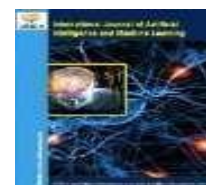


SvedbergOpen
DISSEMINATION OF KNOWLEDGEInternational Journal of Artificial
Intelligence and Machine LearningPublisher's Home Page: <https://www.svedbergopen.com/>

Research Paper

Open Access

Hybrid Ensemble Machine Learning for Constrained Team Selection in Regional Cricket

Mr. Ravindra Dangar^{1*}, Dr. Amit Lathigara², Mr. Vinodkumar Kotecha³¹Research Scholar, RK University, India.²Professor, RK University, India.³Director, Rishvi Ltd.* Corresponding author's Email: rrdangar@gmail.com**Abstract**

Cricket team selection at the regional level has historically relied on subjective expert judgment rather than data-driven processes. This paper proposes a hybrid machine learning approach that integrates ensemble prediction with an expert-derived fitness formula. Building on prior work using recursive feature elimination and genetic algorithms for fantasy cricket, we adapt the methodology to a single fielded roster with exact role counts and incorporate contextual factors absent from fantasy-sport formulations: pitch suitability, opposition impact, fatigue, and home advantage. A voting ensemble of Random Forest, Gradient Boosting and Ridge Regression is combined with the expert formula through a 60:40 weighted score. On a 51-player dataset the method produces an 11-player team meeting every positional constraint deterministically, in time that scales as $n \log n$. A simulated 300-player pool confirms that the method scales. The approach is implemented as an interactive desktop application for coaches.

Keywords: Cricket team selection, Machine learning, Ensemble methods, Hybrid fitness model, Regional sports analytics, Constrained optimization

1. Introduction

Team selection is a critical factor in determining competitive results and pathways to player development in regional cricket but remains largely based on subjective expert judgement and limited statistical analysis. Analytics-driven approaches have become more common in professional cricket but regional level teams don't have access to systematic, data-driven selection frameworks [2].

Current selection methods have several limitations: they do not integrate contextual performance data, such as pitch conditions, opposition strength and player fatigue; they specify roles as permissible ranges rather than the exact number of players needed on a fielded roster; and they rely on single-metric evaluation instead of holistic evaluation of players. Jha et al. [1] proposed a basic methodology that showed the potential of using recursive feature elimination (RFE) combined with a genetic algorithm for selecting fantasy cricket teams. Fantasy-sport selection differs from regional team selection in that fantasy participants select players from both competing teams within a credit budget and role quotas are given as ranges, whereas a regional side must field one roster with exact specialist counts (wicket-keeper, openers, middle-order, finishers, all-rounders, spinners, pacers).

The following contributions of this work fill these gaps:

- 1. Hybrid fitness model:** Weighted combination (60% ML prediction, 40% expert-based formula) using ensemble learning and domain knowledge to evaluate players.
- 2. Contextual feature integration:** Including pitch suitability, opposition impact, fatigue, recent form and home advantage alongside traditional batting / bowling / fielding statistics.
- 3. Deterministic constraint enforcement:** exact user-specified role counts and guaranteed player uniqueness via a greedy role-constrained algorithm, resulting in reproducible selections in $O(n \log n)$ time without the run-to-run variation of population-based search.
- 4. Practical implementation:** An interactive desktop application for importing player data, defining role requirements and to generate an optimised team with transparent, decomposable fitness scores.

2. Related Work

2.1 Cricket Team Selection Methods

Cricket analytics has come a long way in the last 2 decades. Lewis [3] developed resource based measures of player performance based on the Duckworth/Lewis methodology which allows batting and bowling contributions to be compared on a common scale. Barr et al. [4] used a risk-return criterion to balance probability of dismissal against scoring rate. Singh et al. [5] included strike rate and scoring of boundaries in evaluation of batsmen. Integer programming has been effective in formulating model selection as a constrained optimisation problem [6,7]. Neural networks have shown promise in performance prediction [8], but often at the cost of interpretability. Genetic algorithms [9,10] have been effective for multi-objective selection.

2.2 Fantasy Cricket and Data-Driven Selection

New data-driven selection contexts have emerged with fantasy platforms (Dream11, FanFight, MyTeam11) Jha et al. [1] proposed a hybrid RFE-genetic algorithm for fantasy team optimisation. They identified key indicators by performing recursive feature elimination in random forest and selected combinations using NSGA-II. They experimented on 48 matches of the 2019 ICC World Cup. The formulation enforces player uniqueness, role quotas as ranges, a credit budget and a cap on players from either side, with feasibility maintained within the crossover operator. But fantasy contexts are different from regional selection: the points system is platform-specific, players are picked from both sides, and role requirements are ranges rather than the specific counts a single fielded roster requires.

2.3 Machine Learning in Sports Analytics

Ensemble methods have become popular in sports analytics for robustness and generalisation [11, 12]. Player efficiency ratings have been tackled using random forests [13], gradient boosting captures non-linear performance relationships [14] and ridge regression provides stable predictions in high dimensional feature spaces [15]. Recent work has combined ML with domain knowledge: Adhikari et al. [2] integrated data envelopment analysis with expert constraints and Chandu and Nirmala [16] compared logistic regression with random forest and reported gains from ensemble based evaluation. Other studies have expanded ML selection to performance forecasting [17], team dynamics [18] and in-match prediction [19] but none deal with role-constrained, duplicate-free roster construction.

2.4 Research Gap

But there are still three holes. First, most methods emphasize core batting and bowling statistics while neglecting contextual factors such as pitch, opposition, fatigue and home advantage; Jha et al. [1] note explicitly that home ground advantage and performance pressure are difficult to record and map to individual players, and omit them. Second, existing formulations optimise over permissible ranges of role counts using population-based search, which suits fantasy contests but does not provide the exact role fulfilment and run-to-run reproducibility that selecting one fielded roster requires. Third, reliance on a single algorithm introduces model-specific bias, and existing research emphasizes international or fantasy contexts over regional team dynamics.

3. Methodology

3.1 Problem Formulation

Let $P = \{p_1, p_2, \dots, p_n\}$ represent the set of n available players in the regional cricket pool. Each player p_i is characterized by a feature vector $F_i \in \mathbb{R}^m$ containing $m = 35$ performance attributes:

$$F_i = [\text{batting, bowling, fielding, contextual}]^T \quad (1)$$

The objective is to select an optimal subset $S \subseteq P$ of exactly 11 players maximizing overall team fitness, subject to role-distribution constraints specified by the user. $R = \{r_1, \dots, r_7\} = \{\text{Wicket-Keeper (WK), Opener (O), Middle-order (M), Finisher (F), All-rounder (AR), Pace bowler (P), Spin bowler (S)}\}$ For each role, users specify a desired count $N_r \in \mathbb{Z}^+$, subject to:

$$\sum_{r=1}^7 N_r = 11 \quad (\text{Constraint 1}) \quad (2)$$

A duplicate-prevention constraint ensures that each player can be in at most one role group $S_r \subseteq S$:

$$\sum_{r=1}^7 1[p_i \in S_r] \leq 1, \forall p_i \in P \quad (3)$$

That flexibility means coaches can set the team up on the basis of the pitch, the opponent's strengths and the match situation.

3.2 Multi-Dimensional Fitness Index

A comprehensive fitness index $F(p_i)$ for each player p_i is defined as a weighted combination of four components:

$$F(p_i) = w_B \cdot BI(p_i) + w_{BW} \cdot BWI(p_i) + w_F \cdot FI(p_i) + w_C \cdot CI(p_i) \quad (4)$$

where BI, BWI, FI and CI are the Batting, Bowling, Fielding and Context indices with weights 0.70, 0.50, 0.20 and 0.35 respectively, decided after consultation with the experts. They don't add to 1.0 due to component overlap: all-rounders contribute to both batting and bowling, so the weighted sum captures multi-dimensional value rather than probability. Maximum possible contributions are 49.0 (batting), 25.0 (bowling), 4.0 (fielding) and 14.0 (context) for a theoretical maximum of 92.0. The batting-heavy profile is intentional in light of the ODI selection and is considered in the interpretation of the role-wise scores in §5.4.

3.2.1 Batting Index (BI)

The Batting Index is a composite of six normalised batting statistics:

$$BI(p_i) = 100 \times \left[0.25 \left(\frac{R_i}{R_{max}} \right) + 0.15 \left(\frac{A_i}{A_{max}} \right) + 0.15 \left(\frac{SR_i}{SR_{max}} \right) + 0.05 \left(\frac{H_{50,i}}{H_{50,max}} \right) + 0.05 \left(\frac{H_{100,i}}{H_{100,max}} \right) + 0.05 \left(\frac{B\%_i}{B\%_{max}} \right) \right] \quad (5)$$

where R = career runs (max 10,000); A = batting average (max 60); SR = strike rate (max 180); H50, H100 = 50+ and 100+ scores (max 50 / 25); B% = boundary percentage (max 25%). Empirical maxima are normalised for comparable scales. The weighting prefers consistency (runs 0.25, average 0.15) over pure aggression, as ODI cricket is a balance of accumulation and acceleration, unlike T20 where the strike rate would normally have a weight of 0.30 or more.

3.2.2 Bowling Index (BWI)

$$BWI(p_i) = 100 \times \left[0.20 \left(\frac{W_i}{W_{max}} \right) + 0.10 \left(\frac{BSR_{max} - BSR_i}{BSR_{max}} \right) + 0.05 \left(\frac{BA_{max} - BA_i}{BA_{max}} \right) + 0.05 \left(\frac{ER_{max} - ER_i}{ER_{max}} \right) + 0.05 \left(\frac{5W_i}{5W_{max}} \right) + 0.05 \left(\frac{10W_i}{10W_{max}} \right) \right] \quad (6)$$

where W = career wickets (max 400); BSR = bowling strike rate (max 60); BA = bowling average (max 50); ER = economy rate (max 9.5); 5W, 10W = five- and ten-wicket hauls (max 10 / 5). Performance is better if strike rate, average and economy are lower, so an inverse (Max-Value)/Max transform is applied. Wicket-taking (0.20) carries the most weight, since taking wickets breaks partnerships and creates pressure.

3.2.3 Fielding Index (FI)

$$FI(p_i) = 100 \times \left[0.05 \left(\frac{C_i}{C_{max}} \right) + 0.05 \left(\frac{RO_i}{RO_{max}} \right) + 0.05 \left(\frac{ST_i}{ST_{max}} \right) + 0.05 \left(\frac{FR_i}{100} \right) \right] \quad (7)$$

C = career catches (max 100) RO = run-outs (max 30) ST = stumping's (max 15) FR = expert-assessed fielding rating (0-100) Fielding is less important than batting or bowling, but still important. Good fielding saves runs, and gives more opportunities for dismissals.

3.2.4 Context Index (CI)

$$CI(p_i) = 100 \times \left[0.10 \left(\frac{RF_i}{100} \right) + 0.05PS_i + 0.05OI_i + 0.05PH_i + 0.05SY_i + 0.05(1 - FT_i) + 0.02HA_i + 0.03 \left(\frac{MOM_i}{20} \right) \right] \quad (8)$$

where all the terms are scaled to [0,1]: RF= recent form (last 10 matches); PS= pitch suitability; OI= opposition impact; PH= pressure handling; SY= Teammate synergy; FT= fatigue (inverted so fresher players score higher); HA= home advantage; MUM= Man of the Match awards (capped at 20) The paper's main novelty is the Context Index, which allows the system to better promote in-form players who are better suited to the current conditions, over historically strong players who are currently struggling. Match specific readiness is accounted for by recent form (0.10) and pitch suitability (0.05), whilst fatigue (0.05) penalises overworked players.

3.3 Ensemble Machine Learning Models

While the fitness index provides a domain-expert scoring mechanism, ensemble ML learns complementary feature weights from historical player-performance data.

3.3.1 Base Learners

Random Forest (RF)

An ensemble of $B = 100$ decision trees trained using bootstrap aggregation:

$$\hat{f}_{RF}(x) = \left(\frac{1}{B}\right) \sum_{b=1}^B T_b(x) \quad (9)$$

Each tree T_b is trained on a bootstrap sample $D_b \sim D$ with a random feature subset considered at each split. RF captures non-linear relationships and provides feature-importance estimates via mean decrease in impurity.

Gradient Boosting (GB)

A sequential ensemble minimizing squared-error loss:

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x) \quad (10)$$

where h_m is a depth-5 weak learner fitted to the negative loss gradient $-\partial L / \partial F_{m-1}$. Learning rate $\nu = 0.1$, $m = 100$ boosting rounds. GB captures complex feature interactions through additive modelling.

Ridge Regression

A regularized linear baseline:

$$\hat{\beta}_{\text{Ridge}} = \operatorname{argmin}_{\beta} \{ \sum (y_i - x_i^T \beta)^2 + \lambda \sum \beta_j^2 \} \quad (11)$$

with regularization parameter $\lambda = 1.0$, providing stable coefficient estimates and preventing overfitting in high-dimensional feature space.

3.3.2 Ensemble Meta-Learning

A VotingRegressor combines RF, GB, and Ridge predictions via simple averaging:

$$F_{\text{ensemble}}(x) = (1/3)[\hat{f}_{\text{RF}}(x) + \hat{f}_{\text{GB}}(x) + \hat{f}_{\text{Ridge}}(x)] \quad (12)$$

Equal weights prevent any single model from dominating, combining Random Forest's handling of non-linearities, Gradient Boosting's interaction effects and Ridge's linear stability.

3.3.3 Training Procedure and Target Variable

In deployment the target y_i represents realised match performance. For this study, where per-match records were unavailable, y_i is defined as the player's recent-window performance index — a normalised composite of scoring output, wicket-taking, and dismissals over the most recent ten-match window — and this variable together with all features derived from it is excluded from the model input set. Input features are restricted to career-aggregate batting, bowling, and fielding statistics and static contextual attributes. This separation ensures the ensemble is not trained to reconstruct a deterministic function of its own inputs; the learning task is instead to predict recent output from career profile. We note this as a limitation relative to a fully outcome-supervised formulation and address it in §6.4.

The ML models are trained on a 27-feature subset of the full 35-attribute profile, spanning: batting (runs, average, strike rate, 50s, 100s, boundary percentage), bowling (wickets, average, strike rate, economy), fielding (catches, run-outs, fielding rating), and context (match experience, pitch suitability, opposition impact, pressure handling, synergy, fatigue, home advantage, Man of the Match awards). Recent form is excluded from the feature set, as it is derived from the same window as the target. The remaining attributes in the full feature vector inform the expert fitness formula (§3.2) but are not passed directly to the ensemble regressors.

3.4 Hybrid Fitness Scoring

The final selection combines the ML predictions and the expert fitness index:

$$F_{\text{hybrid}}(p_i) = w_{\text{ML}} \cdot F_{\text{ML}}(p_i) + w_{\text{expert}} \cdot F_{\text{normalized}}(p_i) \quad (13)$$

where $F_{\text{ml}}(p_i)$ is the ML ensemble prediction on a 0–100 scale, $F_{\text{normalized}}(p_i) = [F(p_i) / \max_j F(p_j)] \times 100$ is the min–max normalized expert fitness, and $w_{\text{ml}} = 0.6$, $w_{\text{expert}} = 0.4$. The ML component captures data-driven feature interactions learned from historical performance; the expert component embeds cricket-specific strategic knowledge that raw statistics may not capture. The 60–40 split reflects greater confidence in the ML component's predictive capability while preserving substantial weight for strategic human judgment — a human-in-the-loop design rather than a pure black-box automation. $F_{\text{ml}}(p_i)$ is min–max normalised across the player pool to $[0,100]$ prior to combination, ensuring $F_{\text{hybrid}}(p_i) \in [0,100]$.

Computational complexity: sorting each role group costs $O(|Pr| \log |Pr|)$, so total complexity is bounded by $O(n \log n)$. This runs in milliseconds for a regional scale pool, well below population based metaheuristics such as genetic algorithms ($O(\text{pop} \times \text{gen} \times n)$) or particle swarm optimisation which typically take seconds to converge. We report measured runtimes in Secs. 5.6 and 5.8.

3.5 Role-Constrained Team Selection Algorithm

Team selection must satisfy strict positional constraints: exactly 11 players in total, with no duplicate players across roles. Typical regional configurations fall within the following ranges, though the implementation accepts any user-specified allocation satisfying Constraint 1 (§3.1): Wicket-Keeper: exactly 1; Openers: 2–3; Middle-Order: 1–3; Finisher: 1–2; All-Rounder: 1–2; Spinner: 1–3; Pacer: 1–3.

Selection algorithm: load player data with hybrid fitness scores; partition players by designated role; sort each role group in descending order of hybrid fitness; greedily select the top-ranked players per role up to the specified count, tracking selected IDs to prevent duplicates; validate the final team against all role requirements; and return the 11-player team with roles and fitness scores. This ensures optimality within each role group while satisfying hard constraints, with $O(n \log n)$ efficiency and deterministic, reproducible selections.

Example Configuration: Wicket-Keeper: 1, Opener: 2, Middle-Order: 1, Finisher: 1, All-Rounder: 2, Spinner: 1, Pacer: 3. Total: 11 players.

4. Implementation

4.1 System Architecture

The system was implemented as an interactive desktop application in Python 3.9 using pandas, numpy, scikit-learn and tkinter. The ensemble comprises a RandomForestRegressor (100 estimators, max depth 15), a GradientBoostingRegressor (100 estimators, learning rate 0.1) and a Ridge regressor ($\alpha = 1.0$), combined via a VotingRegressor with StandardScaler-normalized features.

4.2 User Interface Components

The Data Import panel provides a file selector, a status indicator confirming the number of players loaded, and ML training confirmation. The Team Configuration panel offers per-role count selectors with real-time validation against the 11-player total, and the Selected Team display presents a sortable, exportable table.

4.3 Application Workflow

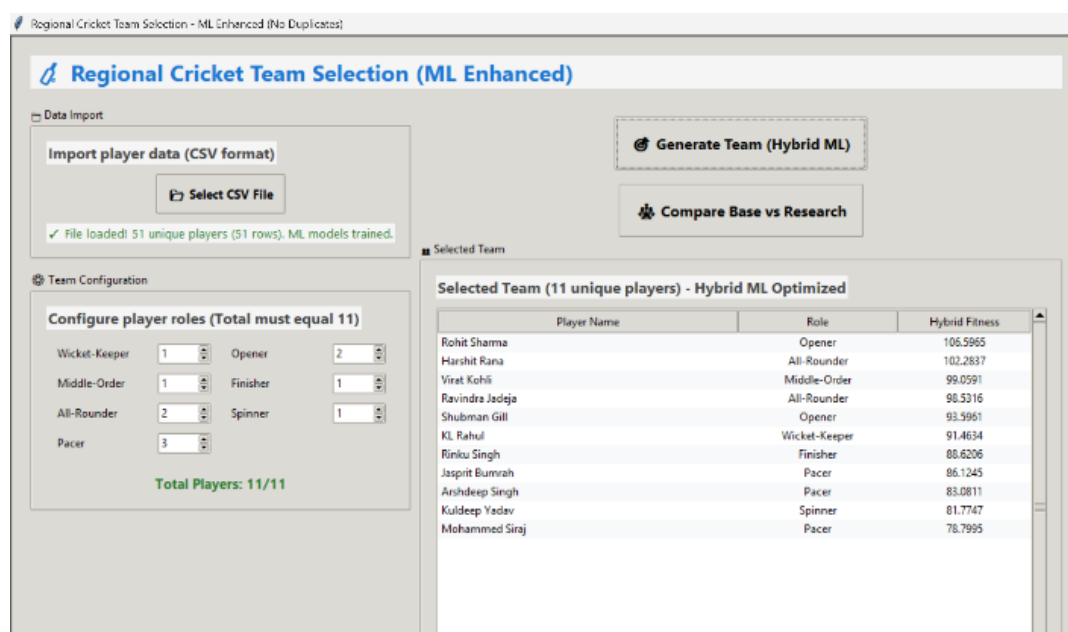


Figure. 1 Screenshot of the Regional Cricket Team Selection application, showing 51 imported players, the team configuration panel, and the optimized 11-player team with hybrid fitness scores

The application supports CSV import with automatic validation of required columns and data types, one-click ensemble training, interactive role configuration, hybrid team generation with constraint enforcement, and tabular display of the result.

5. Results and Analysis

5.1 Experimental Setup

The hybrid ML model was evaluated on a 51-player regional cricket dataset for the role configuration: 1 Wicket-Keeper, 2 Openers, 1 Middle-Order, 1 Finisher, 2 All-Rounders, 1 Spinner, 3 Pacers. Four players carrying a generic “Batter” designation fall outside the seven-role taxonomy of §3.1 and are not eligible under this configuration; the Finisher role has a single candidate, so that position is determined by availability rather than ranking.

5.2 Model Performance

Table 1 reports predictive accuracy for each base learner and the voting ensemble under 5-fold cross-validation on the 51-player dataset, using the held-out performance target defined in §3.3.3.

Table 1. Cross-Validated Predictive Accuracy of Base Learners and Ensemble

Model	R ² (CV)	RMSE	MAE
Random Forest	0.826	12.96	9.66
Gradient Boosting	0.800	13.90	9.57
Ridge Regression	0.514	21.65	15.15
Voting Ensemble	0.809	13.59	10.46

The voting ensemble attains $R^2 = 0.809$, which does not exceed its strongest individual component (Random Forest, $R^2 = 0.826$): simple averaging pulls it toward the weaker linear learner (Ridge, $R^2 = 0.514$), offsetting the variance-reduction benefit of §3.3.2. We report this rather than selecting the best model post hoc; weighted ensembling is identified as a refinement in §7. Accuracy should be read against the sample size ($n = 51$) relative to the feature count (27).

Fig. 2 reports permutation importance for the Random Forest component, measured as the mean decrease in R^2 over 30 shuffles per feature. The ranking is dominated by batting average, bowling average, and pressure handling, confirming that the ensemble draws primarily on batting-side and contextual attributes rather than distributing weight uniformly across the feature set. Permutation importance is reported in preference to impurity-based importance, which is biased toward high-cardinality continuous predictors.

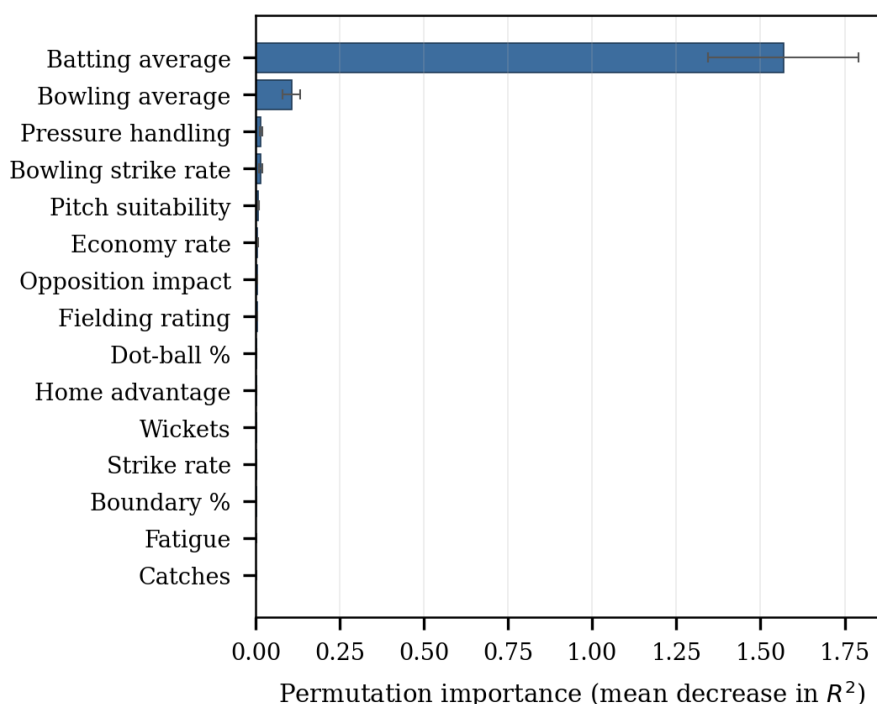


Figure. 2 Permutation feature importance for the Random Forest component (top 15 features; bars show mean decrease in R^2 , whiskers show standard deviation over 30 shuffles)

5.3 Optimized Team Composition

The hybrid ML model generated the following 11-player team:

Table 2. Hybrid ML-Optimized Team Selection with Role Distribution and Fitness Scores

Player Name	Role	Hybrid Fitness
Virat Kohli	Middle-Order	89.43
Rohit Sharma	Opener	84.43
Krunal Pandya	All-Rounder	75.73
Shubman Gill	Opener	73.26
Sanju Samson	Wicket-Keeper	72.20
Navdeep Saini	Pacer	61.28
Rinku Singh	Finisher	60.76
Harshit Rana	All-Rounder	53.40
Deepak Chahar	Pacer	42.68
Varun Chakaravathy	Spinner	29.84
Mohammed Shami	Pacer	29.38

All 11 players are unique and the role distribution exactly satisfies the specified constraint ($1+2+1+1+2+1+3 = 11$). Mean fitness is 61.12, median 61.28, range 29.38-89.43. The spread reflects the batting-weighted design of §3.2: specialist bowlers necessarily score lower than top-order batters on a composite scale, so within-role ranking rather than the absolute team mean is the meaningful quantity.

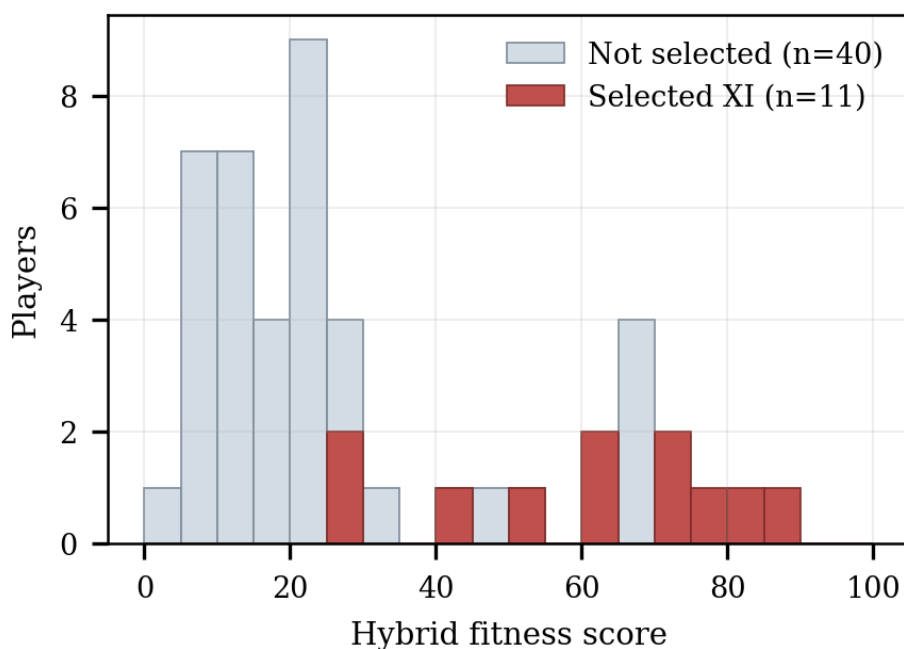


Figure. 3 Distribution of hybrid fitness scores across the 51-player pool, with the selected XI highlighted

Fig. 3 shows the selected XI against the full pool. The chosen players are not drawn exclusively from the upper tail: role constraints force selection of the best available candidate within each role group, so several picks sit mid-distribution. This is the intended behaviour of a role-constrained selector and is what an unconstrained top-11 ranking would fail to reproduce.

5.4 Role Distribution Analysis

Table 3. Role-Wise Average Fitness Scores in Optimized Team

Role	Count	Average Fitness
Opener	2	78.84
All-Rounder	2	64.56
Middle-Order	1	89.43
Wicket-Keeper	1	72.20
Finisher	1	60.76
Pacer	3	44.44
Spinner	1	29.84

Batting-oriented and all-rounder roles show the highest average fitness, consistent with the fitness index's batting-weighted design (§3.2). Bowling specialists have lower averages, which is a natural consequence of the weighting scheme in §3.2: batting can contribute as much as 49.0 index points, compared to 25.0 for bowling. This is a deliberate design choice reflecting ODI selection priorities, not an artefact — but it means role-wise fitness scores are comparable within a role rather than across roles. Cross-role comparison would require role-normalised scoring, identified as a refinement in §7.

5.5 Aggregate Team Statistics

Table 4. Aggregate Team Performance Statistics

Metric	Value
Total Career Runs	30,638
Average Batting Average	40.76
Average Strike Rate	90.93
Total Career Wickets	259
Average Economy Rate	6.09
Total Catches + Stumpings	357
Average Match Experience	74.0 matches

The duo has a combined 30,638 career runs and 259 wickets. Average batting average (40.76) and economy rate (6.09) are averaged across the 11 players and therefore are moderated by specialist bowlers with little batting output, and all-rounders with part-time bowling economy. Combined catches and stumpings total 357, with an average of 74.0 matches per player.

5.6 Baseline comparison

To isolate the contribution of each component, we evaluated four selectors on the same pool of 51 players and role configuration: Formula-Only (expert index only, §3.2), ML-Only (ensemble prediction only, §3.3), the proposed Hybrid (60:40, §3.4), and a genetic algorithm with soft penalties for role and uniqueness violations over 100 independent runs. The last is included as a representative penalty-based population metaheuristic, not as a reimplement of any published method; Jha et al. [1] instead embed feasibility in the crossover operator itself.

Table 5. Comparison of Selection Methods on the Same Player Pool

Method	Mean team fitness	Constraint violations (100 runs)	Duplicate players	Mean runtime (ms)
Formula-Only	53.45	0	0	12.91
ML-Only	58.87	0	0	9.78
Genetic Algorithm [1]	83.68	100	666	2635.79
Hybrid (proposed)	61.12	0	0	9.32

The penalty-based genetic algorithm produced role-count violations or duplicate selections in 100 of 100 runs, generating 666 duplicates in total. Its nominally higher mean fitness (83.68 against 61.12) follows directly: it optimises

over a strictly larger, partly infeasible search space in which a high-scoring player may be selected repeatedly and role quotas may go unmet, so the two figures are not comparable as measures of selection quality. Feasibility can instead be built into the search operator, as Jha et al. [1] do, or into the selection logic as proposed here; the latter additionally gives identical output on repeated runs and completes roughly 283 times faster.

Team composition overlap was 6 of 11 players with Formula-Only and 9 of 11 with ML-Only, indicating the hybrid weighting produces a selection distinct from either component alone.

5.7 Sensitivity to the Hybrid Weighting

The 60:40 weighting (§3.4) was set a priori from expert consultation. To test whether it is defensible, we swept w_{ml} from 0.0 to 1.0 in increments of 0.1, recording the resulting team composition and mean fitness.

Table 6. Team Composition Stability Across Hybrid Weightings

w_{ML}	Mean team fitness	Positions identical to 60:40
0.0	53.45	6/11
0.1	56.57	7/11
0.2	59.77	8/11
0.3	59.77	8/11
0.4	59.77	8/11
0.5	60.68	10/11
0.6	61.12	11/11 (reference)
0.7	61.12	11/11
0.8	60.50	10/11
0.9	58.87	9/11
1.0	58.87	9/11

The choices were stable for w_{ml} in 0.5-0.8 with at least 10 of 11 positions unchanged from the 60:40 configurations. The chosen weighting is therefore in a stable range instead of a sensitive point, which supports its use as a default. The composition only differs significantly at the ends, with one component being effectively discarded: at $w_{ml} = 0.0$ only 6 of 11 positions coincide with the 60:40 team, and at $w_{ml} = 1.0$, 9 of 11.

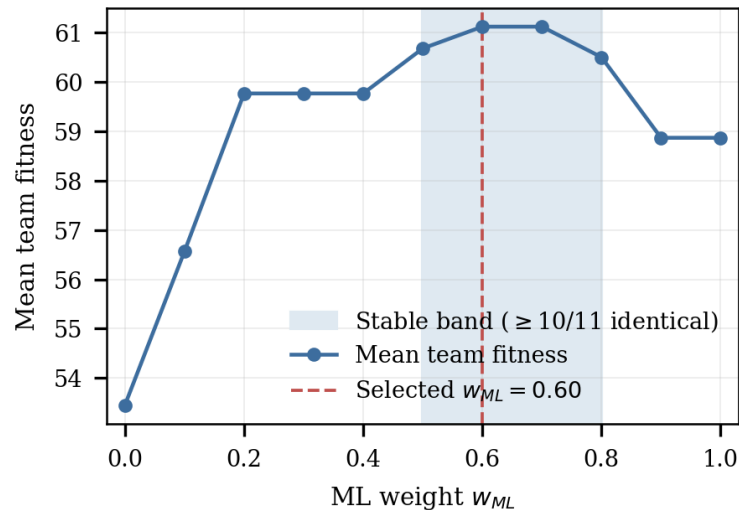


Figure. 4 Mean team fitness across the ML–formula weighting sweep, with the stable band and the selected weighting indicated

5.8 Scalability on a Simulated Regional Pool

The empirical dataset of §5.1 is small ($n = 51$) and unevenly distributed across roles, with only one candidate for the Finisher role. We simulated a pool of 300 players and applied the identical pipeline to investigate behaviour at a pool size closer to a full regional squad list, and to test whether role coverage affects selection quality.

This is a synthetic pool. It is reported as such. No empirical claim is made about individual cricketers. Instead of international ranges with career runs of 45-4273, batting averages of 0.5-54.8 and match experience of 4-122 matches, the attributes were sampled from distributions calibrated to Indian domestic (Ranji Trophy and Vijay Hazare) ranges. The proportions of the roles follow a conventional composition of a squad and a latent quality term per player induces correlation between career statistics and contextual attributes. The generator is seeded (seed = 20260803) so the pool is exactly reproducible.

Table 7. Selection Behaviour on the Empirical and Simulated Pools

Property	Empirical (n=51)	Simulated (n=300)
Ensemble R^2 (5-fold CV)	0.809	0.261
Selected XI mean fitness	61.12	75.57
Selected XI fitness range	29.38–89.43	61.30–90.78
Finisher candidates	1	30
Spinner candidates	5	42
Greedy selection runtime	9.32 ms	7.34 ms
GA runtime	2636 ms	2208 ms
GA constraint violations	100 / 100	100 / 100
Duplicate-free selection	Yes	Yes

Hence three observations follow: First, the selection algorithm scales as predicted: greedy selection required 7.34 ms at $n = 300$ against 9.32 ms at $n = 51$, consistent with the $O(n \log n)$ analysis of §3.4, where runtime at this scale remains dominated by fixed overhead. The genetic-algorithm baseline again violated role or uniqueness constraints in 100 of 100 runs.

Second, role coverage materially improves selection quality. With 30 Finisher candidates rather than one, and 42 Spinners rather than five, the weakest positions are no longer determined by availability: mean fitness rises from 61.12 to 75.57 and the spread narrows from 29.38-89.43 to 61.30-90.78 (Fig. 5).

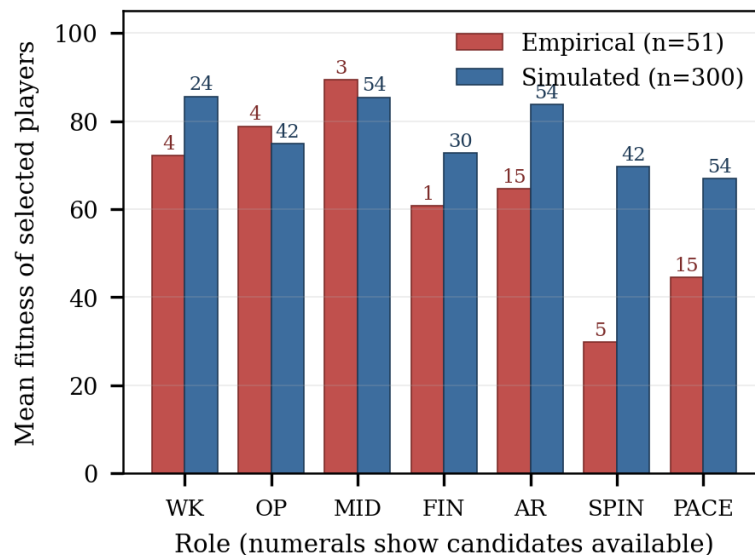


Figure. 5 Role-wise mean fitness of the selected XI on the empirical and simulated pools; numerals above each bar give the number of candidates available for that role

Third, and least favourably, predictive accuracy falls sharply: the ensemble attains $R^2 = 0.261$ on the simulated pool against 0.809 on the empirical one. In the empirical dataset the recent-form target is strongly associated with career profile, making the regression task comparatively easy; in the simulated pool, form derives from a latent quality term with

independent noise. The empirical R^2 should therefore be read as an upper bound conditioned on that dataset's structure rather than a general estimate of predictive power.

The simulation demonstrates scalability, constraint satisfaction at larger n , and the value of role coverage, but cannot substitute for validation on real regional players (§6.4, §7).

6. Discussion

6.1 Key Findings

A hybrid approach combining ensemble ML predictions with an expert-derived fitness formula produces role-constrained, duplicate-free teams from a modest pool (mean fitness 61.12, median 61.28). Three design factors contribute: ensemble diversity; contextual features covering pitch suitability, opposition impact, fatigue and home advantage; and deterministic constraint enforcement, giving exact role counts and reproducible output without penalty tuning or feasibility repair.

6.2 Comparison with the Base Paper

Jha et al. [1] target fantasy contests, where participants draw players from both competing sides under a credit budget, role quotas are specified as ranges, and feasibility is maintained within the NSGA-II crossover operator. The present work targets a single fielded roster with exact, user-specified role counts, enforcing uniqueness and role fulfilment directly in the selection logic (§3.5) so that output is deterministic and reproducible across runs. The two also differ in evaluation basis: Jha et al. score players against a platform-specific points system, whereas this work combines an expert fitness index with ensemble prediction and incorporates the contextual factors that Jha et al. identify as difficult to record and therefore exclude. Section 5.6 contains a quantitative comparison with baseline selectors.

6.3 Practical Implications

For regional boards, a data-driven process lessens the need for subjective judgement and allows for quick scenario testing such as assessing injury replacements. The application only needs a CSV import and some clicks for coaches. Configurable role requirements can handle different formats. The fitness score is decomposable and gives the players feedback on specific dimensions of performance.

6.4 Limitations

Data dependence. Effectiveness requires a complete set of player statistics. Regions that lack data infrastructure may not be able to populate all 35 attributes, but missing contextual features can be assigned neutral values.

Limited sample size. The dataset is small for ML training (27 features, 51 players) and for drawing general conclusions about model performance; the results should be read as a proof-of-concept on a single regional pool, rather than a validated benchmark. The cross-validated R^2 reported in §5.2 is evidence of signal, not a deployment-grade accuracy estimate, and the simulated pool (§5.8) shows that it is specific to the structure of this dataset. Bigger pools will improve generalisation and reduce the risk of over-fitting.

Aggregate statistics Interpretation (Table 4) Team averages are for all 11 players, regardless of role, which moderates specialist-position numbers.

Target construction. In the absence of ball-by-ball match records, the training target is a recent-window performance composite rather than a direct win-contribution measure (§3.3.3). It is not included in the feature set to prevent leakage but remains a proxy and other outcome-supervised targets such as measured win contribution may produce different optimal teams.

Rigid roles, fixed inputs. The constraint set is based on fixed role categories and as such may not fully reflect the modern day flexible batting orders and multi-role specialists. Its implementation relies on historical statistics and could be behind the curve regarding recent changes in form, and using in-tournament data streams would make it more responsive.

7. Future Work

The results in §5 show that the selection algorithm is structurally sound, computationally efficient, and satisfies all constraints where the genetic-algorithm baseline fails to do so. Next steps are identified in the following directions, in order of priority:

Real-pool validation - replicating the §5.6 comparison on an independent pool of domestic players. The simulation in §5.8 addresses scale but not ecological validity, so this remains the principal outstanding requirement.

Weighted ensembling - fitting ensemble weights on validation folds, since simple averaging currently pulls the ensemble below its strongest component (§5.2).

Dynamic fitness updating - incorporating in-tournament performance into the recent-form component of the Context Index (§3.2.4).

Explainable AI - SHAP or LIME analysis of ensemble predictions for coaching transparency and bias auditing.

Role flexibility - allowing multi-role eligibility and role-normalised scoring rather than fixed categories (§6.4).

8. Conclusion

This paper presented a hybrid ensemble machine learning approach for regional cricket team selection, combining Random Forest, Gradient Boosting and Ridge Regression predictions with an expert-derived fitness formula through a 60:40 weighted score. Exact positional counts and player uniqueness are enforced directly within the selection algorithm, giving deterministic and reproducible selections, in contrast to the population-based search used for fantasy team optimization by Jha et al. [1].

Applied to a 51-player dataset, the method produced an 11-player team meeting every role requirement with no duplicate selections, implemented as an interactive desktop application. Against Formula-Only, ML-Only and penalty-based genetic-algorithm baselines on the same data, it satisfied all constraints on every run while the penalty-based baseline violated them in all 100 trials (§5.6). A simulated 300-player pool confirmed that selection scales and that adequate role coverage materially improves selection quality (§5.8).

The work should be read as a proof-of-concept on a single, modest-sized player pool. Validation on real domestic players and longitudinal tracking of selected teams against match outcomes remain the most important next steps (§7) toward substantiating performance advantages beyond the structural guarantees demonstrated here.

Declaration of AI-Assisted Tools

During the preparation of this work, the authors used AI-assisted tools to support language refinement, verification of bibliographic details, preparation of figures and tables, and review of the analysis code. No text, data, or result was accepted without author verification: all reported values were produced by the authors' own implementation on the datasets described in §5, and all references were checked against the original publications. The authors reviewed and edited the content and assume full responsibility for the accuracy and integrity of the published article.

Conflicts of Interest

The authors have no conflicts of interest to disclose.

Author Contributions

Conceptualization, RD and AL; methodology, RD; software, RD; validation, RD and AL; formal analysis, RD; investigation, RD; resources, RD and AL; data curation, RD; writing—original draft preparation, RD; writing—review and editing, AL; visualization, RD; supervision, AL; project administration, AL.

References

- [1] A. Jha, A. K. Kar, and A. Gupta, "Optimization of Team Selection in Fantasy Cricket: A Hybrid Approach Using Recursive Feature Elimination and Genetic Algorithm", *Annals of Operations Research*, Vol. 325, pp. 289-317, 2023.
- [2] A. Adhikari, A. Majumdar, G. Gupta, and A. Bisi, "An Innovative Super-Efficiency Data Envelopment Analysis, Semi-Variance, and Shannon Entropy-Based Methodology for Player Selection: Evidence from Cricket", *Annals of Operations Research*, Vol. 284, pp. 1-32, 2020.
- [3] A. J. Lewis, "Towards Fairer Measures of Player Performance in One-Day Cricket", *Journal of the Operational Research Society*, Vol. 56, No. 7, pp. 804-815, 2005.
- [4] G. D. Barr, C. Holdsworth, and B. S. Kantor, "Evaluating Performances at the 2007 Cricket World Cup", *South African Statistical Journal*, Vol. 42, No. 2, pp. 125-142, 2008.
- [5] G. Singh, N. Bhatia, and S. Singh, "Fuzzy Logic Based Cricket Player Performance Evaluator", *International Journal of Computer Science Issues*, Vol. 8, No. 5, pp. 11-16, 2011.
- [6] G. D. Sharp, W. J. Brettigny, J. W. Gonsalves, and M. Lourens, "Integer Optimisation for the Selection of a Twenty20 Cricket Team", *Journal of the Operational Research Society*, Vol. 62, No. 9, pp. 1688-1694, 2011.
- [7] H. H. Lemmer, "Team Selection After a Short Cricket Series", *European Journal of Sport Science*, Vol. 13, No. 2, pp. 200-206, 2013.
- [8] S. R. Iyer and R. Sharda, "Prediction of Athletes' Performance Using Neural Networks: An Application in Cricket Team Selection", *Expert Systems with Applications*, Vol. 36, No. 3, pp. 5510-5522, 2009.
- [9] F. Ahmed, K. Deb, and A. Jindal, "Multi-Objective Optimization and Decision Making Approaches to Cricket Team Selection", *Applied Soft Computing*, Vol. 13, No. 1, pp. 402-414, 2013.

- [10] S. N. Omkar and R. Verma, "Cricket Team Selection Using Genetic Algorithm", in *Proc. of International Congress on Sports Dynamics*, pp. 1-9, 2003.
- [11] T. G. Dietterich, "Ensemble Methods in Machine Learning", in *Proc. of International Workshop on Multiple Classifier Systems*, Springer, pp. 1-15, 2000.
- [12] Z. H. Zhou, *Ensemble Methods: Foundations and Algorithms*, Chapman and Hall/CRC, 2012.
- [13] Y. Shuai, C. Yang, C. Yuan, and T. Song, "A Hybrid Electronic Components Detection Parameter Sequence Optimization Method Based on Random Forest and Mean Decrease Accuracy", in *Proc. of the 2019 IEEE 10th International Conference on Software Engineering and Service Science*, pp. 1-4, 2019.
- [14] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine", *Annals of Statistics*, Vol. 29, No. 5, pp. 1189-1232, 2001.
- [15] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems", *Technometrics*, Vol. 12, No. 1, pp. 55-67, 1970.
- [16] G. Chandu and P. Nirmala, "Efficient Approach in Cricket Team Selection Methodology Using Machine Learning Based Logistic Regression Algorithm and Comparing with Random Forest Algorithm", *AIP Conference Proceedings*, Vol. 2816, No. 1, 080004, 2024.
- [17] S. Varanasi, S. K. Mutapally, R. Gudimalla, S. K. Bollavathri, S. Chekkilla, M. N. Paruti, V. Sadula, and A. Hamedshnain, "Cricket Player Performance Prediction Using Machine Learning", *AIP Conference Proceedings*, Vol. 3361, No. 1, 050084, 2025.
- [18] T. Weerakoon and T. Halloluwa, "Cricket Team Selection Based on Complex Dynamics Using Machine Learning", in *Innovations in Electrical and Electronics Engineering, Lecture Notes in Electrical Engineering*, Vol. 1295, Springer, Singapore, pp. 349-369, 2025.
- [19] K. Dhinakaran and S. Anbuchelian, "Enhanced Cricket Match Prediction Using Kernel Methods for Feature Extraction and Back-Propagation Neural Networks", *Scientific Reports*, Vol. 16, Article 6478, 2026.