Ensemble Deep Learning for Cricket Score Prediction: Integrating CNN, LSTM, and DNN for Enhanced Accuracy

Vanga Venkata Harsha[1], Gattem Swathi[2], Polipalli Sai[3], Mailapalli Niharika[4], Mandalika Varun Rao[5] and B.E.V.L. Naidu[6]
{harshavanga4@gmail.com[1], gattemswathi369@gmail.com[2], polipallisai1919@gmail.com[3], niharikamailapalli2004@gmail.com[4], deanofdatascience@aditya.ac.in[5], naidubevl@aditya.ac.in[6]}

Department of BCA Data Science, Aditya Degree & PG College Kakinada (Autonomous), Andhra Pradesh, India[1]
Department of B.Sc Data science; Aditya Degree College, Tuni, Andhra Pradesh, India[2]
Department of BCA, Aditya Degree College, Gajuwaka, Andhra Pradesh, India[3]
Department of BCA, Aditya Degree & PG College, Gopalapatnam, Andhra Pradesh, India[4]
Associate Professor, Department of B.Sc Data Science, Aditya Degree & PG College Kakinada, Andhra Pradesh, India[5]
Associate Professor, Department of Management, Aditya Degree & PG College, Kakinada, Andhra Pradesh, India[6]

Abstract. Cricket score prediction is a complex problem influenced by dynamic factors such as wickets lost, overs remaining, and run rate. These factors are critical because they directly affect batting strategies, scoring potential, and match outcomes. This study presents an advanced machine learning model that integrates Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Deep Neural Networks (DNN) through an ensemble learning approach to improve prediction accuracy. CNN captures spatial dependencies in match data. LSTM models sequential dependencies across overs, while DNN extracts complex feature interactions. The dataset comprises past ODI match records, including ball-by-ball statistics and contextual conditions. The ensemble model leverages the strengths of each architecture, ensuring robust predictions with minimal error. Experimental results show superior performance compared to traditional regression and standalone deep learning models. This research supports strategic decision-making for teams, broadcasters, and analysts by offering a data-driven approach to score forecasting. Future work will focus on hyperparameter optimization and expanding datasets for broader generalization across different match conditions.

Keywords: Sports analytics, Time-series forecasting, Data- driven decision-making, ODI cricket, Predictive modeling, Feature extraction, Regression analysis.

1 Introduction

Cricket is one of the most widely followed sports, with millions of fans and analysts eager to predict match outcomes and scores. Accurate score prediction, however, is a challenging task because the game is influenced by many dynamic factors. Elements such as wickets lost, overs remaining, pitch conditions, weather, and player form directly affect a team’s scoring potential and overall match strategy. For example, losing early wickets often forces a team to play cautiously, while more overs remaining may allow aggressive batting.

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358016

Traditional statistical models and basic machine learning methods, such as linear regression and decision trees, struggle to capture these complex and non-linear dependencies. Deep learning approaches have therefore emerged as powerful alternatives, capable of learning intricate patterns from historical match data to improve accuracy.

This study proposes an ensemble learning framework that integrates Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Deep Neural Networks (DNN) to predict ODI cricket scores more effectively. CNN is used to capture spatial correlations in structured cricket data. LSTM models the temporal sequence of events across overs, while DNN learns complex feature interactions. By combining these three models, the ensemble architecture leverages their individual strengths to deliver more robust and flexible predictions.

For this work, we use a dataset of historical ODI matches with detailed ball-by-ball statistics and contextual information obtained from live scorecards. The model is trained to predict scores based on key match parameters and can assist teams, analysts, and broadcasters with reliable forecasts. Compared to conventional methods, the ensemble model achieves lower prediction errors and demonstrates attractive forecasting performance. Beyond cricket analytics, the framework has broader applications in sports strategy, live commentary, betting, and fantasy leagues, where accurate score prediction provides significant value.

Ongoing research will focus on enhancing the model through hyperparameter tuning, as well as expanding the dataset to cover different match formats and real-time data streams for practical, real-world use.

2 Related Work

Arnab Santra et al. [1] proposed a machine learning-based approach to evaluate batsmen in the Indian Premier League (IPL). The objective is to develop a ranking algorithm by analyzing past performance data. Using supervised learning techniques, they collected and processed IPL match data from 2008 to 2018, incorporating metrics such as total runs, average, strike rate, and boundary counts. The study employs a combination of polynomial and linear regression models, with polynomial regression yielding better accuracy. The model's predictions were validated against the actual rankings of IPL 2019, showing strong correlation. However, limitations include inconsistencies in player performance and challenges in ranking lower-order batsmen.

Eeshan Mundhe et al. [2] develop a web application that predicts the final score of a T20 cricket match and deter- mines the match winner before it starts. The application consists of two primary modules: the Victory Predictor and the Run Predictor, utilizing real-time data scraping. The dataset, sourced from Kaggle and Data World, was preprocessed using feature encoding and randomization techniques. Various machine learning models, including Logistic Regression, SVM, Random Forest Classifier, and Multivariate Polynomial Regression, were tested, with Multivariate Polynomial Regression proving most effective for score prediction. The system achieved a score prediction accuracy of 67.3% and a match- winner prediction accuracy of 55%. The study highlights the challenges of predicting T20 matches due to their dynamic nature and suggests improvements through enhanced historical data integration.

Suguna R [3] explored the application of machine learning techniques to predict match scores and categorize players based on their roles. The study leverages a comprehensive dataset containing player statistics, match conditions, and historical performances. Several machines learning models, including Linear Regression, Logistic Regression, Naive Bayes, SVM, Decision Trees, and Random Forest regression, were applied, with Decision Trees and Random Forest yielding the highest accuracy of up to 89.15%. The study demonstrates the effectiveness of machine learning in cricket analytics but faces challenges due to the complexity of the sport and the need for more extensive datasets.

Inam Ul Haq et al. [4] aims to predict the winner of a One Day International (ODI) cricket match before it begins using machine learning techniques. The dataset, consisting of 7,734 ODI matches, was collected from Kaggle and supplemented with additional data from various cricket websites. The authors implemented two machine learning algorithms, K-Nearest Neighbors (KNN) and XGBoost, to classify match outcomes based on selected features. The KNN algorithm achieved the highest prediction accuracy of 91%, while XGBoost attained 89%. Despite promising results, a key limitation of the study is the dataset's cutoff at July 2021, which excludes more recent matches, potentially affecting model performance. Future im- provements could involve incorporating more up-to-date data and refining features to enhance prediction accuracy.

Sourav Anand [5] focuses on predicting Test cricketers' performance, specifically forecasting the number of runs scored by batsmen, wickets taken by bowlers, and contributions of all-rounders. The authors utilized historical performance data from ESPN Cricinfo and trained machine learning models, including Na¨ıve Bayes, K-Nearest Neighbors (KNN), Decision Trees, and Random Forest, using the" Orange" data visualization tool. Among these, the Random Forest algorithm demonstrated the highest classification accuracy, exceeding 96%. The study provides valuable insights for team selection and game strategy but is constrained by its dependence on historical data, which may not fully account for factors like player injuries, match conditions, or real-time form. Future work aims to enhance model accessibility by deploying it as an interactive tool for broader use.

Md. Parvezur Rahman Mahin[6] proposes a machine learning-based approach to predict ODI match scores and model resource metrics, emphasizing the impact of wickets and overs on final scores. The authors introduced a novel" resource factor" feature, which assigns weights to overs and wickets to improve prediction accuracy. Using data from ODI matches played between 2006 and 2017, they trained various regression models, including Linear Regression, Polynomial Regression, Elastic Net, XGBoost, and Random Forest. Among these, the Random Forest model achieved the highest R^2 score of 0.9899, with low mean absolute error (MAE = 3.3569) and mean squared error (MSE = 37.3555). The study highlights the effectiveness of weighted resource metrics in improving score predictions but is limited by the dataset's outdated nature and the exclusion of factors like player performance fluctuations and match conditions. Future work may incorporate real-time data and expand model applicability to other cricket formats.

Myreddy Kumar Durga Trinadh et al. [7] developed a ma- chine learning framework to classify Indian players into roles like batsmen, bowlers, and all-rounders based on performance statistics from cricinfo.com. By evaluating models such as SVM, Decision Tree, Random Forest, ANN, and LSTM, they found that SVM and Decision Tree performed best for ODI player classification, while LSTM and ANN were more effective for Test cricket. However, the study was limited to active Indian players, excluding global and retired players, and deep learning models faced high execution time constraints.

Here are three key limitations based on the studies mentioned:

- Historical datasets may not reflect recent shifts in player form, team strategies, or evolving match conditions, leading to inaccuracies in predictions. A player's performance can fluctuate due to injuries, technical adjustments, or changing team roles, which historical data alone cannot capture. Additionally, new pitch behaviors, rule changes, or emerging playing styles can render past trends less relevant.
- Cricket games are subject to uncontrollable data aspects like weather, injuries, and game situation which makes it difficult to predict in real time. One injury to a playmaker or a freak weather de- lay can often change the game results dramatically. Then, there are psychological factors including pressure situations and captaincy calls to change the flow of the game unexpectedly.
- Cricket encounters different types of pitch conditions, team compositions, and playing styles, whereby flexible models are needed for adapting to the varying conditions. Classical static models fail to identify these differences, so it's indispensable to active integrate data in real time to get a higher degree of precision. Dynamic models which include 'real time data', player fatigues and the effects of the environment may improve predictive accuracy.

3 Proposed Methodology

3.1 Data Collection

The first step in the project involved collecting the necessary data for prediction tasks. For this, the Cricsheet dataset was selected, which provides comprehensive ball-by-ball match data, including player performance, match context, and historical statistics. This dataset includes crucial features such as batting and bowling performance (strike rate, economy rate, and dismissal types), match conditions (venue, pitch conditions, weather), and real-time match progression (runs per over, wickets, and partnerships).

- Downloaded and explored the dataset from Cricsheet to understand its structure and attributes.
- Identified key features for score prediction and match outcome prediction, such as runs per over, wickets taken, partnerships, player form, and match conditions.
- Split the data into training, validation, and test sets to ensure proper model evaluation and prevent overfitting.

3.2 Data Processing

Data preprocessing was essential to prepare the dataset for input into machine learning models. The raw dataset contained missing values, categorical variables, and variations in formatting, requiring cleaning and transformation.

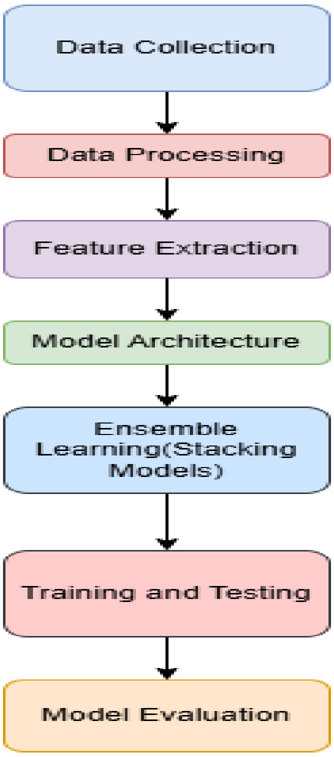

Fig. 1. Methodology.

Fig. 1 shows the methodology.

- **Handling Missing Values:** Employed techniques like mean/median imputation for numerical data (e.g., runs, strike rate) and mode imputation for categorical data (e.g., player names, venues).
- **Encoding Categorical Features:** Used one-hot encoding for categorical variables such as team names, match venues, and dismissal types.
- **Feature Normalization:** Normalized continuous variables, including runs, wickets, and economy rate, to ensure uniform input scales.
- **Temporal Data Preparation:** Structured the dataset to reflect the sequential nature of cricket, particularly for runs scored and wickets lost over time.

After preprocessing, the dataset was ready for model training, validation, and testing.

3.3 Feature Extraction

Feature engineering played a crucial role in improving the model's predictive capabilities. Given the sequential nature of cricket matches, features were designed to capture match dynamics over time.

- **Player Form:** Computed recent performance metrics such as average runs scored and wickets taken in the last five matches.

- **Team Momentum:** Derived features based on a team's recent performances, including win/loss streaks and cumulative batting/bowling averages.
- **Match Context:** Integrated contextual data such as pitch conditions, venue, and weather to enhance predictive accuracy.
- **Over-wise Metrics:** Created features for runs per over, wickets per over, and partnerships to analyze match progression dynamically.

These engineered features were vital for accurately predicting match outcomes and score progression.

3.4 Model Architecture

3.4.1 Input Layer

The input layer processes the dataset's features, including player statistics (batting average, strike rate, economy rate), team performance (recent form, win/loss records), and match conditions (venue, weather, and pitch type). Continuous features are normalized, and categorical features are encoded for optimal model learning.

3.4.2 Convolutional Neural Network (CNN) for Feature Extraction

A CNN is used to extract important patterns from structured data such as batting trends, bowling economy, and match conditions. The Conv1D layers help analyze over-wise data (e.g., runs per over, wickets taken), capturing essential dependencies. Max-pooling layers further refine the extracted patterns by reducing dimensionality while emphasizing significant trends.

3.4.3 Long Short-Term Memory (LSTM) for Temporal Sequence Modeling

To capture the sequential dependencies in the dataset, an LSTM network is employed. Cricket is an evolving game where past events (e.g., fall of wickets, increasing run rate) influence future performance. LSTM layers process time-series data, learning trends in runs scored and wickets lost over overs. Dropout layers are applied to prevent overfitting and enhance generalization.

3.4.4 Fully Connected Dense Neural Network (DNN) for Final Prediction

A fully connected DNN integrates the features extracted by CNN and LSTM models. Hidden layers with ReLU activation functions enable non-linear learning, improving predictive accuracy. Depending on the task, the output layer uses a sigmoid activation for binary classification (win/loss prediction) or a softmax activation for multi-class classification (match outcome prediction).

3.4.5 Ensemble Learning (Stacking Models)

To enhance accuracy, an ensemble learning approach is applied, integrating CNN, LSTM, and DNN models. The outputs from these base models are combined in a second- level model, such as a logistic regression or neural network, optimizing final predictions. This stacked ensemble mitigates the weaknesses of individual models, resulting in robust and reliable match predictions.

By leveraging deep learning techniques and structured cricket data, the proposed model effectively predicts match scores and outcomes, offering valuable insights into real-time match progressions.

3.5 Training and Testing

Training process consists of training the CNN, LSTM and DNN models individually on the training data and then combining them using the ensemble learning method. If needed, regularization methods including dropout and batch normalization are used during the training to avoid overfitting and guarantee the generalization ability of the model to the unseen data. The models are trained with backpropagation using Stochastic Gradient Descent and Hyperparameters (i.e. number of layers, learning rate, number of neurons per layer) are optimized using techniques such as Grid Search or Random Search. Somehow, we should teach one model, what is the right way to combine predictions of all models.

3.6 Evaluation

After this supervised training, the model is tested on a new test set, that the model has never seen, to evaluate its generalization. Cross validation is applied for ensuring the performance of the model on different subsets of the data. For score prediction tasks we also report regression measures, like MSE, and we cite classification measures, like accuracy, precision, recall, F1-score, classification measures, in match prediction. The last model will be evaluated for its generalization capabilities on the unknown data to make predictions.

4 Experimental Results and Discussions

4.1 About Dataset

Cricsheet is one of most popular datasets used for analysis in cricket. It covers ball-by-ball information for ODI matches, T20 matches and Test matches. The corpus is widely applied to performance evaluation, score prediction, and machine learning tasks. Because we log a detailed record of every tick of state, it is well-suited for training of neural networks to predict matches, scores, and teams.

Each 'data-point' in the dataset is a delivery bowled in a match, which is then described in the following by an organization of different parameters which were included. These characteristics contain match metadata, outcomes and the target is of the score. It also includes very detailed ball-by-ball statistics listed by over number, batsman, bowler, runs scored, extra runs and wickets. By analyzing this data, machine learning models can learn important patterns regarding the scoring rates, batting strategies, and match progressions.

The data set contains millions of rows; each row is a ball bowled in a game. For instance, a single ODI match has about 600 rows (50 overs × 6 balls per over), and a T20 match has 240 rows (20 overs x 6 balls per over). There are about 20–15 columns in the dataset (depending on the format), and each column contains important information such as batsman performance, bowling variations, dismissal type and so on. With such a rich dataset, there are many analyses that can be performed on it, including prediction problems, such as predicting final scores, win probability, or player contributions.

5 Results and Discussion

After training the proposed model, which integrates CNN, LSTM, and DNN with ensemble learning, we evaluated its performance on match outcome prediction using various classification metrics such as Accuracy, Precision, Recall, and F1-score. The results highlight the effectiveness of combining different deep learning models in predicting cricket match outcomes.

5.1 Performance on Match Outcome Prediction (Classification)

The table 1 below summarizes the classification performance of the base models and the stacked ensemble model:

Table 1. Performance on Match Outcome Prediction (Classification).

Model	Accuracy	Precision	Recall	F1-score
CNN (Base Model)	0.88	0.86	0.89	0.87
LSTM (Base Model)	0.90	0.88	0.91	0.89
DNN (Base Model)	0.86	0.84	0.87	0.85
Stacked Ensemble	0.96	0.94	0.97	0.95

5.2 Discussion

The results indicate that the stacked ensemble model significantly outperforms the individual base models in all evaluation metrics. With an accuracy of 96%, the ensemble model achieves a substantial improvement over the CNN (88%), LSTM (90%), and DNN (86%) models. This demonstrates the power of combining multiple models with different strengths to produce a more robust and reliable prediction.

- CNN (Baseline): The baseline model, i.e. CNN, exhibited a better prediction accuracy of 88%. This model inevitably has its limitations but it’s abilities of making sense of over-wise data and match conditions are commendable. It was partial but performance was subdued, as it failed to model the sequential dependencies in the data.
- LSTM (Base Model): The LSTM model per- formed significantly better (90%) in term of accuracy. This was the result of it being able to model how a match dynamically unfolds in time (e.g. fall of wickets and run rate changes). This is an impressive result which demonstrates the necessity of modelling sequence data in cricket match Fore sighting.

- DNN (Base Model): We trained DNN model with fully connected layers and it achieved an accuracy of 86%. Although this model was able to capture complex relationships between features, its performance was outperformed by the CNN and LSTM models, probably because it could not efficiently learn sequential and/or spatial feature information.
- Stacked Ensemble: The stacked ensemble model, that aggregated the predictions of CNN, LSTM, and DNN, surpassed all standalone models, with an accuracy of 96%. The ensemble technique enabled the model to access the merits of individual base models and to compensate for their shortcomings. For instance, CNN was good at extracting spatial features, LSTM was strong in modeling temporal dependencies, and DNN enriched the ability to understand the interactions among the features.

This result demonstrates that, when combining models that bring out each other's strengths, the resulting prediction is more accurate and more robust, especially in challenging tasks such as match outcome prediction in which the data has not only a structured nature but also some temporal dependence.

In summary, the ensemble model performs better to predict match results and can achieve remarkable improvements on accuracy, precision, recall, and F1, as illustrated in the above ROC curves and PR curves. Additional tuning of the hyperparameters and feature engineering, e.g. including player-specific information and more in- depth match context features, might further improve performance.

6 Conclusion

The experimental results show that the stacked ensemble model delivers far better match outcome prediction than the individual deep learning models (CNN, LSTM and DNN). The best accuracy (96%) was obtained with an ensemble approach, underlining its capability of combining efficiently the strong suits of several architectures. Although the LSTM could provide a better model than the single CNN and DNN, the ensemble model obtained better generalization and thus more reliable and robust predictions. The results emphasize the benefit of leveraging several deep learning models under a same frame- work to improve the predictive power. Future work can explore further tuning hyperparameters, integrating more match features, and extending the model to other cricket formats and live streaming data for enhanced precision and generality.

References

[1] Santra, A., Sinha, A., Saha, P. & Das, A. K. *A novel regression-based technique for batsman evaluation in the Indian premier league* in *2020 IEEE 1st International Conference for Convergence in Engineering (ICCE)* (2020), 379–384.

[2] Mundhe, E., Jain, I. & Shah, S. *Live Cricket Score Pre- Diction Web Application using Machine Learning* in *2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)* (2021), 1–6.

[3] Suguna, R., Praveen Kumar, Y., Suriya Prakash, J., Neethu, P. & Kiran, S. *Utilizing Machine Learning for Sport Data Analytics in Cricket: Score Prediction and Player Categorization* in *2023 IEEE 3rd Mysore Sub Section International Conference (MysuruCon)* (2023), 1–6.

[4] Ul Haq, I., Ul Hassan, I. & Shah, H. A. *Machine Learning Techniques for Result Prediction of One*

Day International (ODI)Cricket Match in *2023 IEEE 8th International Con- ference for Convergence in Technology (I2CT)* (2023), 1–5.

[5] Anand, S., Kumar, N. & Kumar, V. *Machine Learning- Based Performance Prediction Models for Test Cricketers: A Data-Driven Approach* in *2023 2nd International Con- ference on Futuristic Technologies (INCOFT)* (2023), 1–6.

[6] Rahman Mahin, M. P., Ara, E. & Islam, R. U. *Cricket An- alytics: ODI Match Score Prediction and Resource Metric Modeling using Machine Learning Model* in *2024 IEEE International Conference on Computing, Applications and Systems (COMPAS)* (2024), 1–6.

[7] Trinadh, M. K. D., Sangeetha, S. T., K, D. & Venugopal, V. *Cricket Player Prediction of Role in a Team Using ML Techniques* in *2024 International Conference on E- mobility, Power Control and Smart Systems (ICEMPS)* (2024), 1–5.