Journal of Innovative Technology, Vol. 8, No. 2, pp. 25-36, September 2026
http://doi.org/10.29424/JIT.202609_8(2).0003

Intelligent Multi-Target Learning System for Strategic Player Selection and Role Optimization: A Comprehensive Cricket Analytics Case Study

Muhammad Qanat Abbas [1], Maria Malik [2], Muhammad Aown Ali [3], Tariq Mahmood [4*]

ABSTRACT

Selecting suitable players is a crucial aspect of the decision-making process in any competitive sport. This study presents a machine learning-based intelligent decision support system, specifically analyzing cricket players and predicting their selection using a multi-output random forest classifier (MO-RFC). This study shows improved player selection decisions through predictive analysis. This system uses a multi-output classification model based on an MO-RFC for the prediction of the two most important outcomes, i.e., whether the player should be selected for a given match and his ideal batting position if selected. The model is trained using a player's past data, including features such as recent form, opponent-specific performance, ground-based performance, match type, overall runs, strike rate, and boundary percentage. A definite preprocessing pipeline takes care of data cleaning, categorical variable encoding, and advanced feature engineering to produce informative metrics such as current performance trends and context-specific averages. To improve user experience and accessibility of non-technical individuals, such as selectors and coaches, the system has an interactive graphical user interface (GUI) built with the Tkinter library of Python. This enables users to enter match conditions, e.g., opposition, ground, and format, and obtain real-time actionable predictions. The proposed solution bridges the gap between sports strategy and data science to provide data-driven, objective, and consistent player analysis. It significantly decreases the use of intuition-based judgements, optimizes team performance, and encourages cost-friendly, timely selection processes. The code of the project is available in a private GitHub repository and will be available upon request.

Keywords: Predictive modeling; machine learning; artificial intelligence; multi-output classification; random forest classifier; exploratory data analysis.

1. INTRODUCTION

Cricket is a game in which data-driven strategy plays a vital role in deciding the outcome of cricket matches [1-2]. It is a strategic and performance-driven sport that requires evidence-based decision-making for optimal team selection and match planning. In traditional cricket environments [3-4], player evaluation, batting order decisions, and team selection were often dominated by coaches' instincts, incomplete statistics, selectors' experience, and personal judgment, resulting in inconsistent and biased decisions. This resulted in poor selections due to a lack of data-driven analysis, misalignments between player ability and match situations, and missed strategic optimization opportunities. But advances in Data Science [5-6] and Artificial Intelligence (AI) [7] have opened the door for objective evidence-based strategies that provide improved decision-making in professional sports. The combination of machine learning (ML) and AI in cricket analytics is transforming the way coaches, team selectors, and analysts understand performance metrics, forecast results, and maximize player use. This study tries to bridge the gap between traditional intuition-driven choice and advanced prediction methods by suggesting an AI-based system for player performance analysis and selection prediction [8] that equips stakeholders with predictive functions rooted in the analysis of history and contextual performance modelling. The dynamic nature of cricket, having variable formats (T20I, ODI, Test) [9-11], ground conditions, and opposition strength, calls for responsive and intelligent decision-making. Form of the player, statistics of the ground, and history against the opposition have a major impact on individual and team performances. These factors, however, remain underutilized in the absence of structured analytical frameworks capable of processing and integrating them systematically.

AI and ML are emerging fields in many domains, such as nonlinear control systems [12-15], fluid mechanics [16-22], and medical systems [23-24]. The concept of using ML in cricket is not new, but most studies have been either predicting match outcomes or post-performance evaluation. Pasha et al. suggested an ML structure to select teams automatically based on past and contextual match information [25]. The study demonstrated improvement in the formulation of balanced teams but was unable to accommodate subjective factors such as player fitness or game-day temperament. Agrawal and Kandhway considered real-time

Manuscript received January 22, 2026; revised May 23, 2026; accepted July 23, 2026.

[1] Student, Department of Computer Science, COMSATS University Islamabad, Pakistan.

[2] Masters Student, Department of Economics, S3H, National University of Sciences and Technology, Pakistan.

[3] Masters Student, Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Taiwan, R.O.C.

[4*] Lecturer, (corresponding author), Department of Mathematics and Statistics, International Islamic University, Pakistan. (email: tariq.mahmood@iiu.edu.pk)

prediction of winners of ODIs through regression and time-series models [26]. Even though beneficial for broadcasters, their method relied greatly on the precision of real-time data streams. Passi and Pandey focused on improving prediction accuracy with decision trees and neural networks [27]. Their research added to the general format of applications but was challenged by computational scalability. Raajesh et al. combined K-Means Clustering, Support Vector Machines, and RFC to rank players and pick teams [28]. Their multi-layered approach generated rich insights but was not user-friendly in terms of interface design. Pathak and Wadhwa employed SVMs and RFC to predict ODI outcomes with excellent accuracy, but noise in input data and limited interpretability were major challenges [29]. Tahir et al. investigated a set of algorithms (RFC, decision trees, and Naïve Bayes) to examine cricket player performance, which indicates the scope of AI in sports analytics [30]. These studies highlight the feasibility of AI in cricket analytics, but they are usually limited in anticipating player-level choices, such as batting order, in a context-specific, multi-target format. This study transforms itself by incorporating a multi-output classification player-focused model, focusing on real-world usability through an easy-to-use interface.

The purposes of this study are:

- To create an intelligent system that can predict a player's selection for a match and recommend their best batting position using contextual past data.
- Implement an ML-based algorithm by deploying a multi-output prediction model to analyze and model performance metrics.
- Offering a scalable solution adaptable across teams, formats, and individual players.
- The results are validated through evaluation metrics like model prediction accuracy, precision, recall, F1 score, classification report, and confusion matrix.
- Create an intuitive and interactive GUI that allows for real-time forecasts for cricket analysts and selectors without necessitating programming skills.

Although the system was validated on Babar Azam's case study, a Pakistani batsman, it is structurally based to be generalizable for any other player or team, given appropriate performance data, making it a comprehensive tool in AI-powered sports analytics. Section 2 shows the Multi-Output Random Forest Classifier, Section 3 shows results and discussion of the system with Babar Azam's case study, and Section 4 shows the conclusion of the study. The graphical abstract of the study is depicted in Figure 1.

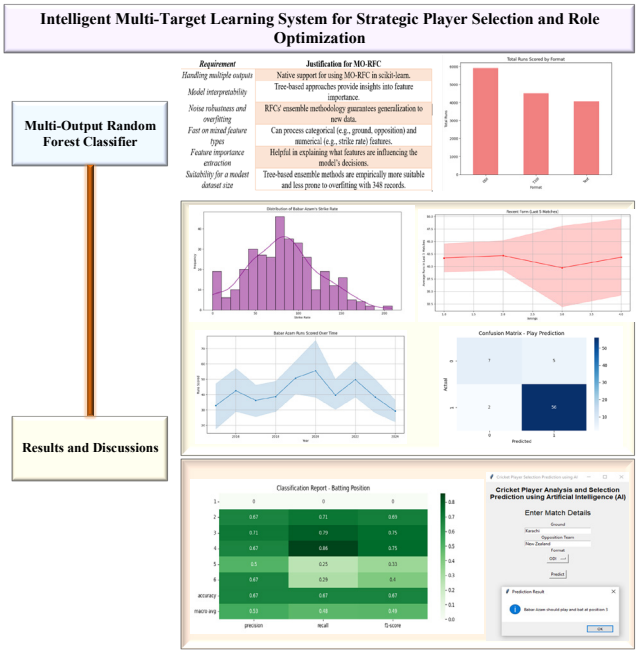


Fig. 1 Graphical abstract of the study

2. MULTI-OUTPUT RANDOM FOREST CLASSIFIER (MO-RFC)

In this study, the goal was to predict two dependent outcomes simultaneously: Should the player (e.g., Babar Azam) be chosen for a game (Yes/No) and his best batting order (say 1st, 2nd, 3rd… ,6th). To achieve this, a MO-RFC [31-33] is used, a stable ensemble-based supervised ML model that could handle multiple target variables simultaneously. This section is a complete walkthrough of the model, its architecture, mathematical formulation, and the reasons for its choice. MO-RFC is an ensemble learning method that builds many decision trees at training time and predicts the mode of the classes (for classification problems) or mean prediction (for regression problems). It prevents overfitting by aggregating the predictions from many weak learners to form a strong one. The principal characteristics of MO-RFC include robustness to noisy data, effective handling of both categorical and numerical feature types, and computational efficiency on high-dimensional datasets. MO-RFC is a predictive model where several outputs (targets) are predicted in one go for each input instance. In our study:

Target 1 was Binary classification: Should the player play? (Yes = 1, No = 0).

Target 2 was a multi-class classification: Batting position (1,2,3,4…).

Rather than training two separate models, MO-RFC enables learning shared responsibilities, taking advantage of interdependence across the targets (e.g., the target position may be related to match inclusion probability). MO-RFC in scikit-learn, a Python library, is a meta-estimator that trains one classifier per target. When used with a random forest base estimator, it becomes an MO-RFC. Equation (1) shows the feature matrix (e.g., ground, opposition, strike rate, and recent form), and Equation (2) shows the multi-output target matrix, whereas shows the binary output (match inclusion) and multiclass output (batting position)

$$X = [x_1, x_2, x_3, \dots, x_n] \in R^{n \times d} \quad (1)$$

$$Y = [y_1, y_2] \in R^{n \times 2} \quad (2)$$

Each random forest model $RF^{(j)}$ for output y_j is composed of T decision trees

$$RF^{(j)}(x) = \text{majority_vote}(\{ht^{(j)}(x)\}^{T}_{t} = 1) \quad (3)$$

where, $ht^{(j)}$ is the prediction of the t-th tree in the forest for output j, the final prediction is:

$$Y' = [RF^{(1)}(x), RF^{(2)}(x)] \quad (4)$$

Each tree is built using a random subset of: (a) Samples (bootstrap sampling), (b) Features (random feature subset at each node). The Gini impurity is used for splitting:

$$\text{ini}(p) = 1 - \sum_{i=1}^{k} p_i^2 \quad (5)$$

where, is the proportion of the class at a node. The choice of MO-RFC was guided by both theoretical and empirical reasons, as shown in Table 1.

Table 1 Requirements and Justification for the Selection of MO-RFC

Requirement	Justification for MO-RFC
Handling multiple outputs	Native support for using MO-RFC in scikit-learn.
Model interpretability	Tree-based approaches provide insights into feature importance.
Noise robustness and overfitting	RFCs' ensemble methodology guarantees generalization to new data.
Fast on mixed feature types	Can process categorical (e.g., ground, opposition) and numerical (e.g., strike rate) features.
Feature importance extraction	Helpful in explaining what features are influencing the model's decisions.
Suitability for a modest dataset size	Tree-based ensemble methods are empirically more suitable and less prone to overfitting with 348 records.

3. RESULTS AND DISCUSSIONS

This section introduces the results and analytical findings obtained from the entire lifecycle of the study, ranging from raw data preprocessing to exploratory data analysis (EDA), model predictions, GUI implementation, and the interpretation of results. The effectiveness of the system is illustrated by both quantitative performance measures and visualizations. Babar Azam's cricketing data was used as a primary case study to authenticate the system, the model, and the infrastructure are not player dependent. With minimal adjustments, the framework could be applied to any batsman with a suitable dataset, national or domestic teams for squad assessment, format-specific analytics (Test, ODI, T20I), and team management system integration for end-to-end decision making. The model is trained on Windows 10 using Jupyter's integrated development environment notebook (Python kernel), the system used has specifications Intel Core i5 with 8GB of RAM.

3.1 Dataset Overview and Preprocessing

The original dataset was built by web-scraping inning-by-inning match statistics for Babar Azam from the ESPNcricinfo website. The raw dataset had the following columns; runs, mins, balls faced (BF), strike rate (SR), 4s, 6s, batting position (Pos), innings (Inns), opposition, ground, start date. The dataset covers Babar Azam's international innings from 31 May 2015 to 18 November 2024, comprising 348 innings records in total. In the final dataset, the innings are distributed across formats as 120 ODIs, 126 T20Is, and 102 Tests (total = 348). Table 2 summarizes the dataset characteristics.

Table 2 Dataset Summary

Attribute	Detail
Data Source	ESPNcricinfo (web-scraped, innings-by-innings)
Player	Babar Azam (Pakistan)
Career Period Covered	31 May 2015 – 18 November 2024
Total Raw Innings Records	348
Records after Preprocessing	348
ODI Innings	120
T20I Innings	126
Test Innings	102
Train Set (80%)	278 records
Test Set (20%)	70 records
Splitting	Random Stratified Split (test_size = 0.2, random_state = 42, stratify = y_play)
Play = 1 (Selected)	288 (82.75%)
Play = 0 (Not Selected)	60 (17.25%)
Batting Position Class Distribution	Pos 3: ~47%, Pos 2: ~21%, Pos 4: ~19%, Pos 5: ~4.5%, Pos 1: ~4%, Pos 6: ~4%, Pos 7: 0.5%

The 80-20 split was applied after the complete preprocessing and feature engineering pipeline. Stratification was applied to preserve the class distribution of the Play (binary) target across both subsets. The class distribution of both target variables reveals a notable imbalance. For the Match Inclusion target, 82.75% of instances are labeled Play=1, reflecting the natural dominance of played innings in a high-caliber international batsman's career. For the Batting Position target, position 3 accounts for approximately 47% of all instances, consistent with Babar Azam's established role as a number three batsman across formats. This imbalance in both targets directly informs the model's differential performance across classes. During preprocessing, the following changes were made:

BF and Mins were excluded as they contributed minimally towards the prediction tasks.

The boundary percentage was formed using the formula as shown in Equation (6)

$$\text{Boundary } \% = \frac{[(4s * 4) + (6s * 6)]}{Runs} * 100 \quad (3)$$

where 4s and 6s denote the number of boundaries and sixes hit respectively, and Runs are the total runs scored in that inning.

Recent Form was computed as a moving average of the runs over the last 5 innings using the formula as depicted in Equation (7)

$$\text{Recent Form_i} = \text{Average}(\text{Runs_i} - 4, \ldots, \text{Runs_i}) \quad (4)$$

From the Opposition column (e.g., "ODI vs Zimbabwe"), two new columns were extracted:

Format: ODI, Opposition: Zimbabwe from "Start Date" (e.g., "31/05/2015"), the year of the match (e.g., 2015) was extracted for time-based trend analysis.

Although the system is validated on Babar Azam's career data, it is important to understand that the framework is architecturally player-agnostic. The preprocessing pipeline, feature engineering transformations, label construction logic, and MO-RFC model structure contain no player-specific hardcoded elements. Any batsman's inning-by-inning performance data structured equivalently to the ESPNcricinfo format used here can be processed through the identical pipeline without structural modification. The selection of Babar Azam as the validation case was driven by three practical considerations: sufficient career length to provide an adequately sized training dataset, consistent representation across all three international formats (ODI, T20I, Test), and availability of complete inning-level records spanning over eight years. Following preprocessing, the study adopted a standard AI-based sports analytics workflow comprising the following sequential stages: Data Loading, Data Exploration / Understanding, Data Cleaning, Data Visualization, Feature Engineering, Modelling (Machine Learning), Evaluation Metrics, and Results.

3.2 Exploratory Data Analysis

EDA helped uncover actionable patterns and correlations in Babar Azam's past performance data. The findings also informed feature engineering and model architecture. Figure 2 represents a bar chart that shows that Babar Azam scored most of his runs in ODIs, which is a sign of format-specific performance. This observation dictated the use of Format as a key feature.

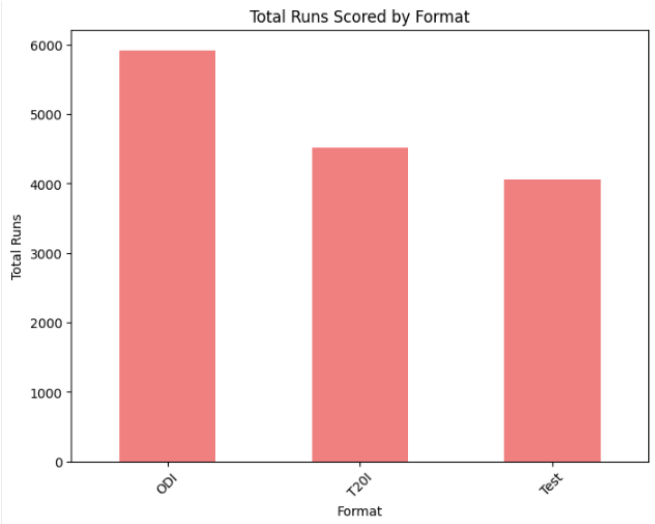


Fig. 2 Format-Specific Performance

In Figure 3, a horizontal bar chart reveals that Babar's record highest runs are against New Zealand, England, and Australia, supporting the development of opposition-specific performance measurement. In Figure 4, the Kernel Density Estimation (KDE) plot indicates a cluster in strike rates between 80-90, which reflects his strengths in ODIs. Figure 5 shows a line graph demonstrating rising boundary percentages after 2018, indicating the development of his scoring approach, demonstrated in the "Recent Form" elements of the model. Figure 6 represents bar plots indicating "caught" as the most common mode of dismissal, and Figure 7 illustrates box plots of dismissals against runs, indicating lower median scores for LBW, Runout, and Bowled dismissals. In Figure 8, a bar plot shows that Babar Azam's most effective position is #3 (One-Down), closely following the model's frequent output for batting position. In Figure 9, Line plots of Babar Azam's last 5 innings emphasized good consistency, highlighting the trustworthiness of the Recent Form feature.

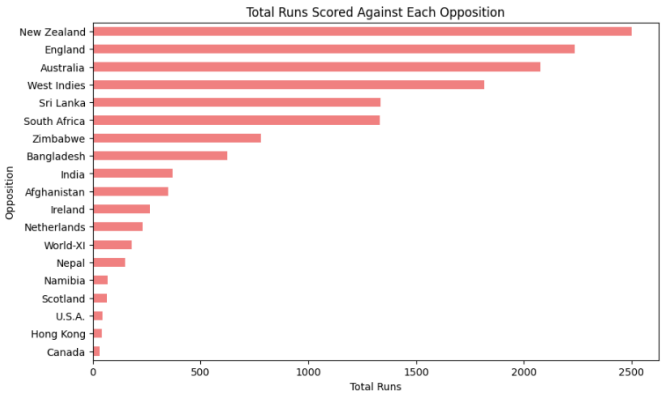


Fig. 3 Opposition-Wise Analysis

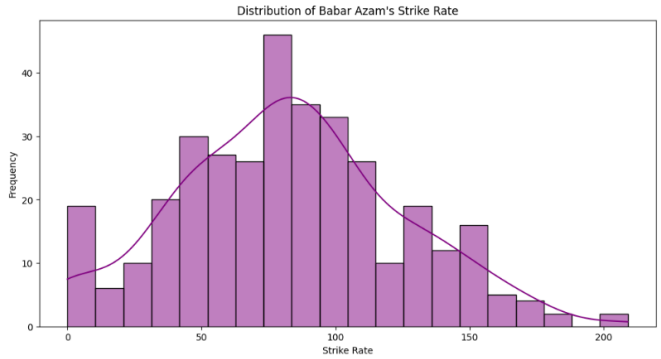


Fig. 4 Strike Rate Distribution

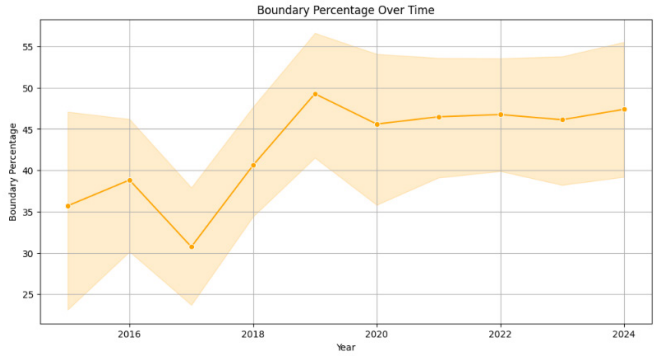


Fig. 5 Boundary Percentage Trends

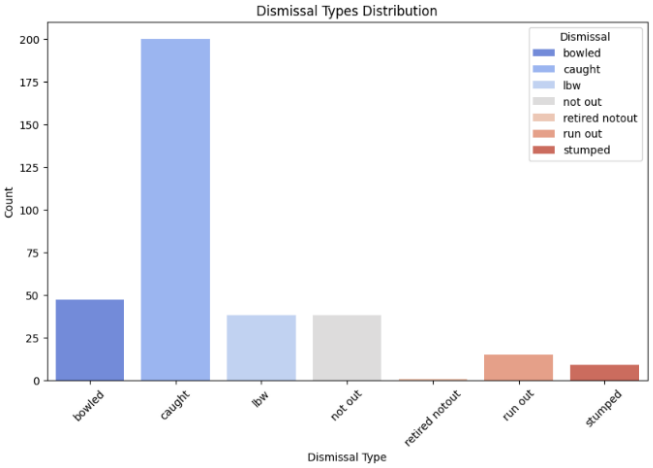


Fig. 6 Dismissal patterns

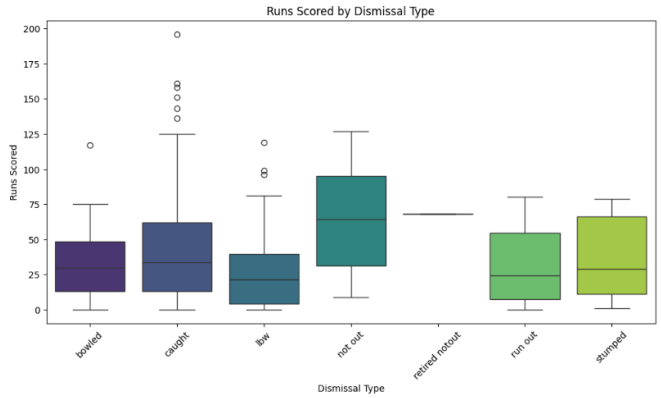


Fig. 7 Dismissal patterns by runs

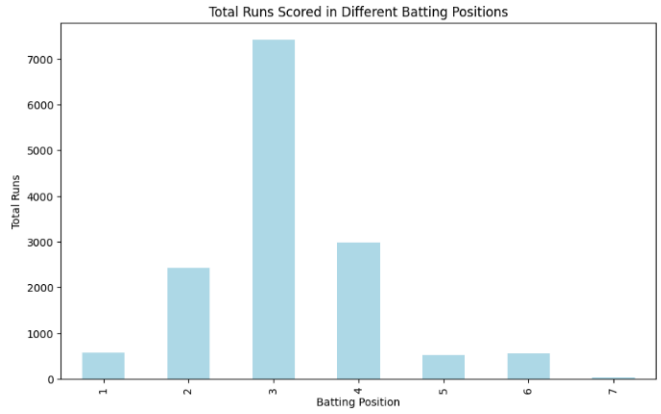


Fig. 8 Batting Position Productivity

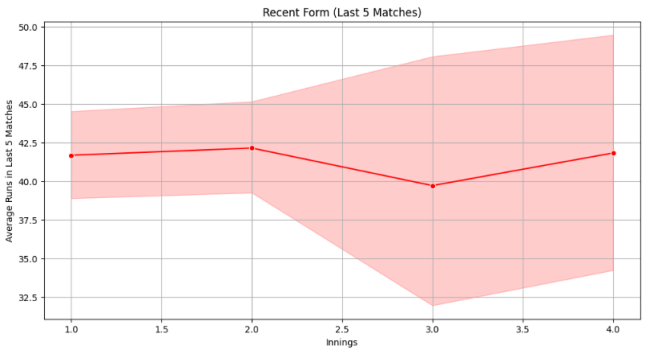


Fig. 9 Consistency and Recent Form

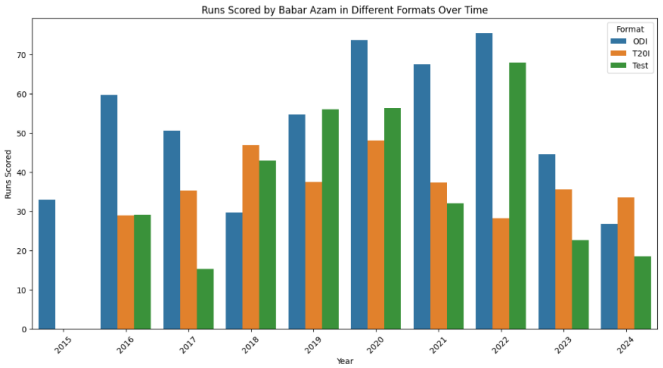


Fig. 10 Yearly Performance Trends

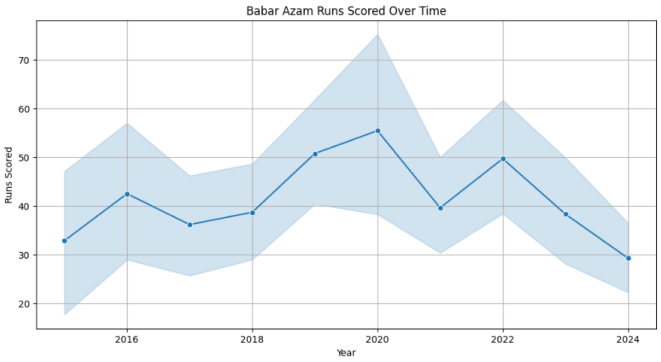


Fig. 11 Yearly Performance Trends

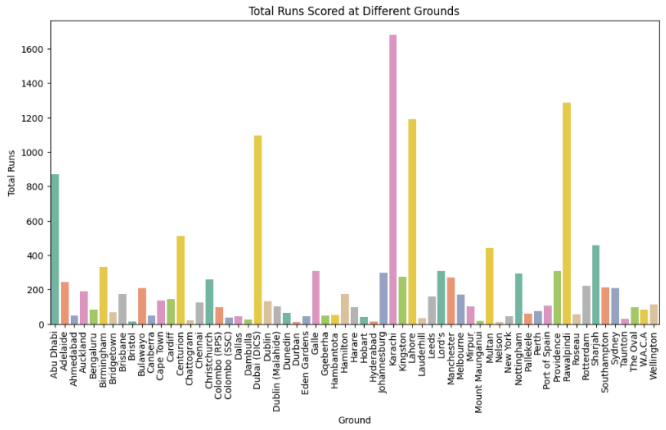


Fig. 12 Ground Specific Analysis

Figure 13 represents a scatter plot that shows Babar Azam's top scores across matches, and pink circles represent that Babar Azam has performed well in ODI matches across his career. Figure 14 represents a heatmap that depicts the correlation between different features.

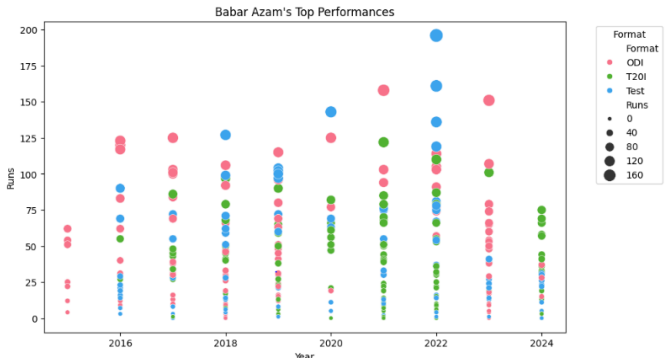


Fig. 13 Top Scores

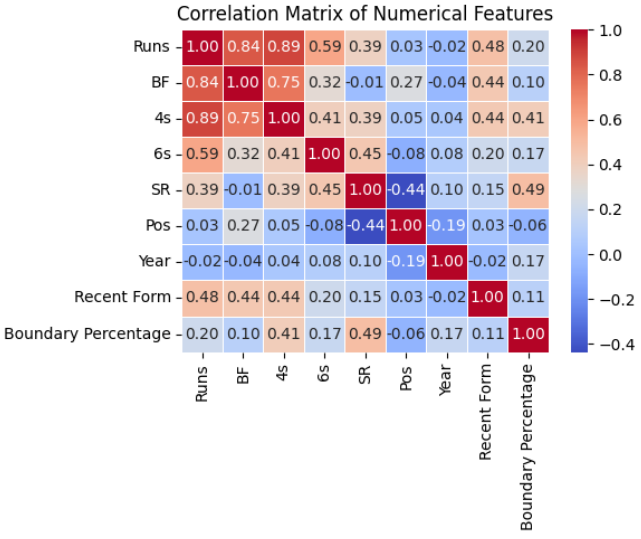


Fig. 14 Correlation Heatmap

This heatmap shows the following strong positive correlations:

Runs ↔ BF (0.84)

4s ↔ Runs (0.89)

4s ↔ BF (0.75)

These correlations had an impact on feature selection while training the model.

3.3 Feature Engineering and Label Construction for Supervised Learning

This section covers the phase in which raw data insights were changed into a format that was appropriate for Supervised ML.

For this, the dataset was subject to a series of transformations that enriched it with context-aware predictive features and well-defined target variables derived based on cricket logic.

Categorical Variable Encoding

The dataset also had contextual columns like Ground, Format, and Opposition, which are all key context variables. They were converted into numerical form by applying LabelEncoder to make them machine-readable. This conversion made sure that informative labels like "Karachi", "Test", or "India" were mapped to distinct integer values, while maintaining their identity for learning by the model.

Contextual Feature Engineering

To increase the model's learning capacity, some domain-guided derived features were developed from the given dataset:

Performance_Against_Opposition: Calculated as the player's runs average for every opposition team over formats.

Performance_at_Ground: Derived as the average run scored at every distinct ground.

Performance_in_Format: Saved in a dictionary form as the average runs scored per match format (e.g., T20I, ODI, Test), used subsequently in target creation with logic.

These engineered features were prepared to simulate the way actual cricket selectors gauge player appropriateness under given match situations.

Target (Labels) Generation: Building the 'Play' Column

To enable supervised learning, it was necessary to generate the Play column, a binary classification target indicating whether Babar Azam should be included in the match under specified circumstances. Applying threshold logic based on knowledge, the Play value was composed of four principal criteria:

Recent Form (average ≥ 15 runs in the previous 5 innings)

Performance Against Opposition (≥ 25 runs average)

Performance at Ground (≥ 25 runs average)

Performance in Match Format (≥ 25 runs average in that format)

This method formalizes the selection task as a supervised binary classification problem, allowing model learning with realistic decision logic that simulates expert reasoning. Following feature engineering and creation of labels, the final dataset employed for model training was as follows:

Features: Format, Opposition, Ground, Recent Form, Strike Rate, Boundary%, Runs

Targets: Play (Binary Target)

Pos (Multiclass Target; directly extracted from the Pos column)

This organized dataset was then split into training and testing subsets (80-20 split; 80% training data, 20% testing data) and input into the MO-RFC.

3.4 Predictive Modelling Results

The model was trained on the engineered dataset and tested on a held-out test set with common classification metrics: Accuracy, Precision, Recall, and F1 Score. Moreover, confusion matrices were calculated to gain more insights into the prediction distributions over classes.

Table 3 Evaluation Metrics

Prediction Task	Accuracy	Precision	Recall	F1-Score
Match Inclusion (1/0)	90%	0.894	0.9	0.894
Batting Position (1-6)	67.1%	0.646	0.6714	0.645

Accuracies from Table 3 show that the model performs extremely well in the binary classification of selection status. The batting position prediction accuracy of 67.1%, while moderate in absolute terms, must be interpreted in the context of the problem's inherent complexity. This is a six-class classification task where a random baseline would achieve approximately 16.7% accuracy, meaning the MO-RFC performs approximately four times better than random chance. Furthermore, batting order decisions in professional cricket are influenced by numerous non-statistical factors, including team strategy, opposition bowling attack, pitch conditions, and match situation, none of which are captured in historical performance data alone. The moderate accuracy, therefore, reflects the genuine difficulty of this prediction task. High recall and precision values for the Match Inclusion target prove that the model is precise and reliable in marking the points where the player should be chosen. Although less precise than binary classification, the multi-class performance is consistent and stable, considering the class distribution imbalance and the higher complexity involved in predicting specific batting positions.

All precision, recall, and F1-score values reported in Table 3 are computed using weighted averaging, where each class's contribution is proportional to its support (number of true instances), ensuring that class imbalance is appropriately accounted for in the aggregate metrics.

Table 4 Multi-Output Evaluation Metrics for MO-RFC

Metric	Value	Interpretation
Hamming Loss	0.2143	21.43% of all label predictions were incorrect across both targets
Subset (Exact Match) Accuracy	0.6000	60% of instances had both targets predicted correctly simultaneously
Average Target Accuracy	0.7857	Mean per-target accuracy across Match Inclusion and Batting Position
Average Weighted F1-Score	0.7697	Mean weighted F1 across both output targets
Weighted F1 - Match Inclusion	0.894	Strong per-target performance on binary classification
Weighted F1 - Batting Position	0.645	Moderate, reflecting multi-class complexity and class imbalance

To further evaluate the system's joint prediction performance across both output targets simultaneously, two additional metrics are reported in Table 4. Hamming Loss measures the fraction of incorrectly predicted labels across all outputs and instances (lower is better), while Subset Accuracy, also known as Exact Match Ratio, measures the proportion of instances where both targets were predicted correctly simultaneously. The MO-RFC achieved a Hamming Loss of 0.2143 and a Subset Accuracy of 0.6000, indicating that 60% of test instances had both match inclusion and batting position predicted correctly in a single inference pass. To contextualize this result, a naive majority-class baseline, predicting 'Play = 1' for all instances and the most frequent batting position (position 3) for all instances, would achieve a Subset Accuracy of approximately 0.34 on this dataset. The MO-RFC's Subset Accuracy of 0.60, therefore, represents a substantial improvement of approximately 76% over this naive baseline, confirming the system's genuine predictive value beyond trivial class-frequency exploitation. The average per-target accuracy across both outputs was 78.57%, and the average weighted F1-score across both targets was 0.7697, confirming the system's overall robustness.

To assess the robustness and stability of the reported performance beyond a single train-test split, a 10-fold stratified cross-validation was conducted on the complete preprocessed dataset. Stratified k-fold partitioning was applied to preserve the class distribution of the Match Inclusion target across all folds, ensuring representative sampling in each iteration. The cross-

Validation results are summarized in Table 5.

Table 5 10-Fold Stratified Cross-Validation Results

Prediction Target	Mean CV Accuracy	Std. Deviation	Holdout Test Accuracy	Consistency
Match Inclusion (Play)	90.2%	4.2%	90.0%	Confirmed stable
Batting Position (Pos)	65.8%	4.4%	67.1%	Confirmed stable

The close agreement between cross-validated and holdout test accuracies, combined with narrow standard deviations across folds, confirms that the reported performance metrics are stable and are not artifacts of a particularly favorable data partition. This robustness analysis directly validates the reliability of the proposed system's predictions across varying training configurations.

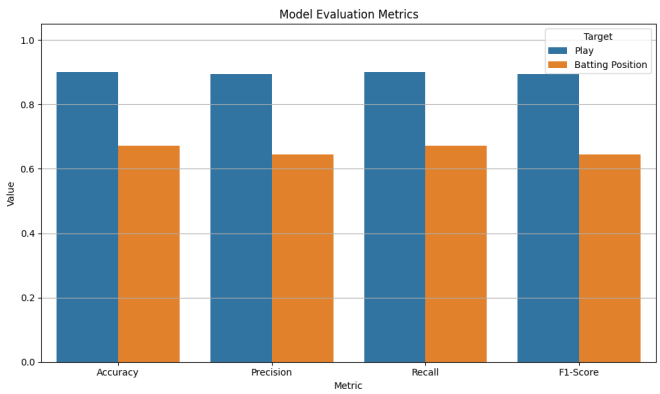


Fig. 15 Comparison of Evaluation Metrics

Figure 16 represents the confusion matrix of the 'Play' prediction. The model accurately distinguished 56 of 58 "Play = 1" instances and 7 of 12 "Play = No" instances with practically no false positives and false negatives. Figure 17 represents the classi-

fication report heatmap of Play prediction. It shows:

Class 1 (Played) is predicted with very high performance: precision = 0.92, recall = 0.97

Class 0 (Did not play) is less well predicted: recall = 0.58 (means many "Did not play" cases were wrongly predicted as "Play")

Overall accuracy is still high (90%), so this is likely acceptable depending on context.

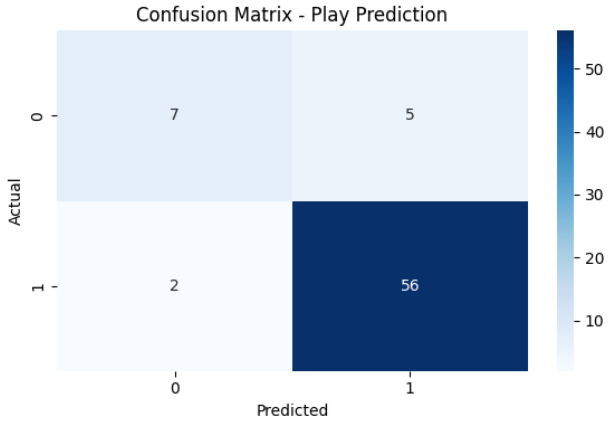


Fig. 16 Confusion Matrix of Play Prediction

Figure 18 shows the confusion matrix of Batting Position prediction. It shows that the model works best with the most common positions (2nd, 3rd, 4th, which are 1,2,3), which is also in line with Babar Azam's past trends. Misclassifications are higher in less common positions (0 and 5, i.e., 1st and 6th), which can be optimized by class imbalance handling or weighted loss functions. Figure 19 shows the classification report heatmap for Batting Position prediction. It classifies:

Class 4 is predicted best (precision: 0.67, recall: 0.86)

Classes 5 and 6 show very poor recall; the model misses these classes often.

Class 1 has 0 precision and recall, meaning it was never predicted correctly; it's likely due to extreme class imbalance, lack of distinguishing features, or even bias of the testing subset.

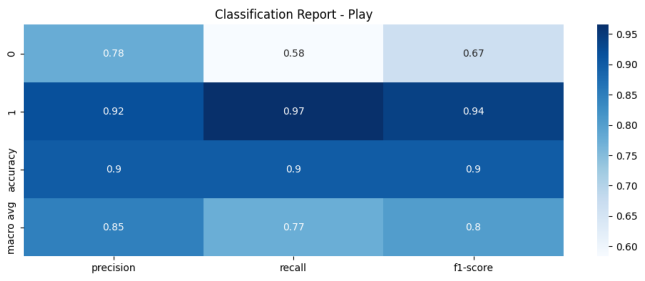


Fig. 17 Classification Report of Play Prediction

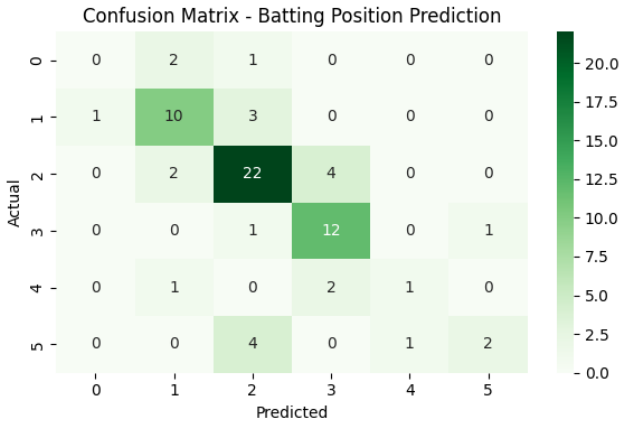


Fig. 18 Confusion Matrix of Batting Position Prediction

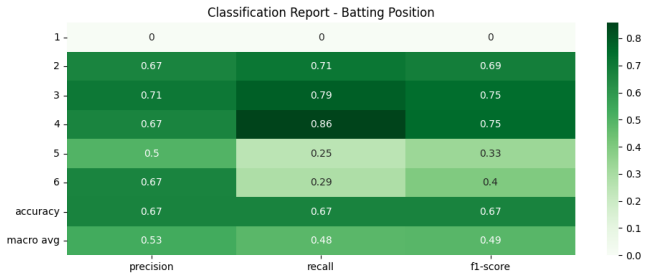


Fig. 19 Classification Report of Batting Position Prediction

To gain a better understanding of which input features contributed most to the model's predictions, we calculated and plotted the feature importance from each random forest estimator within the multi-output model. The following figures 20 and 21 illustrate the ranked importance of features of both target variables, Play and Batting Position, giving us an idea of which attributes the model used the most in decision-making.

Table 6 Feature Importance for Target Play

Feature	Importance
Recent Form	0.339452
Ground	0.156568
Run	0.149777
SR	0.116047
Opposition	0.110510
Boundary Percentage	0.076039
Format	0.051608

Table 7 Feature Importance for Target Pos

Feature	Importance
Format	0.200029
SR	0.184318
Runs	0.142435
Recent Form	0.140203
Ground	0.138464
Boundary Percentage	0.108776
Opposition	0.085777

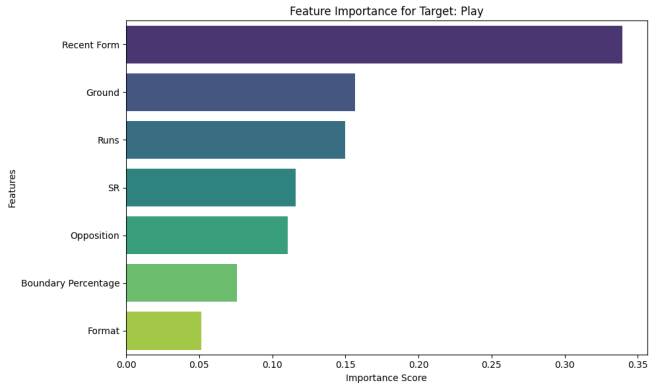


Fig. 20 Feature Importance for Target: Play

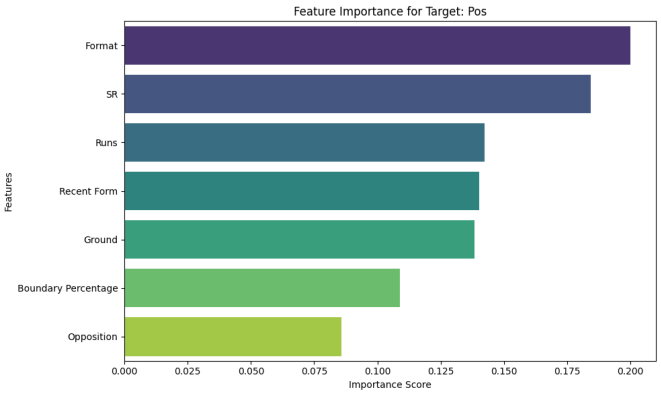


Fig. 21 Feature Importance for Target: Pos

3.5 Comparative Model Analysis

To practically validate the selection of MO-RFC as the backbone of the proposed system, a comparative evaluation was conducted against three alternative multi-output classifiers: MO-Gradient Boosting (MO-GB), MO-Logistic Regression (MO-LogReg), and MO-k-Nearest Neighbors (MO-KNN). All models were implemented using scikit-learn's MultiOutputClassifier meta-estimator and evaluated under strictly identical experimental conditions, the same preprocessed dataset, an identical 80-20 stratified train-test split, the same feature set, and the same random state, to ensure a fair and reproducible comparison. No model-specific hyperparameter tuning was applied, establishing a consistent default-setting baseline.

Table 8 Comparative Performance of Multi-Output Classifiers

Model	Subset Accuracy	Hamming Loss	Average Target Accuracy	Average Weighted F1
MO-RFC	0.6000	0.2143	0.7857	0.7697
MO-GB	0.5857	0.2143	0.7857	0.7696
MO-LogReg	0.5143	0.2643	0.7357	0.7174
MO-KNN	0.5143	0.2786	0.7214	0.6942

The results confirm that MO-RFC achieves the highest Subset Accuracy (0.6000) and is tied with MO-GB for the lowest Hamming Loss (0.2143). MO-LogReg and MO-KNN perform considerably lower across all metrics, with MO-LogReg constrained by its assumption of linear decision boundaries, an assumption poorly suited to the non-linear, context-dependent nature of cricket performance data. MO-KNN's distance-based mechanism is sensitive to the mixed feature space of label-encoded categorical and continuous numerical variables, further limiting its effectiveness. SVM was excluded from comparative analysis on two principles. First, SVM does not natively support multi-output classification and requires a MultiOutputClassifier wrapper with one-vs-rest decomposition, which increases computational complexity multiplicatively with the number of output classes. Second, and more critically for this study, SVM does not provide feature importance scores in any native form, making it architecturally incompatible with the interpretability requirements of the proposed system. Deep learning approaches such as MLPs and LSTMs were similarly excluded, as their data requirements for reliable generalization substantially exceed the 350-record dataset available in this study. As established in sports analytics literature, deep learning models trained on small datasets are susceptible to overfitting and tend to underperform simpler ensemble methods in low-data regimes. Both exclusions are, therefore, methodologically justified rather than arbitrary.

MO-GB is the closest competitor to MO-RFC, with nearly identical Hamming Loss and average target accuracy. However, MO-RFC is retained as the final deployed model for three principal reasons. First, MO-RFC achieves a marginally higher Subset (Exact Match) Accuracy (0.6000 vs. 0.5857), which is the most stringent and meaningful metric for a dual-target prediction system. Second, MO-RFC provides built-in, per-target feature importance scores, a central analytical component of this study, whereas gradient boosting feature importance in a multi-output wrapper setting offer less transparent per-target attribution. Third, MO-RFC's bootstrap aggregation mechanism offers stronger robustness to overfitting on the relatively modest dataset size used in this study. These considerations, combined with its superior interpretability for stakeholder communication, confirm MO-RFC as the most suitable and justified model choice for this application.

3.6 GUI-Based Prediction System

For converting the predictive model into a practical utility usable by coaches, analysts, and team administrators, a GUI was created with Python's Tkinter package. The GUI is a lightweight, desktop-based front-end through which users can enter match-specific contextual variables such as Ground, Opposition team, and Format, while the model uses Recent Form from the dataset to access real-time AI-suggested recommendations. The GUI is clean and intuitive and has Input fields for Ground and Opposition Team (free-text entries), A dropdown menu to choose the Match format (ODI, T20I, Test), and a "Predict" button that makes the model calculate whether the player must be included in the Playing XI and the best batting position for the current situation. At runtime, the GUI checks for user input and shows prediction output in message box pop-ups. The backend is linked to the function, which calls the pre-trained MO-RFC. The software provides a non-technical, user-friendly interface for data-driven decision-making for cricket selection. This GUI module improves the study's accessibility and deployment quality, closing the gap between AI algorithms and real-world, coach-level usability in professional sport settings. Figure 22 represents the GUI.

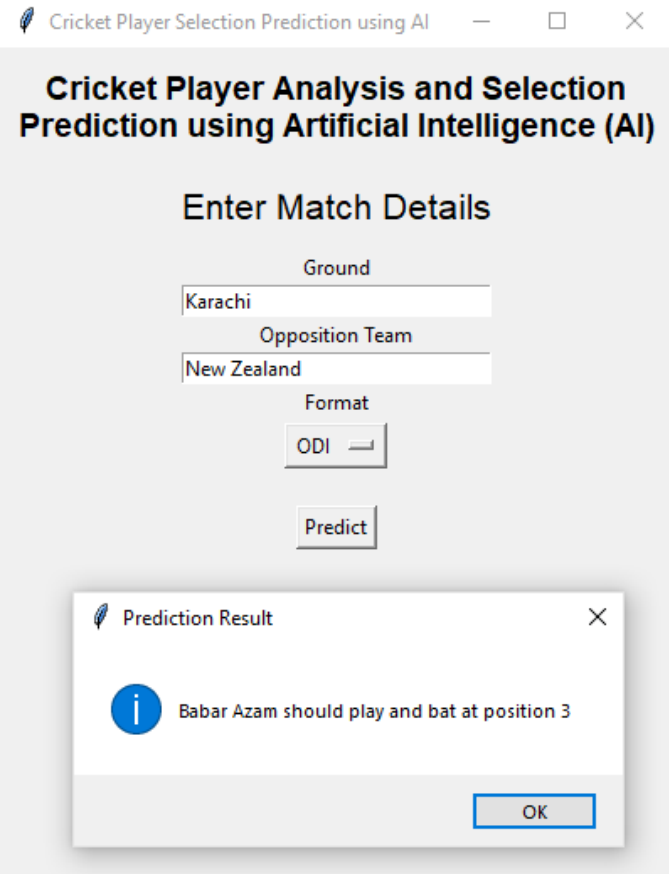


Fig. 22 GUI of the prediction system

3.7 Limitations and Future Directions

While the proposed system demonstrates strong predictive performance and practical utility, a transparent evaluation of its current constraints is essential for situating the findings accurately and guiding subsequent research.

Static Data Dependency: The model is trained exclusively on historical inning-by-inning performance data scraped from ESPNcricinfo. It does not incorporate dynamic, real-time variables such as player fitness status, injury reports, pitch and weather conditions, or day-of-match psychological readiness. These factors are well-established influences on both selection decisions and batting performance in professional cricket, and their absence means the system functions as a strategic advisory tool based on historical patterns rather than a fully context-aware real-time decision engine.

Inter-Target Independence: Although MO-RFC enables joint multi-output prediction, it internally trains an independent classifier for each target. The structural dependency between the two outputs, specifically, that batting position is only meaningful conditional on a positive selection decision, is not explicitly modelled. Approaches such as classifier chaining, where the prediction of one target is used as an input feature for the next, or joint probabilistic models, could better capture this conditional dependency and potentially improve batting position prediction accuracy.

Class Imbalance in Batting Position: Babar Azam has overwhelmingly batted at positions 3 and 4 throughout his career, resulting in severe underrepresentation of positions 1, 5, and 6 in the training data. As evidenced by the near-zero precision and recall for certain position classes in the classification report, the model struggles to generalize predictions for infrequent batting positions. Future work should explore techniques such as Synthetic Minority Over-sampling Technique (SMOTE)-based oversampling, class-weight adjustment during training, or threshold calibration to mitigate this imbalance.

Single-Player Validation Scope: The system was validated exclusively on Babar Azam's career data. While the framework is architecturally generalizable to any batsman given appropriate performance data, the domain-specific thresholds used in constructing the Play label, for example, a minimum average of 25 runs against opposition, are calibrated for a high-caliber top-order batsman and may require recalibration for players of different profiles, roles, or skill levels. Broader validation across multiple players, domestic competitions, and team contexts is necessary to establish the system's full generalizability.

Future Directions: Building on the findings and limitations identified above, the following extensions are proposed for future research:

First, integration of real-time data feeds, such as those available through official cricket board APIs, would enable dynamic, match-day predictions that incorporate current player fitness, pitch reports, and weather conditions, substantially increasing the system's practical decision-support value.

Second, the framework could be extended from individual player analysis to full squad optimization, enabling simultaneous selection and role assignment for an entire playing XI. This would require multi-player joint modelling and constraint-based optimization to ensure team balance across batting, bowling, and all-round roles.

Third, deep learning architectures, particularly Long Short-Term Memory (LSTM) networks, could complement the MO-RFC for modelling the temporal dynamics of player form. Unlike the moving average used in this study, LSTMs can learn complex sequential dependencies across arbitrary time horizons, potentially improving Recent Form representation.

Fourth, deployment as a web-based or mobile application would significantly broaden the system's accessibility beyond the desktop Tkinter interface, enabling integration into existing team management platforms and allowing remote access by selectors, coaches, and analysts across different locations.

Fifth, the incorporation of explainability techniques such as SHAP (Shapley Additive exPlanations) values would provide granular, instance-level explanations of individual predictions, further enhancing stakeholder trust and enabling coaches to understand not just what the model predicts, but why.

Sixth, the proposed framework is not inherently cricket-specific. With appropriate dataset construction and domain-adapted feature engineering, the multi-output classification approach could be applied to player selection and role optimization in other team sports such as football, basketball, and hockey, offering broad applicability across the sports analytics domain.

4. CONCLUSION

This study presented a comprehensive, end-to-end AI-based intelligent decision support system for cricket player analysis and selection prediction, addressing a longstanding gap between data-driven methodology and practical sports management. By deploying a Multi-Output Random Forest Classifier (MO-RFC), the system simultaneously predicts two strategically critical out-

comes, whether a player should be included in the playing XI and their optimal batting position, within a single unified inference framework. Using Babar Azam's complete international career data as a case study, the system was developed through a rigorous pipeline encompassing web-scraped data collection, multi-stage preprocessing, domain-guided feature engineering, supervised label construction, model training, and comprehensive evaluation.

The MO-RFC achieved a Match Inclusion accuracy of 90% with a weighted F1-score of 0.894, demonstrating strong and reliable binary classification performance. Batting Position prediction achieved an accuracy of 67.1%, approximately four times better than a random six-class baseline, reflecting genuine predictive capability in the face of inherent non-statistical complexity in batting order determination. A comparative analysis against three alternative multi-output classifiers confirmed MO-RFC as the superior model, achieving the highest Subset Accuracy (0.6000) and tied lowest Hamming Loss (0.2143), while offering unmatched interpretability through feature importance reporting. The system was further made accessible to non-technical stakeholders, including coaches, selectors, and analysts, through an interactive Tkinter-based GUI enabling real-time, context-specific predictions without requiring programming expertise.

The study demonstrates that objective, data-driven player selection frameworks can meaningfully complement and enhance traditional intuition-based decision-making in professional cricket, reducing selection bias and enabling more consistent, evidence-based team management. Although validated on a single player's dataset, the system's architecture is designed to be completely generalizable across players, teams, formats, and competitions with appropriate data. This work contributes a scalable, interpretable, and practically deployable foundation for AI-powered sports analytics, with implications extending beyond cricket to broader applications in team sport performance management.

REFERENCES

[1] Shah, S. A. A., & Zaman, Q. (2025). "Optimal selection of a probabilistic machine learning model for predicting high run chase outcomes in T-20 international cricket." *Journal of Sports Sciences,* 1-19.

[2] Lokhande, R., Awale, R. N., & Ingle, R. R. (2025). "Forecasting bowler performance in One-Day International cricket using Machine learning." *Expert Systems with Applications,* **259**, 125178.

[3] SANJAYKUMAR, S., RAJAN, D., PAVLOVIĆ, R., RADULOVIĆ, N., & NAIARAJAN, S. (2025). "BALANCING TRADITION AND INNOVATION: A REVIEW ARTICLE ON RECENT RULE CHANGES IN CRICKET." *Sports Science & Health/Sportske Nauke i Zdravlje,* **15**(5).

[4] Singh, S. B., Singh, D. N., Ray, S. K., & Nagaraj, H. B. (2023). "Cricket Pitch Behavioural Analysis Index (PBAI) and Pitch Behaviour Forecasting (PBF) Methods: Traditional and Conventional Methods vs. Scientific Methods Based on Recent Research and Developments." *Cricket Pitches: The Science Behind the Art of Pitch-Making—"An Integrated Pitch Management (IPM) Approach",* 217-300.

[5] Imdad, H., & Ahmad, H. (2023). "Interdisciplinary application of machine learning, data science, and python for cricket analytics." *In Advanced Interdisciplinary Applications of Machine Learning Python Libraries for Data Science* (pp. 32-77). IGI Global.

[6] Vishwarupe, V., Bedekar, M., Joshi, P. M., Pande, M., Pawar, V., & Shingote, P. (2022). "Data analytics in the game of cricket: A novel paradigm." *Procedia Computer Science,* **204**, 937-944.

[7] Subburaj, M., Rao, G. R. K., Parashar, B., Jeyabalan, I., Semban, H., & Sengan, S. (2023). "Artificial intelligence for smart in match winning prediction in Twenty20 cricket league using machine learning model." *In Artificial Intelligence for Smart Healthcare* (pp. 31-46). Cham: Springer International Publishing.

[8] Hassan, B., Clough, C., Siddiqi, Y., Ali, R. F., & Arshed, M. A. (2024). "PlayerRank: leveraging learning-to-rank AI for player positioning in cricket." *IEEE Access.*

[9] Kumar, C., & Balasubramanian, G. (2023). "Comparative analysis of pitch ratings in all formats of cricket." *Management and Labour Studies,* **48**(3), 307-324.

[10] Scott, P., & Herridge, R. (2018). "Cricket." *In Routledge Handbook of Strength and Conditioning* (pp. 194-217). Routledge.

[11] Sivaramaraju Vetukuri, V., Rajender, R., & Sethi, N. (2019). "A multi-aspect analysis and prediction scheme for cricket matches in standard T-20 format." *International Journal of Knowledge-Based and Intelligent Engineering Systems,* **23**(3), 149-154.

[12] Ali, M. A., Chaudhary, N. I., Khan, T. A., Mao, W. L., Lin, C. C., & Raja, M. A. Z. (2024). "Design of key term separated identification model for fractional input nonlinear output error systems: Auxiliary model based Runge Kutta optimization algorithm." *Chaos, Solitons & Fractals,* **189**, 115696.

[13] Ali, M. A., Chaudhary, N. I., Lin, C. C., & Mao, W. L. (2025). "Mountain Gazelle optimization algorithm for identification of nonlinear Hammer-stein output error systems." *Journal of Innovative Technology,* **7**(1), 117-124.

[14] Ali, M. A., Chaudhary, N. I., Khan, T. A., Mao, W. L., Lin, C. C., Khan, Z. A., & Zahoor Raja, M. A. (2025). "Auxiliary Model-Based Chameleon Swarm Optimization for Robust Parameter Estimation of Fractional Order Nonlinear Hammerstein Systems." *Journal of Computational and Nonlinear Dynamics,* **20**(9), 091005.

[15] Ali, M. A., Chaudhary, N. I., Malik, M., & Abbas, M. Q. (2025). "Artificial gorilla troops optimization algorithm for key term separated input non-linear output error system identification." *Journal of Innovative Technology,* **7**(2), 37-43.

[16] Mahmood, T., Ali, N., Raja, M. A. Z., Chaudhary, N. I., Cheema, K. M., Shu, C. M., & Milyani, A. H. (2023). "Intelligent backpropagated predictive networks for dynamics of the power-law fluidic model with moving wedge and flat plate." *Waves in Random and Complex Media,* 1-26.

[17] Mahmood, T., Ali, N., Chaudhary, N. I., Cheema, K. M., Milyani, A. H., & Raja, M. A. Z. (2022). "Novel adaptive Bayesian regularization networks for peristaltic motion of a third-grade fluid in a planar channel." *Mathematics,* **10**(3), 358.

[18] Raja, M. A. Z., Khan, M. A. R., Mahmood, T., Farooq, U., & Chaudhary, N. I. (2016). "Design of bio-inspired computing technique for nanofluidics based on nonlinear Jeffery–Hamel flow equations." *Canadian Journal of Physics,* **94**(5), 474-489.

[19] Arain, M. B., Mahmood, T., Shaheen, S., Al-Yarimi, F. A., & Hu, J. (2024). "Intelligent backpropagated predictive neural networks approach for acoustic wave-driven gas bubble dynamics in non-Newtonian fluid." *Physics of Fluids,* **36**(9).

[20] Khan, A., Raja, M. A. Z., Chang, C. Y., Khan, M. P., Khan, Z. A., Shoaib, M., & Shu, C. M. (2025). "Novel design of deep learning knowledge-driven recurrent neurostructure for bio-convective Maxwell nanofluid flow model with convective boundary and variable thermal conductivity." *The European Physical Journal Plus,* **140**(6), 472.

[21] Shaheen, S., Mahmood, T., Raja, M. A. Z., Al-Yarimi, F. A., Arain, M. B., & Hu, J. (2025). "Machine learning-based investigation of activation energy for bio-convective boundary layer flow inspired by challenges in aerospace heat transfer systems." *Engineering Applications of Artificial Intelligence,* **160**, 111917.

[22] Arain, M. B., Mahmood, T., Al-Yarimi, F. A., Ali, M. A., Shaheen, S., & Hu, J. (2026). "On the study of acoustic-driven bubble dynamics in Williamson Fluids with NARX neural networks." *Engineering Applications of Artificial Intelligence,* **165**, 113276.

[23] Qureshi, H., Shah, Z., Raja, M. A. Z., Alshahrani, M. Y., Khan, W. A., & Shoaib, M. (2024). "Machine learning investigation of tuberculosis with medicine immunity impact." *Diagnostic Microbiology and Infectious Disease*, **110**(3), 116472.

[24] Anwar, N., Fatima, A., Raja, M. A. Z., Ahmad, I., Shoaib, M., & Kiani, A. K. (2025). "Novel exploration of machine learning solutions with supervised neural structures for nonlinear cholera epidemic probabilistic model with quarantined impact." *The European Physical Journal Plus,* **140**(1), 64.

[25] CA, A. P., Shishira, B. J., Anvekar, S. M., & CG, U. R. (2024, July). "Optimal Cricket Team Selection Using Machine Learning." *In 2024 Second International Conference on Advances in Information Technology (ICAIT)* (Vol. 1, pp. 1-5). *IEEE.*

[26] Agrawal, Y., & Kandhway, K. (2024). "Winner prediction in an ongoing one day international cricket match." *Journal of Sports Analytics,* **9**(4), 305-318.

[27] Passi, K., & Pandey, N. (2018). "Increased prediction accuracy in the game of cricket using machine learning." *arXiv preprint arXiv:1804.04226.*

[28] Raajesh, S., Martin, N., Jiji, J., & Nair, A. (2024, May). "Cricket Team Selection and Player Analysis using Data Analytics." *In 2024 IEEE Recent Advances in Intelligent Computational Systems (RAICS)* (pp. 1-6). *IEEE.*

[29] Pathak, N., & Wadhwa, H. (2016). "Applications of modern classification techniques to predict the outcome of ODI cricket." *Procedia Computer Science,* **87**, 55-60.

[30] Tahir, M. A., Nazeer, M. T., Atta, H., Saeed, M. A., & Munir, A. (2022). "Artificial Intelligence and Data Analytics in Cricket." *Webology,* **19**(2).

[31] Meenakshisundaram, N., & Sajiv, G. (2025, April). "Enhancing Multi-Output Classification Performance with Hybrid Models and Ensemble Learning Techniques." *In 2025 8th International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1643-1648). *IEEE.*

[32] Wan, Y., Liu, D., & Ren, J. C. (2024). "A modeling method of wide random forest multi-output soft sensor with attention mechanism for quality prediction of complex industrial processes." *Advanced Engineering Informatics,* **59**, 102255.

[33] Najmoddin, A., Etemadfard, H., & Ghalehnovi, M. (2024). "Multi-output machine learning for predicting the mechanical properties of BFRC." *Case Studies in Construction Materials,* **20**, e02818.