DOI: 10.21917/ijdsml.2025.0163

SPORTS ANALYTICS IN CRICKET - PREDICTING SCORES IN THE INDIAN PREMIER LEAGUE USING DEEP LEARNING TECHNIQUES

Ashish Raghav[1], Neeta Singh[2] and Naresh Kumar[3]
[1,2]Department of Computer Science and Engineering, Gautam Buddha University, India
[3]Department of Mathematical and Physical Sciences, University of Nizwa, Sultanate of Oman

Abstract
Sports analytics, with its dynamic and unpredictable nature, has gained significant attention due to its ability to improve decision-making and performance measurement. This study mainly focuses on predictive analytics in the context of the Indian Premier League (IPL), an annual T20 cricket league held in India. Various factors could influence the outcome of a cricket match, and hence it becomes challenging for analysts to identify patterns within the vast volume of data available. The proposed framework analyzes the most significant factors that have a significant impact on match outcomes and uses the factors to predict final cumulative scores of teams in the IPL after 20 overs. Deep learning techniques, such as Feedforward Neural Networks (FNN), Multilayer Perceptrons (MLP), Long Short-Term Memory (LSTM), and Recurrent Neural Networks (RNN), were applied on a dataset over ten years of ball-by-ball data collected from reliable sources such as ESPN and Cricksheet, covering 10 IPL seasons. The model was evaluated in terms of parameters such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 values. Among the deep learning models, LSTM demonstrated the highest prediction accuracy, achieving an 85% success rate. The findings highlight the advantage of deep learning techniques in IPL score prediction with valuable insight for strategic planning and decision-making in cricket.

Keywords:
Sports Analytics, Feedforward Neural Networks, Long Short-Term Memory, Dynamic Nature

1. INTRODUCTION

Cricket being one of the most widely played sport [1], around the world, and has evolved in different formats such as Test matches, One Day Internationals (ODIs), and the high-speed and fast-paced Twenty20 (T20) format [2]. The T20 format has become extremely popular because of the short length or duration of the match and the thrilling play. The Indian Premier League (IPL) is one of the most desirable leagues featuring the world's top players in the T20 leagues. With its competitive nature and dynamic matches, the IPL is a rich ground for predictive modeling and analytics.

Traditionally, score predictions have relied on simple metrics like run rate and manual calculations, which are often error-prone and fail to capture the dynamic shifts that occur during a match. The limitations of manual prediction methods and run rate-based estimations highlight the need for more sophisticated approaches. To address this problem, the proposed work leverages deep learning models capable of capturing complex, non-linear relationships in the data. By utilizing historical match records, real-time in-game statistics, and advanced architectures networks [16], the system aims to offer more reliable score forecasts. Though large datasets containing ball-by-ball data of IPL matches are available, it is an extremely tough task to analyze such data manually to identify patterns and predict scores. Recent developments in machine learning (ML) and deep learning (DL) techniques have established the feasibility of sports analytics in resolving and overcoming such issues [3]. These techniques can handle large volumes of data, identify hidden patterns or trends, and make precise predictions. In this work, we have addressed deep learning technique for predicting cumulative team scores - a key aspect of the league that contributes to its intense and unpredictable nature. We experimented with a dataset of ten years of IPL ball-by-ball data, which we obtained from ESPN and Cricksheet. We used key features that influence match scores, such as the score at the current moment, batting and bowling team details, overs faced, runs taken in the last five overs, wickets lost in the last five overs, and overall wickets lost batting against the opposing team. These features have a significant influence on match scores.

The study was carried out on four deep learning models: FNN, MLP, LSTM, and RNN, and their performance was evaluated using parameters such as MAE, MSE, RMSE, and R^2. The results showed that the LSTM model consistently gave the best predictions of IPL match scores, offering useful data-driven insights for enhancing match outcome strategies in cricket. The results of this study not only validate the use of DL techniques in sports analysis but can assist cricket fans, sports analysts, and team administrators in decision-making, strategy formulation, and a better understanding of overall match dynamics.

The rest of the paper is structured as follows. Section 2 provides a literature review of ML and DL applications in cricket analytics emphasizing their contributions and significance. Section 3 explains our proposed methodology in detail. Section 4 describes our results and findings, and Section 5 and Section 6 provides a conclusion and the future scope at the end.

2. LITERATURE REVIEW

The rising interest in sports analytics has led to extensive research aimed at enhancing how we assess performance and make decisions across different sports disciplines [5]. In cricket, the availability of large datasets has enabled researchers the opportunity to uncover patterns and trends that were previously difficult to identify manually. Many studies have focused on using ML and DL techniques to analyze the available cricket data, especially for predicting match results, player performances, and scoring outcomes. Given, the dynamic, fast-paced, and sequential nature of cricket data has driven a shift towards more advanced approaches, like DL models, which are better suited to handle the complex relationships and temporal dependencies within the available data. This section will highlight significant research or studies on ML and DL applications in cricket analytics, emphasizing their contributions and significance.

Shah et al. [6] have described the application of machine learning models to predict high run chases in T-20 International cricket matches. They have studied 458 T-20I matches from 2005 until 2024, using different parameters such as venue, toss results, batting position, pitch, and team rankings. The study was conducted on seven different machine learning models such as Random Forest, Support Vector Machine, Decision Tree, and some others and tested their operating performance on parameters such as accuracy, precision, AUC and recall. The results indicated Random Forest and Decision Tree were performing the best to predict high-scoring chases, producing good insights into better data-driven strategies for T-20I cricket.

Sadi et al. [7] explored the potential of deep learning techniques for the prediction of cumulative cricket scores while concentrating their study on hundred-ball cricket fixture. An MLP Regression model was developed to predict scores in hundred-ball-matches with accuracy and efficiency in comparison to other traditional techniques. The study conducted predictive analysis across actual games and showed that the model could achieve up to 99% accuracy, thus demonstrating its effectiveness and potential for improving score predictions in cricket.

Kapadia et al. [8] employed machine learning techniques along with historical data to analyze IPL match outcomes. Feature selection was done using techniques like Correlation-based Feature Selection and Information Gain, and subsets were prepared based on the advantages of home teams and decisions about the toss. Predictive models were designed on Naïve Bayes, Random Forest, K-Nearest Neighbor, and Model Trees. The results of tree-based models such as Random Forest had outperformed concerning home-team data while toss-based models underperformed.

Singh et al. [9] proposed a model for the estimation of the ODI first-innings score using Linear Regression and the outcome of the second innings using the Naïve Bayes classifier. First-innings prediction features included run rate, wickets, venue, and batting team. Second-innings prediction features include target score as well. A marked decrease in error was found with respect to first innings score prediction using the proposed system, while 68% to 91% accuracy for the second-innings outcomes from 0-5 to 45 overs in the matches considered between 2002-2014.

Kumar et al. [10] utilized Decision Trees and Multilayer Perceptron Networks to analyze the impact of pre-game and in-game factors on cricket match outcomes. Some of the key attributes in their work were venue, team strength, toss, run rate, and wickets. Based on these analyses, they developed CricAI, a prediction system that uses pre-game attributes like ground, venue, and innings for predicting the result of a given match.

Asif and McHale [11] tested a generalized nonlinear forecasting model that could estimate runs remaining in an innings given over left and wickets lost for all formats of limited-overs cricket. For T20 Internationals, the model provided estimates of run differentials during a match as a measure of game competitiveness and team ratings. The GNLM embraces the Duckworth/Lewis model and its McHale/Asif variant as special cases. This enhanced the accuracy of forecasting and the analysis of matches.

Parasuraman et al. [12] used Convolutional Long Short-Term Memory (CLSTM) hybrid deep learning model for cricket data analysis with the aim of predicting the score of a match and its outcome. The integration of CNN with LSTM processes both the spatial and temporal dependencies in cricket data, such as the performance of individual players, the conditions of the pitch, and previous outcomes of different matches. This gave high accuracy with prediction and offered many insights on the dynamics of the game that could be availed for cricket analytics using deep learning.

Prakash and Verma [13] presented the Deep Player Performance Index for evaluating T20 cricket players in batting and bowling strengths about the current form of the players, considering the players' role within the team. DPPI will be built by using K-Means clustering and Random Forest algorithms that compare players with similar roles within different teams and estimate team strength by summing the player scores. By applying for the IPL 2019 season, DPPI outperforms the existing performance indexes in practice, which was very valuable to the cricket fans, coaches, and analysts.

A review of the existing literature brings out various important works that have been carried out in cricket analytics, mainly related to the prediction of outcomes of matches, forecasting of scores, and evaluation of players. Based on these studies, our work deals with the problem of cumulative score prediction in IPL matches using deep learning techniques. We analyzed 10 years of ball-by-ball IPL data for influential features to improve predictive accuracy for practical insights into team strategy and decision-making.

3. METHODOLOGY

This section outlines the methodology of the proposed work. The prediction model is designed to provide a cricket team's final score after 20 overs.

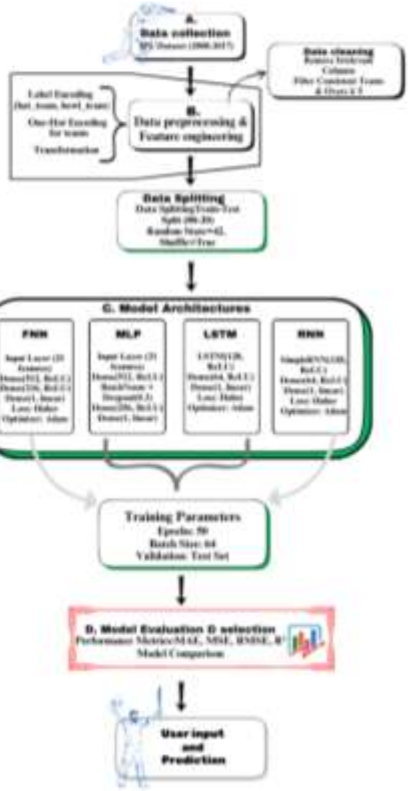

Fig.1. Architecture of the proposed model

It considers various factors, including the current score, data from both the batting and bowling teams, the number of overs played, runs scored in the last five overs, wickets taken in the last five overs, and the number of wickets lost while batting against the opposing team. The Fig.1 represents our methodology and, a step-by-step overview of the approach we have followed to predict the final score of the match once all the overs are completed.

3.1 DATASET

The data for the proposed work is taken from ESPN and the Cricksheet website [4], both of which are known for their comprehensive and detailed cricket statistics. The dataset comprises ball-by-ball information on the IPL matches across ten seasons from 2008 to 2017, comprising over 75,000 entries and 15 different attributes. The dataset includes various attributes such as match ID, date, venue, batting team, bowling team, batter, bowler, runs scored, wickets taken, overs bowled, runs and wickets in the last five overs, on-strike batter, non-striker, and total score. To build and validate the predictive models, 80% of the dataset is used for training, while the remaining 20% is employed for testing. This approach ensures that the models are trained on a substantial amount of data and evaluated on unseen data, resulting in reliable score predictions.

3.2 DATA PREPROCESSING

For the IPL score prediction dataset, the preprocessing stage focused on filtering and organizing the data to facilitate optimal analysis and compatibility with deep learning models. Initially, irrelevant features, including [‘match_id’, ‘venue’, ‘batsman’, ‘bowler’, ‘striker’, and ‘non-striker’], were removed, as they did not contribute significant predictive power. The dataset was refined to highlight the most important features as shown in Fig.2, including batting team, bowling team, runs scored, wickets taken, overs bowled, runs in the last 5 overs, wickets in the last 5 overs, and the target variable total score, ensuring the dataset captures the essential elements for predictive modeling.

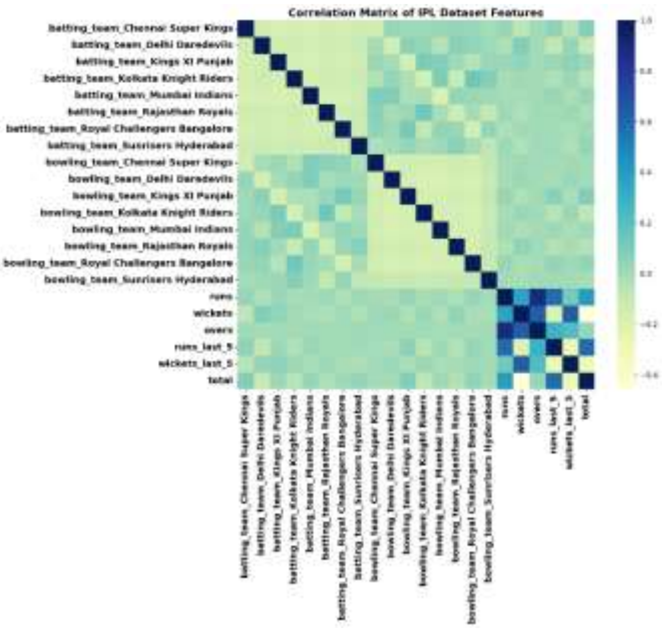

Fig.2. Correlation Matrix of IPL Dataset Features

To handle categorical variables, Label Encoding was first applied to convert team names into numerical values, simplifying their representation. Subsequently, One-Hot Encoding was used to transform categorical features, such as batting_team and bowling_team, into binary columns, enhancing the model’s ability to interpret team-based data. This transformation resulted in a structured dataset, that combined encoded categorical features with numerical data. For the sake of consistency in analysis, the dataset was filtered to keep matches played by consistent teams throughout the league.

The data for the first five overs (overs < 5.0) was not considered, as we are predicting the final score based on some initial values and taking the real-time data into account for final score prediction. These preprocessing techniques ensure the dataset is clean, consistent, and suitable for predictive modeling, providing a reliable foundation for training and testing predictive models.

3.3 CLASSIFICATION

During the analysis phase of this study, various neural network models were implemented to predict IPL match scores based on a preprocessed dataset. The models were constructed using Keras with several hyperparameters and included multiple dense layers. To effectively handle regression tasks, we used a linear activation function in the output layer. The Huber loss function was employed to manage outliers, and the models were optimized using the Adam optimizer, was employed for robust training over 50 epochs with a batch size of 64. To monitor generalization, we utilized a validation dataset, and the performance was assessed using metrics such as RMSE, MSE, MAE, and R^2 scores, which are presented in Table 1.

Table.1. Model Hyperparameters

Hyperparameters	FNN	MLP	LSTM	RNN
Hidden Layers	2 Dense Layers (512 units, 216 units)	2 Dense Layers (512, 256 units) + BN & Dropout	LSTM (128) + Dense (64 units)	SimpleRNN (128 units) + Dense (64 units)
Activation Functions	ReLU	ReLU	ReLU	ReLU
Output Layer	Linear	Linear	Linear	Linear
Loss Function	Huber Loss (delta=1.0)	Huber Loss (delta=1.0)	Huber Loss (delta=1.0)	Huber Loss (delta=1.0)
Optimizer	Adam	Adam	Adam	Adam
Epochs	50	50	50	50
Batch Size	64	64	64	64
Evaluation Metrics	MAE, MSE, RMSE, R^2	MAE, MSE, RMSE, R^2	MAE, MSE, RMSE, R^2	MAE, MSE, RMSE, R^2
Validation Split	20%	20%	20%	20%

A Multilayer Perceptron was developed to enhance predictive capabilities with improvements like Batch Normalization and Dropout Regularization, which help reduce overfitting and facilitate smoother convergence. To capture temporal dependencies, an LSTM network was implemented, with input features reshaped accordingly. The model architecture consisted of LSTM layers followed by dense layers to map the learned patterns to the target output. Additionally, a RNN using Simple RNN layers was constructed to compare its performance against the LSTM. These recurrent models took advantage of the sequential nature of the data, showing how the extraction of temporal features influences prediction accuracy. Consistent performance metrics were used to evaluate each model's effectiveness. We also plotted the loss curves for both the training and validation datasets, as presented in Fig.6, to ensure stable training dynamics. Through a systematic exploration of various architectures, we were able to choose the best neural network model among others implemented, for predicting IPL scores.

3.4 MODEL SELECTION PROCESS

In our selection process, we experimented with different models to predict IPL match scores and compared how well each one performed against the actual real match data. We chose models such as FNN, LSTM, MLP and RNN because they effectively handle complex relationships and patterns in the data over time.

The FNN was initially chosen since it could effectively capture the straightforward relationships or connections between input features and target variables, that is the final score in our case. To enhance its performance further, we used Multilayer Perceptron, which incorporates more layers to the network. To address the challenge of overfitting during our model training, we incorporated techniques like Batch Normalization and Dropout Regularization. These techniques helped in improving the generalization of our models. When it came to analyzing or better understanding the sequential patterns found in cricket data, we implemented the LSTM model, which generally uses memory units to capture long-term dependencies. Although we have also explored RNNs for temporal analysis, we found that their simpler architecture didn't perform well or are less effective when compared with LSTM model.

To ensure a fair comparison and evaluation of all models, we systematically trained and validated each model using the same dataset split. Throughout this process, we reviewed some key performance indicators, such as MSE, MAE, RMSE, and the R^2 score to determine how each model performed upon testing all the models and validated using key matches of IPL 2024. The testing employed actual match data from the final, qualifiers, and eliminators as shown in Table.6.

To further test the performance or hold of the LSTM model, we validated it using actual IPL match data. We set up match situations using inputs from actual games and the features needed to predict match scores. This systematic selection and validation approach ensured the reliability and applicability of the selected model in predicting IPL scores.

4. RESULTS AND OBSERVATIONS

We implemented four deep learning models in this study—FNN, MLP, LSTM and RNN to predict the IPL match scores using specific dataset [4]. We utilized reliable Python libraries such as Keras, TensorFlow, and scikit-learn in our study. To enhance the speed and the performance, we employed GPU acceleration, particularly while training the LSTM model, which enabled us to obtain the results quickly.

4.1 IMPLEMENTATION OF FNN MODEL FOR IPL SCORE PREDICTION

Initially, we have implemented the FNN regression model using the parameters presented in Table 1. The FNN model considers significant features such as the batting Team, Bowling Team, Runs scored, Wickets taken, Overs bowled, Runs in the last 5 overs, and Wickets in the last 5 overs. The network structure is an input layer that corresponds to the features, two hidden layers of 512 and 216 neurons, respectively, both with the ReLU activation function to enhance the capacity to identify complex patterns in the data. There is then an output layer with a linear activation function that gives continuous score predictions. The model uses the Huber loss function [14], a robust loss function that is a blend of both MSE and MAE. We used a loss function that is less sensitive to outliers than traditional MSE, which is a nice advantage. To optimize the model, we utilized the Adam optimizer with a batch size of 64, running it for 50 epochs. After training, the model demonstrated strong performance, achieving evaluation metrics including an MAE of 8.69, RMSE of 13.12, MSE of 172.39, and an R^2 score of 0.80, as presented in Table 2. This means it was able to explain about 80% of the variation in the IPL score dataset.

Table.2. Predicted Evaluation Metric Values Using FNN

Evaluation Metrics	MAE	MSE	RMSE	R^2
Values	8.69	172.39	13.12	0.80

For a more detailed analysis, refer to the line plot presented in Fig.2.

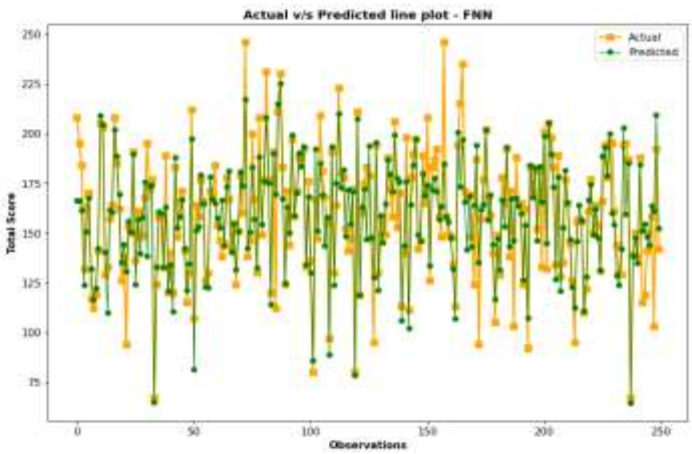

Fig.3. Graph showing analysis of line plot using FNN model

The graph in Fig.3 clearly shows that the FNN model effectively captures the variability in IPL score predictions, with predicted scores fluctuating between 75 and 250. The model demonstrates a strong ability to align predicted values with actual scores, even in cases with higher variability. Clusters of data points, particularly in the range of 125 to 190, highlight the model's consistent performance in predicting scores within this range, where IPL match scores are more frequent. The close alignment of predicted and actual lines further confirms the robustness of the model in handling diverse scenarios, including outliers, showcasing its ability to generalize well across different matches.

4.2 IMPLEMENTATION OF MLP MODEL FOR IPL SCORE PREDICTION

The MLP model we're using for predicting IPL scores aims to improve accuracy by using techniques like Dropout and Batch Normalization. It uses ReLU activation for the hidden layers. The model structure begins with an input layer that matches the number of features we have. Then, there are two dense hidden layers - one with 512 neurons and the other with 256. We incorporate Batch Normalization to prevent overfitting and apply a Dropout rate of 0.3 after each hidden layer. Finally, the output layer has a linear activation function, which also is suitable for the regression task. This careful designing helps us make better predictions on IPL scores.

Table.3. Predicted Evaluation Metric Values Using MLP

Evaluation Metrics	MAE	MSE	RMSE	R^2
Values	10.73	226.48	15.04	0.74

The model was constructed using the Adam optimizer used with the Huber loss function, set by a delta value of 1.0. This approach effectively minimizes sensitivity to outliers while ensuring a smooth learning process. After training for 50 epochs as shown in Fig.6, with a batch size of 64, the model delivered some promising performance metrics as shown in Table 3. The R^2 value, a statistical measure of model fit, was established to be 0.74. Essentially, this means that the multi-layer perceptron model captures about 74% of the variability in IPL scores, showcasing its strong capability in predicting match outcomes accurately.

The Fig.4 shows the correlation between the predicted scores and the actual scores, further highlighting the model's effectiveness.

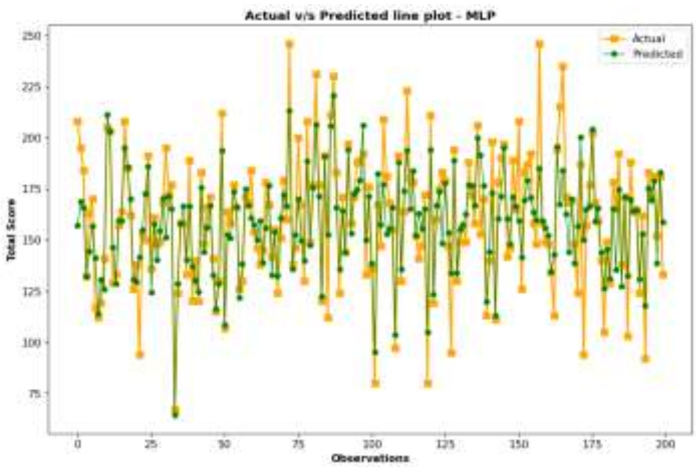

Fig.4. Graph showing analysis of line plot using MLP model

From Fig.4, the MLP model shows predicted values ranging from 75 to 240, which means it can make a variety of predictions in different scoring situations. However, the predicted line exhibits higher variability and inconsistencies compared to other algorithms, indicating a relatively less robust performance in capturing the underlying patterns of the data. This suggests that the MLP model isn't as good at recognizing the patterns in the data. While it offers reasonable estimates, its accuracy and reliability are not as strong as the other approaches we looked at.

4.3 PROPOSED MODEL FOR IPL SCORE PREDICTION USING LSTM

The proposed model for predicting IPL match scores takes advantage of the sequential nature of the match data to deliver reliable forecasts. Reshape the input feature vectors into a three-dimensional format for compatibility with the LSTM layer. The LSTM layer has 128 units, using a ReLU activation function, so that it can be used to learn temporal relationships in the data effectively. Finally, a dense hidden layer with 64 units further refines features learned from the previous layer, thus enhancing the predictive ability of the model. The output layer uses a linear activation function for the generation of continuous score predictions. Training is achieved by the Adam optimizer and the use of a Huber loss, effective in dealing with outliers within the dataset.

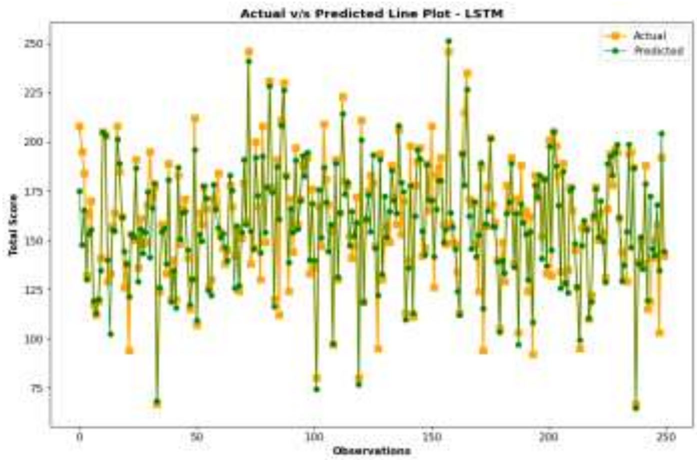

Fig.5. Graph showing analysis of line plot using LSTM model

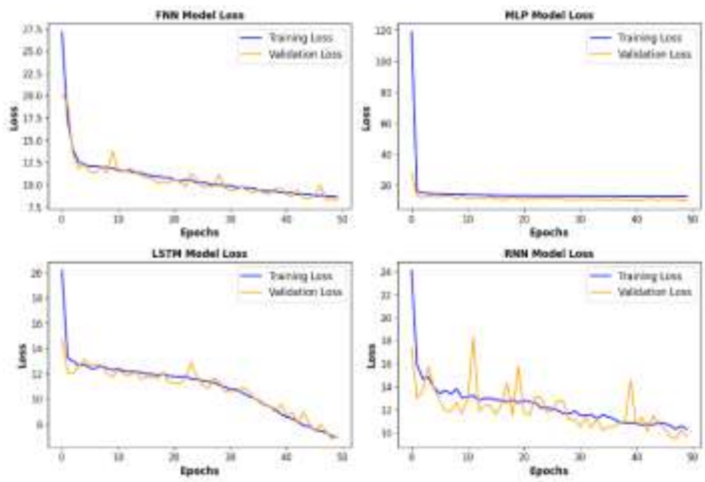

Fig.6. Training and validation loss of models over 50 epochs

The results from the graph (Fig.5) clearly show that the LSTM model excels in predicting IPL scores, achieving impressive accuracy with the predicted values closely matching the actual scores across a range of 70 to 240. The model's performance is strong, as it successfully captures the underlying patterns in the dataset, even in extreme cases. A notable highlight is the dense clustering of predictions between 125 and 185, indicating the model's precision in handling scenarios that fall within this frequent score range.

From Fig.6, it is observed that after training the model for 50 epochs, we observed significant convergence, which effectively minimized the loss function and boosted its predictive capabilities. The choice of a ReLU activation function within the LSTM layer played an important role in efficient extraction of non-linear features, while the linear activation in the output layer ensured the accuracy and continuity of the score predictions. Employment of the Huber loss function for training also ensured balanced sensitivity to outliers, thus making the model stable. Generally, this establishes the LSTM model as the most accurate and reliable of the models we tested. The predicted line consistently follows the actual values, showing the reliability of the model in capturing different scenarios.

From Table.4, the performance metrics emphasize the strength of the LSTM model compared to the different algorithms applied in this work.

Table.4. Predicted Evaluation Metric Values Using LSTM

Evaluation Metrics	MAE	MSE	RMSE	R^2
Values	7.50	131.84	11.48	0.85

The MAE, which is a measure of the average difference between the actual and predicted values, was approximately 7.50. This means that, when averaged, the model's prediction is away or deviates from the actual output by about 7.50 units. The MSE, which is the average of the squared differences between predictions and actual values, was 131.84. The RMSE, calculated from the MSE, was determined to be 11.48, indicating that this is the typical size of the errors in our forecasts. The value for, R^2 indicating how well the model fits the data, was obtained as 0.85. The R^2 value obtained would suggest that our model is able to explain around 85% of the score variations in IPL, thus demonstrating a strong ability to capture complex patterns correctly compared to alternative methods.

4.4 IMPLEMENTATION OF RNN MODEL FOR IPL SCORE PREDICTION

The selected RNN model for the prediction of IPL score captures sequential dependencies in historical match data very well. Using the ReLU activation function, the model represents an RNN layer with 128 units followed by a fully connected Dense layer of 64 units and an output of linear shape to attain precise and continuous score predictions. Optimal gradient update is achieved through the Adam optimizer, and robustness against outliers is improved with the Huber loss function. The input data is shaped again to be compatible with the RNN structure to allow the best learning. Training is performed on the model for 50 epochs with a batch size of 64, and validation data are incorporated to track generalization performance.

Table.5. Predicted Evaluation Metric Values Using RNN

Evaluation Metrics	MAE	MSE	RMSE	R^2
Values	10.06	209.08	14.45	0.76

The scores predicted are measured against actual values using standard performance metrics, presented in Table 5. The convergence plot of loss justifies consistent improvement in learning, and the actual vs. predicted score plot (Fig.7) indicates the robustness of the model in predicting actual match results. The output proves the capability of RNN to learn temporal patterns and thus is a feasible solution for the prediction of IPL scores.

By looking at Fig.7, it's clear that the predicted scores match up nearly with the actual scores. This implies that the RNN model does a great job of understanding and interpreting the sequences and patterns in IPL match data. The predicted values lie within the expected range, which means the model can adapt well to any conditions that can occur during a match. It is to be noted that the model excels in capturing score ranges, with minimal deviation from actual scores, which indicates that the model follows a well-generalized learning pattern.

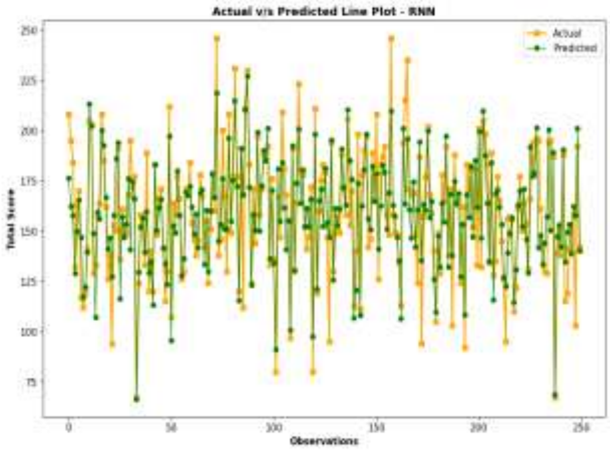

Fig.7. Graph showing analysis of line plot using RNN model

The ReLU activation function helps in capturing non-linear relationships, while the Huber loss function makes it robust against any outliers we might encounter. The model's performance metrics shown in Table 5, validate its predictive accuracy for estimating IPL scores.

4.5 COMPARATIVE ANALYSIS OF THE PROPOSED FRAMEWORK ACROSS ALL THE MODELS

The Table.6 presents a comparative analysis of actual and predicted scores across different models, including FNN, MLP, LSTM, and RNN, for key matches of IPL 2024 and the visual representation of the same is shown in Fig.8**.**

Table.6. Comparative Analysis of Actual Vs. Predicted Scores Across Models

Models	FNN	MLP	LSTM	RNN
Actual Score(F)	**113**	**113**	**113**	**113**
Predicted Score(F)	103	119	108	101
Actual Score(Q2)	**139**	**139**	**139**	**139**

Predicted Score(Q2)	131	145	136	147
Actual Score(E)	**172**	**172**	**172**	**172**
Predicted Score(E)	164	160	179	184
Actual Score(Q1)	**159**	**159**	**159**	**159**
Predicted Score(Q1)	165	151	163	149

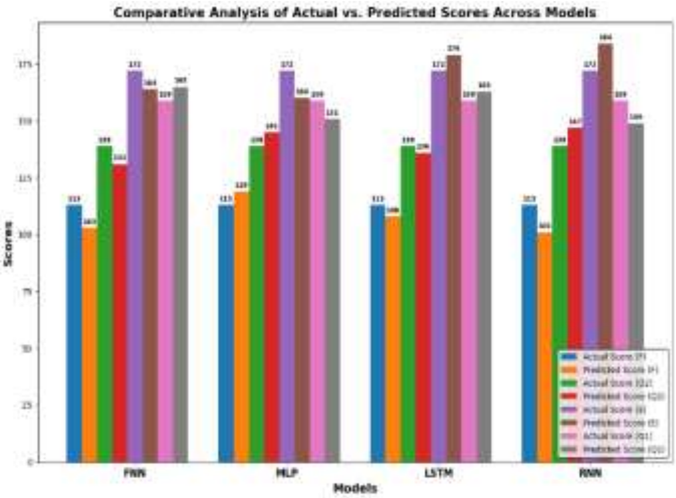

Fig.8. Comparative analysis of actual vs. predicted scores across models

The evaluation used real match data from the Final(F), Qualifier 2(Q2), Eliminator(E), and Qualifier 1(Q1) matches validated from Cricbuzz [15]. The models were trained on various input features such as the batting and bowling teams, overs completed, current runs, wickets lost, and performance in the last five overs in terms of runs scored and wickets taken. For the Final (SRH vs KKR) at 12.4 overs, 77 runs, and 7 wickets down, with 20 runs scored in the last five overs and 3 wickets lost, the model captured the innings progression effectively. Similarly, for Qualifier 2 (RR vs SRH), at 16.1 overs with 114 runs and 6 wickets lost, alongside a strong finish of 34 runs in the last five overs with only one wicket lost, the model provided predictions reflecting the actual trend. The Eliminator (RCB vs RR) scenario, where 9.3 overs were completed with 74 runs and 2 wickets down, followed by 30 runs in the last five overs with one wicket lost, showed predictions aligning well with the actual score. Likewise, for Qualifier 1 (SRH vs KKR) at 17 overs, 133 runs, and 9 wickets lost, with a lower finish of 16 runs in the last five overs and 4 wickets lost, the model reflected match conditions accurately. This comparison highlights the strengths and predictive accuracy of different models, in estimating final scores based on critical in-match parameters.

4.6 ANALYSIS OF THE PROPOSED FRAMEWORK USING EVALUATION METRICS WITH RESPECT TO ALL THE MODELS

The performance of our DL models – FNN, MLP, LSTM, and RNN, has been analyzed using key evaluation metrics: MAE, MSE, RMSE, and R^2 score.

Table.7. Comparison Of Performance Metrics for Different DL Techniques

Metrics	MAE	MSE	RMSE	R^2
FNN	8.69	172.39	13.12	0.80
MLP	10.73	226.48	15.04	0.74
LSTM	7.50	131.84	11.48	0.85
RNN	10.06	209.08	14.45	0.76

The comparative results are summarized in Table 7, and the visual representation of the same is shown in Fig.9. Table 7 clearly shows that the LSTM model outperforms the other techniques, achieving the lowest MSE value of 131.84 and RMSE value of 11.48, which indicate a lower overall error, and more precise predictions compared to the other models. Among all the other models implemented, LSTM has achieved the highest R^2 score of 0.85, signifying a better fit of the model to the actual data.

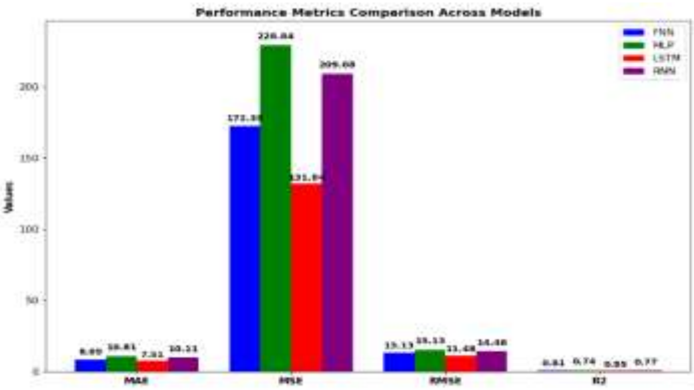

Fig.9. Comparison of performance metrics across the models

The findings indicate that LSTM is the best among the examined techniques for the considered dataset with outstanding predictive performance in both error minimization and model fit.

5. CONCLUSION

In this study, we concentrate on the use of deep learning models in sports analytics to make predictive insights. Mainly focusing on IPL score prediction based on various match conditions. In the study, FNN, MLP, LSTM and RNN are evaluated in terms of key performance metrics. Results indicate that the performance of LSTM is better than other models with the lowest MSE and RMSE and highest R^2 score, being the most suitable model for capturing sequential dependencies in cricket game data. Results indicate the increasing use of machine learning and deep learning in sports analytics to facilitate more accurate forecasting and strategic decision-making.

The application of ML and DL in the future cannot be ignored since such models assist in the interpretation of complex patterns and trends of match data to enable teams, analysts, and fans to make better decisions. Based on historical match data, in-match data, and sequence-based models, there can be an improved process of prediction, which enables teams to prepare or strategize accordingly based on predicted outcomes.

Although the results are promising, there is scope for improvement. Incorporating additional contextual features like individual player performance, pitch or surface conditions, weather conditions, and current game dynamics could further improve predictive accuracy. The use of ensemble methods combined with deep learning and traditional statistical approaches, or hybrid models could improve the predictive model. Additional work can extend this study by adding real-time score prediction, athlete performance prediction, and match result analysis, thereby extending the application of AI-based sports analytics.

Overall, this study improves sports analysis in cricket through the applications of deep learning techniques. This study provides a foundation for future studies in-game performance analysis, match outcome forecasting, and strategic decision-making in cricket and other sports.

5.1 FUTURE WORK

This study highlights the future application of deep learning techniques. The scope of this research can be Real-time prediction during the live ongoing matches with dynamic updates based on the current game conditions is future prospects or enhancements. Utilization of advanced machine learning and deep learning models, including hybrid models, can assist in capturing intricate relationships within the data, thereby enhancing prediction accuracy. Increasing the dataset to incorporate additional seasons, individual player performance statistics, varying playing conditions, and player statistics can further significantly enhance the overall model flexibility.

These techniques can also be adapted to apply to other formats of cricket, specifically Test cricket and One-Day Internationals (ODIs), and thereby extending their application into cricket analytics. There is still scope for the development of a user-friendly mobile or web interface that will be further useful and increase the accessibility for cricket analysts, fans, and teams, providing strategic recommendations and real-time insights. Furthermore, this approach can be expanded beyond cricket to other sports analytics applications, making it more useful for broader predictive modeling in sports. By continuously refining the framework and experimenting with innovative techniques, this research aims to contribute to the advancement of sports analytics, benefiting team management and coaches but also enriching the experience for analysts and fans.

REFERENCES

[1] "India Constitutes 90 Percent of one Billion Cricket Fans: ICC Research", Available at https://www.icc-cricket.com/media-releases/first-global-market-research-project-unveils-more-than-one-billion-cricket-fans, Accessed in 2018.

[2] "What is Cricket?-The 3 Formats", Available at https://www.icc-cricket.com/about/cricket/game-formats/the-three-formats, Accessed in 2023.

[3] N.H. Nguyen, D.T.A. Nguyen, B. Ma and J. Hu, "The Application of Machine Learning and Deep Learning in Sport: Predicting NBA Players' Performance and Popularity", *Journal of Information and Telecommunication*, Vol. 6, No. 2, pp. 217-235, 2022.

[4] "Cricsheet", Available at https://cricsheet.org/downloads/, Accessed in 2024.

[5] T. Xu and S. Baghaei, "Reshaping the Future of Sports with Artificial Intelligence: Challenges and Opportunities in Performance Enhancement, Fan Engagement and Strategic Decision-Making", *Engineering Applications of Artificial Intelligence*, Vol. 142, pp. 1-7, 2025.

[6] S.A.A. Shah, Q. Zaman, D. Wasim, J. Allohibi, A.A. Alharbi and M. Shabbir, "Optimal Model for Predicting Highest Runs Chase Outcomes in T-20 International Cricket using Modern Classification Algorithms", *Alexandria Engineering Journal*, Vol. 114, pp. 588-598, 2025.

[7] M.T.A.H. Sadi, Z.A. Bhuiyan, S. Yun, M.E. Haq, M.S. Rahman, M. Khan and A.K. Al Azad, "Deep Learning to Reliable Score Prediction in Hundred Ball Cricket Matches", *Proceedings of International Conference on Computer and Information Technology*, pp. 1-6, 2023.

[8] K. Kapadia, H. Abdel-Jaber, F. Thabtah and W. Hadi, "Sport Analytics for Cricket Game Results using Machine Learning: An Experimental Study", *Applied Computing and Informatics*, Vol. 18, No. 4, pp. 256-266, 2022.

[9] T. Singh, V. Singla and P. Bhatia, "Score and Winning Prediction in Cricket through Data Mining", *Proceedings of International Conference on Soft Computing Techniques and Implementations*, pp. 60-66, 2015.

[10] J. Kumar, R. Kumar and P. Kumar, "Outcome Prediction of ODI Cricket Matches using Decision Trees and MLP Networks", *Proceedings of International Conference on Secure Cyber Computing and Communication*, pp. 343-347, 2018.

[11] M. Asif and I.G. McHale, "A Generalized Non-Linear Forecasting Model for Limited Overs International Cricket", *International Journal of Forecasting*, Vol. 35, No. 2, pp. 634-640, 2019.

[12] T. Parasuraman, L. Marina, L. Mohandass, B. Selvamuthukrishnan, N.B. Naik and R. Ramakrishnan, "Enhancing Cricket Match Analysis through Deep Learning Techniques for Digital Spin", *Proceedings of International Conference on Electronics, Communication and Aerospace Technology*, pp. 1112-1117, 2023.

[13] C.D. Prakash and S. Verma, "A New in-Form and Role-based Deep Player Performance Index for Player Evaluation in T20 Cricket", *Decision Analytics Journal*, Vol. 2, pp. 1-18, 2022.

[14] T. Wang, X. Lai and J. Cao, "A Highly Efficient ADMM-based Algorithm for Outlier-Robust Regression with Huber Loss", *Applied Intelligence*, Vol. 54, pp. 5147-5166, 2024.

[15] E.E. Cust, A.J. Sweeting, K. Ball and S. Robertson, "Machine and Deep Learning for Sport-Specific Movement Recognition: A Systematic Review of Model Development and Performance", *Journal of Sports Sciences*, Vol. 37, No. 5, pp. 568-600, 2019.