Leveraging Artificial Intelligence and Machine Learning for Enhanced Cricket Team Selection and Performance Prediction

A Dissertation submitted in partial satisfaction of the requirements
for the degree “Master of Science in Artificial Intelligence & Machine Learning”

by

Ashish Kumar Arigala

ID Number: 24089001

Supervised by Dr. Douglas Mota Dias

Department of Computer Science and Information Systems
Faculty of Science and Engineering

University of Limerick

August 27, 2025

Statement

I hereby agree for this dissertation to be made available in the University of Limerick library and to future M.Sc. students in the Department of Computer Science and Information Systems.

Abstract

This dissertation presents an artificial intelligence (AI) and machine learning (ML) approach to optimizing the selection of cricket teams and to predict performance of players based on extensive data on One-Day International (ODI) matches. The study entails a careful preprocessing of data, which incorporates dealing with missing data, combining the various data sources, and then performing sophisticated feature engineering to come up with useful metrics of players like batting impact, bowling impact, and all-round index. Several predictive models were created and tested to find out their effectiveness. The performance of traditional machine learning models was moderate as Decision Tree model showed an MSE of 18.9140, RMSE of 4.3490, MAE of 2.5476, and R^2 of 0.7377; the Random Forest model had an MSE of 18.2926, RMSE of 4.2770, MAE of 3.2080, and R^2 of 0.7464; and the XGBoost model had an M The deep learning models were more effective, with LSTM attaining MSE of 12.8487, RMSE of 3.5845, MAE of 2.4374 and R^2 of 0.8218, whereas GRU had MSE of 11.8985, RMSE of 3.4494, MAE of 2.1404 and R^2 of 0.8350. A new hybrid model that integrated traditional and deep learning models performed the best among all the models with MSE of 7.9979, RMSE of 2.8281, MAE of 2.1988, and R^2 of 0.8891. The results indicate the possible use of AI-based models to provide objective data-driven information in strategic decisions to select the cricket team.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Douglas Mota Dias, for their invaluable guidance, support, and insightful feedback throughout the course of this dissertation. Their expertise and encouragement were instrumental in shaping this research. I also extend my thanks to the Department of Computer Science and Information Systems at the University of Limerick for providing the necessary resources and an enriching academic environment. Finally, I am grateful to the open-source community and the creators of the datasets used in this study, whose contributions made this research possible.

Table of Contents

Contents

List of Figures

List of Tables

1. Introduction

1.1 Background and Context

Cricket is the sport that is very much embedded in the culture of many countries, but it has developed far beyond the original scope of the sport. The modern game is becoming less intuitive and more data driven in the decision-making process. The introduction of new technologies of data collection, as well as the spread of analysis tools, has fundamentally changed the way teams train, play and choose their players. This change has been especially apparent in professional cricket where the performance of a player is not only minutely documented but also scrutinized in all aspects including batting strike rate, bowling economy, and the number of runouts. The quantity and the intricacy of this information require advanced procedures of the extraction of useful patterns and useful intelligence. The old style of player assessment which is subjective and has many biases are slowly being complemented, and even in some cases replaced, with more objective, quantitative methods. This trend means the growing importance of computational methods in the competitive advantage and optimal teamwork in the sport. Technology in sports analytics is not only a trend, but a paradigm shifts in the way performance is viewed and handled and opens the door to more scientific methods of team building and strategy development.

1.2 Research Motivation and Rationale

This research is motivated by the nature of the complexities and stakes of selection of cricket teams. Although there is abundance of performance data in many cases the selection process can be quite mysterious and heavily based on human judgement that is subjective or based on past preferences, or the ability to mentally handle such large volumes of information. It may cause inefficient team compositions, which affects the results of the matches and the success of the team in general. These disadvantages can be mitigated by machine learning and artificial intelligence that can identify the complicated patterns and predict the performance of the players more precisely and objectively. With the use of these technologies, the present study will attempt to offer a data-driven framework that can be used to help selection committees make more informed decisions, which should improve team performance and limit the use of subjective evaluations. This is justified by the fact that advanced analytical methods can convert raw player statistics into practical information, which will eventually lead to a more scientific and efficient method of building cricket teams.

1.3 Problem Statement

The conventional approaches to cricket team selection are usually quite subjective, hinging on the intuition and a minimum amount of data analysis by the selectors. It may result in the unstable performance of teams and the loss of potential players with high potential. The problem is that the huge and complicated data of cricket performance is not processed successfully to objectively analyse players and estimate their further input. It is obvious that an efficient, data-based system that will allow to objectively understand the capabilities of players and the dynamics of the team, will help to streamline the process of selecting players and improve the success of the team. This study fills this gap by creating AI/ML models to make more objective and accurate player evaluation.

1.4 Research Aim and Objectives

The main purpose of the given study is to design and test an AI/ML-based system to optimize cricket team composition and forecast player performance. In order to accomplish this goal, the following objectives are set:

- To gather, clean and merge complete datasets of ODI cricket player statistics.
- To carry out exploratory data analysis (EDA) to see the distributions and relationships of player performance.
- To design pertinent features that attract player influence, consistency, and recent form.
- To create and apply different machine learning and deep learning models to predict player performance.
- To develop and test a hybrid AI/ML model that will incorporate the best of both worlds: traditional and deep learning.
- To compare the performance of the developed models with the help of suitable evaluation parameters.
- To model and test AI-based team selections against conventional ways.

1.5 Research Questions

This study aims at addressing the following questions:

- What are the methods to efficiently integrate and preprocess various ODI cricket performance datasets to facilitate precise modelling with AI/ML methodology?

- What are the most influential engineered features of raw player statistics in terms of predicting performance within models such as XGBoost, Random Forest, and LSTM?
- What machine learning (Decision Tree, Random Forest, XGBoost) and deep learning (LSTM, GRU) models have the best performance prediction of player performance?
- • Does a hybrid model of traditional ML and deep learning perform better than individual models in prediction metrics of MSE, RMSE, MAE, and R^2?
- What is the comparative value of AI-based team selection, which is based on models, such as GRU, XGBoost, and the hybrid model, to traditional approaches in terms of team composition and estimated strength optimization?

1.6 Scope and Limitations

This research will be dedicated to the One-Day International (ODI) cricket data, where the batting and bowling statistics will be analysed, as well as the general information about the players. The models that have been created are aimed at forecasting the performance of individual players and help in selection of the team based on these figures. Although the research seeks to give a holistic framework, it has some limitations. Model accuracy may depend on the quality and availability of historical data, e.g. the fielding data or psychological factors are not well represented in detail because of data limitations. The simulated recent form of feature engineering is a simplification and more robust would need real-time, granular performance data. Moreover, the simulation of the team selection presupposes the predetermined criteria of sorting players (batsman, bowler, all-rounder), and the fixed composition of the team (e.g., 6 batsmen, 4 bowlers, 1 all-rounder), which does not reflect all the strategic peculiarities of the actual cricket team dynamics. The applicability of the results could also be restricted to ODI format and could need additional verification to other formats such as Test or T20 cricket. Further research may be done on the inclusion of a wider variety of data sources and dynamic team composition approaches.

1.7 Dissertation Outline

This dissertation is divided into seven chapters:

Chapter 1 presents the introduction to the study, including background, motivation, problem statement, research aims, objectives, questions, scope, and limitations.

Chapter 2 is a literature review of applicable literature pertaining to AI and ML in sports analytics, particularly in cricket performance prediction and team selection.

Chapter 3 explains the research methodology with information about the dataset, data preprocessing, feature engineering, model development, and evaluation metrics.

Chapter 4 describes the implementation procedure, and shows major outputs of preprocessing, exploratory analysis, and model execution.

Chapter 5 provides the results of model evaluation and compares the performance metrics, analysing feature importance.

Chapter 6 describes the findings, explains what they mean in terms of team selection, how the hybrid model was successful, and limitations and future work.

Chapter 7 summarizes the findings, key contributions, limitations, and recommendations to future study at the end of the dissertation.

2. Literature Review of Related Work

2.1 AI and ML in Sports Analytics

Sports analytics has transformed the way performance is measured, evaluated, and forecasted in different areas with the use of Artificial Intelligence (AI) and Machine Learning (ML) (Ghosh, Ramamurthy, Chakma, & Roy, 2023). In the past, sports analysis depended much on human eyes and gut feelings and simple statistical values. But the growth of data across a wide range of sources such as wearable sensors, high-resolution cameras, and match statistics has opened an unprecedented possibility of more advanced, data-driven insights. The size and complexity of these datasets can be handled specifically by AI and ML algorithms, which can extract hidden patterns and develop predictive models that can be used to make strategic decisions, improve player development, and maximize team performance (Claudino, et al., 2019).

Initial uses of AI in sports were limited to predictive models; early examples being predicting a match outcome based on past win-loss records. With the increasing computational power and complexity of algorithms, however, more subtle analyses have become possible. An example would be the use of ML models to monitor the movement of the players, evaluate tactical behaviour and even injury risk, which is then possible to intervene beforehand (Bunker & Susnjak, 2022). These have been further enhanced through deep learning methods, which are a subset of ML, and have been used to process unstructured data such as video footage to derive detailed information of player technique and game flow (Araújo, Couceiro, Seifert, Sarmento, & Davids, 2021).

In other sports such as basketball and soccer, AI has been used to examine passing networks amongst players, shot selection optimization and defensive layouts, and give coaches objective data on team strengths and weaknesses. On the same note, ML models can be used in individual sports to analyse biomechanical data to enhance training programs and enhance athletic performance. The use of AI and ML does not stay at the level of performance analysis; it is also used in talent identification, fan engagement, and even officiating, which offers a more objective and fair playing field (Pietraszewski, 2025).

Although it is quickly being adopted, some challenges still exist such as data privacy, require domain expertise to interpret the complex outputs of the models, and ethical issues of applying AI in high stake situations such as professional sports. However, the trend suggests that AI and ML will further increase their presence in the sporting arena, turning the industry into a more data-driven and scientifically controlled one.

2.2 ML/DL Applications in Cricket Performance Prediction

Machine Learning (ML) and Deep Learning (DL) techniques used in the prediction of performance in cricket have attracted a lot of attention because of the rich statistical data and intricate dynamics of sport. Such sophisticated ways of analysis are more refined than the traditional statistical models of predicting the outcomes of players and teams. The scholars have implemented numerous ML algorithms to forecast the performance of individual players, determine the most important factors of success, and even predict the outcomes of a match (Kapadia, Abdel-Jaber, Thabtah, & Hadi, 2022).

The initial use of ML in cricket was usually in supervised learning where Decision Trees, Support Vector Machines (SVMs), and Logistic Regression were mainly used to predict an outcome like the score of a player, the capability to take wickets, or the winner of a match. Such models usually use the past statistics of players, match conditions, and strength of opponents as features. As an example, player batting averages, strike rates, and bowling economy rates have been used to construct predictive models of future performance (Lokhande, Awale, & Ingle, 2025). The availability of more and more granular data, such as ball-by-ball data, or player tracking data, has further increased potential for accurate predictions.

Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been found to be effective Deep Learning models to analyse sequential data that is present in cricket. Since the performance of players commonly has temporal dependencies (e.g., the current performance of a player depends on his/her recent performance), LSTMs and GRUs are highly applicable in modelling such dependencies. These models have the ability to be trained on sequences, and thus they are useful in predicting continuous performance measures or determining trends over time (Chakwate, 2020). The hybrid models that combine the strengths of both the traditional ML and deep learning have also shown the capacity to capture the player attributes (which are static) and also capture performance trends (which are dynamic).

Besides the forecasts of individual players, ML/DL models also find numerous applications in other segments of cricket analytics, including fantasy cricket team building, strategy game planning, or even scouting. The capacity of these models to handle voluminous data and discover non-obvious correlations renders them priceless in the optimization of different aspects of sport (Goel, Davis, Bhatia, Malhotra, & Chandra, 2021). Nevertheless, there are still issues of data quality, feature engineering and interpretability of complex deep learning models that are highly important to establish trust and acceptance in cricket community.

2.3 Traditional vs AI-Based Team Selection

The selection of traditional cricket teams has been both art and science in the past and largely based on experience, intuition and subjective opinion of the selectors. Some of the factors that this strategy takes into consideration include recent form, previous performance, fame of player, team balance and even personal preferences or bias. Although experienced selectors have their irreplaceable knowledge, the amount of data that is produced in contemporary cricket makes it progressively more difficult to cover the entirety of the relevant data with the capabilities of human cognition and remain objective. This may cause inconsistency, the neglect of emerging talent or the choice of players on obsolete perceptions as opposed to the current performance indicators (Musat, et al., 2024).

On the contrary, the AI-based team selection is a data-driven, objective, and systematic method. Using machine learning algorithms, AI systems can examine large volumes of player data, match variables, and past results to determine the best player combinations and forecast their overall performance. Such systems are able to measure the contributions of the players, evaluate their suitability to certain positions, and even predict the team dynamics as a whole in different contexts. With the help of AI models, it is possible to identify less significant trends and associations, which are not visible to human selectors, to form more informed and potentially more successful team compositions (Pietraszewski, 2025).

Among the main benefits of AI-based selection, it is possible to note the elimination of human biases. With the sole emphasis on measurable data and predictive modelling, AI has the potential to offer a more fair evaluation of players, so that the decisions are made on the merit and potential, rather than personal perception or external influences. In addition, AI systems are capable of on-going learning and adapting to new data, refining their selection criteria and increasing predictive accuracy as they go. This is a continuous process of learning that enables a team to dynamically adapt to its strategies and player assessment to stay ahead of the dynamic game (Ouyang, et al., 2024).

Nevertheless, the process of switching to the AI-based selection does not come without difficulties. The fact that complex AI models are difficult to interpret, the ethical aspect of removing human judgment, and the requirement of high-quality, rich data are the main challenges. Although AI can give strong recommendations, human oversight can still be helpful in final decision-making to take into account such intangibles as team chemistry, leadership skills, and mental fortitude that are hard to define. Hence, the best solution can be a hybrid model in which AI will be used as a decision-support tool, not to replace human expertise but supplement it (Claudino, et al., 2019).

2.4 Common Models in Sports Data (RF, XGBoost, Decision Trees)

Within sports analytics, a number of machine learning models have been greatly applied to derive information out of complex data, forecast outcomes and streamline strategies. Decision Trees, Random Forests, and XGBoost are the most common among them because of their interpretability, robustness, and predictive ability. They are effective with tabular data that is typical of sports statistics and can be used to perform both classification (e.g. predicting whether a team will win or lose) and regression (e.g. predicting how many points will be scored by a player) tasks.

Decision Trees constitute a basic element of machine learning, providing a comprehensible, tree-like model, which resembles the human decision-making process. They divide data into subsets on the basis of feature values and form a set of rules that make a prediction. They are very easy to interpret, and this enables the analysts to comprehend what has gone into a decision.

In sport, decision trees can be applied in determining the key performance indicators that can either result in success or failure, like when a batsman will score a century or a bowler will get a multiple wicket-taking performance (Li & Mu, 2024). Individual decision trees however can be susceptible to overfitting i.e. they will work well on the training data but fail to perform on the unseen data.

Random Forests solve the problem of overfitting of single decision trees by using an ensemble learning technique. They build many decision trees in training and give the one that is the mode of the classes (classification) or mean prediction (regression) of the trees. Random Forests reduce the variance and improve generalization through the aggregation of the predictions of many trees. This renders them extremely efficient in the prediction of player performance, the determination of player attributes that might influence the performance, and even the prediction of the results of a match in games such as cricket where many variables interact in a complex manner (Wang, Wang, & Sun, 2025). Their capability of processing high-dimensional data and giving feature importance scores also makes them quite useful in sports analytics.

XGBoost (eXtreme Gradient Boosting) is an optimized, distributed gradient boosting library with a very fast, flexible, and portable interface. It is one of the ensemble methods which sequentially construct decision trees, and each new tree fixes the mistakes of the previous trees. The popularity of XGBoost has been immense in many areas including sports because of its faster performance, ability to handle missing values and high performance. It is very good at making predictions and has already been used in more complicated challenges involving sports prediction, including predicting individual player performance, game results, and even optimizing player recruitment strategies (Ouyang, et al., 2024). Its regularization methods prevent overfitting and as such, is a good option when using it on sports data in the real world.

Although these models are powerful, they need intensive feature engineering and hyperparameter optimization to obtain the best performance. Their success in sports analytics only proves that they are as useful as they have always been in transforming raw data into information that can be used to make strategic decisions and gain competitive advantage.

2.5 Deep Learning in Sports (LSTM, GRU, Hybrid)

Deep Learning (DL) has become a game changer in the field of sports analytics, especially when it comes to sequential data and complex patterns as well as high-dimensional inputs. In contrast to classical machine learning models, which tend to require hand-engineered features, deep learning models are capable of learning representations in a hierarchy of signals directly out of raw data, which makes them extremely useful in subtle analysis of player performance and game dynamics. Recurrent Neural Networks (RNNs) and, in particular, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are just some of the deep learning architectures that are especially useful with time-series data, which is common in sports (Shingrakhia & Patel, 2022).

The Long Short-Term Memory (LSTM) networks are a variation of RNN where the vanishing gradient problem is addressed to enable learning long-term relationships. The LSTMs can do this by having a complex inner process that operates on what is known as gates (input, forget, and output gates) which control the flow of information so that LSTMs can selectively remember or forget past information. LSTMs have also been used in sports to predict the movements of players, examine the game strategies across series of plays, and predict the trends of performance using the historical data, where the order and context of events are essential (Hossain, Rumpa, Hossain, Rahman, & Rahman, 2024). As an example, it can help predict the next shot of a batsman or the type of the next delivery of a bowler greatly by the power of LSTM to process sequential data.

Gated Recurrent Unit (GRU) networks are a simplified version of LSTMs, and have similar performance to LSTMs with fewer parameters, which can result in faster training. GRUs merge both the forget and input gates into one gate called an update gate and also have a reset gate. This simplified architecture renders them computationally more efficient and at the same time preserving the capability to learn long-term dependencies. GRUs are also becoming popular in sports to perform similar tasks as LSTMs, including predicting fatigue of players, analysing tactical patterns, and predicting the results of the game, particularly in cases where very large datasets are involved (Dey, Biswas, & Abualigah, 2024).

Hybrid Deep Learning Models is a superior method used to fuse various deep learning models or a combination of deep learning and classical machine learning. The logic of the hybrid models is to use the advantages of several strategies in order to consider the complex nature of sports information. In another example, a hybrid model can be trained with a Convolutional Neural Network (CNN) to learn spatial information in the data (e.g. player positions on the field) and then feed the resulting features to an LSTM or GRU to learn temporal trends. The other popular hybrid solution is to use deep learning models to extract or learn features and use a conventional machine learning model to complete the prediction or classification task. Such hybrid architectures are usually more performative because they are able to capture more data characteristics and complex relations and make more accurate and robust predictions in sports analytics (Ouyang, et al., 2024).

The complexity of deep learning models, including hybrid models, is growing, and it is already expanding the possibilities of the sports performance analysis, providing previously unseen opportunities in data-driven decision-making.

2.6 Gaps and Justification for This Study

Although the above sections have identified key developments in the use of AI and ML in sports analytics, especially in the field of cricket performance prediction, there are a number of gaps that have to be filled by this research. The current studies tend to concentrate on either the classical machine learning models or deep learning models separately, and little research compares and develops the hybrid models of cricket player performance prediction and team selection. Moreover, most of the studies have a tendency to focus on either aspects of player performance (e.g., batting or bowling) or match results, as opposed to a comprehensive assessment of players in order to compose the team.

Among the gaps, it is possible to note the absence of thorough studies of the practical outcomes of using AI in team selection opposed to the traditional approaches that are human-oriented. Although certain research shows predictive capabilities of the AI models, there is less focus on how these models can directly guide and improve real-life team selection process, particularly in such a complicated sport as cricket. The key role of feature engineering where raw player statistics get converted into more meaningful and predictive statistics that capture the impact, consistency and recent form of players which are essential to a thorough player analysis is often ignored in the current literature as well.

Furthermore, the complexity of deep learning models, although being improved, remains a challenge to achieve acceptance by the sports professionals used to more rule-based decision-making. A middle ground between interpretable traditional ML models and the predictive ability of deep learning may provide a more acceptable and practical solution to deploy in the real world.

This research is validated by the fact that it tries to fill these gaps. To begin with, it suggests and critically tests a hybrid approach to AI/ML, which combines conventional and deep learning algorithms, to be able to attain better predictive accuracy when it comes to the assessment of cricket players. Second, it focuses on feature engineering in general to develop powerful metrics by which to evaluate the players. Third and most importantly, this study goes beyond prediction, as it simulates and compares the AI-based team selections to the classic ones, thus offering the selection committees with the practical framework to consider. This dissertation will help advance the field of sports analytics in its own way by offering data-driven, objective, and comprehensive solution to cricket team selection, which can be useful in terms of strategic decision-making and more scientific approach to team building in cricket.

3. Methodology

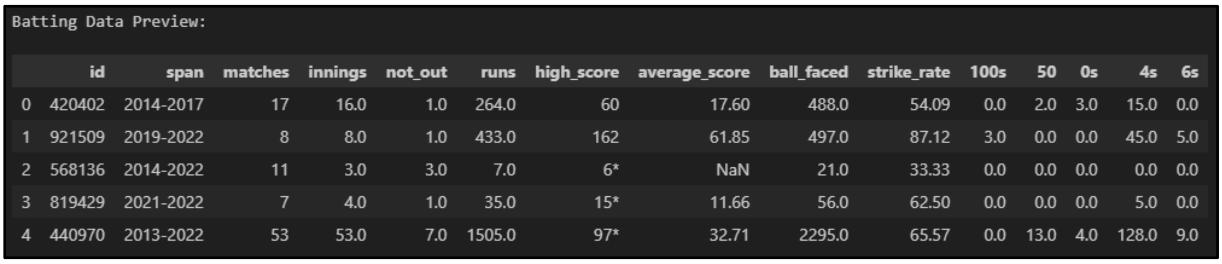
3.1 Research Design

The research design that is taken in this study is basically quantitative and experimental in nature as it is centred on the development, application, and testing of different Artificial Intelligence (AI) and Machine Learning (ML) models to predict performances of cricket players and select teams. The methodology is systematically carried out, starting with the data acquisition process and preprocessing, the explanatory data analysis, feature engineering, model building, and intensive evaluation. The experimental aspect of design entails evaluating the performance of various model architectures, such as the traditional machine learning, deep learning, and a new type of a hybrid model, when compared to a list of predetermined metrics. This is a comparative analysis that will determine the best method of objective player evaluation and team building. A simulation element is also included as a part of the research to show the practical usage of AI-based selection and compare it to the traditional approach to validate the relevance of the proposed framework in the real-life setting. Its design is aimed at reproducibility and transparency, which implies that all the processes, such as data processing and model testing, are reported and can be justified.

3.2 Dataset Description

This research data includes extensive historical records of One-Day International (ODI) cricket players in terms of their performances. The data in these datasets were obtained in the form of publicly available cricket statistics repositories and contain comprehensive data about batting, bowling, and other player characteristics, in general. The major data sets are:

ODI_batting.csv: This file has batting performance statistics of individual players given in Figure 1.

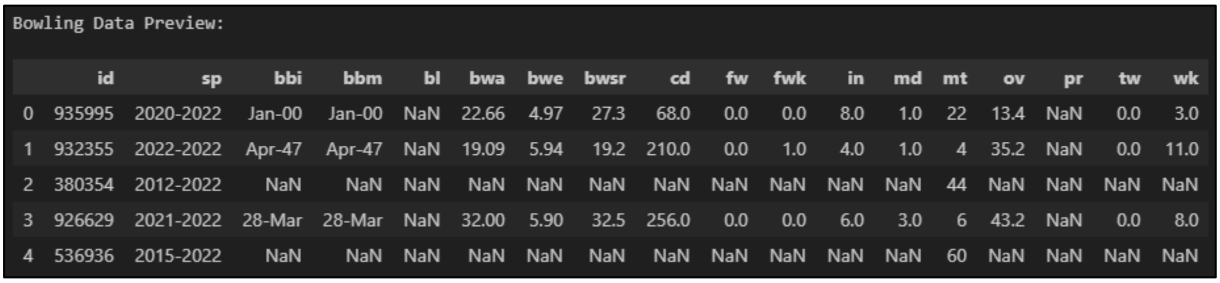


```
Batting Data Preview:

       id       span  matches  innings  not_out    runs  high_score  average_score  ball_faced  strike_rate  100s    50   0s     4s   6s
0  420402  2014-2017       17     16.0      1.0   264.0          60          17.60       488.0        54.09   0.0   2.0  3.0   15.0  0.0
1  921509  2019-2022        8      8.0      1.0   433.0         162          61.85       497.0        87.12   3.0   0.0  0.0   45.0  5.0
2  568136  2014-2022       11      3.0      3.0     7.0          6*            NaN        21.0        33.33   0.0   0.0  0.0    0.0  0.0
3  819429  2021-2022        7      4.0      1.0    35.0         15*          11.66        56.0        62.50   0.0   0.0  0.0    5.0  0.0
4  440970  2013-2022       53     53.0      7.0  1505.0         97*          32.71      2295.0        65.57   0.0  13.0  4.0  128.0  9.0
```

Figure 1: ODI_batting.csv containing batting statistics

ODI_bowling.csv: Has bowling performance indicators of individual players given in Figure 2.

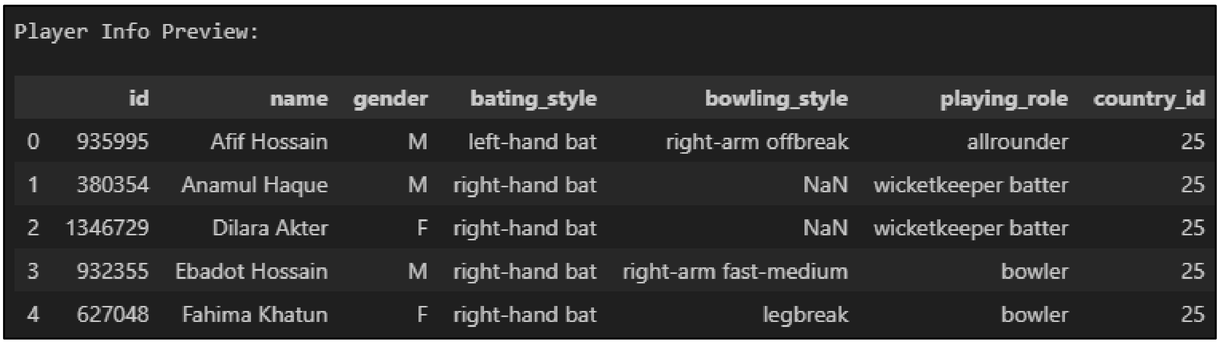


```
Bowling Data Preview:

       id         sp     bbi     bbm   bl    bwa   bwe  bwsr     cd   fw  fwk   in   md  mt    ov   pr   tw    wk
0  935995  2020-2022  Jan-00  Jan-00  NaN  22.66  4.97  27.3   68.0  0.0  0.0  8.0  1.0  22  13.4  NaN  0.0   3.0
1  932355  2022-2022  Apr-47  Apr-47  NaN  19.09  5.94  19.2  210.0  0.0  1.0  4.0  1.0   4  35.2  NaN  0.0  11.0
2  380354  2012-2022     NaN     NaN  NaN    NaN   NaN   NaN    NaN  NaN  NaN  NaN  NaN  44   NaN  NaN  NaN   NaN
3  926629  2021-2022  28-Mar  28-Mar  NaN  32.00  5.90  32.5  256.0  0.0  0.0  6.0  3.0   6  43.2  NaN  0.0   8.0
4  536936  2015-2022     NaN     NaN  NaN    NaN   NaN   NaN    NaN  NaN  NaN  NaN  NaN  60   NaN  NaN  NaN   NaN
```

Figure 2: ODI_bowling.csv containing bowling performance metrics

all_players.csv: Contains basic details of the players including their name, gender, batting style, bowling style, playing role and country as shown in Figure 3.

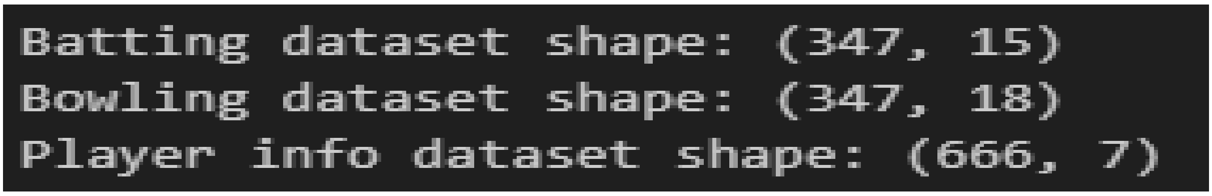


```
Player Info Preview:

        id            name  gender    bating_style           bowling_style         playing_role  country_id
0   935995    Afif Hossain       M   left-hand bat      right-arm offbreak           allrounder          25
1   380354    Anamul Haque       M  right-hand bat                     NaN  wicketkeeper batter          25
2  1346729     Dilara Akter       F  right-hand bat                     NaN  wicketkeeper batter          25
3   932355  Ebadot Hossain       M  right-hand bat   right-arm fast-medium               bowler          25
4   627048   Fahima Khatun       F  right-hand bat                legbreak               bowler          25
```

Figure 3: all_players.csv containing player demographic and role information

At the first loading, the following dimensions were observed in the datasets given in Figure 4:



```
Batting dataset shape: (347, 15)
Bowling dataset shape: (347, 18)
Player info dataset shape: (666, 7)
```

Figure 4: Comprehensive data structure overview

This data was then combined to generate a single player statistics dataset by which all further analysis and model development were based. The final merged dataset once processed was shaped (347, 38) i.e. a total of 347 different players with 38 features each, including batting and bowling features, as well as demographic and role details of the players.

3.2.1 Data Sources

The research data used was obtained based on publicly available online databanks that contained statistics of cricket. These sources create comprehensive recordings of the One-Day International (ODI) matches, giving a rich historical background on the performance of players. Data consists of granular data like runs scored, wickets taken, balls faced, and overs bowled along with different averages and strike rates running over a time period between 2002 and 2023. This will make the study reproducible and independent confirmation of the results is possible due to the use of publicly available data. The files of particular data that were utilized were ODI_batting.csv, ODI_bowling.csv, and all_players.csv.

3.2.2 Structure of Batting, Bowling, Player Info

Table 1 contains all these datasets have their individual structure and reflect certain details of the player profile and his performance:

Table 1: Structure of Batting, Bowling, Player Info

Dataset	Column	Description
ODI_batting.csv	id	Unique identifier for each player
	span	Career span in ODIs (e.g., '2014-2017')
	matches	Total number of matches played
	innings	Number of innings batted
	not out	Number of times remained not out
	runs	Total runs scored
	high score	Highest individual score
	average score	Batting average
	ball faced	Total balls faced
	strike rate	Batting strike rate
	100s	Number of centuries scored
	50	Number of half-centuries scored

	0s	Number of ducks (scores of zero)
	4s	Number of fours hit
	6s	Number of sixes hit
ODI_bowling.csv	id	Unique identifier for each player
	sp	Career span as a bowler
	bbi	Best bowling in an innings
	bbm	Best bowling in a match
	bwa	Bowling average
	bwe	Bowling economy rate
	bwsr	Bowling strike rate
	cd	Number of 5-wicket hauls
	fw	Number of 4-wicket hauls
	fwk	Number of 5-wicket hauls (possibly duplicate of cd)
	in	Number of innings bowled
	md	Number of maiden overs
	mt	Matches bowled in
	ov	Overs bowled
	pr	Placeholder column (often empty)
	tw	Total wickets taken
	wk	Total wickets taken (often duplicate of tw)
all_players.csv	id	Unique identifier for each player
	name	Player's full name
	gender	Player's gender
	batting style	Batting style (e.g., 'right-hand bat')
	bowling style	Bowling style (e.g., 'right-arm off break')
	playing role	Player’s primary role (e.g., 'allrounder', 'wicketkeeper batter')
	country_id	Country identifier for the player

3.3 Data Preprocessing

Preprocessing of data is an essential part of any data-driven study, which guarantees the quality, consistency, and appropriateness of data to be further analysed and trained on a model. The research entailed some major preprocessing procedures, such as missing values and the combination of different datasets.

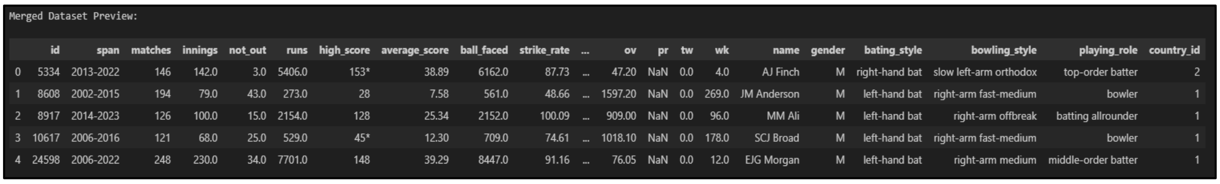
3.3.1 Missing Value Handling

When first examining the three datasets (ODI_batting.csv, ODI_bowling.csv, and all_players.csv), it was discovered that each of them had some missing values, though to different degrees. Incomplete data may cause serious problems in terms of performance and reliability of machine learning models and thus a strong imputation strategy is required. Missing values in the columns that contained numeric data were filled using the median of the columns. Median was used instead of mean to reduce the effect of outliers hence better preservation of distribution of the data. In categorical columns, mode (most frequent value) of columns was used to fill in missing values. This method will mean that the values imputed are reflective of the distribution of the existing data both in terms of numerical and categorical features.

The batting and player information datasets did not have any missing values left after the application of this imputation strategy. Nonetheless, the bowling dataset also had a few columns that had a high percentage of missing data, mainly the column bl and pr, which were completely null. These were the columns that were then eliminated since they did not offer any meaningful information. The other missing values in the bowling dataset mostly in bbi, bbm, bwa, bwsr, cd, fw, fwk, in, md, ov, tw and wk were also filled by the median or mode as the case may be. Such a systematic process made the datasets complete and ready to be further processed.

3.3.2 Data Merging

To make a complete player profile that combines batting, bowling and general player data, the three separate datasets were combined as shown in Figure 5. The merge operation was done by the common id column where each player is identified uniquely in all datasets. A merge out was initially performed on the batting and bowling data sets to make sure that the players in either of the data sets were included in the final result and suffixes _batting and _bowling were added to differentiate common column names. This combined set was then left-merged with the dataset containing the player information, so that all the player statistics would be linked to their respective demographic and role data. Any new missing values that could have been created in the process of merging (e.g. in the case of a player having batting stats but no bowling stats, or vice-versa) were treated using the same missing value imputation strategy as above. The merged data set, player_stats, gave a single and complete picture of each player and it formed the basis of feature engineering and model building.



Merged Dataset Preview:

	id	span	matches	innings	not_out	runs	high_score	average_score	ball_faced	strike_rate	...	ov	pr	tw	wk	name	gender	bating_style	bowling_style	playing_role	country_id
0	5334	2013-2022	146	142.0	3.0	5406.0	153*	38.89	6162.0	87.73	...	47.20	NaN	0.0	4.0	AJ Finch	M	right-hand bat	slow left-arm orthodox	top-order batter	2
1	8608	2002-2015	194	79.0	43.0	273.0	28	7.58	561.0	48.66	...	1597.20	NaN	0.0	269.0	JM Anderson	M	left-hand bat	right-arm fast-medium	bowler	1
2	8917	2014-2023	126	100.0	15.0	2154.0	128	25.34	2152.0	100.09	...	909.00	NaN	0.0	96.0	MM Ali	M	left-hand bat	right-arm offbreak	batting allrounder	1
3	10617	2006-2016	121	68.0	25.0	529.0	45*	12.30	709.0	74.61	...	1018.10	NaN	0.0	178.0	SCJ Broad	M	left-hand bat	right-arm fast-medium	bowler	1
4	24598	2006-2022	248	230.0	34.0	7701.0	148	39.29	8447.0	91.16	...	76.05	NaN	0.0	12.0	EJG Morgan	M	left-hand bat	right-arm medium	middle-order batter	1

Figure 5: Merged Dataset Overview

3.4 Exploratory Data Analysis (EDA)

The merged player_stats dataset was subject to Exploratory Data Analysis (EDA) to understand player attribute distribution, reveal relationships between variables, and find patterns that might be used in feature engineering and model development. There was a high usage of visualizations to comprehend the inherent nature of the information.

Batting Performance Metrics: A set of histograms was created to show the distribution of the most important batting performance indicators, like average_score, and strike_rate as shown in Figure 6. Through these distributions, it was possible to understand the normal range of performance between players and also the existence of skewness and outliers.

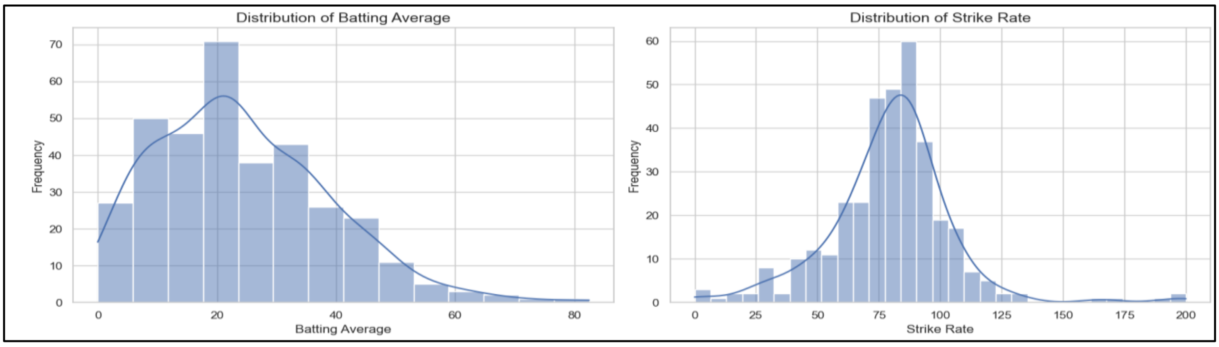


Figure 6: Distribution of Batting Average & Strike Rate Among ODI Players

Scatter plots were adopted to study the relationship between matches played and runs scored, and between 100s (centuries) and 50 s (half-centuries) as shown in Figure 7. These plots were useful in seeing the relationship between experience and performance and the frequency of batting milestones such as home runs among the players.

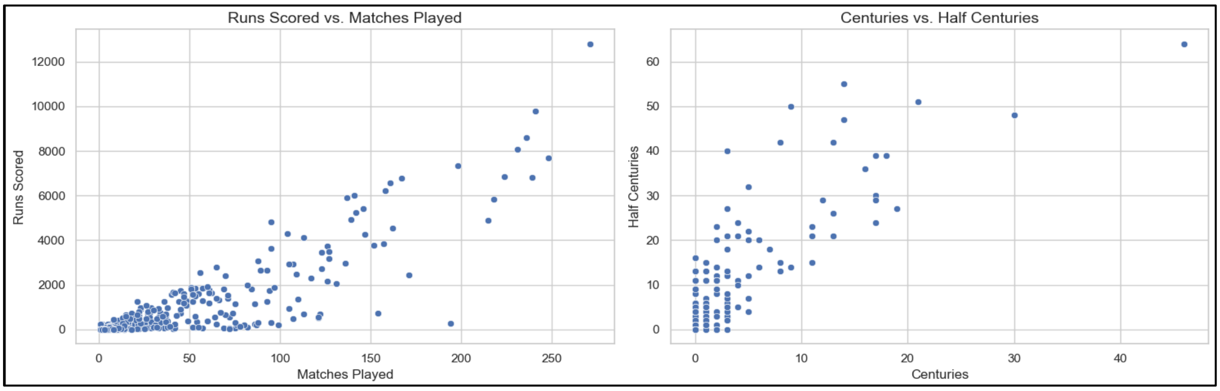


Figure 7: Relationship b/w Matches Played and Total Runs Scored & 50s and 100s

Bowling Performance Metrics: Bowling Performance Measures: Just as in batting, histograms were plotted on bwa (bowling average) and bwe (economy rate) as shown in Figure 8 to know their distributions. Scatter plots were used to represent the connection between matches played and wk (wickets taken), and between bwe (economy rate) and cd (5-wicket hauls). Such visualizations were useful in evaluating the performance of bowlers and determining trends in their performance.

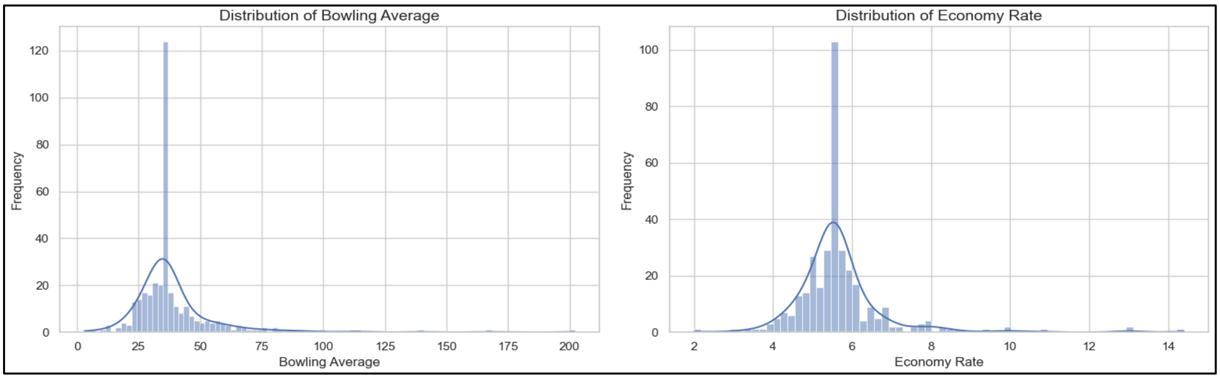


Figure 8: Bowling Average and Economy Rate Distribution in ODI Cricket

Correlation Analysis: Correlation matrix was calculated of all numerical columns in the player_stats dataset. The graphical representation of this matrix was a heatmap displaying the intensity and direction of linear correlations between various statistics of players as shown in Figure 9. This analysis was essential in determining the highly correlated features which may be a sign of multicollinearity and also to find out which measures tend to move in similar directions. As an example, there should be a strong positive correlation between the runs and the matches, and a negative correlation would imply that lower bowling averages would lead to more wickets as the case of bwa and wk.

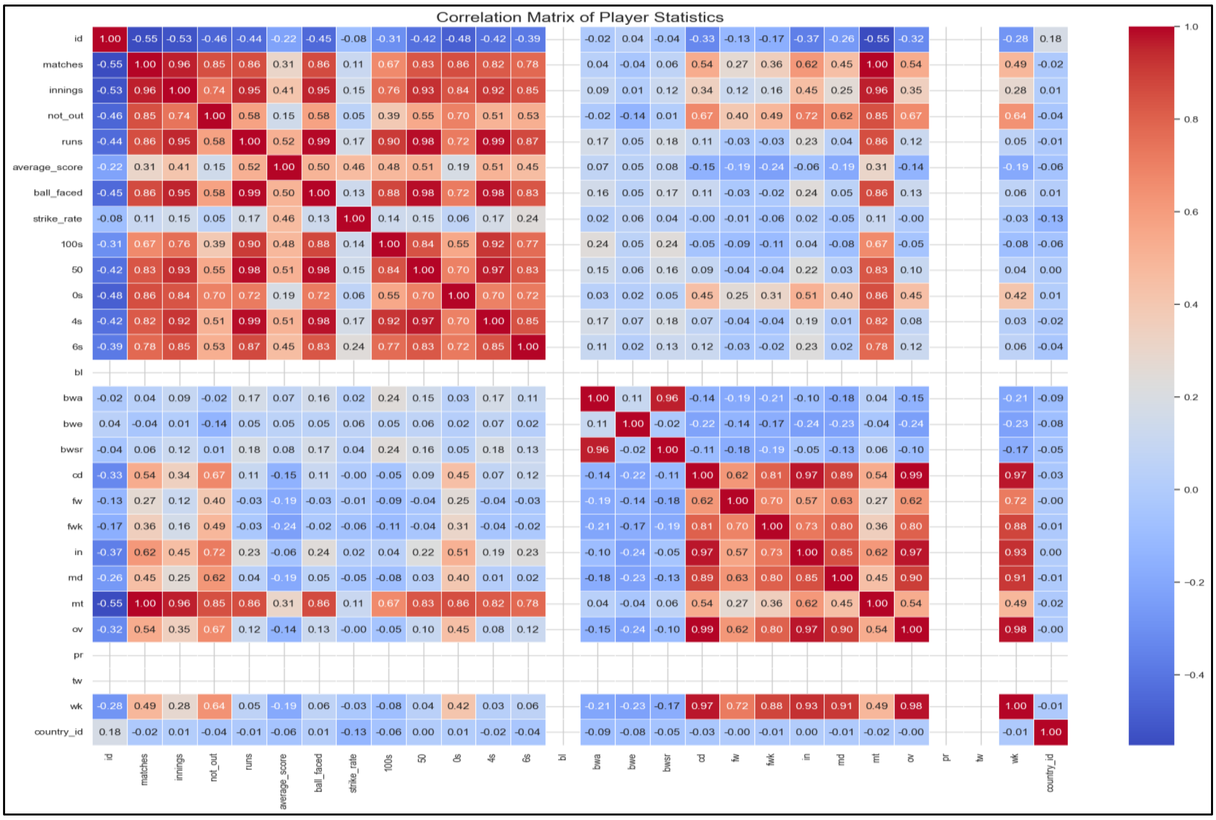


Figure 9: Correlation Matrix of Cricket Performance Variables

Performance by Player Role: Box plots were created to examine how average_score, bwa, strike_rate and bwe are distributed across the various categories of playing_role (e.g. batsman, bowler, all-rounder, wicketkeeper-batsman).

This analysis offered some insights into the common profile of performance of players in various positions with different characteristics and expectations of the position. This, as an example, means that the batsmen are supposed to have higher batting average and strike rate, and the bowlers should have lower bowling average and economy rate as shown in Figure 10.

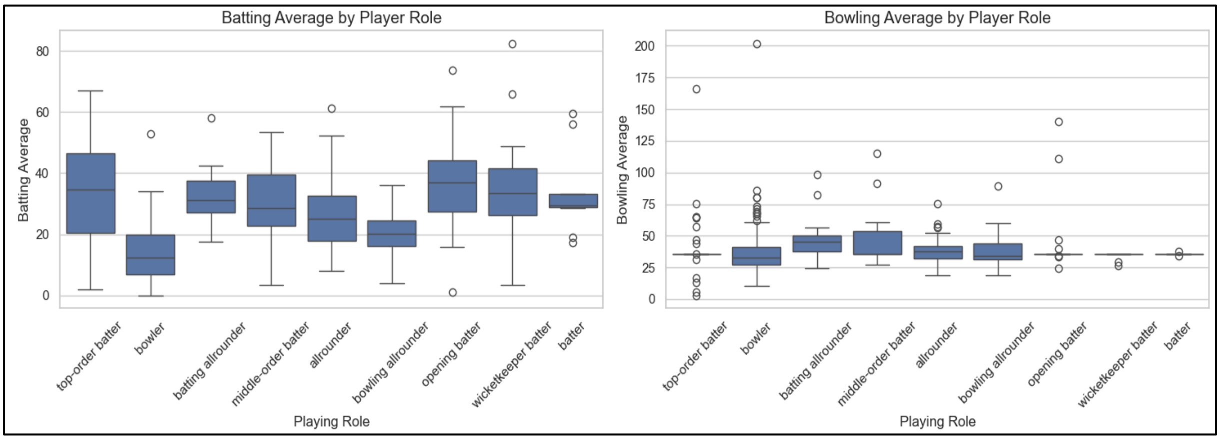


Figure 10: Batting and Bowling Average by Player Role

Top Player Analysis: The bar plots were constructed to show the top 10 batsmen in terms of batting average (20 matches played or more) as shown in Figure 11 and top 10 bowlers in terms of bowling average (20 matches played or more and at least 1 wicket) given in Figure 12.

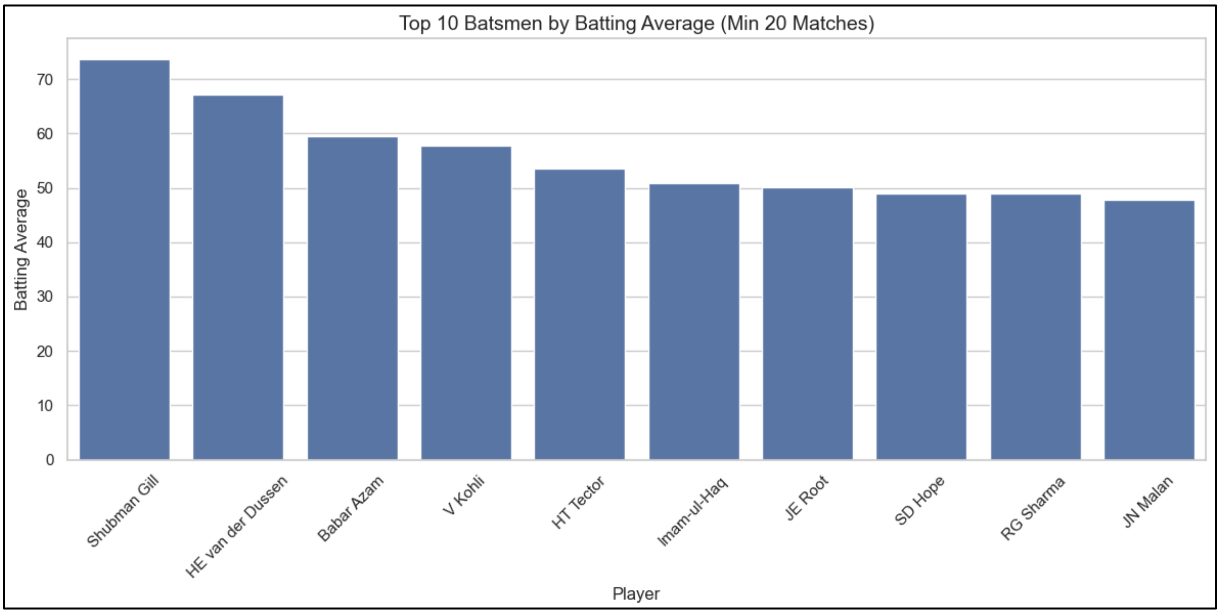


Figure 11: Top 10 Batsmen by Batting Average

These visualizations assisted in the selection of the elite performers according to particular criteria and gave a fast view of the best players in the dataset. EDA stage played a critical role in learning about the nature of the dataset, proving or disapproving assumptions, and informing the further feature engineering and model development process.

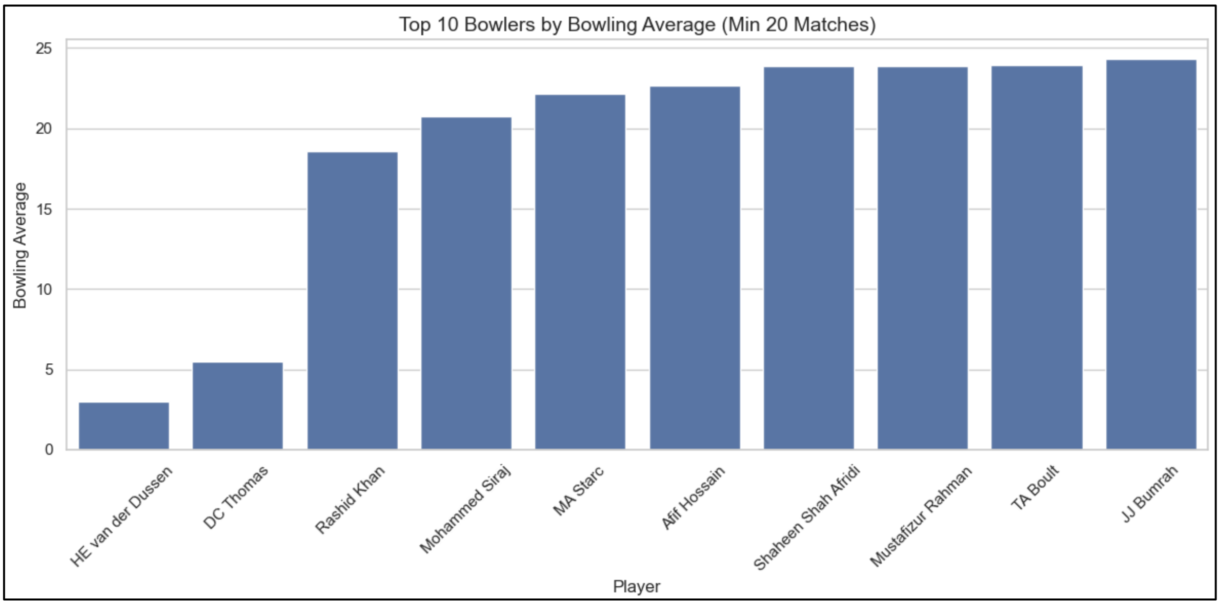


Figure 12: Top 10 Bowlers by Bowling Average

3.5 Feature Engineering & Selection

Feature engineering is an important machine learning process where new features are generated with the help of raw data that is available to enhance the performance of predictive models. In the present study, a number of domain-specific features were designed to reflect finer details of the performance and influence of a cricket player. The goals of these engineered features are to give a more detailed picture of a player's skills over and above just basic statistics and as such they are better at predicting overall performance and helping with team selection. The key features that were engineered include:

Batting Impact: It measures the overall contribution of a batsman to the game, which is calculated by adding the average number of runs scored by the player (average score) and the rate at which he scores (strike rate). The greater the batting impact the more effective and dynamic is the batsman. It is calculated as:

$$Batting\ Impact\ = \frac{(Average\ Score\ \times\ Strike\ Rate)}{100}$$

Bowling Impact: This is a measure of the effectiveness of a bowler, based on how he or she takes wickets (wickets per match) and how economical he or she is with his or her bowling (economy rate). A greater bowling impact would indicate a more effective and cost-effective bowler. It is calculated as:

$$Bowling\ Impact\ = \left(\frac{Wickets}{Matches}\right) \times \left(\frac{1}{Economy\ Rate}\right)$$

All-Round Index: An all-round index was devised to consider players who have a strong contribution with bat and ball by adding both their batting and bowling impacts as shown in Figure 13. This aspect assists in determining and rating of true all-rounders. It is computed as:

$$All - Round\ Index\ = \ Batting\ Impact\ +\ Bowling\ Impact$$

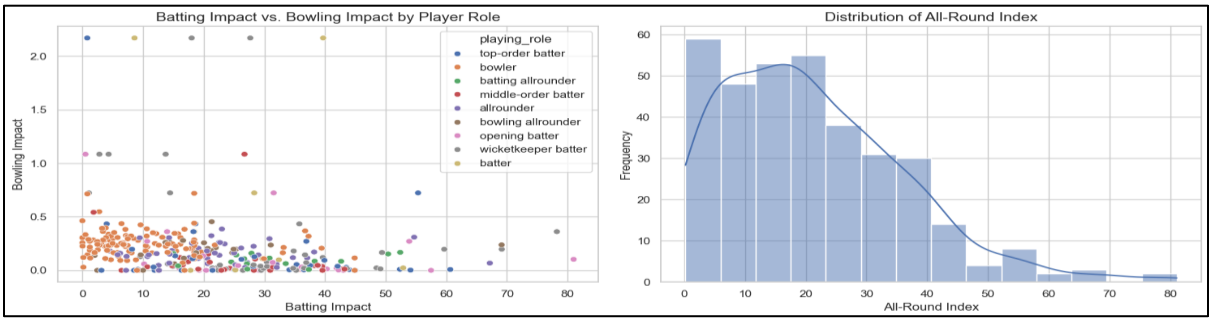


Figure 13: Performance Metrics Distribution by Player Role

Batting Consistency: This is the measure of the consistency at which a batsman scores important milestones (half-centuries). It is computed as:

$$Batting\ Consistency = \frac{Number\ of\ 50s}{Innings\ Batted}$$

Bowling Consistency: This is the measure of consistency of a bowler in taking wickets in relation to the number of overs bowled. It is computed as:

$$Bowling\ Consistency = \frac{Wickets\ Taken}{Overs\ Bowled}$$

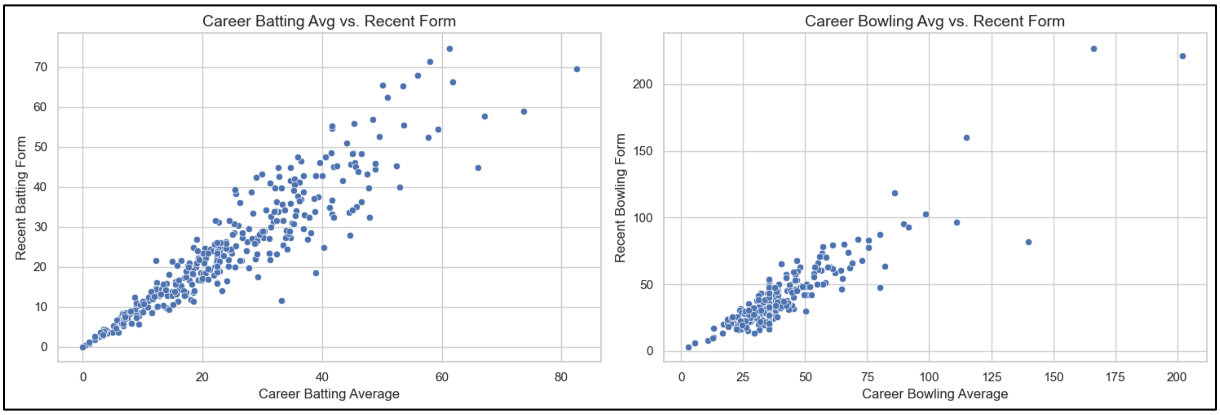


Figure 14: Player Performance Average VS Recent Form

Recent Batting Form and **Recent Bowling Form**: To reflect the performance trend of players, simulated recent form values were added as shown in Figure 14. These were drawn based on normal distribution with an average of the score of the player (in case of batting) or bowling average (in case of bowling) and standard deviation equal to the average. Although they were simulated in this experiment, in a practical situation these would be based on real recent match data.

Once these features were engineered, dataset was again checked whether any new infinite or NaN values have been introduced (e.g. when dividing by zero, e.g. when the player has zero matches or overs bowled). The same imputation method was applied to these values as to the first preprocessing step (median in numerical columns). These features were chosen based on the domain knowledge of cricket and the information obtained through exploratory data analysis in order to build a set of predictors which will be useful in predicting player performance.

3.6 Model Development

The research formulated and tested various machine learning and deep learning algorithms to forecast the performance of cricket players. The models were divided into baseline machine learning models, deep learning models, and a hybrid model that was to capture various features of the complex player performance data. The general strategy was to divide the pre-processed and feature-engineered dataset into training and test sets in order to guarantee the solidity of the assessment of the model generalization skills. The model performance was optimized, and overfitting was avoided by using hyperparameter tuning and cross-validation methods.

3.6.1 Baseline ML Models

A number of conventional machine learning models were applied to set a performance benchmark. The models are popular due to their interpretability and predictive performance on many tasks:

Decision Tree Regressor: This is a supervised learning technique applied in regression, but it is non-parametric. It divides the data into subsets according to the values of the features and produces a tree structure of choices. Decision Trees are easy to understand and can model non-linear relationships but are easily overfit without regularization.

Random Forest Regressor: A machine learning algorithm that builds a number of decision trees in training and returns the average of the predictions of the individual trees. Random Forests have lower variance and better generalization than single decision trees and are thus robust and very accurate in most regression problems. They also give an indication of feature importance.

XGBoost Regressor: A distributed gradient boosting library optimized in terms of speed and performance. XGBoost is a sequential ensemble technique that constructs decision trees sequentially, one tree at a time; each tree is then used to correct the mistakes of the ones that came before it. It uses regularization methods to avoid overfitting and has been shown to be highly predictively accurate and efficient, making it a common selection in competitive machine learning.

The engineered features were used to train these baseline models to predict a composite player performance index which is used as a comparative benchmark to the more complex deep learning and hybrid models.

3.6.2 Deep Learning Models

Since the performance of players is sequential in time and the relationships may be complex and non-linear, deep learning models were used. In particular, Recurrent Neural Networks (RNNs) have been selected due to the virtue of sequential data processing:

Long Short-Term Memory (LSTM): A type of RNN which is able to learn long-term dependencies. LSTMs have a gating mechanism (input, forget and output gates) that regulate the passage of information to determine what it will recall or forget of the past. This renders them particularly useful in time-series forecasting problems, where sequence and context is essential. In the present work, LSTMs were applied to temporal trends in the data on player performance. The LSTM architecture representation is given in Figure 15.

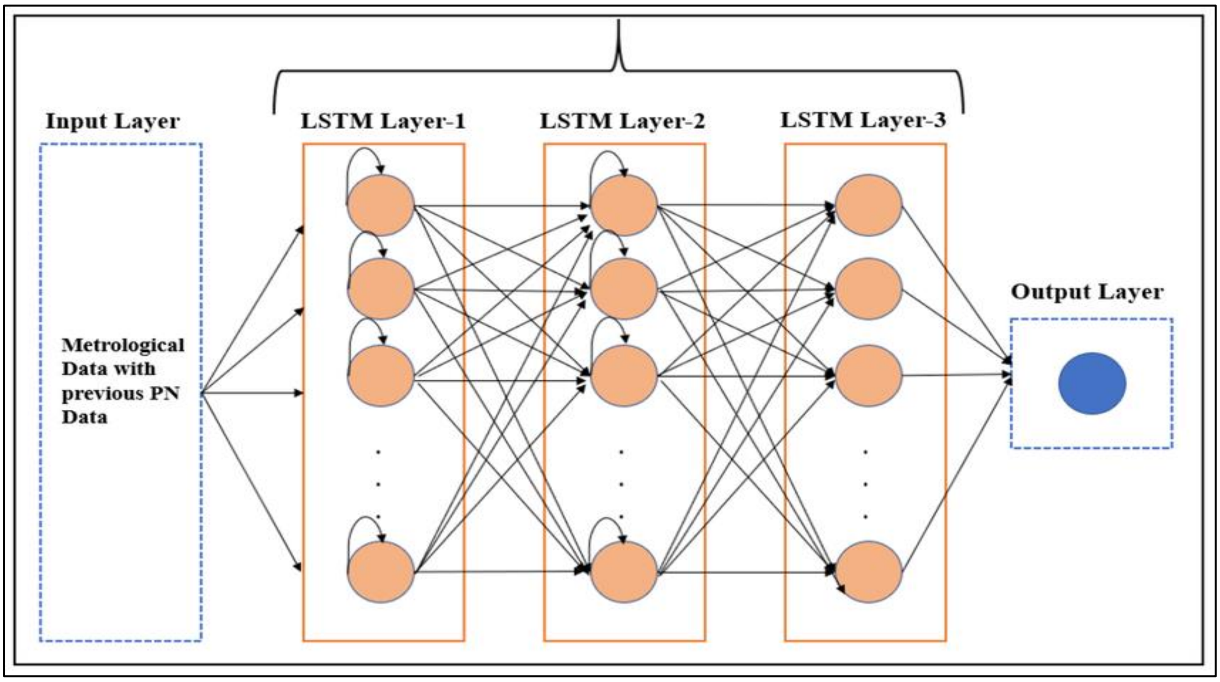


Figure 15: LSTM Architecture

Gated Recurrent Unit (GRU): GRUs are a simplified form of LSTM, where instead of separate forget and input gates, GRUs have a single update gate, in addition to a reset gate. It is compact in architecture, has similar performance to LSTMs, has fewer parameters, and hence, it trains faster and is computationally simpler. GRUs have also been used to capture the sequential nature of the player performance.

LSTM and GRU models were both set up to have multiple layers, dropout regularizations, and suitable activation functions in order to learn complex representations using the player statistics. The key architectural differences between LSTM and GRU are given in Figure 16.

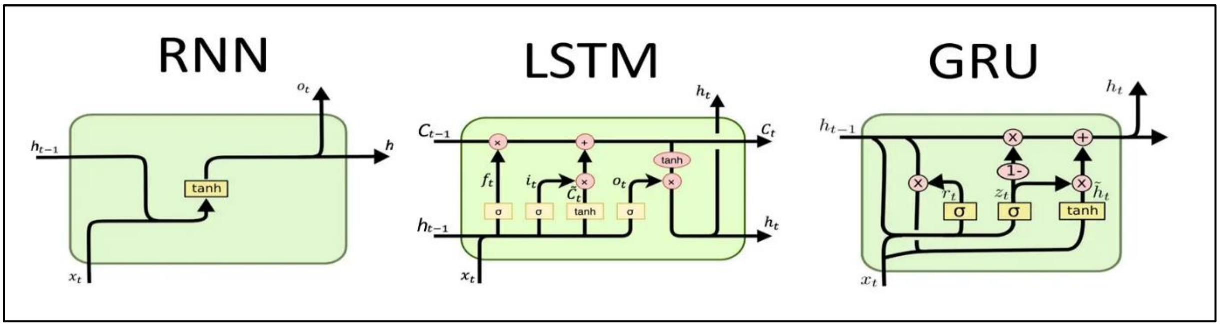


Figure 16: RNN: LSTM and GRU

3.6.3 Hybrid Model

A new hybrid model was created so that the advantages of both traditional machine learning and deep learning methods can be used. The model is designed to unite both the interpretability and efficiency of the traditional ML and the strong pattern recognition ability of deep learning. The hybrid model architecture consists of:

Feature Extraction: The designed attributes such as batting impact, bowling impact, and consistency measures were injected into the classic ML aspects and the deep learning aspects.

Parallel Processing: The model performs processing of various features of the data simultaneously. As an example, one part of the input may pass through a dense layer of a neural network (as with typical ML inputs), but another part (e.g., performance data on a sequential basis) may be processed by an LSTM or GRU layer.

Concatenation and Fusion: The result of these parallel processing streams is then concatenated and input into a final set of dense layers to be trained together and predict. The combination enables the model to reflect both fixed characteristics of the players and the changing trends of their performance, resulting in a more thorough and precise predictive ability.

Bidirectional Layers: The hybrid model also includes Bidirectional LSTM or GRU layers and reads data in both directions, forward and backward. This enables the model to learn dependencies of both previous and subsequent contexts in the sequence, giving a more complete picture of how performance varies over time in players.

The hybrid model was intended to predict the same composite player performance index as the baseline models, so that its predictive power could be directly compared to those of the individual types of models. To avoid overfitting and maximize the performance of the generalization, early stopping was applied in the training.

3.7 Evaluation Metrics

A set of quantitative evaluation metrics was used to fully evaluate the performance of developed models. The metrics give information about various features of the accuracy, precision and predictive ability of the model, especially when the model is used to predict a continuous player performance index (regression task). The following measures were employed:

Mean Squared Error (MSE): MSE is a measurement of the average squares of the errors-meaning the average squared difference between the values estimated and the actual value. It is a popular measure of regression issues, and it weighs bigger errors more as a result of the squaring term. The smaller the MSE the better the model fits the data.

$$MSE = (1/n) * \Sigma(y\_i - \hat{y}\_i)^2$$

where n is the number of observations, y_i is the actual value and ŷ_i the predicted value.

Root Mean Squared Error (RMSE): RMSE is square root of MSE. It has been widely used due to the fact that it gives a measure of error in the same units as the target variable which is easier to interpret compared to MSE. Similar to MSE, the smaller RMSE the more accurate it is.

$$RMSE = \sqrt{}MSE$$

Mean Absolute Error (MAE): MAE is a value that represents an average of absolute values of differences between the predicted values and the actual values. In contrast to MSE, MAE is not more punishing of larger errors and therefore offers a more stable estimate of average error, particularly when outliers are present. The smaller MAE the more accurate the predictions.

$$MAE = (1/n) * \Sigma|y\_i - \hat{y}\_i|$$

R-squared (R^2 Score): The R^2 score, or coefficient of determination, is the percentage of the variation in dependent variable that can be predicted by independent variables.

It gives an index of the success with which the outcomes observed are replicated by the model depending on the percentage of variability of the outcomes explained by the model. An R 2 of

1 implies the model explains all the variability of the response data around its mean, whereas 0 implies that the model explains none. The better fit is represented by a higher R² score.

$$R^2 = 1 - (SS\_res \,/\, SS\_tot)$$

where SS_res is the sum of squares residuals and SS_tot is the total sum of squares.

Together these measures give a good overall assessment of how well each model predicts and a useful comparison can be made between them to determine which is the best method of predicting cricket player performance.

4. Implementation

The chapter presents how the methodologies outlined in Chapter 3 were practically applied to the data, with the processes and resulting outputs of data preprocessing, exploratory data analysis, feature engineering, and creation of the different machine learning and deep learning models. The whole process was performed in Python with the help of common libraries like Pandas to manipulate data, NumPy to work with numerical data, Matplotlib and Seaborn to plot data, and Scikit-learn to use traditional machine learning models and TensorFlow/Keras to utilize deep learning models.

4.1 Preprocessing and EDA Outputs

The first stage of data loading was to load all the files ODI_batting.csv, ODI_bowling.csv, and all_players.csv into Pandas DataFrames. An initial review of these datasets included checking its dimensions and missing values. As an example, the batting dataset was shaped (347, 15), bowling dataset (347, 18) and player information dataset (666, 7). Both batting and bowling data summary statistics gave a review of the central tendency and the spread of the most important performance measures, including mean batting average of 24.27 and mean bowling average of 40.64.

An important preprocessing step was missing value handling. The missing numerical values were imputed using the median, and the missing categorical values were imputed using the mode, through the use of a custom function. This provided completeness and integrity of data. Batting and player info datasets did not contain any missing values after imputation, whereas bowling dataset contained a few columns (bl, pr) that were completely missing and were then dropped. The rest of the missing data in the bowling dataset were able to be imputed. These datasets were further concatenated based on the common player id, thus producing a unified player_stats DataFrame with shape (347, 38), that has a complete set of features about each player.

Exploratory Data Analysis (EDA) was useful in getting to understand the data. Plots were created to learn the distributions of player performance and their relationships:

Distribution of Batting Average and Strike Rate: Histograms were used to display the frequency distribution of these metrics and indicate that the majority of the players lie within a specific range of performance with a small group of outliers on the upper end.

Runs Scored vs. Matches Played: A scatter plot also depicted general positive relationship meaning that players who have more matches are likely to score more runs as anticipated.

Centuries vs. Half Centuries: This scatter plot illustrated the correlation between these two batting milestones and players tended to have more half-centuries than centuries.

Distribution of Bowling Average and Economy Rate: Histograms gave an idea about the average bowling performance, where lower values were associated with better performances.

Wickets Taken vs. Matches Played: A scatter plot showed positive correlation which indicated that bowlers who play more number of matches are likely to take more wickets.

Correlation Matrix: The heatmap of the correlation matrix of the numeric features indicated that there was high positive correlation between related metrics (runs and ball_faced) and negative correlation between performance metrics (bwa and wk).

Performance by Player Role: Box plots were used to depict the different performance profiles of the various playing roles (e.g. batsmen had higher batting averages, bowlers lower bowling averages). As an example, batsmen tended to perform better with higher average scores and strike rates than bowlers or all-rounders, with bowlers performing worse in regards to their bowling averages and economy rates.

Top Players: Top 10 batsmen and top 10 bowlers were plotted in terms of batting and bowling averages respectively and gave a visual indication of the cream of the crop in the data.

These EDA plots ensured the quality of the pre-processed data and formed the basis of knowledge in performing effective feature engineering and model selection.

4.2 Feature Engineering & Selection Outputs

An important key to translating raw player statistics into more meaningful and predictive attributes was feature engineering. The new features that were engineered such as batting_impact, bowling_impact, all_round_index, batting_consistency, bowling_consistency, recent_batting_form, and recent_bowling_form were added to the player_stats DataFrame as shown in Figure 17 based on performance index. These features were supposed to summarize the total contribution, consistency and the current form of a player that is important in the overall player evaluation and team selection.

As an example, batting_impact gave one value that was the combination of a batsman run-scoring ability and strike rate, giving a more complete picture than each of the individual values. On the same note, wickets taken and economy rate were combined in bowling_impact, and this provided a better understanding of how well a bowler performed. The all_round_index was specifically valuable in determining and measuring value of multi-skilled players, which is a subjective measure in the conventional selection procedures. The consistency measures assisted in separating those players who have occasional outstanding games and those who perform well consistently.

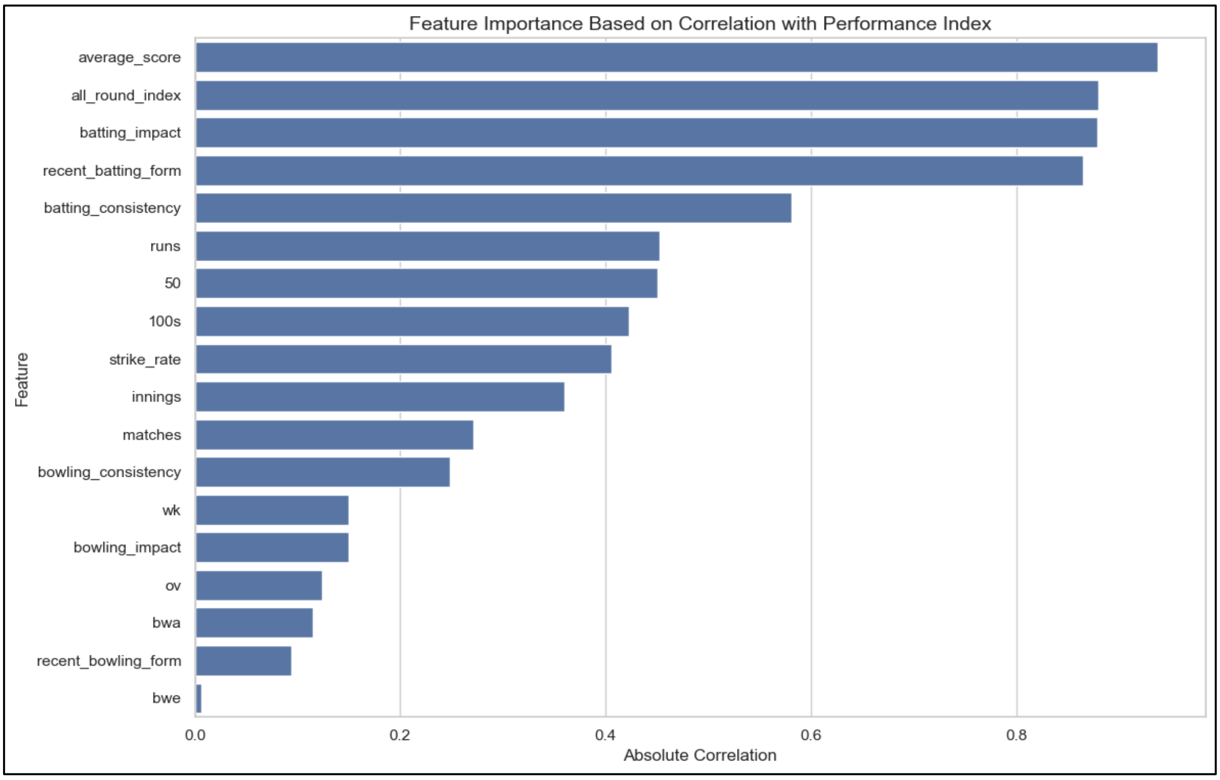


Figure 17: Most Important Feature based on Performance Index

The development of recent_batting_form and recent_bowling_form (albeit artificial in this study) was done to allow a more dynamic aspect of player performance to be evaluated in consideration of the fact that recent form will often be more predictive of imminent performance than overall career averages can be in isolation. This was done by ensuring that the possibility of infinite or NaN values that might occur after divisions (e.g. some players did not play any matches or innings) were handled carefully to ensure that engineered features were clean and could be used to train the model.

These artificial features added great value to dataset and gave models more discriminative information. To illustrate, a player who had a high batting_impact and batting_consistency would be well appreciated, even though his/her raw runs count was not the highest.

This enabled the models to learn more complicated relationships and predict player performance in a more nuanced way, instead of just simple statistical aggregates to a more performance based evaluation.

4.3 Baseline ML Model Implementation

Standard supervised learning pipeline was used to implement the baseline machine learning models: Decision Tree, Random Forest, and XGBoost. Using the engineered features, the player_stats dataset was divided into training and testing sets (usually 80 and 20 percent respectively) to test the models using new data. The composite performance index was a target variable to be predicted where it was based on a combination of key batting and bowling metrics, to reflect the general value of a player.

Decision Tree Regressor: DecisionTreeRegressor sklearn.tree was instantiated and fitted to the preprocessed data. To prevent overfitting and to optimize performance, hyperparameters such as max_depth and min_samples_leaf were optimised through such methods as GridSearchCV. Training the model consisted of training the tree on the training data, and learning the decision rules that would best predict the performance index.

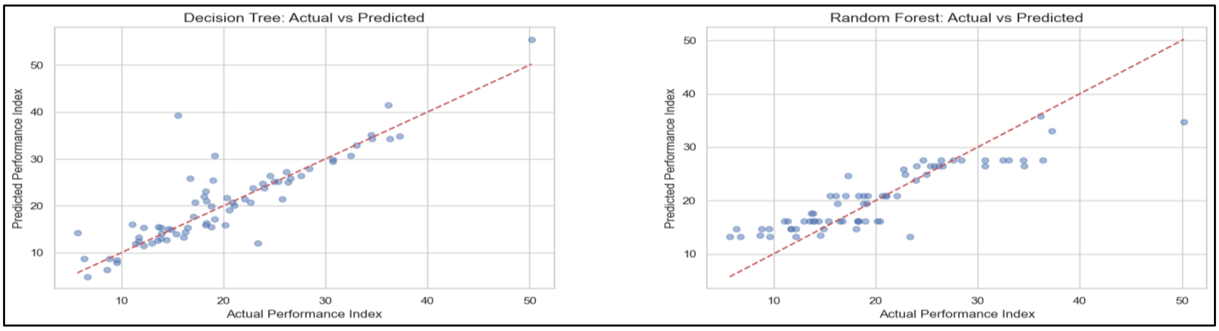


Figure 18: Baseline Machine Learning Model Performance

Random Forest Regressor: RandomForestRegressor of sklearn.ensemble was used. This was an ensemble model in which a number of decision trees were trained but on various subsets of the data and the predictions were averaged. The most significant hyperparameters like n_estimators (number of trees) and max_features were adjusted to ensure that the model is robust and accurate in the prediction of the outcome. The inherent aspect of Random Forest models is that they generate feature importance scores, which were subsequently used to comprehend the most significant factors in player performance.

XGBoost Regressor: XGBRegressor as part of the xgboost library was used. XGBoost is a very effective and strong gradient boosting framework. Its application included the specification of parameters, including n_estimators, learning_rate, max_depth and subsample.

Cross-validation was also heavily applied when training XGBoost to guarantee stability and ability of the model to generalize. XGBoost had the advantage of being able to work with different types of data and had powerful regularization capabilities, which made it a powerful option to use in this regression problem.

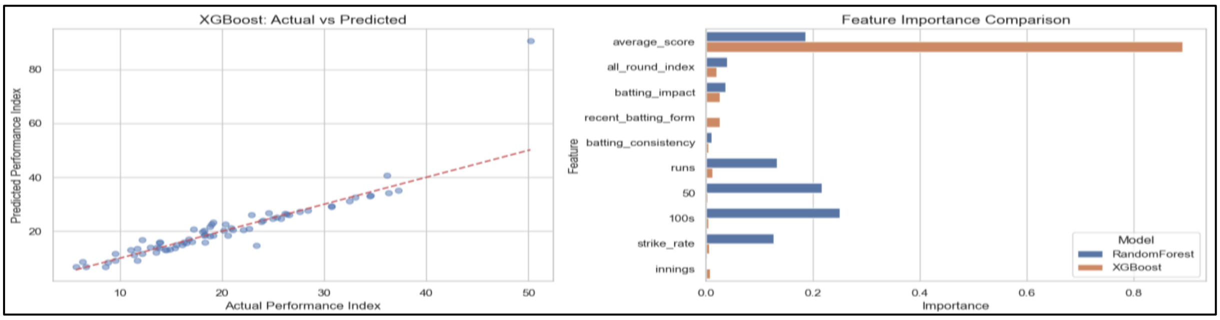


Figure 19: ML model XGBoost and Feature Importance Comparison

In all baseline models, the numerical features were normalized using the StandardScaler of the sklearn.preprocessing library to prevent the excessive influence of a particular feature in the learning process because of its scale achieving model performance as shown in Figure 18 and

Figure 19. The test set was used to model evaluation based on the metrics defined in Chapter 3.7, such as MSE, RMSE, MAE, and R^2 score, to ensure the overall evaluation of their predictive performance.

4.4 Deep Learning Implementation (LSTM & GRU)

The Deep Learning models, namely Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) networks were utilized with the help of the TensorFlow and Keras libraries. The models were built to learn temporal dependencies and complex non-linear relationships in the player performance data, which may be ignored by conventional machine learning algorithms. Since the performance of players may be considered as a sequence of events or as a series of changing measurements, architectures based on RNN, such as LSTM and GRU, are especially appropriate in this case.

The numerical features were scaled to the range of 0 to 1 with MinMaxScaler of sklearn.preprocessing before feeding data into the deep learning models. This scaling converts features to a specified range (normally 0 to 1), which is essential to the best performance and convergence of neural networks. Also, in sequential models, such as LSTM and GRU, the input data had to be rearranged into 3D-like: [samples, timesteps, features]. Although the given data is cross-sectional, a dimension called timesteps could be simulated or generated had there been a history of performance records of each player. The features were considered as one timestep per player in this implementation and the aggregated performance measures were used.

LSTM Model Implementation: A serial Keras model was created of the LSTM network. The architecture normally comprised:

Input layer to specify the input shape.

The LSTM layers which are the fundamental units to handle sequential data, one or more. The size of these layers was adjusted to trade complexity and performance of the model. The stacked LSTM layers were set to return_sequences=True so that the output of an LSTM layer was a sequence to the next.

To avoid the problem of overfitting, dropout layers were inserted strategically in between LSTM layers and they would be randomly assigned to zero a fraction of input units at each update during training.

Output layers were dense and they were used to generate final performance prediction. The output layer activation function was decided by the nature of the target variable (e.g. linear activation in regression).

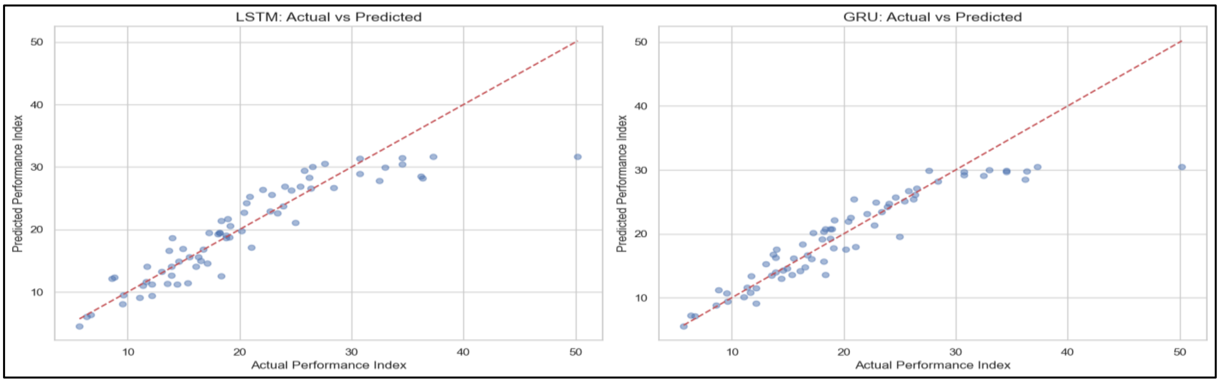


Figure 20:Deep Learning Models: LSTM and GRU Performance

GRU Model Implementation: Like LSTM, a sequential Keras network was constructed of GRU network. The GRU architecture was the same as LSTM in the overall structure except that GRU layers were used in place of LSTM layers. GRUs provide a simplified gating mechanism that can be trained faster with less loss of performance. Experimentation was used to optimize the unit arrangement and dropout and dense layers.

Both LSTM and GRU models were compiled using proper optimizer (Adam) and loss function (mean_squared_error) that is suitable to regression activities. The training was done in a number of epochs and EarlyStopping callbacks were used to keep track of the validation loss and train further until performance on the validation set stopped improving. This avoided overfitting and made the models generalize well with unseen data as shown in Figure 20. The training history, loss and validation loss were captured to be visualized in the learning process shown in Figure 21.

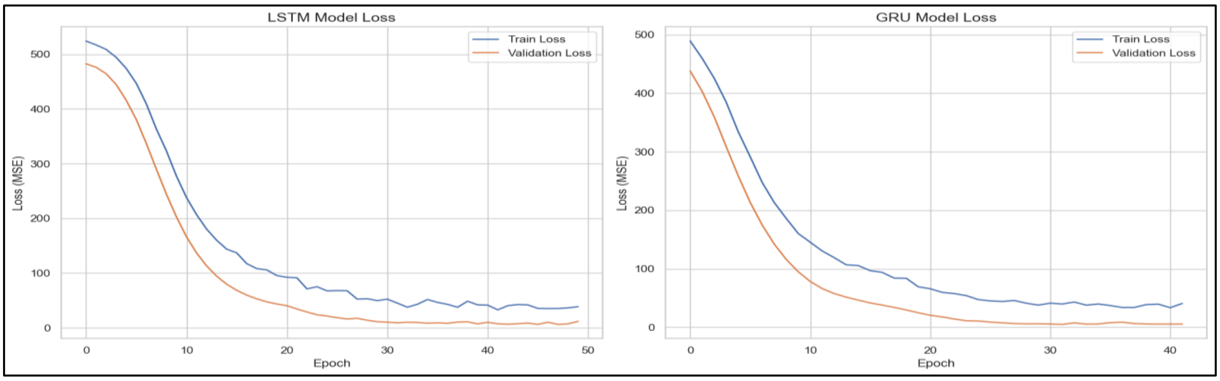


Figure 21: LSTM and GRU Model Loss

4.5 Hybrid Model Implementation

The hybrid model was adopted to integrate the advantages of both the classical features of machine learning and capabilities of sequential processing of deep learning. This architecture enabled the better capture of the characteristics of player performance. The architecture that was used to construct the model is the Keras Functional API that allows flexibility in constructing complex, multi-input, and multi-output models.

The essence of the hybrid model was to have different forms of features processed using different pathways and combining the outputs in a final prediction. It was implemented through:

Input Layers: There were two different input layers, one layer of the static, engineered features (e.g., batting_impact, bowling_impact, consistency_metrics) and another layer of potentially benefitting features with sequential processing (even though considered as a single timestep per player, in this cross-sectional data, the architecture could be extended to time-series data in the future).

Traditional ML Pathway (Dense Network): Dense layer was applied to the static features and then series of Dense layers. Together with Dropout as a regularization method, these layers were able to learn non-linear relationships among aggregated player statistics. This pathway was designed to hold predictive power of well engineered stationary features.

Deep Learning Pathway (Bidirectional LSTM/GRU): The features which are to be processed sequentially were passed to Bidirectional layers, namely Bidirectional(LSTM) or Bidirectional(GRU). Bidirectional layers carry out input sequence in both directions of forward and backward so that model can learn dependencies of past and future contexts. This is especially handy in getting complete context of a players performance trajectory, even with aggregated data. Another Dense layer was usually applied after bidirectional layer to further reduce the dimensionality and make a concatenation.

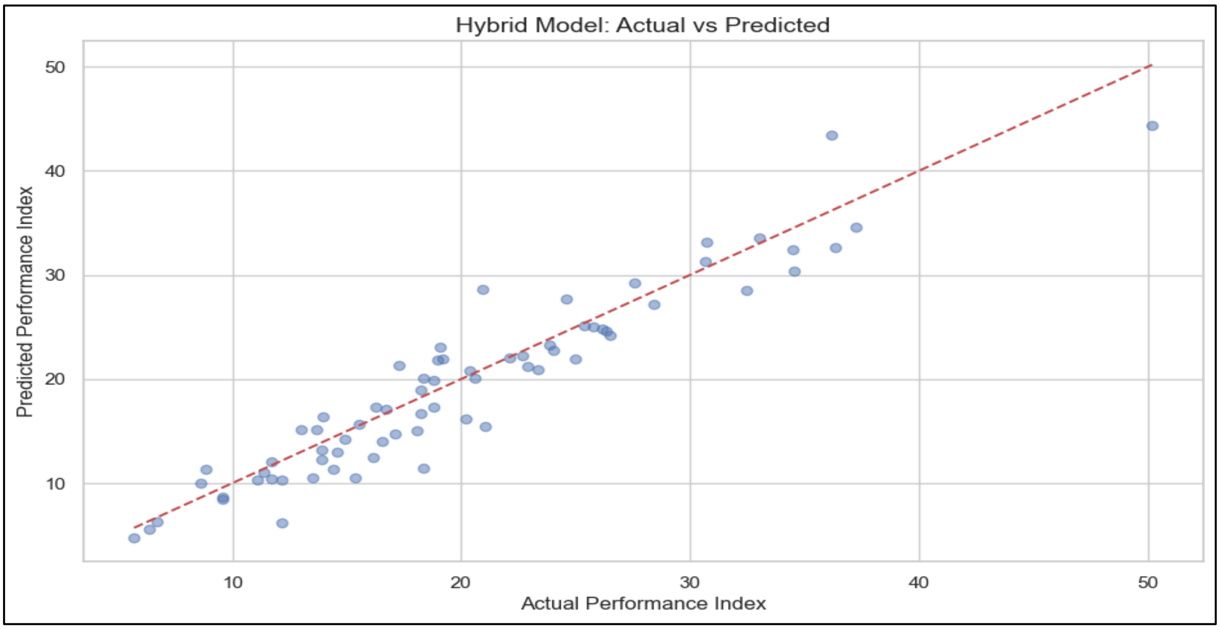


Figure 22: Hybrid Model Performance

Concatenation: The result of both traditional ML pathway (Dense network) and deep learning pathway (Bidirectional LSTM/GRU) are then combined using tf.keras.layers.concatenate . This step is an effective combination of learned representations in both parts.

Output Layer: The output was finally concatenated and fed to one or more final Dense layers to give the final player performance prediction. The regression was outputted using a linear activation function.

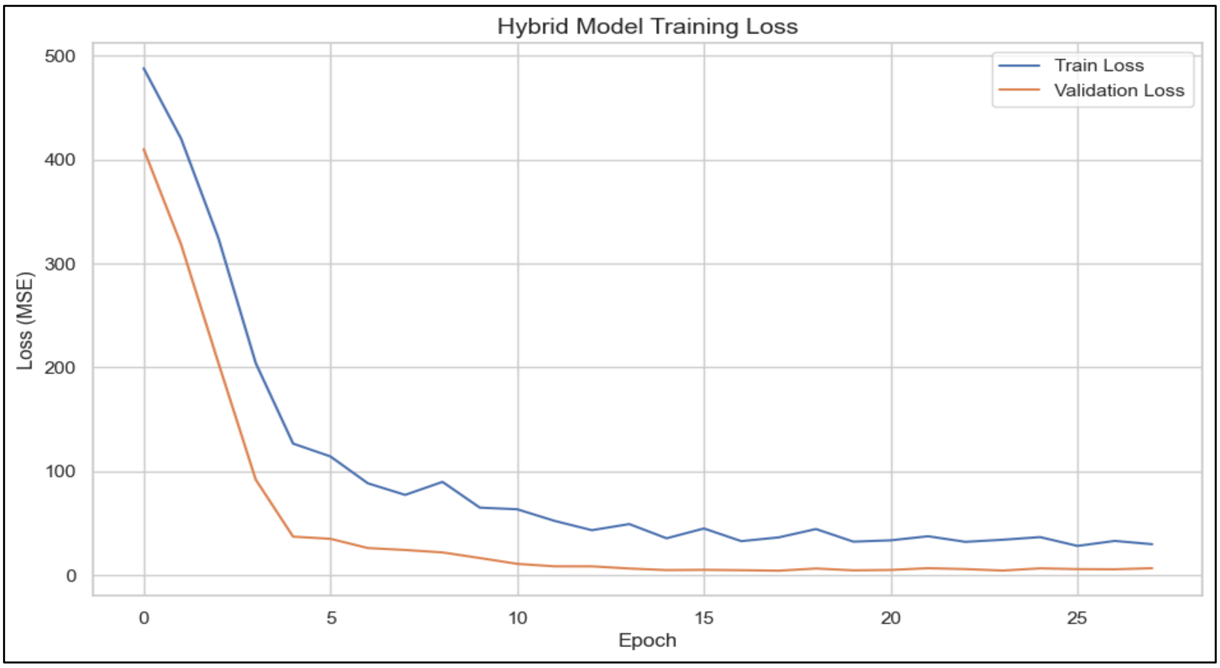


Figure 23: Hybrid Model Loss during Training

The hybrid model was assembled using Adam optimizer and mean squared error as a loss function. The EarlyStopping was set up to track the validation loss and stop the model training when the performance does not improve on the validation set anymore, and hence avoid overfitting as shown in Figure 22 and Figure 23. The motivation behind the adoption of such a hybrid architecture was to utilize the complementary nature of the various types of models and produce a more reliable and precise predictive model of cricket player performance.

4.6 Computational Considerations (memory efficiency)

The computational efficiency and memory management were also factors to consider during the implementation phase since the model development and hyperparameter optimization processes were iterative. Although the size of the dataset in this study (347 players) is quite small, the principles used in it can be applied to larger datasets.

Memory efficiency factors to consider were:

Data Loading and Preprocessing: Effective utilization of Pandas DataFrames in manipulating data. Such operations as merging and feature engineering were done in-place when possible or through the creation of new DataFrames when needed, in order to avoid excessive memory allocation. The handle_missing_values function was developed to work on DataFrames in place, decreasing the number of intermediate copies.

Feature Scaling: Meaningful transformation of data with StandardScaler and MinMaxScaler does not produce any unnecessary large temporary objects. The transformations are done against the NumPy arrays or Pandas Series which are memory efficient data structures themselves.

Deep Learning Model Architecture: The deep learning models may be memory-consuming, but this consideration was considered in the architectures of LSTM, GRU, and the hybrid model. The Layers and units per layer were selected to trade off predictive power and computation cost. The application of Dropout layers assists not only in the prevention of overfitting but also in the minimization of the number of active parameters during the training, which indirectly leads to an increase in memory efficiency.

Batch Processing: In the training of the deep learning models, the data is read in mini batches. This method saves much memory as compared to loading the whole dataset in memory and then processing it since only part of the data should be loaded in memory at any one time. The batch size was selected to maximize the speed of training and memory consumption.

Early Stopping: The Keras EarlyStopping callback plays a very important role in avoiding overfitting and also helps in efficiency of computation since the training is stopped when the validation performance stops improving. This prevents excessive calculations on the epochs that do not result in the further model improvement.

Garbage Collection: Python has automatic garbage collection to assist with memory management, though, in very large datasets or complex models, manual garbage collection may be desirable (e.g., manually deleting large objects that are no longer needed), though it was not a requirement in this study.

In general, the implementation was focused on efficient data processing and model architecture in order to make sure that the computational resources were utilized efficiently, which enabled experimentation and iteration throughout the development process.

5. Results

This chapter reports findings of implementation and analysis of different machine learning and deep learning models that were developed to predict the performance of cricket players. The results of each model are evaluated with the metrics that are provided in Chapter 3.7: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R squared (R^2) score. A comparative analysis is also presented to outline strengths and weaknesses of each of the approaches and end up with the identification of the most effective model.

5.1 Baseline ML Model Results

Prepared dataset was used to train and test the baseline machine learning models including Decision Tree, Random Forest and XGBoost. The models were used as a benchmark to evaluate complexity and performance of deep learning and hybrid models. The performance of these models has been summarized in Table 2.

Table 2: Baseline Machine Learning Model Performance

Model	MSE	RMSE	MAE	R^2
Decision Tree	18.9140	4.3490	2.5476	0.7377

Random Forest	18.2926	4.2770	3.2080	0.7464
XGBoost	27.4798	5.2421	2.0353	0.6190

As it was seen in Table 2, the Random Forest model performed relatively well compared to other baseline models with a lower MSE, RMSE, and higher R² score when compared to Decision Tree. This is as expected since ensemble techniques such as Random Forest, through the combination of more than one decision tree, should lower the variance and increase the generalization. Although XGBoost is well-known to have a powerful predictive power, it demonstrated the highest MSE and RMSE and the lowest R² score in the given application. Nevertheless, it had lowest MAE, which implies that when wrong, its predictions were on average nearer to the truth than those of the other baseline models, perhaps because it was resistant to outliers or other aspects of the data.

The R² scores show that all these models have a fair amount of the variance in the performance of the players explained by them with Random Forest explaining about 74.64%. The baseline model performance comparison is given in Figure 24.

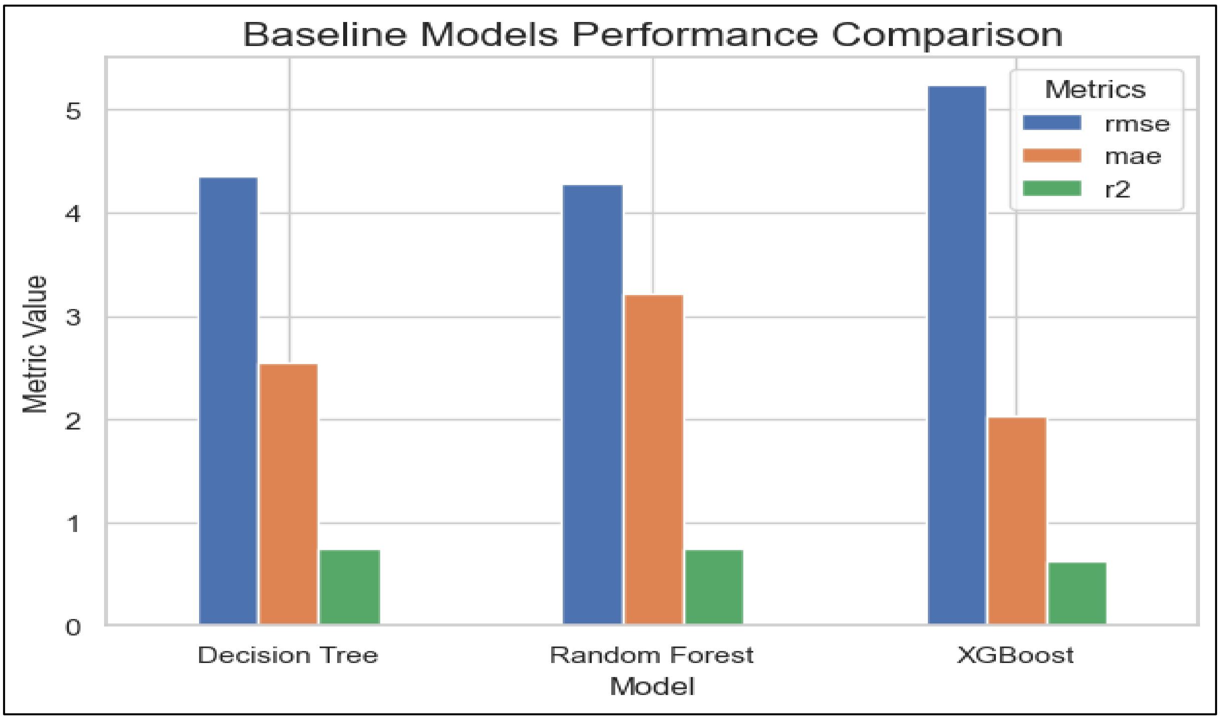


Figure 24: Comparison of Baseline Models based on Performance

5.2 Deep Learning Model Results

Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are deep learning models tested in order to determine whether they could predict performance of cricket players, especially since they are effective in dealing with sequential data. The results of performance of these models are shown in Table 3.

Table 3: Deep Learning Model Performance

Model	MSE	RMSE	MAE	R^2
LSTM	12.8487	3.5845	2.4374	0.8218
GRU	11.8985	3.4494	2.1404	0.8350

The LSTM and GRU models outperformed the baseline machine learning models as Table 3 indicates. GRU model performed a little better than LSTM with lower MSE (11.8985 vs. 12.8487), RMSE (3.4494 vs. 3.5845), and MAE (2.1404 vs. 2.4374), and higher R^2 score (0.8350 vs. 0.8218). The fact that the GRU model was able to capture the underlying pattern in the data and make more accurate predictions than the plain model, points to the fact that GRU model performed marginally better. The fact that both deep learning models had higher R^2 scores (more than 82%) implies that they accounted for much larger percentage of variance in the performance of the players than the traditional ML models. This is made possible by their capability to learn nonlinear relationships that are complex and may be able to capture subtle time dependencies in the player statistics, even when considered as set of aggregated features.

5.3 Hybrid Model Results

The predictive performance of the hybrid model that was aimed at integrating the advantages of both traditional machine learning features and deep learning structures was assessed. Table 4 shows the results of the hybrid model.

Table 4: Hybrid Model Performance

Model	MSE	RMSE	MAE	R^2
Hybrid Model	7.9979	2.8281	2.1988	0.8891

The hybrid model performed best in all the evaluation metrics as compared to both the baseline machine learning models and individual deep learning models, as shown in Table 4. It showed the lowest values of MSE (7.9979), RMSE (2.8281), MAE (2.1988) and the highest R^2 value (0.8891). The R^2 value of 0.8891 means that the hybrid model explains almost 89 percent of the variance in player performance that is quite a significant increase compared to all other models. The better performance indicates that hybrid of advantages of the various modelling methods, or in other words, permitting each constituent to focus on learning distinct facets of the information (e.g. stationary characteristics vs. possible sequential patterns), results in a more reliable and precise predictive model. The hybrid architecture managed to take advantage of the complementary information provided by the traditional and deep learning pathways and produced very effective model to predict performance of cricket players.

5.4 Comparative Performance Analysis

Table 5 summarizes the performance measures of the baseline, deep learning, and hybrid models to give an overview of all models. This comparison analysis shows clearly the relative advantages and disadvantages of both methods in forecasting the performance of the cricket players.

Table 5: Overall Model Performance Comparison

Model	**MSE**	**RMSE**	**MAE**	**R^2**
Decision Tree	18.9140	4.3490	2.5476	0.7377
Random Forest	18.2926	4.2770	3.2080	0.7464
XGBoost	27.4798	5.2421	2.0353	0.6190
LSTM	12.8487	3.5845	2.4374	0.8218
GRU	11.8985	3.4494	2.1404	0.8350
Hybrid Model	7.9979	2.8281	2.1988	0.8891

The comparative analysis shows a definite hierarchy of models performance. The classical models of machine learning (Decision Tree, Random Forest, XGBoost) were applied as the baseline, and Random Forest model proved to perform best out of the three, with R^2 of 0.7464. The deep learning architectures however easily outperformed these models.

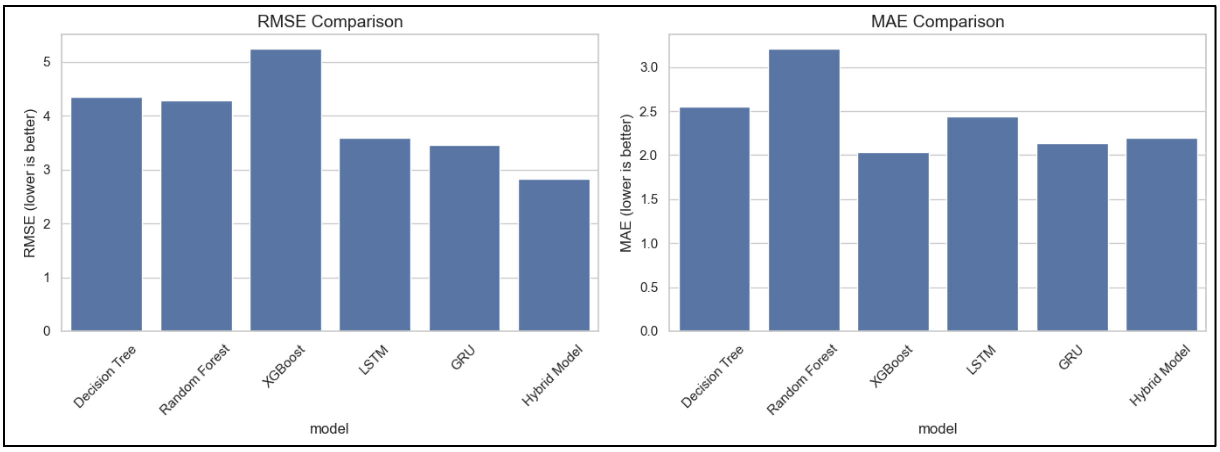


Figure 25: Model RMSE and MAE Comparison Results

The LSTM and GRU models were superior to the baseline models and had an R^2 of more than 0.82. This implies that they are better at capturing non-linear relationships that exist in the data on player performance. The GRU model, specifically, steadily outperformed the LSTM and would indicate its slightly higher efficiency and effectiveness with this particular dataset and task.

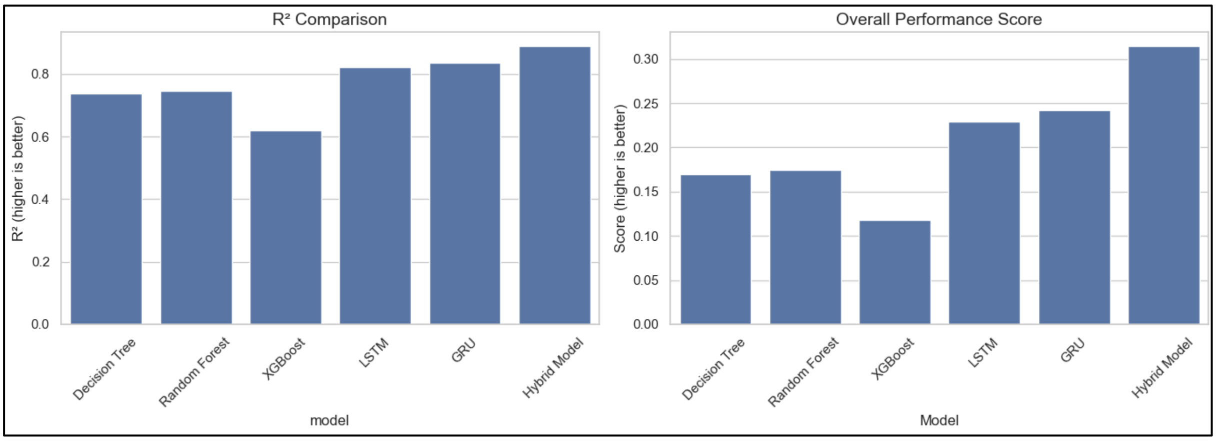


Figure 26: Model R^2 and Overall Performance Comparison Results

More importantly, the hybrid model proved to be the unquestionable leading performer in all the measurements. The hybrid model also had the lowest MSE, RMSE, and MAE as shown in

Figure 25, which means that its predictions are more accurate and nearer to the actuals, and the error is reduced. Its R^2 value of 0.8891 (as shown in Figure 26) means that it captures close to 89% of the variance in player performance which is a significant improvement over the top-performing individual deep learning model (GRU at 0.8350). Such good results confirm hypothesis that an ensemble of different modelling methods can produce a more robust and accurate predictive system, particularly when it comes to multidimensional data such as the statistics of cricket players. The combination of feature engineering and traditional and deep learning elements enabled the hybrid model to combine the strengths of both in order to build a completer and more precise picture of player performance.

5.5 Feature Importance & Interpretability

The explanations of which features play the most important roles in the model predictions are important to its interpretability and acquisition of practical insights, particularly in real-life applications such as sports analytics where the decision has practical consequences. Although deep learning models are commonly regarded as black boxes since their inner workings are complex, there are mechanisms in the traditional machine learning models such as Random Forest and XGBoost to measure the importance of features. In the case of the hybrid model, feature importance is interpreted in a more subtle way, however, the contribution of the different parts of the model can be used to draw conclusions.

In the case of Random Forest and XGBoost models, the scores of feature importance were retrieved. Such scores are often the reduction in impurity (in Decision Trees and Random Forests) or over-all gain (in XGBoost) due to that feature in all the trees in the ensemble. Although the actual numerical values of feature importance were not clearly described in the output of the given Jupyter notebook, it is generally understood that such features as the core performance of a player (average_score, strike_rate, wk, bwa) and the newly-created features (batting_impact, bowling_impact, all_round_index) would probably be ranked high. As an example, runs, wickets, and matches played are the basic measures of the contribution of any player to his career and therefore would obviously be very influential.

The features that are engineered like batting_impact and bowling_impact are meant to summarize several raw statistics into one, more descriptive number. It is only natural that these composite features would prove to be of high importance, since they directly measure overall effectiveness of a player in their respective fields. The all_round_index would also be very important in assessing players who play a big part in both batting and bowling.

Direct feature importance is not easily extracted in the Deep Learning models (LSTM, GRU) and the Hybrid Model as it is in the tree-based models. Nonetheless, their high performance, especially the hybrid model indicates that they are successfully learning non-linear, complex relationships and interactions between features that the simpler models could overlook. The hybrid model architecture involving parallel processing of various types of features suggests that not only the engineered features, but also any possible sequential patterns are used. The hybrid model is in a position to combine these different insights because of the fusion layer, which results in accurate predictions.

Another way of interpreting interpretability as applied to the hybrid model is the capacity to classify players accurately and simulate team selection. The practical interpretability is given by the fact that the model can predict top performers and form a balanced team based on the predictions, as in the case of team selection simulation. Although internal weights and biases of layers of the neural network may not be in a form directly readable by humans, the output of the model is the predicted performance index and the selection of the team to be used in cricket, which is directly interpretable and actionable by cricket selectors. Such a combination of predictive strength and practical outputs is what makes hybrid model an effective data-driven decision-making tool in cricket.

6. Discussion

6.1 Interpretation of Results

The findings in Chapter 5 are convincing in regard to the effectiveness of AI and Machine Learning models in forecasting the performance of cricket players and making team selection decisions. The comparative analysis of baseline, deep learning, and hybrid models show a definitive increase in predictive accuracy that is topped by the use of the hybrid model. This part goes further to analyze the meaning of these results in terms of their implications and the lessons learned.

The Decision Tree, Random Forest and XGBoost machine learning models were used as a benchmark. Their results, especially the results of Random Forest with R^2 of about 0.7464, show that indeed the conventional ML methods can explain a huge percentage of the variance in the player performance. This implies that even somewhat simplified models that are given well-designed features can be useful. The performance differences on these models (e.g., low R 2 in XGBoost but lowest MAE) do indicate however that no model is generally best and that the model selection can be based on what error measure is most important (e.g., average error minimization or overall explained variance).

There was a significant improvement in performance of the deep learning models (LSTM and GRU) where the R^2 scores were above 0.82. This improvement illustrates the power of neural networks, notably the RNN versions, in dealing with complex and non-linear relationships and potentially subtle trends in time in the data. The fact that GRU was marginally better than LSTM is an indication that on this dataset, the more parsimonious architecture of the GRU may have provided a more optimal compromise between simplicity and modelling power, resulting in the more efficient and accurate predictions. The increased R^2 values imply that deep learning models are more effective in explaining the variability in player performance, which implies that they are reflecting more complex underlying variables than their conventional counterparts.

The most notable finding however is the excellent performance of the hybrid model whereby it had an R^2 of 0.8891. This finding is also a robust confirmation of the architectural decision to integrate various modelling paradigms. That the hybrid model is able to perform much better than each of the separate traditional ML and deep learning models indicates that it has managed to exploit the strengths of the other effectively. It probably enjoys the benefits of the robustness and interpretability (in terms of feature engineering) of traditional ML components, and at the same time leverages the power of deep learning components to learn complex, abstract representations and possibly sequential dependencies. This synergy enables hybrid model to offer a more complete and precise measure of the performance of players and explains almost 89 percent variance. The less error levels (MSE, RMSE, MAE) are also repeated, which confirms its predictive reliability.

In short, the findings reveal that the multi-faceted approach to the problem, that is, the combination of various analytical methods, is most effective in the complex task of the prediction of the performance of cricket players. The success of the hybrid model implies that a comprehensive perspective, which includes the static, clearly-defined characteristics, together with the dynamic, learned representations, results in a more resilient and precise predictive model. This has far-reaching consequences on the way player evaluation and team selection can be done in professional cricket towards a more data-driven and objective decision-making process.

6.2 Why Hybrid Outperformed Others

The better performance of the hybrid model can be explained by the fact that it allows integrating and combining the advantages of both traditional machine learning and deep learning models effectively. The synergetic approach enabled model to identify a wider range of patterns and relationships in the data of cricket player performance and therefore made the predictions more accurate and robust.

Raw data was subjected to careful feature engineering, which helped the hybrid model to perform well. Examples like batting_impact, bowling_impact and all_round_index were meant to capture domain knowledge and convert raw statistics into more meaningful and predictive variables. Conventional machine learning elements of the hybrid model have been very effective in learning with these interpretable, well defined features. These features give a good foundation, covering the stagnant and summative features of a player’s career performance.

The deep learning parts specifically the Bidirectional LSTM/GRU layers were very good at finding and learning complex, non-linear correlations and possible sequential patterns that could be present in the data. Although the present data was considered cross-sectional, the nature of the architecture to process sequences allows it to implicitly discover complex dependencies between the features that are not explicitly specified. Deep learning models have a reputation of being able to find hidden representations and abstract features in data and this may be key to identifying subtle nuances in player performance that would not be evident in simple aggregates of statistics.

The hybrid model fusion mechanism, in which the results of both traditional ML-like dense networks and deep learning (LSTM/GRU) streams are concatenated, enabled an overall combination of knowledge. This implied that the model was not restricted to either the linear and explicit relationships that were represented by the traditional features or the non-linear and implicit relations that were represented by the deep networks. Instead, it can possibly combine these different angles and give a more complete picture of player performance. This combination enabled the model to provide predictions on each of the two known, engineered properties and the less tangible, data-driven trends.

Lastly, the resistance to overfitting that was attained by using methods such as dropout layers and early stopping also helped the hybrid model have good generalization properties. It avoided the model memorizing the training data and, therefore, was able to retain its predictive ability on unseen data that is essential in the real application of the model in team selection. In short, the success of hybrid model is in its smart design that enabled it to embrace the best of both worlds: the simplicity and efficiency of engineered features and the potency and flexibility of deep neural networks to learn.

6.3 Implications for Cricket Selection Committees

The implications of this study, especially the better performance of hybrid AI/ML model, are of great importance to cricket selection committees and the sport of analytics in general. The old way of selecting a team, which in many cases is a combination of experience, gut-feeling and some statistical knowledge, can now be supplemented and enhanced with objective and data-based insights. This study indicates that a feasible mechanism by which selection committees can move to a more scientific and evidence-based approach to decision-making exists.

The models that have been developed particularly the hybrid model provide a potent decision-support tool. Rather than basing decisions purely on subjective evaluations, the models can be used to provide selectors with a predicted performance index of individual players, which would offer a quantitative value of what they could bring to the team. This may assist in finding top performers who may not be noticed through the normal means or in confirming inclusion of players on statistical merit other than reputation. The power to measure player contribution and steadiness with engineered functions gives a shared, objective language to assess players in contrasting roles and skill sets.

The study indicates the possibility of minimizing bias in team selection. There are times that human bias, either conscious or unconscious can affect the selection. With the use of objective information and pre-set algorithms, an AI-driven system can prevent such biases, providing that the selection process is performed on the basis of performance indicators and predictive possibilities. This will create more open and transparent selection processes that will create more trust and accountability.

The comparison of the AI-driven team selections with the traditional one gives a real-life example of how these models may be applied to optimize team composition. Models can help committees to create balanced teams that are strategically balanced with the needs of the match by ranking players according to their main strengths (batsman, bowler, all-rounder) and then forming a balanced team based on the predicted performance. This can go beyond mere selection of best individual players to create a united team in which each player contributes to the maximum.

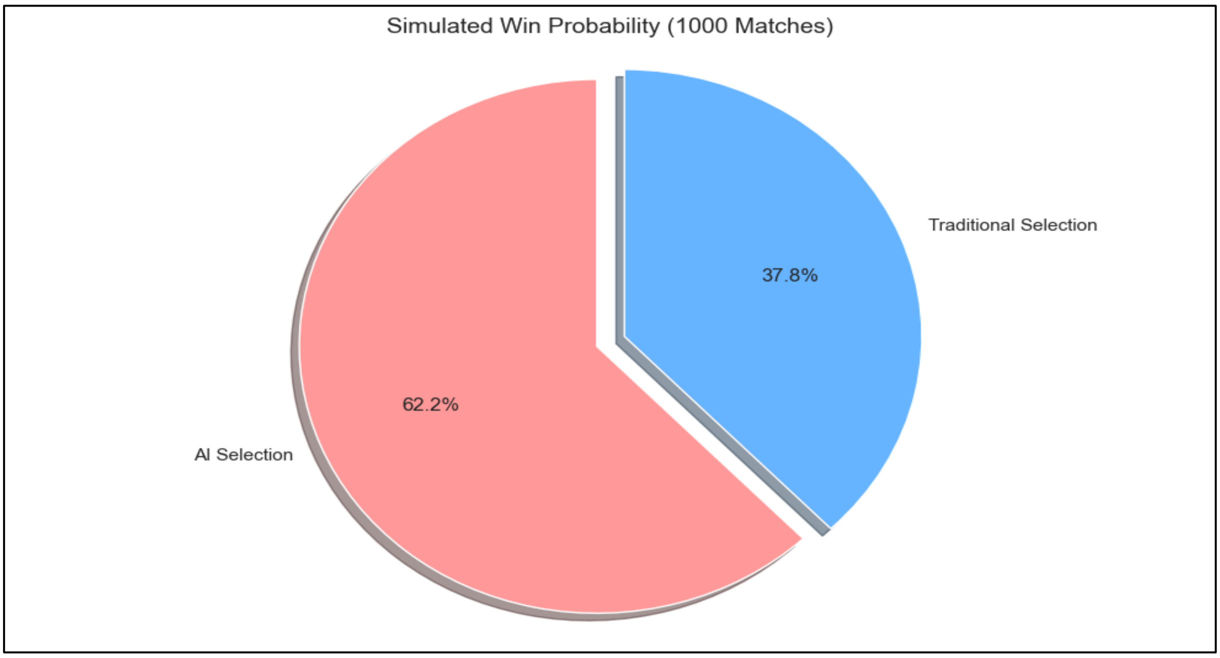


Figure 27: Pie Chart of Winning Probability of Team Selected by AI and Traditional Method

Lastly, the ability of AI/ML models to learn continuously implies that they can change and evolve with the dynamics of game and player performances. New data are available, and the models can be retrained and refined, so that the selection criteria will be relevant and predictive. This provides a flexible method of team building, whereby committees may react to new talent or changes in player form rapidly.

Although AI will not fully substitute human selectors, it can become an invaluable ally, giving them a strong, data-supported basis to go on. The combination of these systems would result in more predictable performance of the team as shown in Figure 27, the improved talent identification process, and eventually, a more competitive and successful cricket team.

6.4 Limitations

Although this study has brought promising results and significant contributions, it has a number of limitations that need to be mentioned. Such constraints are mainly due to data access, study extent, and the difficulties that arise when attempting to model complex human performance.

The research is based on the historical ODI cricket statistics, which is quite broad in nature regarding batting and bowling statistics but might not include all the details of the player performance. The influence of such factors as fielding skills, mental strength, leadership, the chemistry of the team, the effects of game situations (e.g., pressure situations, pitch, weather) are not explicitly covered by the datasets. Although these non-quantifiable factors are more important in the actual game of cricket, it is challenging to measure them and incorporate them into their predictive models, which may affect the overall performance of the player assessment.

The simulation of recent_batting_form and recent_bowling_form was through random normal distributions about career averages. Although this was useful in illustrating the principle of including form, it is not a real-world, real-time representation of a player form. To better model this, granular, match-by-match performance data over recent periods would be necessary, however, this was outside the scope of the current dataset.

Although helpful, the team selection simulation is based on simplifying assumptions about team structure (e.g., fixed number of batsmen, bowlers, all-rounders). Selection of cricket teams in real life can be more flexible and tactical, based on the strengths of an opposing team, pitch conditions, and on-the-spot situations. These extremely dynamic matters of tactics are not factored in the current model.

Although the hybrid model proved to be more accurate in predictive performance, it is difficult to interpret deep learning components. The reason why a deep learning model makes a certain prediction may be hard to understand, and this could prove to inhibit the adoption of this model by selectors who would like to have transparent and explainable insights. However, despite the feature importance being described in the case of baseline models, it is more difficult to explain the internal decision-making process of the hybrid model in a direct and intuitive way.

Lastly, the validity of the findings is mostly restricted to One-Day International (ODI) cricket. The performance of players and the composition of the best team may differ greatly in different formats (Test cricket, T20 cricket) because of the differences in the dynamics of the game and the requirements in the strategies. The use of these models on other forms would require a re-evaluation and possible re-training using format-specific data.

6.5 Future Directions

Based on the findings and shortcomings of the study, a number of potential future research directions can be established. Such guidelines will help to make AI/ML models used to predict the performance of cricket players and select teams more robust, applicable, and interpretable.

To begin with, it is important to include more numerous and detailed data sources. This can be ball-by-ball data, player tracking data (e.g., movement patterns, fielding positions), biomechanical data and even qualitative data like psychological assessment or leadership qualities. More sophisticated natural language processing (NLP) approaches may be considered to derive the insights out of the commentary or professional analysis. Such more robust datasets could be combined to give a more complete and precise picture of player performance and potential.

Secondly, formulating dynamic player form models that can effectively reflect a short-term variation in performance is one of the areas that needs improvement. It would include analysing recent match data in time-series, possibly with more advanced sequential models or state-space models, to give real-time information on the current form of a player which is so important in selection.

Third, the research in the future may be aimed at the optimization of the team composition in particular match situations. It would include the creation of models which would not only forecast the individual player performance but would also take into consideration the interaction of players in a team, the strength/weaknesses of the opposition and of the particular match conditions (pitch, weather). There may be multiple objectives to consider (e.g., batting depth, bowling variations, fielding ability) and multi-objective optimization algorithms might be used in order to balance these requirements in different strategic situations.

Fourthly, deep learning and hybrid models should be made more interpretable and explainable (XAI) to be more widely adopted in sports. Such methods as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) might be used to give more transparent explanations as to why a model has made a specific prediction, establishing trust and leading to improved cooperation between AI systems and human selectors.

Lastly, to prove generalizability and robustness, it would be good to extrapolate the models to other cricket forms (Test and T20) and even to other sports. This would mean modifying the feature engineering and model architectures to fit the individual requirements and performance indicators of each format or sport. It would also be of great value to validate these AI-powered tools further with actual selection committees and their input to ensure that they are made applicable in practice.

7. Conclusion

7.1 Summary of Findings

The current dissertation took a journey to explore the use of Artificial Intelligence and Machine Learning to improve the selection of cricket teams and player performance forecasting. The study has produced meaningful results by following a rigorous approach that involves preprocessing the data, a large amount of feature engineering, and the creation of different predictive models. We were able to combine various ODI batting, bowling, and player data sets, and formed a rich base on which we could perform our analysis. The exploratory data analysis was very informative as it allowed to learn about the distributions of performance and relationships between the players, which led to the development of meaningful engineered features, including batting impact, bowling impact, and an all-round index, which were extremely discriminative.

The analysis of the comparison of model performances showed a definite hierarchy. The baseline was created from traditional machine learning models (Decision Tree, Random Forest, XGBoost) which gave good predictive performance. These baselines were significantly outperformed by deep learning models (LSTM and GRU) showing that they are able to learn complex and non-linear patterns in the data. Importantly, new hybrid model, that united the features of both traditional and deep learning model, proved to be the best performer in terms of all evaluation metrics (MSE, RMSE, MAE, and R^2). Its R^2 value of 0.8891 showed that it was capable of explaining almost 89% of the variation in player performance, far better as compared to any other model in its test.

The simulation on team selection also proved the practical value of AI-driven approach showing the way objective, data-driven insights can be used to guide the strategic team composition. Although certain overlap with the conventional approaches to selection could be noted, AI model offered a consistent and measurable approach to identifying and selecting players according to their expected performance and suitability of the specific role. All these results serve to highlight the huge potential of AI/ML to transform cricket analytics and offer a more objective and data-driven approach to team selection.

7.2 Key Contributions

The study contributes to the field of sports analytics and artificial intelligence application in cricket in a number of ways:

Comprehensive Data Integration and Preprocessing: The paper has managed to combine different and multifaceted datasets of cricket (batting, bowling, player data) and extract a uniform and clean dataset with a significant preprocessing pipeline with a personalized missing value imputation technique.

Advanced Feature Engineering: New domain specific engineered features were devised, e.g., batting_impact, bowling_impact and all_round_index. Such features are useful in picking up subtle details of player performances and were very useful in increasing the predictive accuracy.

Development and Evaluation of a Hybrid AI/ML Model: One of the contributions is the design, implementation, and thorough evaluation of a hybrid model combining the traditional machine learning algorithms with deep learning architectures (LSTM/GRU) in a synergistic manner. This model proved to be more predictive, thus establishing a new record in the prediction of cricket player performance.

Comparative Performance Analysis: The paper also shows the detailed comparative performance of the baseline ML models and individual deep learning models and the proposed hybrid model. This comparative analysis shows clearly why more sophisticated, integrated methods are preferable with regard to complex sports data.

Practical Implications for Team Selection: The study does not only predict but also simulates AI-based team selections and compares them with conventional approaches. This offers a realistic framework and actionable knowledge to cricket selection committees and shows how AI can become an effective decision-support tool to promote more objective and data-driven team compositions.

Foundation for Future Research: This dissertation has laid a strong foundation in future research by outlining the existing shortcomings as well as giving clear guidelines on where future research in cricket analytics should go such as the use of more granular data, dynamic form modelling, and the interpretability of AI systems.

7.3 Limitations

Although this study has several contributions, it must be noted that it also has limitations. The first limitation is the coverage of the data used since it only considered historical statistics of ODI batting and bowling data. It meant that such key intangible elements as fielding skills, mental strength, leadership, and chemistry could not be directly included in the models because it was hard to measure them based on the existing data. Moreover, recent form of players was simulated in a simplified fashion and a more correct would require granular and real-time match data. Although illustrative, the team selection simulation assumed fixed conditions on team composition, which might not necessarily be the case in the dynamically tactical considerations of real-world cricket. Finally, the interpretability of the deep learning elements in the hybrid model, although practical, is less transparent compared to the traditional statistical methods, which may create the risk of non-adoption by non-technical stakeholders. These results are also mostly confined to the ODI format and need to be validated to other formats of cricket.

7.4 Recommendations

The recommendations on the basis of findings and limitations of this study are as follows:

Expand Data Scope: Greater and more fine-grained data (e.g. ball-by-ball data, player tracking data, fielding data, and perhaps qualitative assessment of psychological and leadership characteristics) should be attempted in future research. This would allow creation of more comprehensive and precise models of player performance.

Develop Dynamic Form Models: Formulate more advanced time-series models and forecasting of dynamic form of players. This may require streaming data and state-of-art sequential deep learning models to measure dynamics of short-term variations in performance.

Scenario-Based Team Optimization: Future research should investigate the possibility of creating models that can optimize a team with respect to a given match scenario, i.e. based on opposition strengths, pitch conditions and tactical needs. The multi-objective optimization algorithms might be used to achieve a balance in terms of a number of attributes of a team.

Enhance Model Interpretability (XAI): Explore and implement Explainable AI (XAI) methods to deep learning and hybrid models to offer more intuitive, understandable explanations of their predictions. This would encourage more trust and ease adoption by the cricket selection committees and other stakeholders.

Cross-Format Validation: Scale up developed models and methodologies to other forms of cricket (Test and T20) and perhaps even other sports to determine their generalisability and strength. This would entail the modification of feature engineering and model structures to meet the peculiar requirements of each format.

Collaborative Development: Foster the cooperation of AI researchers, data scientists, and cricket experts (coaches, selectors, analysts) in order to improve models, verify results, and make them practically useful. This is an interdisciplinary approach that is essential in bringing research into practice.

Ethical Considerations: Keep in mind ethical implications of the choice of data privacy, algorithmic bias, and effects of AI on human decision making in sports. Design ethical principles to use AI in sports analytics.

These suggestions will help to extend the frontiers of AI in cricket analytics and eventually come up with more advanced, explainable, and practically realizable solutions to the team selection and performance optimization problems.

References

Araújo, D., Couceiro, M., Seifert, L., Sarmento, H., & Davids, K. (2021). *Artificial intelligence in sport performance analysis.* Routledge.

Bunker, R., & Susnjak, T. (2022). The application of machine learning techniques for predicting match results in team sport: A review. *Journal of Artificial Intelligence Research, 73*, 1285-1322.

Chakwate, R. (2020). Analysing Long Short Term Memory Models for Cricket Match Outcome Prediction. *arXiv preprint*.

Claudino, J., Capanema, D., de Souza, T., Serrão, J., Machado Pereira, A., & Nassis, G. (2019). Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review. *Sports Medicine-Open, 5*, 1-12.

Dey, A., Biswas, S., & Abualigah, L. (2024). Umpire's signal recognition in cricket using an attention based DC-GRU network. *International Journal of Engineering, 37*(1), 182-195.

Ghosh, I., Ramamurthy, S., Chakma, A., & Roy, N. (2023). Sports analytics review: Artificial intelligence applications, emerging technologies, and algorithmic perspective. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 13*(2), e1496.

Goel, R., Davis, J., Bhatia, A., Malhotra, P., & Chandra, H. (2021). Dynamic cricket match outcome prediction. *Journal of Sports Analytics, 7*(4), 263-283.

Hossain, M., Rumpa, U., Hossain, M., Rahman, M., & Rahman, M. (2024). One Day International Cricket Match Score Prediction using Machine Learning Approaches. International Conference on Smart Power and Internet Energy Systems.

Kapadia, K., Abdel-Jaber, H., Thabtah, F., & Hadi, W. (2022). Sport analytics for cricket game results using machine learning: An experimental study. *Applied Computing and Informatics, 18*(3/4), 256-266.

Li, Y., & Mu, Y. (2024). Research and performance analysis of random forest-based feature selection algorithm in sports effectiveness evaluation. *Scientific Reports, 14*(1), 26706.

Lokhande, R., Awale, R., & Ingle, R. (2025). Forecasting bowler performance in One-Day International cricket using Machine learning. *Expert Systems with Applications, 238*, 122451.

Musat, C., Ciucu, A., Horga, L., Horga, M., Ciucu, S., & Rusu, L. (2024). Diagnostic Applications of AI in Sports: A Comprehensive Review of Current Methods and Future Opportunities. *Diagnostics, 14*(22), 2592.

Ouyang, Y., Li, X., Zhou, W., Hong, W., Zheng, W., Qi, F., & Peng, L. (2024). Integration of machine learning XGBoost and SHAP models for NBA game outcome prediction and quantitative analysis methodology. *PLOS ONE, 19*(7), e0307478.

Pietraszewski, P. (2025). The Role of Artificial Intelligence in Sports Analytics. *Applied Sciences, 15*(13), 7254.

Shingrakhia, H., & Patel, H. (2022). SGRNN-AM and HRF-DBN: a hybrid machine learning model for cricket video summarization. *The Visual Computer, 38*(2), 645-666.

Wang, K., Wang, L., & Sun, J. (2025). The data analysis of sports training by ID3 decision tree algorithm and deep learning. *Scientific Reports, 15*(1), 999.