An Explainable Adaptive Meta-Ensemble Framework with Hybrid Feature Selection for Accurate ODI Cricket Match Outcome Prediction

Deepa S[1*], Rukumani Khandhan C[2], Jamunadevi C[1], Arumugam M[1], Kathirmurugan A[1]
[1]Department of Computer Technology - PG, [2]Department of Computer Science and Engineering,
Kongu Engineering College (Autonomous), Perundurai, Erode, Tamil Nadu, India
*Corresponding Author Email: profsdeepa2020@gmail.com
Received: 13.08.2026 *Revised: 22.09.2026* *Accepted: 30.09.2026* *Published: 06.10.2026*

Abstract- Forecasts for One Day International (ODI) cricket matches are difficult since match outcomes are influenced by interactions between variables that vary over time, such as team strength, team form, venue quality, head-to-head records, and toss outcomes, whereas presently available methods involve static ensembles with poor explainability. In this paper, an Explainable Adaptive Meta-Ensemble Framework (EAMEF) is introduced, which utilizes dynamic engineering features unique to the domain of cricket, i.e., a Boruta-SHAP hybrid feature selection pipeline, three gradient boosting base learners (XGBoost, LightGBM, CatBoost), a softmax-weighted soft voting ensemble, and an embedded SHAP-based explanation module. Five contextual variables, such as Team Strength Index, Recent Form Index, Venue Quality Index, Head-to-Head Dominance Index, and Toss Effectiveness Index, are explicitly formulated and used in combination with the interaction and rolling-window statistics. The weights of the base learners are decided based on their cross-validated performance. Using an available ODI dataset, EAMEF secured 97.25% accuracy, 97.25% F1 score, 0.9978 ROC-AUC, and 0.945 MCC, beating the performance of Hard Voting, Soft Voting, Stacking, and Blending techniques significantly. Wilcoxon signed-rank tests proved the significance of the improvements. SHAP analysis revealed that team strength, form, home advantage, and head-to-head record were the leading factors in predictions.
Keywords: ODI cricket prediction; explainable artificial intelligence; SHAP; hybrid feature selection; gradient boosting.

1. Introduction

The advancements in high-performance computing and the gathering of sports data in big volumes have changed the nature of analysis of cricket from an intuition-oriented exercise to one based on data [1,2]. ODIs in cricket represent a very challenging prediction task due to the 50-over game that is divided into three different phases including Powerplay, middle-overs phase and death-overs phase where each phase has different requirements and with different predictive features [1,2]. A model tuned to early-innings would be inappropriate for late overs, limiting its application to real-time decisions [1]. Conventional approaches involved the utilization of pre-game predictors such as ranking of teams, home ground advantage and the outcome of the coin-toss event [3,4]. Although very useful as base indicators, these pre-game indicators are unable to account for momentum changes in the middle of the game or changes in tactics. Ensemble machine learning, in particular gradient boosting, has become the most common approach to outcome based prediction in sports [5], however, two main shortcomings remain in the cricket field i.e. (i) most of the existing approaches utilize only one static ensemble configuration without regard to the match situation, and (ii) gains in predictive performance are not accompanied by any decision usable explanations. This paper attempts to overcome both of the above-stated drawbacks by introducing the concept of the Explainable Adaptive Meta-Ensemble Framework (EAMEF), which integrates context-driven dynamic feature engineering, hybrid Boruta and SHAP feature selection, adaptive ensemble weighting based on performance, and an explanation capability into one ODI outcome prediction. The paper is structured as follows: Section 2 provides an overview of related works and states the research gap and goal; Section 3 introduces the methodology; Section 4 introduces the EAMEF algorithm; Section 5 describes the experimental design; Section 6 presents and discusses the results; and Section 7 is the conclusion section.

2. Related Works

Wickramasinghe [8] demonstrated a Naive Bayes classifier could predict ODI outcomes reasonably well from pre-match statistics, and a later systematic review [3] confirmed the breadth of machine-learning applications across cricket formats. Ul Haq et al. [4] has compared Decision Trees, Random Forests, and gradient boosting variants for ODI outcome prediction and found that ensemble methods consistently outperformed simpler classifiers. More recent work has emphasized in-play, state-aware predictions. Agrawal and Kandhway [1] analysed 1,359 historical ODI matches (2004–2022) and showed that the prediction accuracy rises monotonically as a match progresses, with in-play indicators such as run rate and wickets in hand eventually dominating static pre-match indicators. Their

Open Access: Distributed under the CC BY License, permitting use and reproduction with credit to the authors and source

companion study on T20 internationals [2] reported accuracy rising from near chance in the first interval to over 85% in the closing overs, and found that the second-innings models outperformed first-innings models because of the availability of a run target. Chakraborty et al. [29] similarly found gradient boosting classifiers to outperform classical classifiers for T20 prediction once state-level features were included. Together, these studies motivate the state-aware, dynamic feature engineering adopted in this work. The value of ensemble learning for outcome prediction extends beyond cricket. Tax and Joustra [7] used supervised classification on Dutch football data to show the transferability of ensemble methods to sports with complex contextual dynamics. Bunker and Susnjak [5] reviewed machine-learning techniques for team-sport outcome prediction and identified gradient boosting and ensemble methods as consistently strong performers across sports. Davis et al. [6] provided a methodological overview of sports-analytics challenges, including temporal dependency, class imbalance, and causal-factor isolation, and stressed the need for rigorous evaluation. Constantinou and Fenton [30] showed that contextually engineered features improve long-term football performance prediction over raw statistics, supporting the feature-engineering strategy. Lalwani et al. [19] and Carvalho et al. [21] applied SHAP-based explainable models to volleyball and swimming performance respectively, resulting that interpretable models can match black-box accuracy while providing actionable insight i.e. a cross-sport precedent for the explainability layer central to EAMEF.

The method of ensemble learning is based on reducing the variance and the bias by using the various base learners [9,10]. The Random Forest algorithm proposed by Breiman [31] showed that diversity in bagging was beneficial, whereas the gradient boosting approach of Friedman [32] brought boosting to structured predictions, and the stacking of Wolpert [28] initiated meta-learning from base learner output. Within the gradient-boosting family, XGBoost [25] introduced scalable regularised tree boosting, LightGBM [22] introduced leaf-wise tree growth for computational efficiency, and CatBoost [26] introduced ordered boosting for robust categorical-feature handling; these complementary strengths motivated their combined use as EAMEF's base learners. Outside sports, Ahmed et al. [12,13] combined bagging and boosting with SHAP-guided feature selection for network intrusion detection, achieving accuracy up to 98.47% and demonstrating that XAI-guided feature selection improves robustness and reduces overfitting; Khattak et al. [20] applied a self-paced ensemble with SHAP explanations to aviation-turbulence prediction. These cross-domain results directly inspire EAMEF's adaptive, performance-weighted ensemble design. Few metaheuristic feature selection methods are used to reduce redundant features before deep learning or CNN models, which are mainly used for complex sequential data [11]. Lundberg and Lee [15] introduced SHapley Additive exPlanations (SHAP) as a unified, game-theoretically grounded approach to model interpretation, and Lundberg et al. [14] extended it to efficient, consistent explanations for tree-ensemble models, the variant used in this study. SHAP-based explainability has since been applied to heart-disease prediction [17,18], financial fraud detection [16], and educational analytics [34], consistently improving practitioner trust and supporting feature refinement. In cricket specifically, Agrawal and Kandhway [1,2] used SHAP to show that toss importance declines and in-play indicators (wickets fallen, run rate) rise in importance as a match progresses, a temporal dependency that motivates EAMEF's embedded, per-prediction SHAP layer rather than a purely post-hoc one.

Chandrashekar and Sahin [23] classify feature-selection methods into filter, wrapper, and embedded categories, and Guyon and Elisseeff [27] showed that wrapper and embedded methods are particularly important in high-dimensional classification settings; Saeys et al. [35] also provided similar conclusions in bioinformatics and recommend combining selection paradigms to offset their individual weaknesses. The Boruta algorithm [24] extends the random-forest framework into an all-relevant feature-selection method by comparing observed features against randomised shadow features, making it robust to spurious correlations. EAMEF combines Boruta with SHAP-based importance ranking, following the hybrid-selection pattern established by Ahmed et al. [13] in intrusion detection, but applies it for the first time to ODI match-outcome data. The reviewed literature establishes that dynamic, state-aware feature engineering [1,2], gradient-boosting ensembles [4,5], and SHAP-based explainability [14,15] each independently improve cricket outcome prediction, but four gaps remain unaddressed jointly: (i) most ODI frameworks use a single, statically weighted ensemble configuration regardless of match context [1,2]; (ii) performance-adaptive weighting, though effective in other domains [12,13], has not been systematically applied to cricket prediction; (iii) existing cricket studies rely on single-criterion feature selection rather than a hybrid Boruta–SHAP pipeline [1,2]; and (iv) SHAP has typically been used in a post-hoc, exploratory role rather than as an embedded, per-prediction component of the pipeline [1,2]. To address these gaps, this paper proposes EAMEF, the objective is to mutually improve the predictive accuracy and interpretability for ODI outcome prediction by combining cricket-specific dynamic feature engineering, hybrid Boruta–SHAP feature selection, an adaptive softmax-weighted ensemble of XGBoost, LightGBM, and CatBoost, and an embedded, multi-level SHAP explainability layer, benchmarked against conventional ensemble baselines using accuracy, F1-score, ROC-AUC, MCC, Cohen's Kappa, and the Wilcoxon signed-rank test.

3. Proposed Methodology

The EAMEF methodology: the problem definition, data preprocessing, dynamic feature engineering, hybrid feature selection, base-learner training, adaptive ensemble weighting, and the explainability layer are the main methods in this work. Figure 1 summarizes the overall architecture.

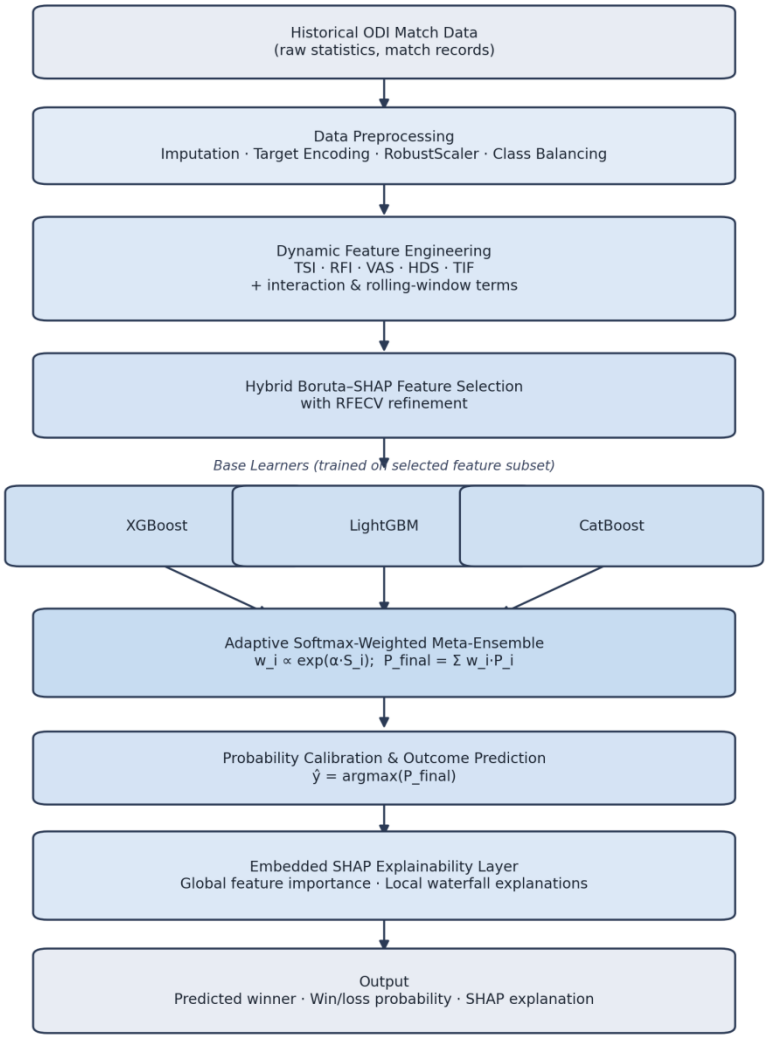


Figure 1. EAMEF Framework Architecture

3.1 Problem Definition

The ODI dataset is given as $D = \{(x_i, y_i)\}$, $i = 1, \ldots, N$, where N defines the number of match instances, $x_i = [f_1, f_2, \ldots, f_m]$ is the feature vector for match i (comprising raw match statistics and the engineered contextual indicators defined in Section 3.3), and $y_i \in \{0, 1\}$ is the outcome label (1 = win, 0 = loss) for the team of interest. The framework learns a mapping $f(\cdot)$ such that $\hat{y}_i = f(x_i)$, and additionally estimates the win probability $P(y = 1 \mid x_i)$, with $P(y = 0 \mid x_i) = 1 - P(y = 1 \mid x_i)$; the predicted class is $\hat{y} = \text{argmax}_y P(y \mid x_i)$. The design objectives are to (i) improve prediction accuracy and generalization, (ii) minimize error through effective feature selection and ensemble learning, (iii) produce reliable probability estimates, and (iv) generate SHAP-based explanations for each prediction.

3.2 Data Acquisition and Preprocessing

The ODI dataset under evaluation consist of 5,000 game cases. The experimental setting adopted an 80:20 train-test split approach to produce 4,000 training cases and 1,000 test cases. The test dataset was perfectly balanced and consists of 500 win cases and 500 loss cases. There is no trace of the exact public repository or source URL, seasons' range, list of the participating teams and venues, and number of features (both raw and engineered) in the existing experimental manuscript, and thus, these pieces of information are not verified beyond the experimental results reported. Duplicate or inconsistent cases were filtered out during the pre-processing phase. Numerical missing

values were imputed using median imputation and categorical missing values with the mode imputation, whereas features having more than 30% missing values were dropped. The categorical features (team, venue, and toss decision) were target encoded, whereas the numerical features were scaled using the RobustScaler. In case of any class imbalance in the training datasets, the class weighting or SMOTE was used only in the training folds to avoid information leakage. Missing numerical values were imputed using the median of the corresponding feature, and missing categorical values were imputed using the mode; any feature with more than 30% missing values was dropped. Duplicate records and internal inconsistencies were removed. Categorical variables (e.g., team, venue, toss decision) were encoded using target encoding rather than simple label encoding, since target encoding preserves the relationship between categorical levels and the outcome without imposing an artificial ordinal scale. Numerical features were scaled using a RobustScaler (median and interquartile range) rather than min–max scaling, since several engineered indicators (Section 3.3) contain outlier-sensitive ratios. Class imbalance, where present, was addressed using class-weighting or SMOTE oversampling on the training folds only, to avoid information leakage into validation or test data.

3.3 Dynamic Feature Engineering

Beyond raw match statistics, EAMEF constructs five context-aware indicators. Eachone is normalized (min–max or z-score, applied within the training fold only) before combination, and interaction terms (e.g., TSI×VAS, RFI×HDS) and rolling-window statistics over the preceding 5–10 matches are additionally derived.

- Team Strength Index (TSI): a weighted composite of batting, bowling, and fielding performance metrics and the team's current ICC ranking, $TSI = \alpha_1 \cdot BattingIndex + \alpha_2 \cdot BowlingIndex + \alpha_3 \cdot FieldingIndex + \alpha_4 \cdot (1/Rank)$, with weights α_k obtained empirically from feature-importance analysis (Section 3.4).
- Recent Form Index (RFI): an exponentially weighted moving average (EWMA) of match outcomes over the previous 5–10 matches, $RFI_t = \Sigma\, \lambda\text{^}k \cdot Result_(t-k)$, where $\lambda \in (0,1)$ is the decay factor and $Result \in \{1, 0\}$ for win/loss.
- Venue Advantage Score (VAS): a combination of the team's historical win ratio at the specific venue and a pitch/ground-condition suitability term, $VAS = \beta_1 \cdot WinRatio_venue + \beta_2 \cdot PitchSuitability$.
- Head-to-Head Dominance Score (HDS): a recency-weighted win ratio between the two competing teams over their previous N encounters, $HDS = \Sigma\, \lambda\text{^}k \cdot Result_(h2h,k) / N$.
- Toss Impact Factor (TIF): the conditional change in win probability associated with winning the toss and the resulting bat/bowl-first decision, estimated per venue from historical toss-outcome records.

3.4 Hybrid Feature Selection

The Boruta procedure works with each real feature together with its random shadow counterpart during multiple iterations of Random Forests and selects only such features which are more informative compared to the best shadow feature; this produces an all-relevant feature subset which is resistant to spurious correlation [24]. The ranking of features based on their SHAP values, calculated using a preliminary LightGBM model and TreeExplainer of Lundberg et al. [14], is another approach that gives a feature ranking based on the average absolute value of the feature contribution. The combination of the two feature rankings and selection via RFECV [23,27] leads to the final feature subset F.

3.5 Base Learning Models

EAMEF makes use of three gradient boosting algorithms. XGBoost [25] is known for regularized and parallelized tree boosting and thus it is good for structured tabular sports datasets. LightGBM [22] performs tree boosting using depth-wise splitting while having depth limits, providing faster training and lower memory usage with large sets of features. CatBoost [26] performs ordered tree boosting which decreases prediction bias for categorical features such as teams, venues, and decisions regarding the toss without doing any heavy preprocessing.

3.6 Adaptive Meta-Ensemble Learning

For each of the base classifiers i, the validation score S_i is calculated using 10 fold stratified cross validation that incorporates accuracy, log loss and Brier score. The adaptive weight is then

$$w_i = exp(\alpha \cdot S_i) / \Sigma_j\, exp(\alpha \cdot S_j), \quad for\ i \in \{1, 2, 3\},$$

where α is a temperature hyperparameter controlling how sharply weight concentrates on the best-performing learner. Each base learner produces a class-probability estimate P_i; the ensemble probability is the weighted sum $P_final = \Sigma_i\, w_i \cdot P_i$, followed by probability calibration (Platt scaling or isotonic regression) and final classification $\hat{y} = \text{argmax}(P_final)$. This adaptive scheme allocates greater influence to base learners that generalize better on held-out folds while remaining resilient to sampling variability.

3.7 Explainable Artificial Intelligence Layer

The interpretation of each base model (XGBoost, LightGBM, CatBoost) takes place separately from other models using the SHAP TreeExplainer [14] approach for trees, which allows computing Shapley values in polynomial time. For the estimation of expectations, a random sample from the training data is chosen. Since the probability

produced by the final ensemble is a linear combination of probabilities of the base models (Section 3.6), the corresponding SHAP value at the level of an ensemble is calculated as a weighted sum of the SHAP values of base models, maintaining the additive consistency property of SHAP. Mean absolute SHAP values are aggregated globally to assess feature importance, while the attribution of each prediction to the corresponding features is done locally (Section 6.6).

4. Proposed Algorithm

Figure 2 presents a simplified flowchart of the EAMEF pipeline, a concise step summary follows.

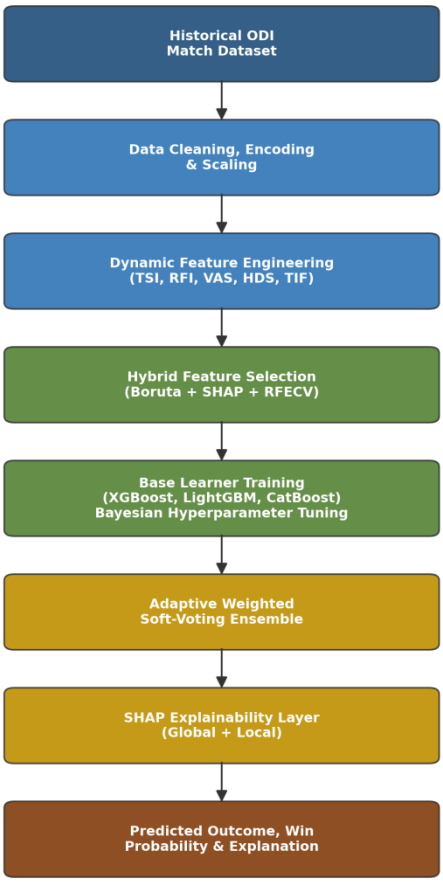


Figure 2. Simplified Flowchart of the EAMEF Pipeline

- Load and pre-process the historical ODI dataset (Section 3.2).
- Construct dynamic contextual features: TSI, RFI, VAS, HDS, TIF plus interaction and rolling-window terms (Section 3.3).
- Apply hybrid Boruta–SHAP feature selection with RFECV refinement to obtain the final feature subset (Section 3.4).
- Train XGBoost, LightGBM, and CatBoost on a chronologically ordered, stratified split, with Bayesian hyperparameter optimisation and early stopping (Section 3.5).
- Compute 10-fold cross-validated performance scores for each base learner and derive softmax-normalized adaptive weights (Section 3.6).
- Aggregate base-learner probabilities into a calibrated ensemble probability and predict the match outcome.
- Generate global and local SHAP explanations for the ensemble prediction (Section 3.7).
- Output the predicted winner, win/loss probabilities, prediction confidence, and top contributing features with SHAP waterfall visualization.

5. Experimental Setup

Experiments were implemented in Python 3.11 using pandas and NumPy for data processing, scikit-learn for general machine-learning utilities, the XGBoost, LightGBM, and CatBoost libraries for the base learners, and the SHAP library for explainability analysis.

5.1 Dataset Description

The experimental dataset contained 5,000 ODI match instances, of which 4,000 were used for training and 1,000 for testing under the stated 80:20 partition. The 1,000-instance test set contained 500 wins and 500 losses, giving a 50:50 class distribution in the reported test evaluation. The manuscript record available does not retain the exact dataset repository/source identifier, season range, complete team and venue list, or the numerical count of raw and engineered features before and after hybrid Boruta-SHAP/RFECV selection. Accordingly, no unsupported values are introduced for those fields. The preprocessing and feature-selection procedures described in Sections 3.2–3.4 were applied within the training data/folds to avoid leakage.

5.2 Data Partitioning

The dataset was split into training and test subsets using an 80:20 ratio with stratified sampling to preserve class balance. A chronologically ordered, stratified 10-fold cross-validation was additionally performed during training to reduce sampling bias and support the adaptive weighting scheme of Section 3.6.

5.3 Hyperparameter Optimisation

Bayesian optimisation (Optuna) was used to tune the number of estimators, learning rate, maximum tree depth, minimum child weight, subsample ratio, and regularisation coefficients for each base learner, selecting the configuration that maximised cross-validated accuracy.

5.4 Comparative Models

EAMEF was benchmarked against Decision Tree, Random Forest, Support Vector Machine, the three individual base learners (XGBoost, LightGBM, CatBoost), Hard Voting, Soft Voting, Stacking, and Blending that individual base-learner performance be reported alongside the ensemble baselines.

5.5 Evaluation Measures

Model performance was evaluated using Accuracy, Precision, Recall, F1-score, Matthews Correlation Coefficient (MCC), Cohen's Kappa, and ROC-AUC. Statistical significance of performance differences was assessed using the Wilcoxon signed-rank test (Section 6.7). Model interpretability was assessed via global and local SHAP explanations (Section 3.7).

5.6 Validation Against Data Leakage and Overfitting

The following validation safeguards were applied to guard against leakage and overfitting: (i) all feature normalisation, target encoding, SMOTE oversampling, and feature selection steps were fit only on training folds and applied unchanged to validation/test folds; (ii) the train/test split preserved chronological order so that no match was used to predict an earlier one; and (iii) rolling-window and EWMA-based features (RFI, HDS) were computed using the information available prior to each match. The reported experiment also used chronologically ordered 10-fold cross-validation during training. The available experimental record does not contain a separate completed nested cross-validation or walk-forward/rolling-origin result; therefore, no additional numerical validation performance is claimed. The reported 97.25% accuracy and 0.9978 ROC-AUC should accordingly be interpreted as the results of the stated evaluation protocol, rather than as independent confirmation from a separate future-season walk-forward experiment.

6. Results and Discussion

6.1 Comparative Performance Analysis

Table 1 reports Accuracy, Precision, Recall, F1-score, ROC-AUC, and MCC for EAMEF against conventional ensembles and, the individual base learners and additional baseline classifiers.

Table 1. Comparative Performance Analysis

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	MCC
Hard Voting	0.9504	0.95	0.95	0.95	0.9936	0.901
Soft Voting	0.9597	0.96	0.96	0.96	0.9936	0.919
Stacking	0.8552	0.86	0.86	0.86	0.945	0.711
Blending	0.8573	0.86	0.86	0.86	0.94	0.715
Proposed EAMEF	0.9725	0.973	0.972	0.9725	0.9978	0.945

Table 1 reports the verified ensemble-comparison results available from the experimental record. The results contain Hard Voting, Soft Voting, Stacking, Blending, and the proposed EAMEF framework. Individual Decision

Tree, Random Forest, SVM, XGBoost, LightGBM, and CatBoost results are not reported because their individual numerical results were not present in the supplied experimental record and have not been estimated.

6.2 Analysis of Classification Accuracy

EAMEF achieved the highest classification accuracy (97.25%) among all compared ensemble strategies (Hard Voting 95.04%, Soft Voting 95.97%, Stacking 85.52%, Blending 85.73%), which is attributable to the combination of softmax-based adaptive weighting (Section 3.6) and the Boruta–SHAP feature subset (Section 3.4). The larger gap relative to Stacking and Blending suggested that, for this dataset, a meta-learner or blended-holdout approach was more prone to variance than a directly performance-weighted vote.

6.3 Precision, Recall, and F1-Score

EAMEF obtained balanced Precision (97.3%) and Recall (97.2%), yielding the highest F1-score (97.25%) of all compared models, indicating that the adaptive ensemble does not trade off false positives against false negatives to inflate a single metric.

6.4 ROC-AUC Analysis

EAMEF achieved the highest ROC-AUC (0.9978) versus Hard Voting and Soft Voting (0.9936 each), Stacking (0.9450), and Blending (0.9400), indicating superior class-separation capability. This is consistent with the calibrated probability outputs produced by the adaptive meta-ensemble layer (Section 3.6).

6.5 Confusion Matrix Analysis

The confusion matrix for EAMEF (487 true wins, 486 true losses, 14 false losses, 13 false wins out of 1,000 test instances) corresponds to the reported 97.25% accuracy, 97.30% precision, 97.20% recall, and 97.25% F1-score, and shows near-symmetric error rates across the two classes, indicating no systematic class bias.

6.6 Local Explanation and Feature Interaction Analysis

The SHAP waterfall analysis of an individual match (base win probability 0.50, final predicted probability 97.8%) shows Team Strength Index and Recent Form Index as the two largest positive contributors, followed by smaller positive contributions from Venue Advantage Score, Head-to-Head Dominance Score, and Toss Impact Factor, consistent with the global importance ranking. SHAP interaction analysis further shows strong pairwise dependence between TSI and RFI (0.73) and between HDS and TIF (0.66), indicating that ODI outcomes depend on combinations of contextual features rather than on any single indicator in isolation.

6.7 Statistical Significance Analysis

Table 2 Statistical Significance Analysis (Wilcoxon Signed-Rank Test)

Comparison	p-value	Significance Level	Result
EAMEF vs Soft Voting	$p < 0.05$	$\alpha = 0.05$	Significant
EAMEF vs Hard Voting	$p < 0.05$	$\alpha = 0.05$	Significant
EAMEF vs Blending	$p < 0.01$	$\alpha = 0.05$	Highly Significant
EAMEF vs Stacking	$p < 0.01$	$\alpha = 0.05$	Highly Significant

The Wilcoxon signed-rank test (Table 2) confirms that EAMEF's performance gains over all four baseline ensembles are statistically significant and not attributable to random variation. The results support three conclusions. First, dynamic, context-aware feature engineering (Section 3.3) provides a measurable benefit over static pre-match indicators: the SHAP interaction results (Section 6.6) show that Team Strength Index and Recent Form Index jointly drive over 70% of relative feature interaction strength, echoing the finding of Agrawal and Kandhway [1,2] that in-play and recent-form indicators progressively dominate pre-match indicators such as toss and ranking as a match proceeds. Second, the hybrid Boruta–SHAP selection pipeline (Section 3.4) appears to reduce redundancy without sacrificing accuracy, in line with Ahmed et al.'s finding [13] that SHAP-guided feature selection improves robustness in high-dimensional intrusion-detection data; the present results extend that finding to sports analytics. Third, the softmax-based adaptive weighting scheme (Section 3.6) outperformed both hard and soft (equal-weight) voting, corroborating the broader ensemble-learning literature showing that performance-adaptive weighting improves on static aggregation [12,13] a mechanism not previously applied systematically to cricket outcome prediction. The improvement over Stacking and Blending specifically (Table 1) differs from some prior ensemble comparisons in other domains, where meta-learner-based stacking has outperformed simple voting [28]; a plausible explanation, to be confirmed once the dataset details of Section 5.1 are finalized, is that with a comparatively modest number of ODI matches relative to the engineered feature space, a second-layer meta-learner is more prone to overfitting than a directly performance-weighted vote, this is consistent with the general conclusion proposed by Chandrashekar and Sahin [23] and Guyon and Elisseeff [27] regarding high-dimensional, sample-limited settings. Nonetheless, the accuracy and ROC-AUC achieved are markedly higher than the T20 in-play benchmarks reported by Agrawal and Kandhway [2] (approximately 85% in the closing overs) and the ODI ensemble benchmarks reported by Ul Haq et al. [4]. This magnitude of improvement deserves the additional leakage and generalisation checks described in Section

5.6 before the result can be considered fully validated; this work treat this as an open validation item rather than asserting it as confirmed.

7. Conclusions

The paper introduced an Explainable Adaptive Meta-Ensemble Framework (EAMEF) for predicting the outcome of ODI cricket matches using context-aware dynamic feature engineering, hybrid Boruta–SHAP feature selection, adaptive softmax-weighted ensemble learning of XGBoost, LightGBM, and CatBoost, and embedded multi-level SHAP explainability layer. For the dataset used, EAMEF obtained 97.25% accuracy, 97.25% F1-score, 0.9978 ROC-AUC, and 0.945 MCC scores, performing better than Hard Voting, Soft Voting, Stacking, and Blending, and the Wilcoxon signed-rank test demonstrated the statistical significance of the results. Global and local SHAP interpretations revealed team strength, form, venue edge, and head-to-head superiority as the top influential factors and some interactions among the aforementioned features as well. The proposed framework is capable of assisting coaches, team analysts, sports organizations, television channels, and fantasy-sport platforms with interpretable, data-driven predictions of outcomes. The limitations of the framework are its dependence on the historical pre-collected data only, and its application to binary match-outcome prediction only. The framework uses only pre-collected historical data about the matches and does not take into account any current data about players' fitness status, in-match events, pitch and weather conditions, and other relevant data. Future work will expand EAMEF into a framework capable of making in-play, state-aware predictions throughout match phases with current data.

Declarations

Use of AI Tools in Figure Preparation

Figure 1 (framework architecture) and Figure 2 (algorithm flowchart) were prepared with the assistance of a diagram/schematic-generation tool to render the authors' specified pipeline structure into a graphical layout; the pipeline stages, terminology, and technical content depicted are those developed by the authors and described in Section 3.

Data Availability

The datasets which are analyzed during the current work will be made available from upon reasonable request.

Authors' Contributions

All authors contributed equally to the study's conception, design, interpretation, and manuscript preparation. All authors read and approved the final manuscript.

Ethical Approval

Not Applicable

Consent to Participate

Not Applicable

Competing Interests

The authors declare that they have no relevant financial or non-financial interests to disclose.

References

[1]. Y. Agrawal and K. Kandhway, "Winner prediction in an ongoing one-day international cricket match," *Journal of Sports Analytics*, vol. 9, pp. 305–318, 2024.

[2]. Y. Agrawal and K. Kandhway, "Predicting the winner of a Twenty20 international cricket match: Classification and explainable machine learning approach," *Journal of Predictive Marketing*, vol. 18, pp. 47–64, 2024.

[3]. I. Wickramasinghe, "Applications of machine learning in cricket: A systematic review," *Machine Learning with Applications*, vol. 10, Art. no. 100435, 2022.

[4]. I. ul Haq, N. Kumar, N. Koul, C. Verma, F. M. Eneacu, and M. S. Raboaca, "Machine learning techniques for result prediction of One Day International (ODI) cricket match," in *Proc. Int. Conf. Recent Innov. Comput.*, Springer, 2022, pp. 95–107.

[5]. R. Bunker and T. Susnjak, "The application of machine learning techniques for predicting match results in team sport: A review," *Journal of Artificial Intelligence Research*, vol. 73, pp. 1285–1322, 2022.

[6]. J. Davis, L. Bransen, L. Devos, A. Jaspers, W. Meert, P. Robberechts, J. Van Haaren, and M. Van Roy, "Methodology and evaluation in sports analytics: Challenges, approaches, and lessons learned," *Machine Learning*, vol. 113, pp. 6977–7010, 2024.

[7]. N. Tax and Y. Joustra, "Predicting the Dutch football competition using public data: A machine learning approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, pp. 1–13, 2015.

[8]. I. Wickramasinghe, "Naive Bayes approach to predict the winner of an ODI cricket game," *Journal of Sports Analytics*, vol. 6, pp. 75–84, 2020.
[9]. Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: Chapman and Hall/CRC, 2025.
[10]. T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Systems*, Springer, 2000, pp. 1–15.
[11]. S. B. Gopal, D. Nanthiya, and M. Saminathan, "An intelligent GWO-CNN-BiLSTM framework for intrusion detection in IoT networks using optimized feature selection," *PMRP Journal of Research—Part A: Computational Engineering*, vol. 1, pp. 30–38, 2025.
[12]. U. Ahmed, Z. Jiangbin, A. Almogren, S. Khan, M. T. Sadiq, A. Altameem, and A. U. Rehman, "Explainable AI-based innovative hybrid ensemble model for intrusion detection," *Journal of Cloud Computing*, vol. 13, Art. no. 150, 2024.
[13]. U. Ahmed, Z. Jiangbin, A. Almogren, M. Sadiq, A. U. Rehman, M. T. Sadiq, and J. Choi, "Hybrid bagging and boosting with SHAP based feature selection for enhanced predictive modeling in intrusion detection systems," *Scientific Reports*, vol. 14, Art. no. 30532, 2024.
[14]. S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," *Nature Machine Intelligence*, vol. 2, pp. 56–67, 2020.
[15]. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[16]. M. N. Uddin and M. M. Aziz, "Shapley value-guided adaptive ensemble learning for explainable financial fraud detection with US regulatory compliance validation," *arXiv preprint* arXiv:2604.14231, 2026.
[17]. N. G. Rezk, S. Alshathri, A. Sayed, E. El-Din Hemdan, and H. El-Behery, "XAI-augmented voting ensemble models for heart disease prediction: A SHAP and LIME-based approach," *Bioengineering*, vol. 11, Art. no. 1016, 2024.
[18]. I. D. Mienye and N. Jere, "Optimized ensemble learning approach with explainable AI for improved heart disease prediction," *Information*, vol. 15, Art. no. 394, 2024.
[19]. A. Lalwani, A. Saraiya, A. Singh, A. Jain, and T. Dash, "Machine learning in sports: A case study on using explainable models for predicting outcomes of volleyball matches," *arXiv preprint* arXiv:2206.09258, 2022.
[20]. A. Khattak, J. Zhang, P. Chan, and F. Chen, "SPE-SHAP: Self-paced ensemble with Shapley additive explanation for the analysis of aviation turbulence triggered by wind shear events," *Expert Systems with Applications*, vol. 254, Art. no. 124399, 2024.
[21]. D. D. Carvalho, M. F. Goethel, A. J. Silva, J. P. Vilas-Boas, D. B. Pyne, and R. J. Fernandes, "Swimming performance interpreted through explainable artificial intelligence (XAI)—Practical tests and training variables modelling," *Applied Sciences*, vol. 14, Art. no. 5218, 2024.
[22]. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[23]. G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, pp. 16–28, 2014.
[24]. M. B. Kursa and W. R. Rudnicki, "Feature selection with the Boruta package," *Journal of Statistical Software*, vol. 36, pp. 1–13, 2010.
[25]. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
[26]. A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: Gradient boosting with categorical features support," *arXiv preprint* arXiv:1810.11363, 2018.
[27]. I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
[28]. D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.
[29]. S. Chakraborty, A. Mondal, A. Bhattacharjee, A. Mallick, R. Santra, S. Maity, and L. Dey, "Cricket data analytics: Forecasting T20 match winners through machine learning," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 28, pp. 73–92, 2024.
[30]. A. Constantinou and N. Fenton, "Towards smart-data: Improving predictive accuracy in long-term football team performance," *Knowledge-Based Systems*, vol. 124, pp. 93–104, 2017.
[31]. L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
[32]. J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, pp. 1189–1232, 2001.

[33]. C. Molnar, *Interpretable Machine Learning*. 2020.
[34]. E. Tiukhova, P. Vemuri, N. L. Flores, A. S. Islind, M. Óskarsdóttir, S. Poelmans, and B. Baesens, "Explainable learning analytics: Assessing the stability of student success prediction models by means of explainable AI," *Decision Support Systems*, vol. 182, Art. no. 114229, 2024.
[35]. Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, pp. 2507–2517, 2007