Strike Rate (SR):

Python

```
# Filter data for Head-to-Head (H2H) matchup
h2h_data=df[(df['Batsman_Name'] ==batsman) & (df['Bowler_Name'] == bowler)].copy()

if not h2h_data.empty:
  # Aggregate performance metrics
h2h_runs = h2h_data['Runs_batsman'].sum()
  h2h_balls = h2h_data['Balls'].sum()
h2h_dismissals= h2h_data['Dismissal'].count()
  # Calculate Strike Rate (SR)
  h2h_strike_rate = (h2h_runs / h2h_balls) * 100 if h2h_balls > 0 else 0
  # Identify most common dismissal type using Pandas mode()
  if 'Dismissal' in h2h_data.columns and not h2h_data['Dismissal'].dropna().empty:
    common_dismissals = h2h_data['Dismissal'].mode()
    # ... (further logic to format output)
```

Similar aggregation methods are applied to assess Overall Form (calculating averages for Strike_Rate, Wickets, and Economy) and Venue Performance for both players, establishing a comprehensive analytical baseline.

Tactical Logic Implementation : The system translates the derived statistics and fundamental player styles into a qualitative advice through a rule-based inference engine. This is implemented using conditional logic based on the extracted battingStyle_batsman and bowlingStyle_bowler attributes.

The following snippet illustrates the deterministic mapping of player styles to tactical recommendations:

Python

```
# Extract styles (using .mode() for robustness)
batsman_style = batsman_all_data['battingStyle_batsman'].mode().iloc[0]
bowler_style=bowler_all_data['bowlingStyle_bowler'].mode().iloc[0]

# Tactical Advice Logic Snippet
if "Right-hand" in batsman_style and "Fast" in bowler_style:
  style_and_against_strategy += "- Focus on playing the ball under your eyes and being prepared for short-pitched deliveries. Be decisive against the swinging ball.\n"
elif "Left hand" in batsman_style and "Spin" in bowler_style:
  style_and_against_strategy += "- Facing spin will see the ball turning away. Play close to the body, watch the ball carefully, and look to score square of the wicket when opportunities arise.\n"
# ...
```

This hybrid methodology ensures the generated strategy is data-backed and practically useful for on-field decision-making by integrating statistical inferences with qualitative, domain-specific rules.

4) Simulated Blockchain Implementation: The system implements a simulated, multi-node blockchain ledger for tamperproof storage and verification of generated strategies.

- Block Structure: Each strategy is encapsulated in a Block that includes its data, index and timestamp, and two important cryptographic fields: the previous_hash and its calculated hash.

- Immutability: The calculate_hash function uses SHA-256 to generate a unique digest of the block's content, Tampering becomes immediately obvious—the system simply refuses to play along..
- Decentralization: The ledger is managed across a simulated multi-node network, where each node stores its chain state locally in a node-ID-specific JSON file.
- Integrity: The is_block_valid and is_chain_valid functions strictly enforce cryptographic linkage. It verifies that H_{i-1} matches the previous_hash of Block_i.
- Consensus Simulation (The Network Vote): When a new strategy is created, it's broadcast instantly. Our Consensus Rule is strict: for that strategy to be finalized, every single configured peer node must validate and accept it. If even one node flags an issue, the block doesn't stick.

This design demonstrates the use of a Distributed Ledger Technology (DLT) for verifiable, transparent, and secure management of sensitive data.

5) User Interface and Workflow (RBAC and DLT Integration): Access to the system is provided through a web interface built using the Streamlit framework, using a strict Role-Based Access Control (RBAC) model. Authentication and Access Control: The workflow begins with an authentication gate separating two types of users:

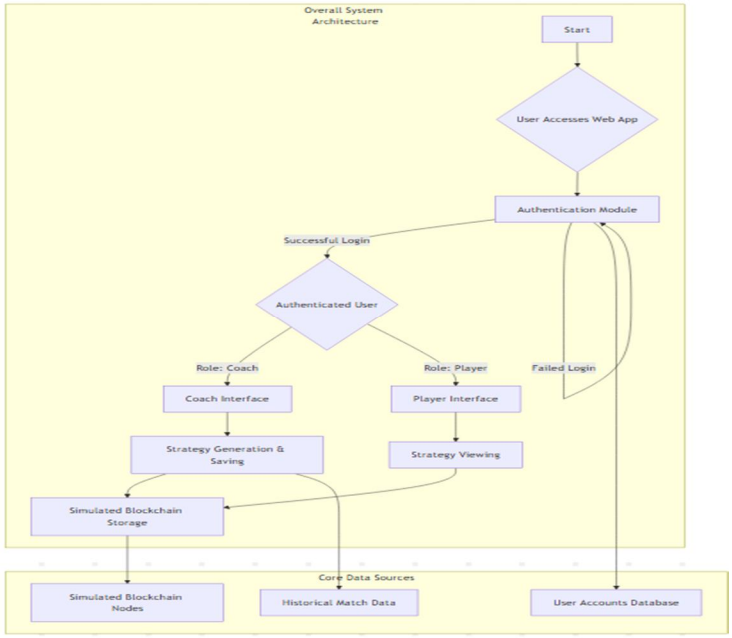
6) Coaches possess comprehensive privileges:

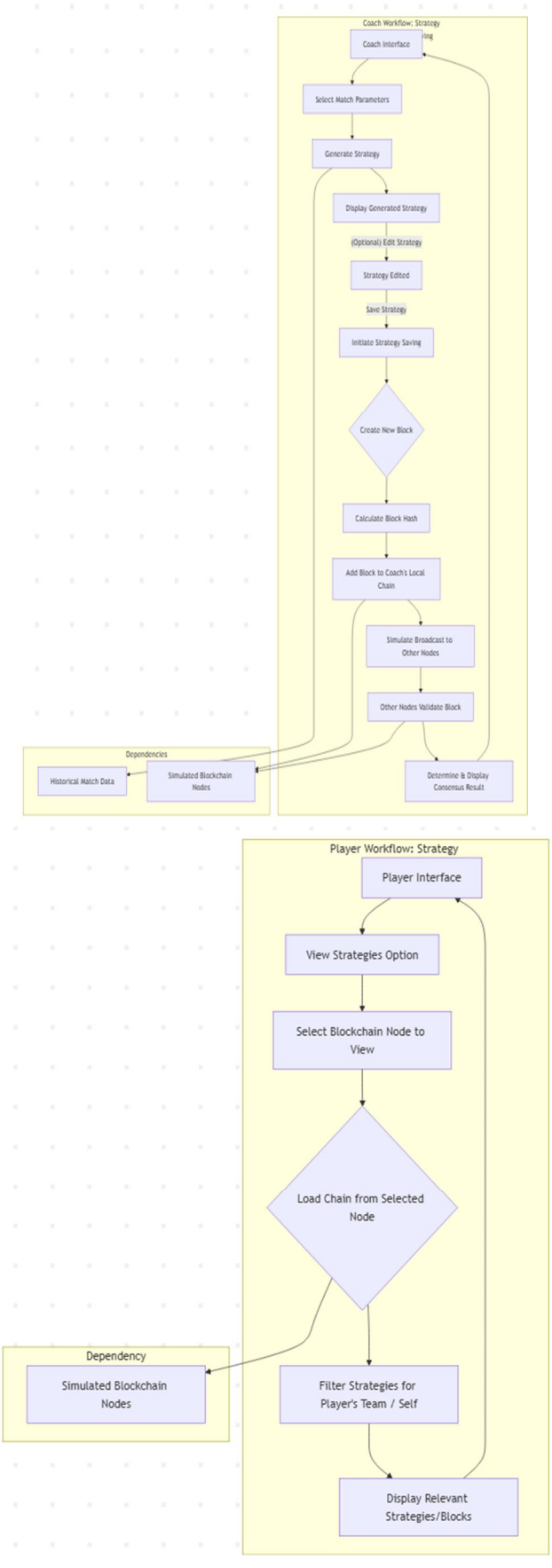
- Strategy Generation: Select the input parameters like batsman, bowler, venue, and run the data-driven analysis module.
- Data Persistence (DLT): When someone saves a generated strategy, the add_to_chain function simulates the adding, broadcasting, and achieving consensus of the entire blockchain (see below).
- Management: Inspect details of the simulated multi-node blockchain network (e.g. number of blocks, chain validity) and raw block data.

7) Players have restricted, read-only access:

- Strategy Retrieval: Retrieve strategies specifically applicable to that profile (i.e., when the Batsman or the Bowler becomes the selected player) from the chain at the simulated node.
- No Mutability: They cannot create, edit, or store new strategies for themselves.

System State Visualization: The graphical user interface is updated in real time, with the total number of blocks present in the chain and its validity shown for each simulated node, to enable real-time visualization of the entire state of the simulated DLT network.




Coach Workflow: Strategy
Coach Interface
Select Match Parameters
Generate Strategy
Display Generated Strategy
(Optional) Edit Strategy
Strategy Edited
Save Strategy
Initiate Strategy Saving
Create New Block
Calculate Block Hash
Add Block to Coach's Local Chain
Simulate Broadcast to Other Nodes
Other Nodes Validate Block
Determine & Display Consensus Result
Dependencies
Historical Match Data
Simulated Blockchain Nodes
Player Workflow: Strategy
Player Interface
View Strategies Option
Select Blockchain Node to View
Load Chain from Selected Node
Filter Strategies for Player's Team / Self
Display Relevant Strategies/Blocks
Dependency
Simulated Blockchain Nodes

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

IV. TECHNOLOGIES USED

The cricket strategy management application uses an array of technologies including data analysis and a blockchain simulation environment to perform its function.

Python was chosen to be the primary language for the code base because it is versatile and supports many libraries for data science and the web.

Streamlit, an open-source Python library, acts as the main framework toward the interactive user interface. Streamlit's designers designed it so it became popular for data scientists who may lack deep training or experience with front-end web development since they can quickly create data-driven web applications through pure Python code. Streamlit has user interface components that include buttons, dropdowns, text inputs, sliders, and other inputs. The components let the user interact with the application. The application running in the web browser updates in real time whenever the code or data gets modified.

In order to perform pre-processing of the cricketing data, and the subsequent analysis of it, the Pandas library, for Python, has been used. This library provides data structures and operations to manipulate data, by filtering, sorting, grouping and processing large data sets, to produce information which is later used to generate strategies..

To ensure the security of user accounts, the bcrypt library is integrated for hashing passwords during the authentication process.Bcrypt is a hashing function that is specifically designed for securely hashing passwords. It uses salting (adding random data to an input of a hash function) to create a hashed password. The purpose of salting is to make it harder for a hacker to gain back the password from the hashed password. In addition to this work factor, bcrypt also uses a one-way hashing function, increasing the difficulty of recovering the original password from the hash to a minimum.

A. Block Integrity via SHA-256 Hashing

Each block contains a cricketing strategy and is linked to the previous block via the SHA-256 hash function to ensure that the entire block cannot be modified without detection. The hash is created from the contents of the block (excluding the hash) serialized with the keys in deterministic order, following blockchain norms to ensure consistency across node implementations.

The hash calculation function is the backbone of block integrity:

```
def calculate_hash(block):
    """Calculates the SHA-256 hash of a block."""
    block_copy = block.copy()
block_copy.pop("hash", None)
#Sort keys for deterministic hashing across nodes
    block_string = json.dumps(block_copy, sort_keys=True).encode()
    return hashlib.sha256(block_string).hexdigest()
```

JSON (JavaScript Object Notation) is used as a data storage format for the user's accounts and the simulated blockchain in the local files. Its light and human-readable structure makes it suitable within the scope of the application.

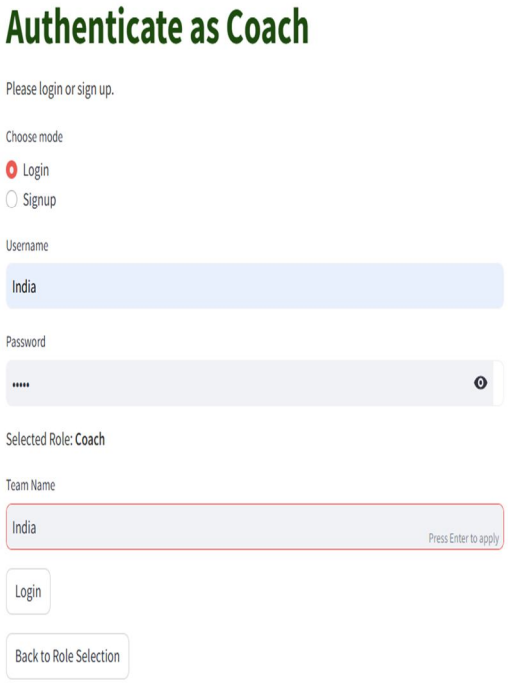
Finally, the Simulated Blockchain Concepts are used in the application as a means to explain decentralized ledger technology, including data structured into blocks, each containing the cricket strategy data with metadata such as a timestamp, index, and cryptographic hashes. Each block will then link together in a chain through the use of the hash of the previous block, creating a chronological and tamper-evident record. Each block's content is hashed using SHA-256 for integrity. The simulation includes a multi-node architecture whereby each simulated node maintains its own copy of the blockchain in separate JSON files. Basic functions that validate the integrity of individual blocks and the entire chain are implemented, along with a simulated broadcast mechanism that mimics the propagation of new blocks across the network. There is also a rudimentary consensus mechanism emulated to reflect how, at the core of a real blockchain network, many nodes agree on the validity of new additions to the ledger.

V. RESULTS AND DISCUSSION

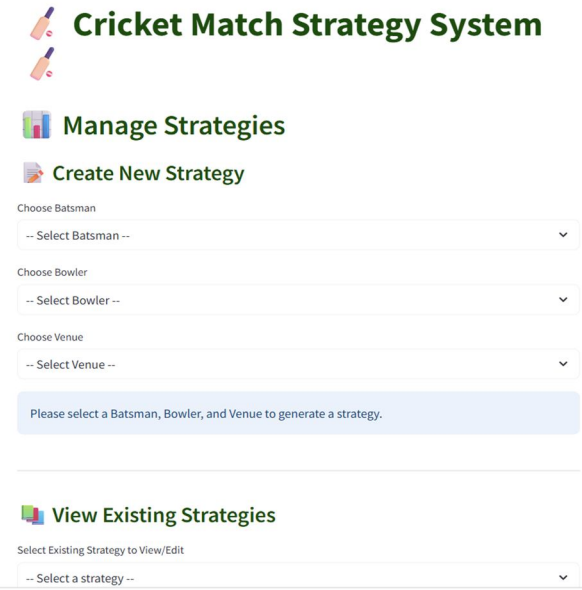
1) Application Overview and User Management: The application developed is a Python-based tool for the management of cricket strategy, integrating data science with a simulated blockchain. The application offers a user-friendly interface through which coaches can quickly create and manage strategies, while players can easily access relevant strategic information. It integrates a sound role-based access control system wherein sensitive strategy generation and editing can only be conducted by the coach, allowing players to only view information relevant to their roles and team.

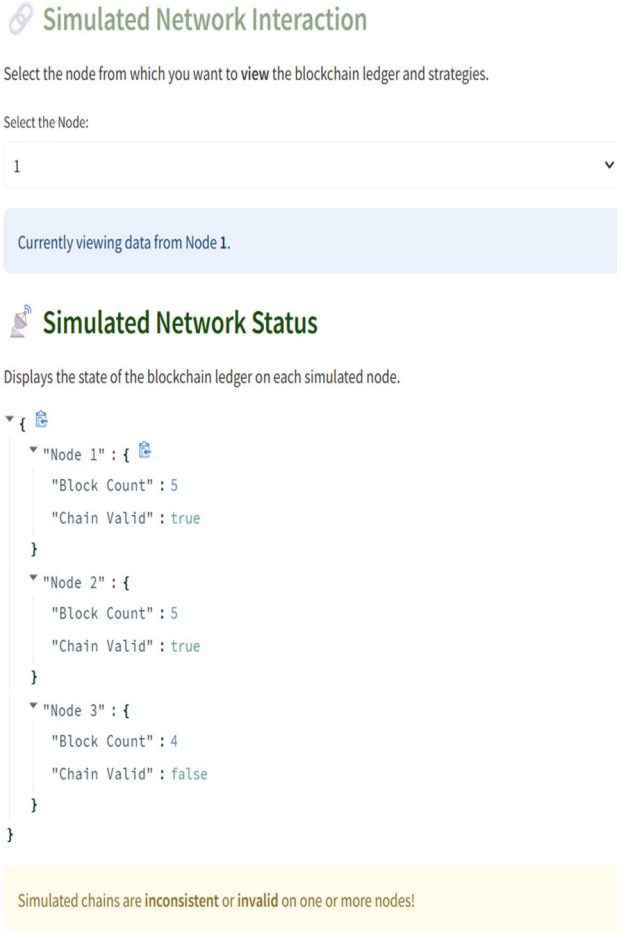
2) Data-Driven Strategy Generation (AI Aspect): The application uses a data-driven, strategy-generating engine operating on historic cricket data on user-selected batsman, bowler, and venue parameters. Generated strategies would be provided in tabulated form, including several sections of analysis, such as "Head-to-Head Analysis" and "Venue Performance." For example, the choice made about a batsman, a bowler, and a venue would automatically trigger the processing of past match data to get information about how two teams or players have fared in the past, their performance trend at that venue, amongst other important statistics, thereby developing evidence-based recommendations for coaches targeting specific match-ups and conditions.
3) Blockchain Simulation for Strategy Storage and Sharing (Core Innovation):The core novelty here is the simulation of a multi-node blockchain for the decentralized storage and sharing of these generated strategies.
4) Block Creation and Chaining: Every time a coach decides to save a strategy, it gets converted into a new block and is added to the blockchain in their simulated node. This involves computing the cryptographic hash of the new block and securely linking the hash to the hash of the previous block in the chain.
5) Simulated Broadcast and Validation: The application simulates the broadcasting of this newly minted block across other nodes in the simulated network. Each receiving node proceeds to verify the newly incoming block against predefined validity criteria that guarantee its integrity and proper linkage.
6) Consensus Mechanism: A validation process such as that among all simulated nodes will determine whether a consensus has been reached on the acceptance of this new block within the distributed ledger.
7) Transparency and Observability: This simulation provides real-time visibility over the status of each node's blockchain through this interface, their current block count, and the overall validity of the chain. Users are also allowed to inspect the raw data of each separate block on different simulated nodes, offering a concrete demonstration of decentralized data storage and sharing.

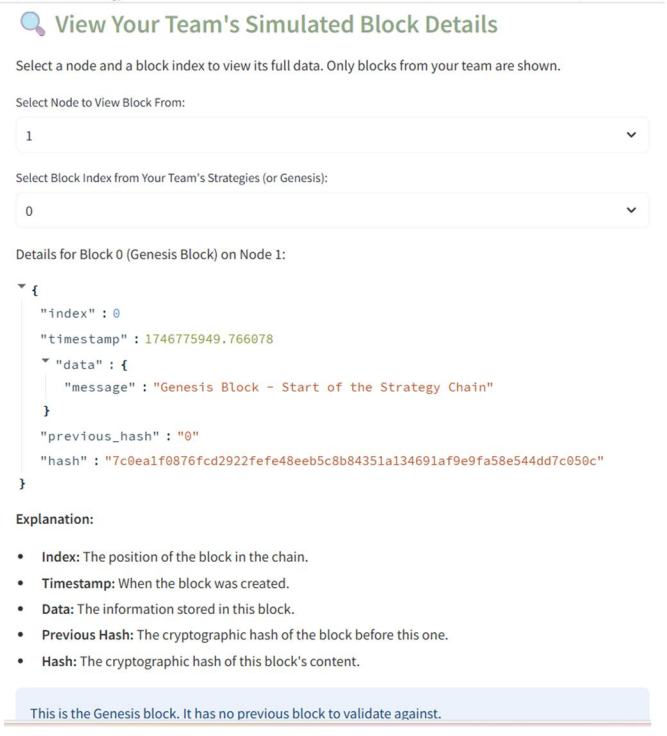
The simulation allows for the demonstration of basic concepts of blockchain technology such as the generation and linking of blocks, cryptographic hashing, block validation, and a basic consensus algorithm, by managing cricket strategies.

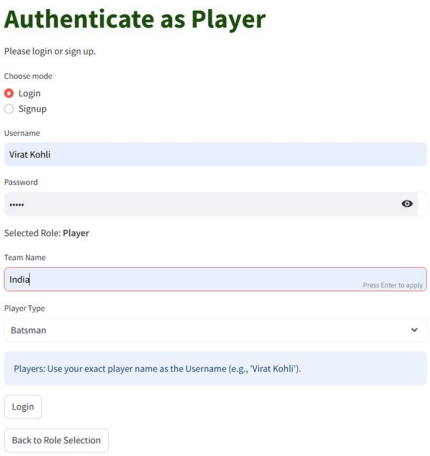


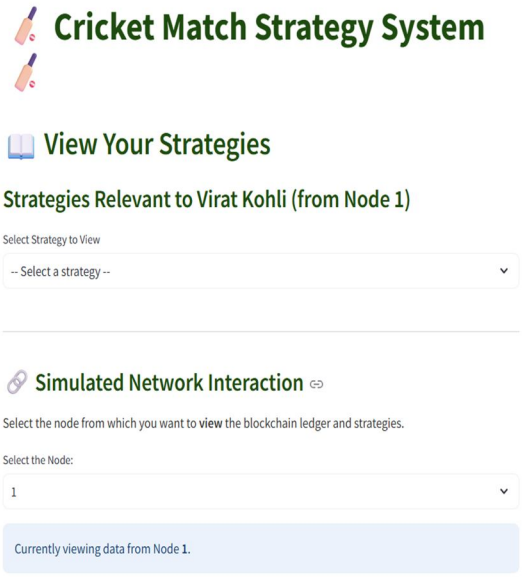
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com











VI. CONCLUSION

In this research we combined data science techniques for creating cricket strategies with a simulated blockchain for secure data management. The Python based application developed in this project serves as a practical example of this idea, offering role based access for coaches and players. It generates data driven strategies using past cricket data and securely storing them in a tamper proof blockchain simulation. This approach shows how technology can make managing team strategies safer and more efficient, offering an improvement over traditional methods.

VII. FUTURE SCOPE

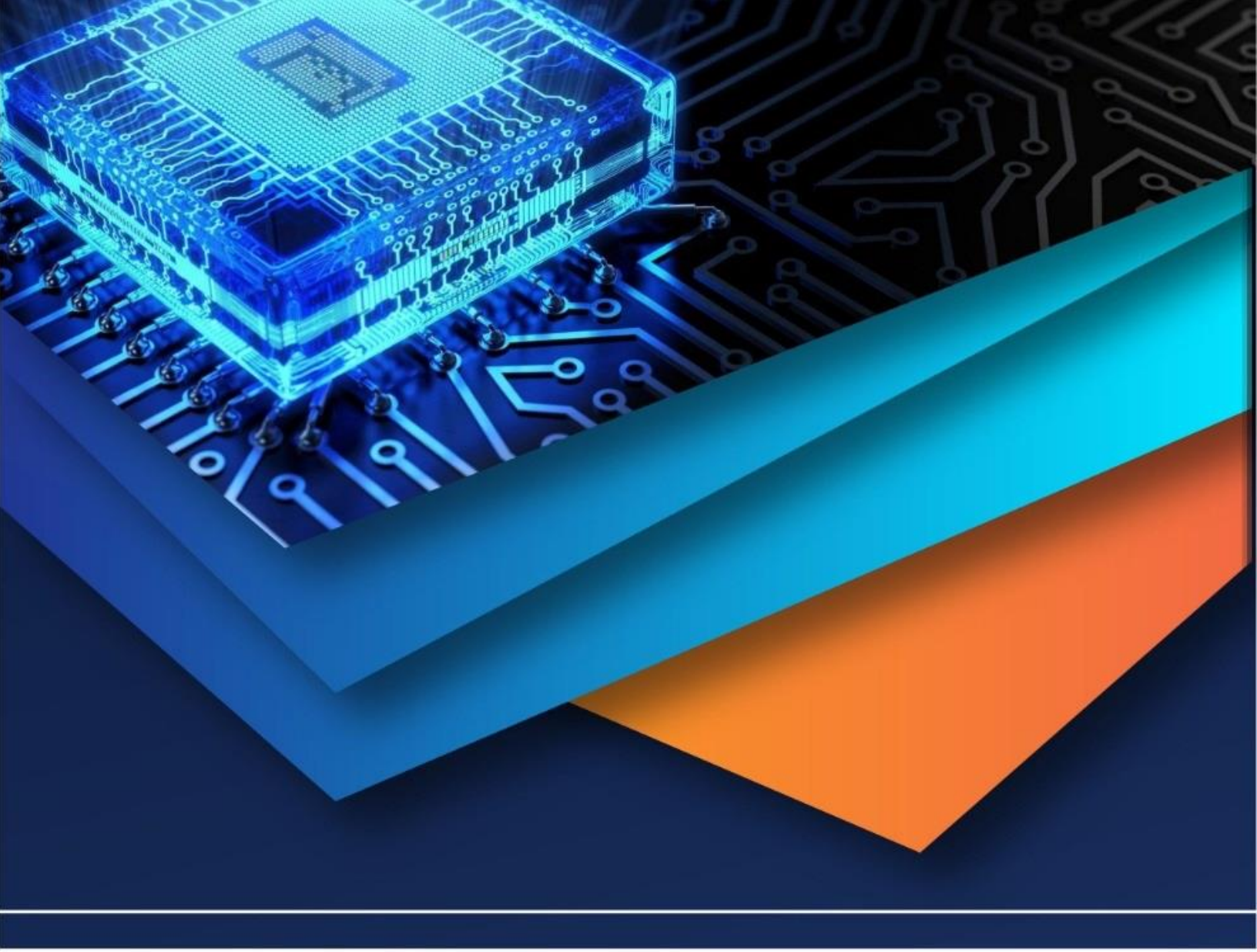
Future work can build upon this:

1) Integrate real-time data streams to enable live strategy updates during ongoing matches.
2) Use advanced data analysis and machine learning techniques—such as deep learning and reinforcement learning—to create more refined strategies and uncover deeper performance patterns.
3) By connecting the system to real-world cricket data APIs, we'll ensure it always has the latest match stats and player information.
4) We also want to study how blockchain-based strategy management actually affects a team's real performance on the field.
5) In the future, we'll look into applying this kind of AI and blockchain-powered system to other sports that rely heavily on data.
6) Finally, we'll dig deeper into how blockchain can be made more secure, scalable, and efficient when used for managing large amounts of sports data.

REFERENCES

[1] M. W. A. Baji, M. M. A. Baji, and M. Suhail, "CRICKET SCORE DATA ANALYSIS," *IRJMETS*, vol. 5, no. 5, pp. 1–6, 2023, doi: ((((https://www.doi.org/10.56726/IRJMETS41152))))
[2] P. Dalal, H. Shah, T. Kanjariya, and D. Joshi, "Cricket Match Analytics and Prediction using Machine Learning," *Int. J. Comput. Appl.*, vol. 186, no. 26, pp. 27–33, 2024, doi: 10.5120/ijca2024923744.
[3] [3] M. Tahir Nazeer, Hira Atta, and Muhammad Awais Saeed, "Artificial Intelligence and Data Analytics in Cricket," *Webology*, vol. 19, no. 2, pp. 10124–10131, 2022.
[4] Kavya Shah and Ammaar Saeed, "How Data Science is Transforming Cricket," *Harvard Data Sci. Rev.*, 2024, doi: [https://doi.org/10.1162/99608f92.g7fzhuzj](https://www.google.com/search?q=https://doi.org/10.1162/99608f92.g7fzhuzj).
[5] Predicting Cricket Outcomes A Comparative Analysis of Machine Learning Models for Optimal Accuracy. (2025). *IJAEM*.
[6] [6] M. Foysal et al., "Shot-Net: A convolutional neural network for classifying different cricket shots," in *2018 21st Int. Conf. Comput. Inf. Technol. (ICCIT)*, 2018, pp. 1–6, doi: 10.1109/ICCITECHN.2018.8631968.
[7] Cricket Match Analytics Using the Big Data Approach. (2021).
[8] Cricket Data Analysis and Visualization using Python and Power BI. (2024).

[9] [9] Z. U. Rehman, M. M. Iqbal, H. Safwan, and J. Iqbal, “Predict the Match Outcome in Cricket Matches Using Machine Learning,” *Manch. J. Artif. Intell. Appl. Sci.*, vol. 3, no. 1, pp. 1–7, 2022.
[10] AI-Driven Analysis of Cricket Match Trends under Varying Environmental Conditions. (2025).
[11] Preetham HK, Prajwal R, Prince Kumar, and Naveen Kumar, “Cricket Score Prediction Using Machine Learning,” *Int. J. Innov. Res. Technol.*, vol. 9, no. 8, 2023.
[12] Applications of Machine Learning in cricket: A systematic review. (2022).
[13] [13] Predicting Players' Performance in One Day International Cricket Matches Using Machine Learning. (2018).
[14] Ishwarya, S., & Gayathri, T. (2021). Relative Analysis and Performance of Machine Learning Approaches in Sports. *ACM Sports Data Journal*, *10*(4), 145–160.
[15] Data Science and AI in Cricket: Revolutionizing Performance Analysis and Decision-Making. (2023).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY