IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 4 September 2024, accepted 13 October 2024, date of publication 11 November 2024, date of current version 5 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3495528

RESEARCH ARTICLE

PlayerRank: Leveraging Learning-to-Rank AI for Player Positioning in Cricket

BILAL HASSAN[1], CLARE CLOUGH[2], YUSRA SIDDIQI[3], RAO FAIZAN ALI[4], (Member, IEEE), AND MUHAMMAD ASAD ARSHED[5]

[1]School of Computing and Digital Media, London Metropolitan University, N7 8DB London, U.K.
[2]School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS London, U.K.
[3]METICS Solutions Ltd., IG3 9JA London, U.K.
[4]School of Computing, University of Kent, CT2 7NZ Canterbury, U.K.
[5]Department of Software Engineering, University of Management and Technology, Lahore 54770, Pakistan

Corresponding author: Bilal Hassan (b.hassan@londonmet.ac.uk)

ABSTRACT Player prioritization is crucial in sports analysis, yet prioritizing based on playing position is underexplored. This paper focuses on using learning-to-rank machine learning models to select the best players for slots within a cricket team's batting order in Twenty20 International (T20I) matches. The aim is to build and train position-specific models to rank potential players for each position in the batting order. These models will use listwise ranking algorithms and an artificial neural network architecture to provide data-driven player rankings, enhancing impartiality and performance focus. Each position-specific model is trained to rank players based on their suitability for that position in the batting order, considering factors like performance metrics and specialization. The models are designed to increase impartiality and focus on player performance. The models achieve an average ordered pair accuracy of over 94%, demonstrating their effectiveness in ranking players for specific batting positions. The specialization of positions enhances the utility of the recommendations, providing a more informed approach to player selection. This study highlights the value of using machine learning models to prioritize players based on their suitability for specific batting positions in T20I matches. The models offer an im-partial and performance-focused approach, enhancing the overall quality of player selection in cricket teams.

INDEX TERMS Cricket player prediction, deep learning, prediction, ranking algorithms, learning to rank.

I. INTRODUCTION

Cricket is a sport played with a bat and ball, featuring two teams of eleven players each. The focus here is on T20I matches, a format where each team has 20 overs to bat. In T20I matches, teams aim to field to take ten wickets while minimizing the opposition's score, and when batting, teams aim to score as many runs as possible. The batting order in cricket is crucial because not all players may get the chance to bat in a T20 match. It's not just about putting the best batsmen first; the early overs are challenging, with more movement in the ball and uncertain pitch conditions. Despite its potential impact on team selection and strategic analysis, there's limited research on optimizing the batting order using deep learning and machine learning techniques.

This paper focuses on building models for the first six positions in the batting order, as each position requires different skills and characteristics. Later-order batsmen are often specialist bowlers, so their batting statistics are less crucial for team selection. Additionally, the order of bowlers within a team is less critical in T20I matches, where they may not even get a chance to bat. Using supervised machine learning approaches, Learning to Rank (LTR) algorithms train a model to accurately determine the best arrangement for a given list of components. A scoring function that has been trained on training data converts the input list into scores that, when sorted downward, indicate the input list's model's rank [1]. Metrics, like Ordered Pair Accuracy (OPA) and Normalized Discounted Cumulative Gain (NDCG), are used to analyze

The associate editor coordinating the review of this manuscript and approving it for publication was Hai Dong.

this ordering by comparing the ranking to a given ground truth ranking.

Pointwise, pairwise, and listwise algorithms are the three primary categories of LTR algorithms. The items are ranked based on the pointwise algorithms' optimization of each element's score against the ground truth label, which approximates a conventional regression or classification issue. When using pairwise techniques, the ordering of the ground truth labels is compared to the relative ranking of each pair of items by the model. Lastly, the listwise takes into account the overall list's order, with each element's score only being significant concerning the other items.

The objective of this study is to rank cricket players for particular matches using a listwise ranking algorithm. Listwise ranking is most appropriate in this situation, where players' relative rankings are more significant than their scores since it has been demonstrated to outperform pairwise and pointwise techniques [2]. The data and methods portions of this article will include further information on the model's architecture and input data.

These algorithms are frequently employed in information retrieval tasks, which involve ranking a collection of documents according to how relevant they are to a certain question. Here, the players are the cases that need to be arranged according to how well they fit the match, and the questions are the specifics of the match. Normalized Discount-ed Cumulative Gain, the primary assessment metric, penalizes errors more severely at higher ranks, emphasizing precision in the top rankings. This is relevant to the selection of cricket players since, similar to information retrieval, only highly rated papers are usually read by users, only the best players are going to be considered for particular positions within the squad.

II. BACKGROUND STUDY

Cricket performance analysis and prediction is a broad and developing discipline. machine learning and Deep learning approaches have been used in earlier research to rate players, estimate player performance, resolve missing fixtures, predict match results, and improve on existing performance measurements. But there hasn't been much attention paid to a cricket team's batting order or the use of ranking models in this situation. Although information retrieval techniques have been used in certain sports studies to rank, Learning Rank models have not been widely used in cricket analysis.

Features that can be utilized to compare cricket players are discussed in several previous research studies. By using network analysis, the study [3] questioned the notion that a batsman's effectiveness can only be determined by his or her batting average by evaluating the effects of player interactions within a team on an individual basis. The Indian Premier League (IPL) batting rankings were predicted by [4] using newly discovered characteristics. The study emphasized the significance of the batsman's scoring rate throughout those innings as well as the total number of innings played. Shah [5]) proposed a new performance index that takes into account the opposition's performance index in each match, arguing that the opposition's calibre is overlooked in the present metrics used to evaluate batters and bowlers. Prakash and Patvardhan [6] pointed up shortcomings in the existing ranking systems, emphasizing the strain on bowlers in the game's shorter forms and the importance of aggressive batting for batsmen's evaluations. To determine the most important characteristics for winning a cricket match, Somaskandhan et al. [7] relied more on data analysis than domain expertise. These papers provide a variety of methods for examining past performance data, which is an essential component of player rating algorithms.

The selection of a cricket squad has also been predicted using machine learning. Shah et al. [8] found using principal component analysis that batting skill was more important than bowling in One Day International (ODI) match results. Based on this finding, he recommended that team selectors give all-rounders priority when it is practical. The Bangladesh national team was projected by Hossain et al. [9] to use a genetic algorithm to choose an ideal 14 players from a pool of 30, whereas Shetty et al. [10] selected possible batters and bowlers for the Indian cricket team using a random forest method. Both initiatives recognized that their strategies would disadvantage all-round players, whose batting and bowling averages would probably be lower than those of specialists. Jadhav et al. [11] claimed that a data-driven strategy reduces bias concerns and guarantees impartiality in selection by using 35 characteristics to assign a score to a player and provide the top players for a certain fixture.

There is a dearth of research devoted to forecasting the performance of individual players. Passi and Pandey [12] predicted runs and wickets in various classes using machine learning techniques including random forests, support vector machines, and naive Bayes classifiers. They discovered that the random forest was a highly precise approach and that as training data increased, so did performance. Similarly, Anik et al. [13] used machine learning algorithms to forecast Bangladeshi batsmen and bowlers' performance in One-Day Internationals (ODIs), but they acknowledged certain limitations in their findings.

In cricket, machine learning methods have been used to forecast match results. Jaylath [14] employed regression analysis and categorization to demonstrate the relevance of home advantage in international one-day matches, which vary in significance throughout nations. These methods were also used by Irvine and Kennedy [15] to determine important variables for international Twenty20 matches. They emphasized the significance of high strike rates and an aggressive style of play in general.

Although it is still in its infancy, deep learning applications to cricket analysis show great promise. Vetukuri et al. [16] used a genetic algorithm and a recurrent neural network to achieve 98.5% accuracy for team selection. Despite having a limited training set of just 83 bowlers, Saikia et al. [17] employed a neural network to predict bowling performance in the IPL, categorizing performance into three bands with

5/7 correct predictions on their test data. While Vohra and Gordon's [18] model's goals were less important, the deep learning and neural network method applies to our research. Vohra and Gordon used neural networks and Markov chains to predict the outcomes of cricket matches.

Players in other sports have also been ranked using machine learning. Using machine learning classifiers, Shahriar et al. [19] ranked Bangladeshi football players, discovering that Decision Trees were the most useful method for predicting player performance. Similar to the FIFA football player ranking system and the chess Elo Rating System, Peiris and Silva [20] updated player ranks using Bayesian statistics based on performance versus other players.

Because football is such a popular sport and statistics are readily available, a lot of study has been done on player ranking systems [21]. Football players were ranked by Ayedemir et al. [22] using unsupervised approaches, with findings highly accurately validated against football transfers and UEFA rankings. Neural networks were employed by Enyindah [23] to pick football players, and by Al-Shboul et al. [24] to predict match results based on team selection, with an accuracy rate of about 54%.

The main use of Learning to Rank algorithms, developed by Google and Microsoft researchers, is in information retrieval. Prior research by Burges et al. [25] concentrated on pairwise ranking optimization using gradient descent; however, more recent work favors listwise ranking, which improves accuracy and processing efficiency [2]. Deep learning approaches have demonstrated superior performance than gradient-boosting decision tree methods in search engines, email filtering, and online platforms such as Airbnb. These methodologies are primarily utilized in these domains [26]. Support vector machine paired learning was used by Akiyama et al. [27] to use Rank algorithms in a simulated football game, with encouraging results.

The resilience and accuracy of the TF-Ranking library used in this model are demonstrated by its use in Google's Gmail search and Drive document recommendation system [1].

Anuraj et al. [28] predict cricket match outcomes by applying sports data mining techniques, focusing on T20 International World Cup matches. Factors such as venue, toss winning, team ranking, and head-to-head records are analyzed to predict match winners, demonstrating the effectiveness of the approach using real-life data. Lin et al. [29] present a Spatiotemporal-Based Brain Pattern Recognition Network (BPR-STNet) using EEG signals for game player expertise classification, achieving an average accuracy of 82%–86.32% across five frequency bands, outperforming state-of-the-art methods by 1.56%–5.79% and demonstrating interpretability through Gradient-weighted Class Activation Mapping (Grad-CAM). Robel et al. [30] propose an ML-based approach for cricket player selection, utilizing physical fitness data, batting and bowling statistics, and other metrics to create player profiles. Three ML algorithms are employed, with support vector regression showing superior performance. The approach is evaluated using data from ESPNCricinfo and cricbuzz, demonstrating significant improvements in player selection for national and franchise league teams.

III. LITERATURE REVIEW

In the sports domain literature shows the growing interest and applications of ML for the selection of players and prediction of their performance [12]. However, ML approaches and algorithms are just applied to evaluate an individual for specific team role. For example, dash et al. implemented an ML approach to identify the cricket players for the selection of team by emphasizing their bowling and batting skills. However, most of the studies have focused on general ranking of players as compare to the identification of specific position that limits their findings in terms of an individual player role. This research presents a solution that investigate the batting position of each player by providing a state-of-the-art approach for the selection of player.

The challenge in developing a model for player ranking including the complexity of data for each player and extensive need of feature engineering for the accurate monitoring of performance metrics. Liu et al. [40] analyzed the match attributes such as team composition, opposition, and venue to increase the performance of model but position of an individual is ignored within lineup. However, this research overcome such limitations by integrating context specific features of a player for example performance of a player against specific component to ensure a thorough analysis for all batting positions.

The practical deployment of ML models for specific ranking position of players has been constrained with availability of data [41]. A lot of datasets on cricket have ignored the information for specific batting position that makes the trained data unreliable. However, in this research we have applied KNN and MICE to handle the missing values and overcome the data inconsistencies to hamper the research on sports analytics. Furthermore, this research boost the ability of model by integrating listwise ranking to rank a player accurately.

Overall, this research provides a more nuanced development of cricket with the help of unique approaches and need of each batting position. This study integrates ranking methodologies, feature engineering, and ML approaches to develop a state-of-the-art player prioritization approach.

IV. PROPOSED METHODOLOGY

Gathering data for testing, validation, and training is the first step in every deep-learning project. Features on the past performance of certain players as well as data on the toughness of opponents and cricket grounds were required for this study. Since there was no current database accessible, several web scraping algorithms were used to get the required data. The information was gathered from the International Cricket Council (ICC) website [31] and the ESPNCricinfo website [32]. Additional information was obtained from How STAT [33]. The ESPNCricinfo website provides comprehensive player performance data, categorized by position (bowling, batting, fielding), format (ODI, Test, T20, etc.), total averages, match by match, innings by innings, and other aspects like opponent or ground.

A. DATASET DESCRIPTION

The three distinct forms of data that were taken from the ESPNCricinfo website for each player are shown in **Fig 1:** player averages, innings-by-innings statistics, and T20I breakdown data. All of these are kept on separate web pages that are accessible using a distinct player ID.

- **Player IDs:** Since there is no database of player IDs, all players for the ten major cricketing nations (Australia, India, Pakistan, Bangladesh, Sri Lanka, England, South Africa, West Indies, New Zealand, and Zimbabwe) were scanned through using a Selenium WebDriver, which was used to extract the IDs from the URL links to each player.
- **Innings by Innings, Player Averages, and T20I Breakdown Information:** Using each player's unique player ID, the web scraping technique is deconstructed in **Fig. 1** below to obtain the career average statistics for each player. The Innings-by-Innings tables and T20I statistics were obtained using a similar methodology that separated the data according to opposition, position, location, and other variables. The page that was navigated to and the type of table that had to be located in each instance were the only distinctions between the three methods.
- **Ground Data:** The average runs per wicket and over for each cricket ground were retrieved from the How STAT website to create a rating for the various cricket grounds. Once the unique codes for each ground were identified, they were utilized to retrieve the statistics table for each ground. This data could then be combined into a table using BeautifulSoup and a method like the one shown in **Fig. 1**.
- **ICC Rating:** Once again, BeautifulSoup was utilized to sift through the HTML code and extract the T20I national cricket teams' ranking table. This would be used to estimate the opponent's difficulty for a forthcoming match. The algorithm employed was somewhat modified from the algorithm in **Fig. 1**, much like with the ground data.

B. DATASET Cleaning

This study has incorporated opposition bowler data and performed detailed statics economy rate, bowling average and previous performance. This data is obtained from HowStat and ESPNcricinfo databases. We have integrated these stats into our dataset by adding two columns like "Economy rate of opponent bowler" and strike rate of opponent bowler. The model is retrained with enhanced dataset to evaluate that inclusion of stats of bowler enhance the performance metrics such as NDCG and OPA. We have implemented k-Nearest Neighbors (KNN) k-Nearest Neighbors (KNN) and Multiple Imputation by Chained Equations (MICE) algorithms to replace the missing values. By implementing these techniques, existing data stats are leveraged to fill the current gaps that potentially leads to more accurate training of model. Furthermore, at some points predictive values was more complex where we have used mean and median for missing values. This approach decrease the outlier's impact and gives more accurate and natural estimated values rather than zeros. After handling the missing values, the model is retrained and compared its performance by focusing on key metrics NDCG and OPA.

- Averages: From nested lists to a single table with a row for each player, player averages were simplified. Calculations were performed on the columns by converting them to numerical datatypes. Column headings were renamed, and KNN algorithm was implemented to predict the missing values. The chart was also cleared of players who had never participated in T20I matches.
- Innings: A nested dictionary including tables for each player's bowling, batting, and fielding was downloaded that contains innings-by-innings data. The players' name-code-country served as the dictionary's key, making it simple to get the data and link it to their row in the averages table. The column names were sanitized and missing values have been predicted by implementing MICE approach.
- T20I breakdown: Like the Innings data, the T20I breakdowns' nested lists were converted into a dictionary. This dictionary contained the overall T20I averages and the summaries categorized by different criteria for each player. The study required positional groupings (batting performance at various spots in the batting order), oppositional groupings (against different teams), and match locations (nation or continent), totaling 290 distinct groupings.

C. DATASET NORMALIZATION

Subsequently, all relevant player characteristics were consolidated into a single row and standardized before integration into the model. Strike rates and batting averages for players in each of the top six batting positions were extracted from the T20I breakdown data and appended to the existing columns. The issue of batting average for players who had never been out was raised. Since this calculation would normally result in an infinite batting average (as it is calculated by dividing the number of runs by the number of times the player has been out), it could significantly impact the ranking. To address this, for these players, their average was considered to be the total number of runs scored as if they had been out only once.

The players' strike rates and averages of players against opponents in various countries were obtained using a similar methodology, and these variables were then added to the main table.

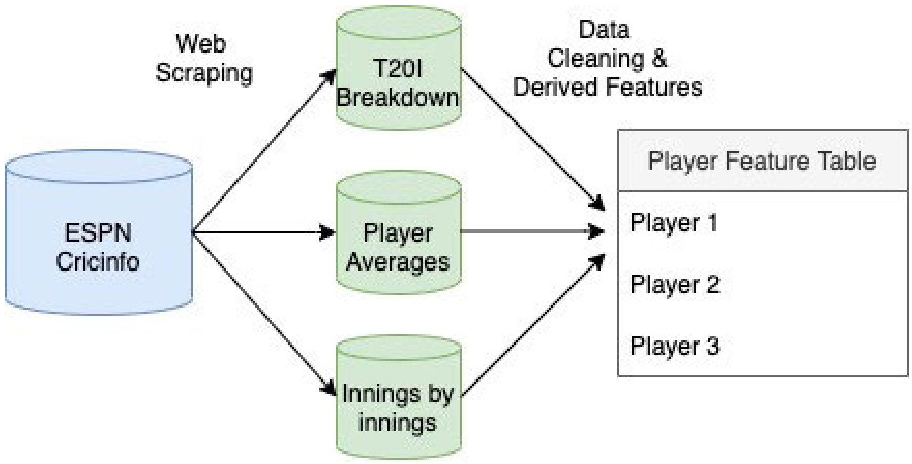


FIGURE 1. Dataset collection and cleaning process abstract diagram.

When a player did not play in a certain country or against a specific opponent, their average and strike rate were identified from selected dataset. For example, if trike rate of a player in data is [130, NA, 150, 120, 140] where NA shows the missing value, then missing value will be calculated and replaced with zero.

After gathering this extra information and fixture-specific characteristics, min-max normalization was used to normalize the data. Initially, derived "rating" features were created by dividing every attribute by the maximum, rather than using normalization. However, testing showed that performance was enhanced when all of the numerical features were subjected to min-max normalization. To determine the hardness of the pitch to bat on, the average runs per wicket and over for each ground were sorted according to the greatest scoring ratio for the contextual variables. In addition, a ranking ratio for teams was generated by dividing each team's ICC rating by the highest rating. The context characteristics in the query-document structure of the ranking models were these two statistics for a certain fixture. To input these features into the ranking model, they were transformed into Example List with Context (ELWC) TensorFlow record datatypes.

To handle the challenge of never gotten out player, we have assumed the batting average of such players with their total number of scores, considering they are out only once. In this way these players were prevented from being considering outliers with predictive averages that was skewing the results of score ranking. Moreover, the missing values have been predicted by using KNN and MICE algorithms where player has never played against other specific team. The implementation of these approaches ensure that each player is evaluated fairly by balancing the dataset.

Furthermore, experiment is executed again and eliminated the potential outliers to improve the record of players. In addition, to predict the missing values, the proposed model estimates the average of player where data is not available directly. This was lead to a high accuracy and fair ranking of player to ensure the validity of results of model.

D. FEATURES FOR PREDICTIVE MODELS

To rate every item, the ELWC format comprises a set of shared contextual (match-specific) attributes as well as several document-specific (player-specific) elements (see **Table 1**). The 100 distinct location-opposition match scenarios were created using the ten main cricketing nations, taking into account all possible combinations of that list.

For instance, playing against England in Sri Lanka and playing against Sri Lanka in India are two different scenarios. A player's performance attributes, as shown in the table below, context features indicating the fixture's complexity, and a ground-truth rating of that player within the list were required for each probable player for each of these 100 match situations. For every 100 queries, one example list including context was generated for each of the six different positions, and random 80/10/10 splits were used to retrieve training, validation, and testing instances from these lists. In each of the six models, the pairs that belong to which split are randomly selected. The unpredictability of these divides should be taken into account when evaluating the performance of different models because of the small number of cases.

1. **Player performance features:** Every player attribute listed above is included in Table 1.
2. **Contextual Features:** These are the two contextual factors unique to the opponent and venue of the forthcoming game.
   - **Location difficulty:** The average number of runs per over at every ground in the specified nation.
   - **Opposition difficulty:** Calculated as the ICC rating of the opposition divided by the highest rating.
3. **Ground-truth Label:** Among the potential players, a rating served as the ground truth label. The players were ranked in descending order based on the average and strike rate of each player for this fixture against the opponent and in the specified venue. This limits the model and the amount of training data that is accessible. First off, regardless of where in the batting order this position falls,

TABLE 1. Player features for predictive models.

FeatureName	Description
Nth position Avg	Batting average when playing at Nth position in the batting order
HS	High score in T20i
100s	Count of 100-run innings in T20I matches
50s	Count of 50-run innings
6s	Count of 6s hit in T20I matches
BF	Count of balls faced
Ct	Count of catches taken
Inns	Count of Innings played
Matches	Count of Matches Played
Average	Count of runs divided by number of dismissals
St	Count of stumpings
NO	Count of times not out
Nth position SR (Strike Rate)	Strike rate when playing at the Nth position in the batting order
SR	Count of runs divided by balls faced
4s	Total 4s shots in T20I matches
Runs	Total count of runs scored

it is assumed that the players with the greatest strike rate and average will always be the best option. Because of game-specific tactical factors, this could not always be the case in practice. Additionally, this restricted the pool of participants for each query because only players who have participated in training against that team and at that venue are eligible.

The ESPNCricinfo website has more than 6000 cricket players info from 10 countries. Just 874 of them have participated in Twenty20 international cricket matches, out of the 3270 men's international players, which reduces the pool of players available for this work. Moreover, 1 innings batted in is the typical number. This could have an impact on the statistics' dependability and the generalizability of the model. The number of players who have batted at each of the first six positions for at least one inning and were used to train the corresponding model is as follows: 1st: 163; 2nd: 192; 3rd: 237; 4th: 229; 5th: 226; 6th: 241 (see **Fig. 2**). The fact that opening batters have a more specialized function within a team and are less likely to have other players shifted to that place within the batting order may be the reason for the growth beyond the first and second spots. In the end, a smaller player pool restricts the variety of players that each model can be trained to rank, which is likely to have a detrimental impact on the model's capacity for broader generalization.

The performance of players at various positions in the batting order can be seen in **Fig. 2**. Being one of the few players in the dataset to have batted at all six positions, Virat Kohli's strike rate and average across all six positions may be compared to determine how well-suited each position is, highlighting the need for specialized models. Since all of the

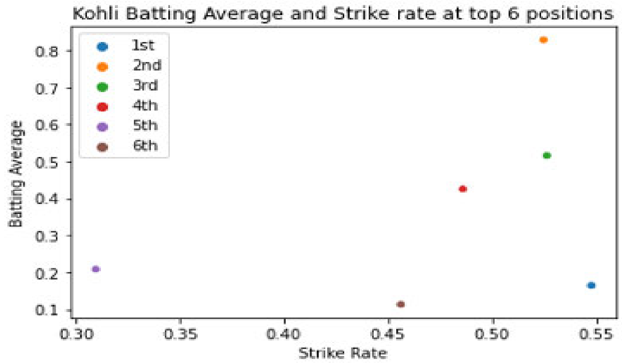


FIGURE 2. Virat Kohli's Batting Average and Strike Rate.

statistics are min-max normalized across all players for that position, his batting average at the second spot is among the best, at almost 0.8, while it is just 0.2 for the fifth.

E. LISTWISE LEARNING TO RANK

From a collection of n elements (x), the number of possible arrangements is n!. The goal of the Learning to Rank (LTR) $\pi*$, approach is to determine which order of these items is optimal for a particular topic. This rating is represented by a list of labels, indicated by the letter y, that are sorted in decreasing order. The way the model operates is that it takes into account both the contextual aspects of the query and the distinctive qualities of each item. The model uses a process called gradient descent together with a loss function to determine the relationship between these features and the best-ranking.

The scoring function adjusts its parameters as the model evolves and is responsible for assigning scores to the input attributes. The most basic type of scoring function is univariate, which assigns a single score to each item separately. The scores can then be ranked in decreasing order. The following equation (1) illustrates this. The score of each item, xi, is represented by the equation g(xi). The model's ranking, denoted by h(x), is then obtained by ordering all of the list's element scores. A multi-layer neural network serves as the scoring function in the TF-ranking library [1] however, additional machine-learning techniques may be employed in other implementations.

$$l(y, \hat{y}) = -\sum_{j=1}^{n} y_j \log\left(\frac{\exp\left(\hat{y}_j\right)}{\sum_{j=1}^{n} \exp\left(\hat{y}_j\right)}\right)$$
$$h(\mathbf{x}) = [g(x1), g(x2), \ldots g(xn.)] \tag{1}$$

Multivariate and univariate scoring functions are employed in this research. Each item is simultaneously mapped to a list of scores that is the same length as the item list [34]. This method uses the concatenated characteristics of every member in the list as the neural network's input, and the list size equals the neural network's output. This listwise method, which is more accurate, assigns scores for each item relative to one another [34]. The specifics of these scoring functions depend on the loss function used at the time of training, there are several choices for the selection of the best loss function.

F. EVALUATION METRICS

The popularity of ranking metrics reflects the demands of this work, especially the fact that appropriately ordering things with higher rankings is more crucial. The Dis-counted Cumulative Gain (DCG) (equation 2) and Normalized DCG (NDCG) (equation 3) are two examples of such metrics. By assigning a heavier penalty to higher-ranked entries that are positioned lower in the list, this strategy prioritizes the algorithm's proper rating of the top-ranked items. The DCG of the ranking at position k in the list, divided by the ideal DCG for that place if all items were ranked correctly, is represented by the NDCG at k. The values that are obtained fall into two categories: 0 (which means that none of the things are appropriately arranged) and 1 (which means that every item is correctly organized).

$$DCG(\pi, y) = \sum_{i=1}^{n} \frac{2^{y_i} - 1}{\log_2(1 + \pi(i))} \tag{2}$$

$$NDCG(\pi, y) = \frac{DCG(\pi, y)}{DCG\,(\pi^*, y)} \tag{3}$$

Ordered Pair Accuracy is another statistic used to evaluate the ranks (OPA). This metric calculates the average of pairs that are appropriately sorted relative to each other, treating the items as pairs of elements. Therefore, an OPA greater than 50% would suggest that more than half of the pairings are rated properly, which is more than what would be predicted by chance.

G. LOSS FUNCTION

The final model is greatly influenced by the choice of loss function. Gradient descent requires a differentiable loss function, even though the evaluation criteria for ranking is-sues are not. It is, therefore, necessary to use a substitute loss [1]. Many alternatives for loss functions used in the listwise ranking are available in the TF-Ranking library, and these options were assessed during the hyperparameter tuning process. As shown in equation (3) earlier. SoftMax Loss is one of the examples, which is the implementation of ListNet in the TF-Ranking Library [2]. This method compares the distribution of probabilities from the model's scores to the variation in actual labels to model the likelihood of a permutation of items using the top one probability of each item [2]:

The approximation of NDCG is another loss function that was established for list-wise LTR [35]. Due to its more direct approach to optimizing the usefulness of the ranked list, prior research has demonstrated its effectiveness in listwise LTR [36]. This method is a differentiable and direct optimization of the NDCG metric as it combines the DCG metric with the sigmoid approximation Qin et al. of an item's rank [1]. A more straightforward function called ListMLE loss [37] uses probability distributions to maximize the likelihood of detecting the ground truth permutation. The last strategy evaluated was Pairwise Logistic Loss, sometimes referred to as RankNet [25]. Its goal is to reduce the number of inaccurate orderings between pairs of items; this strategy performed the lowest, indicating that a listwise method was better suitable for this job.

H. TF-RANKING IMPLEMENTATION

TensorFlow Ranking (TF-Ranking), an intuitive implementation created by Google researchers [1], was utilized to build the model. It offers a wide range of loss functions for model testing. The architecture of the TF-ranking implementation can be seen in **Fig. 3**. The ELWC data, which contains tensors of 2-dimensional form for the context attributes and 3-dimensional individual item features, is sent into the input function.

The input features are then mapped via a transform function to TensorFlow feature columns so that the artificial neural network can process them. The scoring function then concatenates the inputs from the example and context features and feeds them into the neural network to produce the score for each item.

Based on the actual truth labels and the chosen loss function, the ranking head uses these scores to calculate the loss. The model uses backpropagation to modify the scoring function's parameters based on these losses. Stochastic gradient descent is used during training to get closer to the loss minima and, in the end, provide the ideal model parameters.

The model functions as shown in **Fig. 4** when making predictions based on the test data. The neural network receives input from the characteristics of the 15 possible players as well as contextual information that indicates the fixture complexity. A total of fifteen output scores are then produced. The characteristics of all 15 players are assembled and sent across the network concurrently, as opposed to evaluating each player separately. These 15 scores, which are shown in descending order, show how the model ranks the players on the provided list.

I. NEURAL NETWORK TRAINING

Neural networks or multilayer perceptron are the fundamental building blocks of ranking models. Each of the six models used the same architecture, but various training sets of data were used to produce unique parameter sets. After every layer, a ReLU activation layer, batch normalization, and a dropout rate of 0.9 were used to reduce overfitting. According to Ioffe and Szegedy [38], batch normalization helps neural networks converge more quickly and reduces the chance of overfitting the training set [39]. Dropout randomly removes nodes from the model's layer during training, hence reducing the chance of an over-reliance on particular characteristics. The activation layer gives the model non-linearity. Since its introduction, ReLU, an activation function, has been the default in the majority of contemporary neural network designs and has consistently produced good results [39].

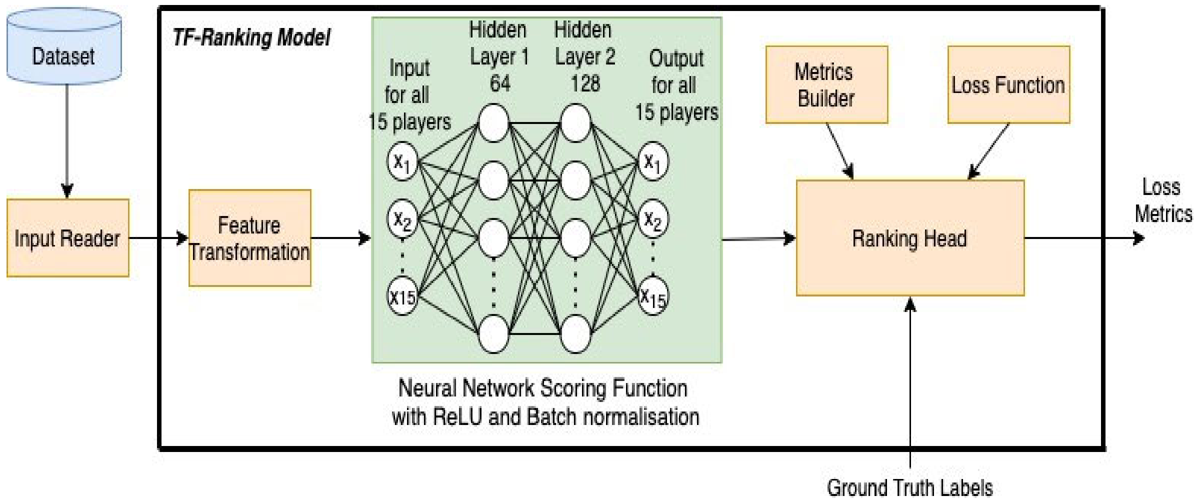


FIGURE 3. Diagram of the model's architecture.

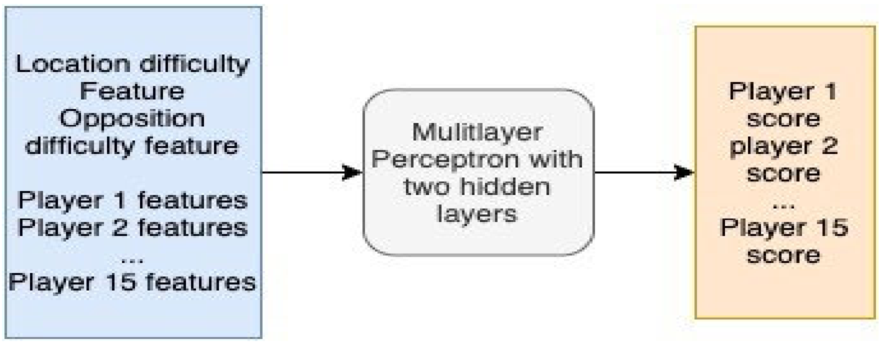


FIGURE 4. Model during predictions.

Training of each model was done using an Adagrad optimizer. There was also the use of a multivariate technique, where the Group Size was set to the number of instances per query, in this case 15. This implies that the model uses listwise ranking, instead of calculating the scores for each item in each list individually.

J. NEURAL NETWORK TRAINING

The first model was used to run a series of tests using validation data to determine the structure of the models. The best parameters were then applied to the remaining five models. Different configurations were assessed against the validation set of the original model, ranging in the number of hidden layers, nodes, learning rates, and loss functions. The models that followed were subsequently built using the best configurations from the earlier testing. Ordered Pair Accuracy (OPA) and Normalized Discounted Cumulative Gain (NDCG) were used to evaluate the models' performance at four different intervals.

1) NUMBER OF HIDDEN LAYERS

The network may capture more complexity by increasing the number of hidden layers, but doing so runs the risk of overfitting the training set. A batch size of 16 and a learning rate of 0.05 were applied to 15,000 training steps in each case as shown in **Fig. 5**. Figure 5 show the comparison of models with multiple hidden layers that specifically highlights that two hidden layer performs excellent.

2) HIDDEN LAYERS NODES

Increasing the number of nodes in each layer enables the model to capture more complexity. However, doing so also increases the risk of overfitting, much like increasing the number of hidden layers. Although the findings were less clear-cut, the results were always rated first or second. Consequently, For the model architecture, 64 and 128 hidden nodes were chosen, see **Fig. 6**. Figure 6 shows the impact of configuration of different nodes among two hidden layer structure. It presents that configuration of 64 and 128 offers best performance as compare to 32 and 64.

3) LOSS FUNCTION

With the first model, several different listwise loss functions were assessed. The results clearly showed that the most effective method was SoftMax loss, which is the ListNet implementation [2]. The pairwise loss function produced the

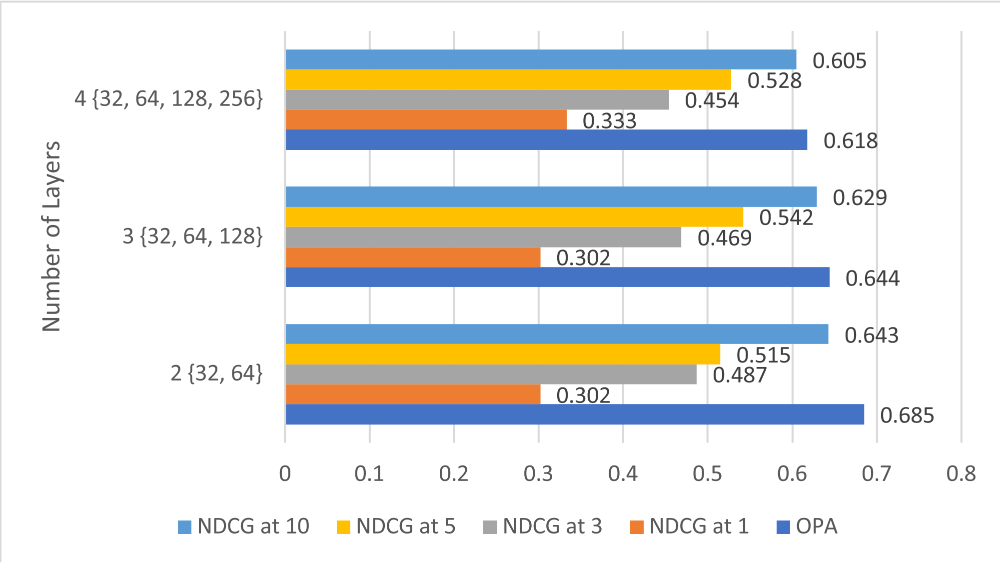


FIGURE 5. Results of layers validation.

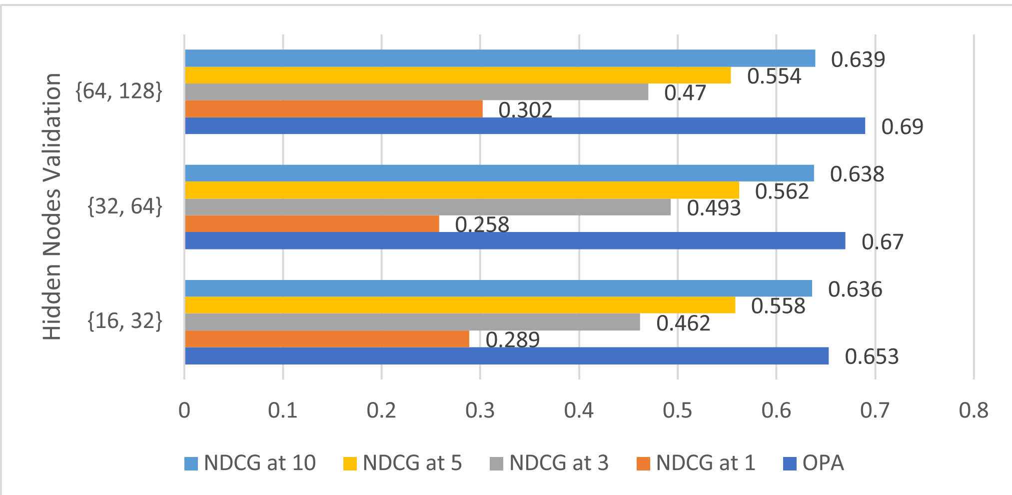


FIGURE 6. Results of hidden nodes validation.

lowest scores for all measures except OPA, highlighting the better performance of listwise algorithms for these ranking tasks, see **Fig. 7**.

4) LR VALIDATION

Although a greater learning rate usually leads to quicker convergence, it can also cause more volatility in the loss and difficulties with optimization at the end of the training phase. The metrics for different learning rates with 15,000 training steps were analyzed. At a learning rate of 0.05, the best results were shown for all but one statistic, see **Fig. 8**. As a result, a learning rate of 0.05 was selected for all models.

5) TRAINING

All six positional models were trained using these settings following the validation of the learning rate, model architecture, and loss function selection on the original model. The training and assessment loss for the original model is shown in **Fig. 9**. To make sure the models had enough steps to find the minima and optimize the parameters, a larger number of training steps—25,000—was used for this training. The training loss was not completely stable as the number of steps got higher, but it did show a general decreasing trend when the parameters were refined via backpropagation. This volatility is probably related to the 0.05 learning rate. The assessment loss decreased over the training procedure, suggesting that the model was still learning and may have continued to do so given more training time. Moreover, it seems that the model was discovering the underlying link between the labels and the document and context characteristics for every query rather than just overfitting. While certain models, like model three, showed higher volatility and model five showed a

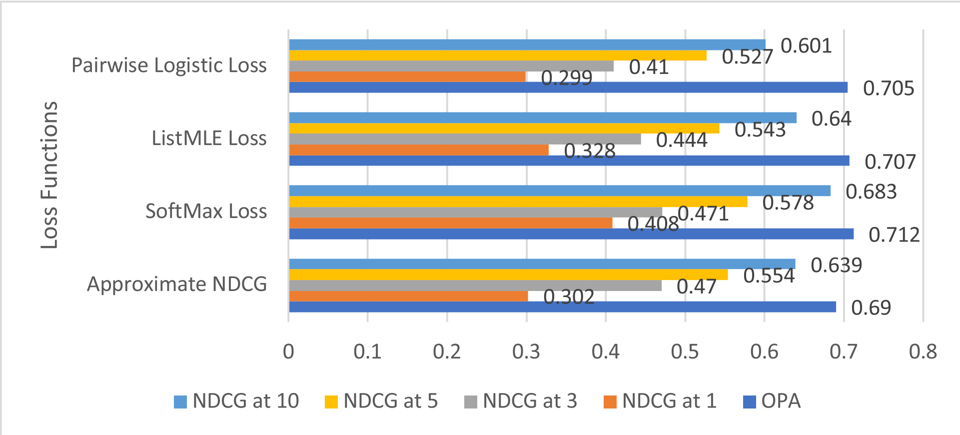


FIGURE 7. Results of loss function validation.

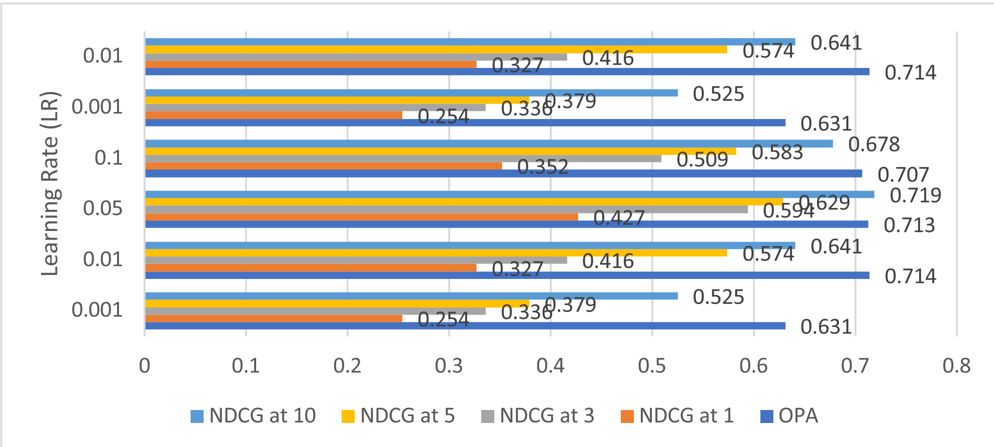


FIGURE 8. Results of learning rate validation.

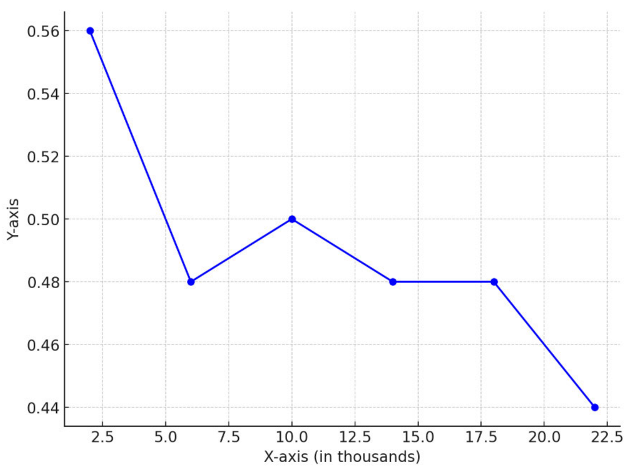


FIGURE 9. Training Loss for the first model.

steadier fall, most models had training losses that were quite similar.

V. EXPERIMENTS AND RESULTS

There is a trend towards higher scores with higher NDCG for all six models. While this is a common issue in ranking tasks, the variability, particularly in NDCG at 1, suggests that the models struggle to accurately select the best player for a given match. Since only the top player is likely to be chosen for that position, this represents a drawback for the model in player selection scenarios.

Moreover, there is significant variation in the performance of the different models, with models two and one outperforming others on all metrics. Despite the limited number of training instances for each model, the range of performance suggests that a more thorough hyper-parameter search, considering each model separately, would have been a preferable approach. **Fig. 10** and **Fig. 11** below depict the NDCG at 1 evolution during the training process for models three and four.

The performance of the various models shows significant variation, with models two and one consistently outperforming the others across all metrics. Despite the limited number of training examples for each model, the wide range of performance indicates that a more detailed hyper-parameter search for each model individually would have been a preferable approach. **Fig. 12** and **Fig. 13** depict the development of NDCG at 1 during training for models three and four.

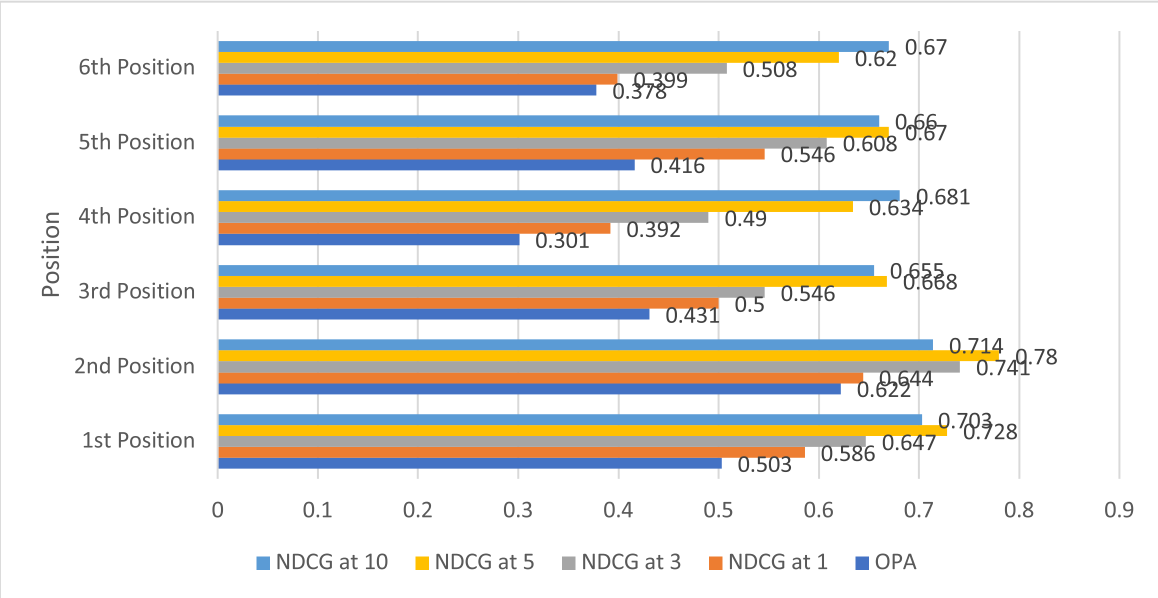


FIGURE 10. Evaluation of all six models.

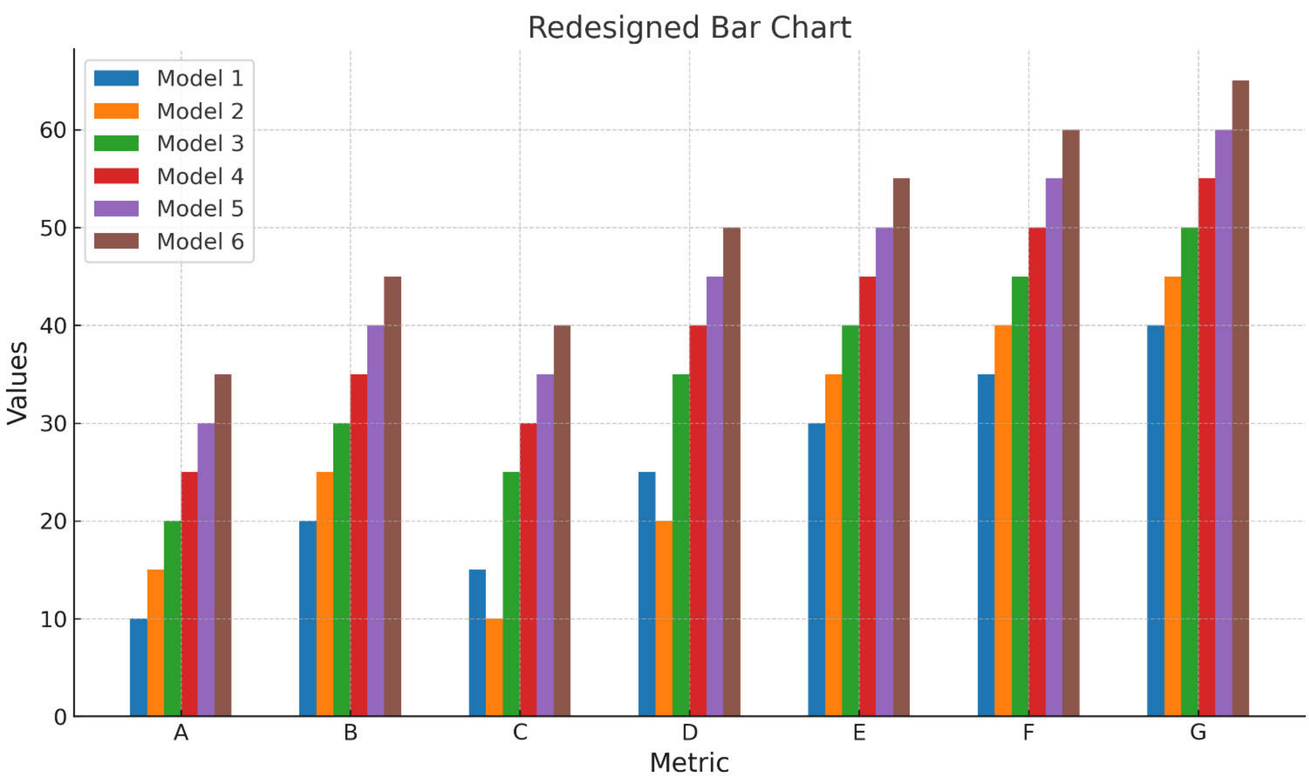


FIGURE 11. Evaluation metrics of all six models.

In contrast to the other five models, the third model's NDCG at 1 decreased during the training. It dropped to a position that was still higher than the fourth model, having begun at a level that was far higher than any of the previous models. This illustrates the variability that results from dividing training and testing data at random and how it affects the models' performance during assessment.

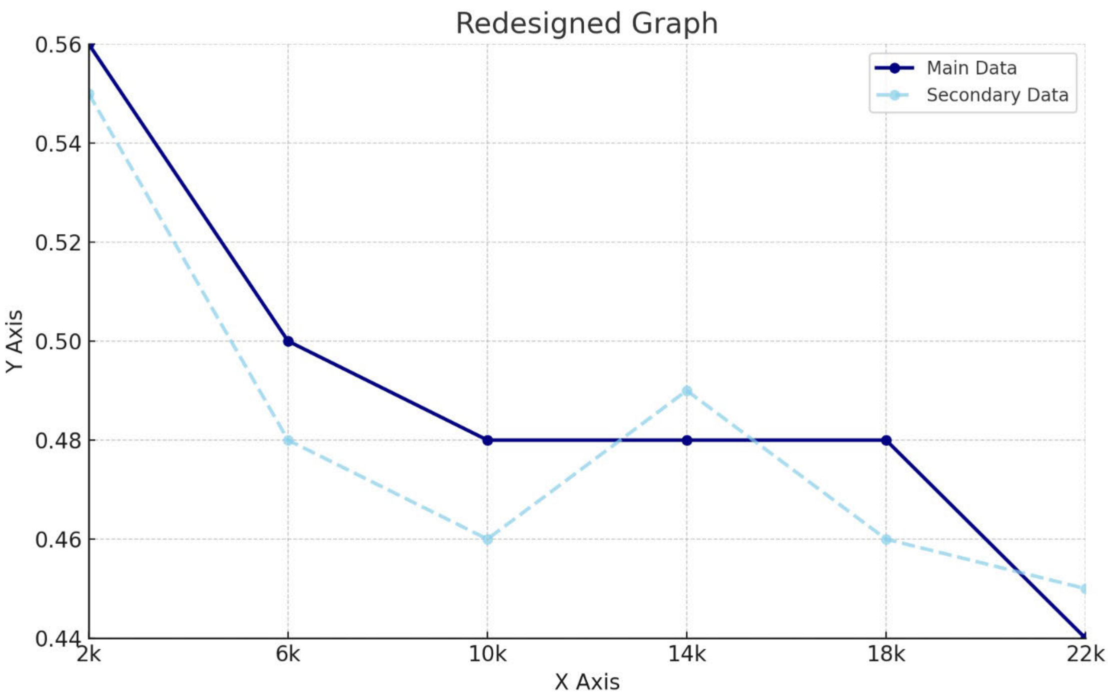


FIGURE 12. Third model NDCG@1.

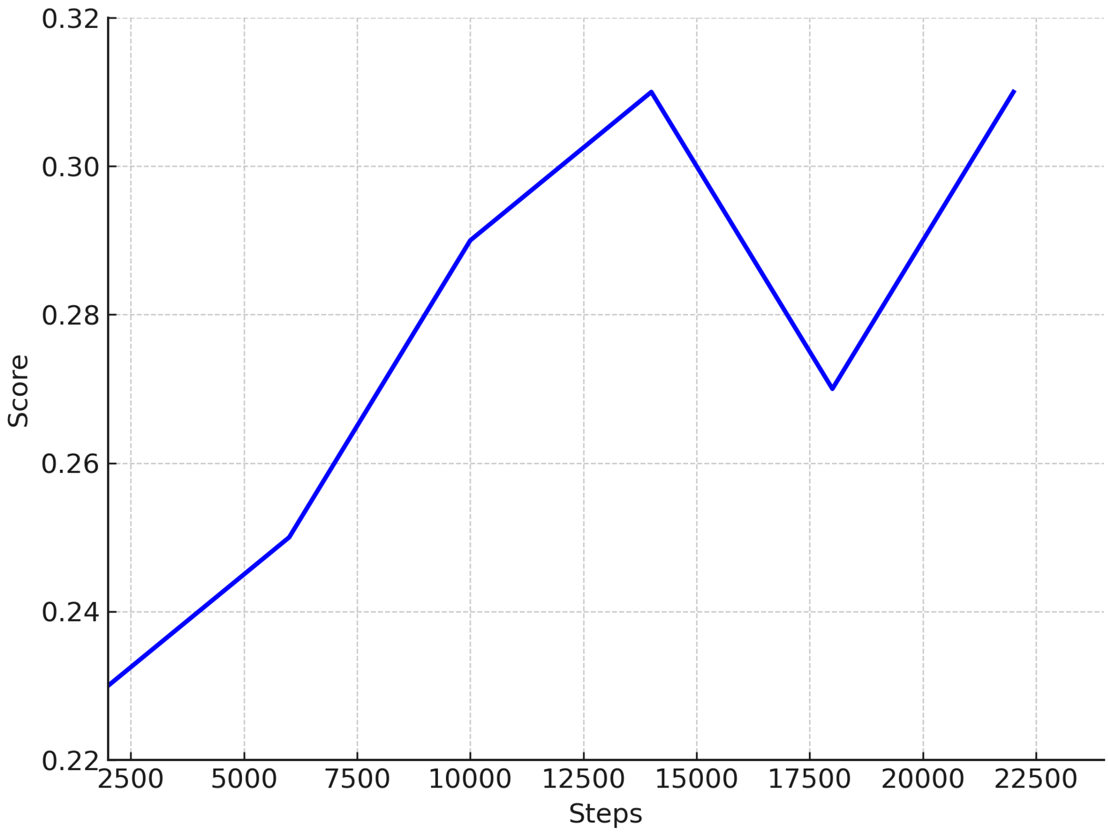


FIGURE 13. Fourth model NDCG@1.

With percentages ranging from 65-71%, the ordered pair accuracy of the six models was more comparable, indicating that each model's accuracy was much higher than random chance. This should be seen as proof that the models can

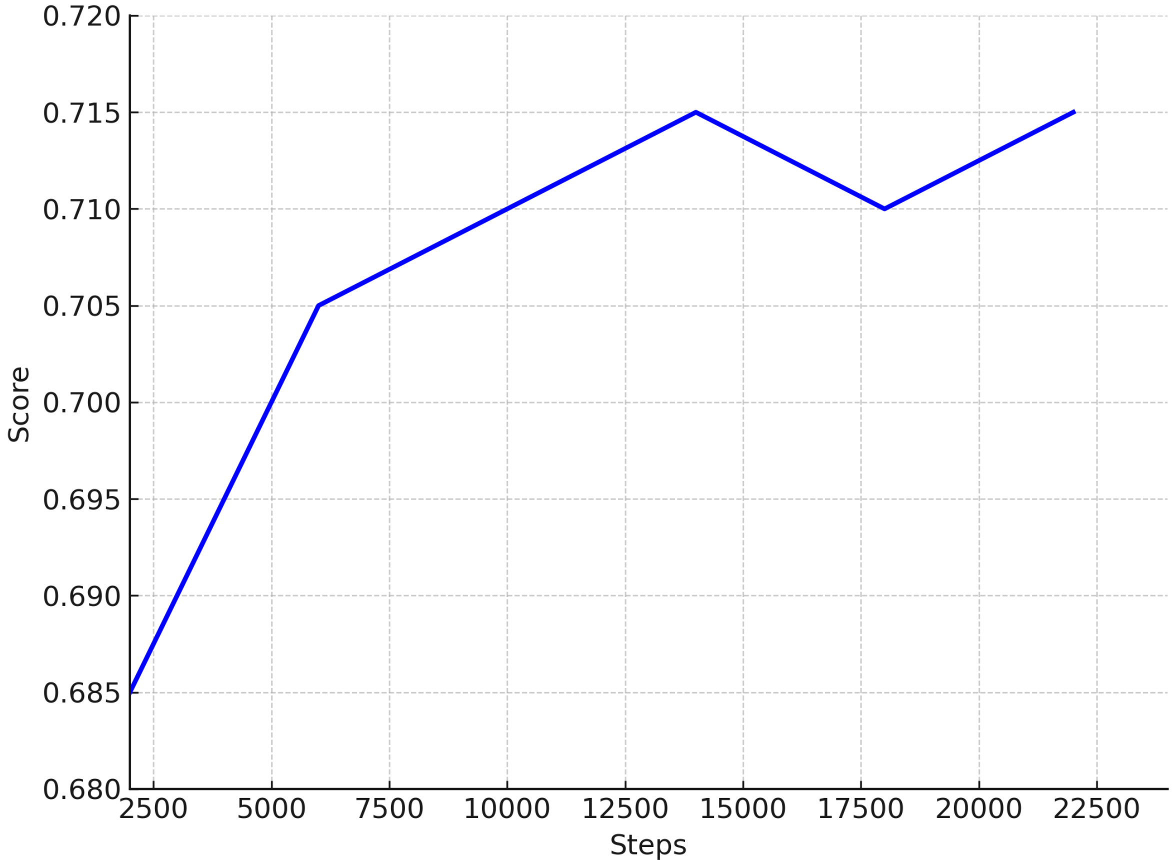


FIGURE 14. Second model OPA.

determine which players are better suited for specific matches and can absorb information from the training set. Figure 14 below depicts the evolution of the OPA for model two throughout training. Though it should be highlighted that the OPA change scale throughout the training was narrower than the NDCG at different levels, there was a noticeable rise throughout.

By considering overall, the models have shown some capacity to translate player group feature vectors to a ranking of how suitable those players are for a certain fixture. The training plots show that most of the model's accuracy on the evaluation datasets in-creased with more training, indicating that the models were likely learning fundamental patterns rather than merely overfitting the training set. Given that the measurements show progress even after 25,000 training steps, it is conceivable that training the models for an extended period will result in even greater performance gains.

VI. DISCUSSION

The experiment's results show a positive development towards higher NDCG score with highest values that indicates improved ranking performance of model. Furthermore, at 1 the variability in NDCG shows some inconsistencies like how every model can perfectly identifies which player is best for a specified position. Therefore, the identification of top player may be a limitation specifically in the selection of player's tasks. Although dataset is limited but model 1 and 2 are integrated as one of the most efficient and effective solutions to gain robustness and highest NDCG scores.

These results recommends that our models can get more benefit from an in-depth hyper-parameter tuning method to refine accuracy. Figure 10 and 11 give the general idea of performance metrics, however figure 12 and 13 showing NDCG to evaluate the model three and four. This limitation shows that partitioning of data influence the performance of model. Additionally, the accuracy of ordered pair of all models demonstrates a significant improvement in accuracy. These results show that all models capture the suitability of a player effectively by identifying their capability to learn from trained data.

Figure 14 shows that gradual improvement in accuracy of ordered pair for two models that affirms that an enhancement in training steps captures underlying data patterns instead overfitting. Overall the proposed solution and results have shown that additional training steps can produce better outcome in terms of players racking for specific matches.

VII. CONCLUSION AND FUTURE WORK

In conclusion, there is proof that a cricket player rating system may be developed using information retrieval techniques, especially Learning to Rank. Given that only the first player would probably be chosen for a team in the real world if such a model were implemented, it is disheartening that the top player in each of the six models performed worse than the others. It's also critical to draw attention to how few training examples were used in these tests. Given that, just 100 questions were generated and just 15 sample players were scored for each query, this raises concerns about the accuracy and repeatability of the outcomes of these models. Furthermore, as already said, creating actual labels for the training data was not the most thorough way to achieve this. Ideally, domain experts would not only offer the Batting Average and Strike Rate, which are limited measures of the value a particular hitter brings to a team but also rank the likelihood that individual players would perform against specific opponents.

Despite these concerns, the foundation has been laid for the use of ranking algorithms in cricket analysis and player selection in the future. The models are now able to map player performance and fixture difficulty level input elements to a rating of the input players. Despite the small number of training instances, there was no sign of overfitting, since the graphs show that as the training progressed, the assessment metrics enhanced. This might be extended to all positions on a cricket team and to further training data that becomes available. Comparisons with other methods are necessary to have a better understanding of this approach's accuracy.

Through this effort, the possibility of ranking athletes for a particular fixture using conventional Information Retrieval domain techniques has been demonstrated. The most apparent way to expand cricket analysis would be to create models for every member of the squad, rather than concentrating only on the top six batting positions. The training data may have been enlarged for that purpose, but it was not large enough to complete the work. Although this issue frequently arises in research publications on cricket performance analysis [6] [12], diversifying into more game forms will provide a larger player pool. Furthermore, as indicated in this paper's findings section, performance is probably going to be enhanced by individually adjusting the hyper-parameters for each model. Even though the Learning to Rank task was almost the same for each of the six sub-models, the different outcomes show that the assessment measures would gain from separate adjustments to the learning rate and other parameters.

VIII. MDEL PERFORMANCE AND LIMITATIONS

The importance of proposed model in this research is evaluated by measuring and reporting the performance metrics like NDCG and OPA. These metrics (NDCG and OPA) are ideal to assess the effectiveness of AI driven model for player ranking, especially for a main batting position. The AI model shows high level of accuracy that is more than 65% with OPA and NDCG approach indicates the performance rank of top players effectively and efficiently. A comparison was made between selection methods of traditional player and AI-driven models. Results proved that AI-driven model is highly objective as well as performance oriented to rank the players in terms of selection rather than traditional ranking methods.

Moreover, research presents the performance of model in multiple scenarios such as different match conditions and profiles of players. However, this study also has some limitations, for example training instances that may affect the generic implications of model. Despite such limitations, AI based models have ensured accurate estimation of player position in cricket matches.

REFERENCES

[1] R. K. Pasumarthi, S. Bruch, X. Wang, C. Li, M. Bendersky, M. Najork, J. Pfeifer, N. Golbandi, R. Anil, and S. Wolf, "TF-ranking: Scalable TensorFlow library for learning-to-rank," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2970–2978, doi: 10.1145/3292500.3330677.

[2] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proc. 24th Int. Conf. Mach. Learn.*, vol. 227, Jun. 2007, pp. 129–136, doi: 10.1145/1273496.1273513.

[3] S. Mukherjee, "Quantifying individual performance in cricket—A network analysis of batsmen and bowlers," *Phys. A, Stat. Mech. Appl.*, vol. 393, pp. 624–637, Jan. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437113008819

[4] A. Santra, A. Sinha, P. Saha, and A. K. Das, "A novel regression based technique for batsman evaluation in the Indian premier league," in *Proc. IEEE 1st Int. Conf. for Converg. Eng. (ICCE)*, Sep. 2020, pp. 379–384. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9290569/

[5] D. P. Shah, "New performance measure in cricket," *IOSR J. Sports Phys. Educ.*, vol. 4, no. 3, pp. 28–30, May 2017, doi: 10.9790/6737-04032830.

[6] C. Deep, C. Patvardhan, and C. Vasantha, "Data analytics based deep mayo predictor for IPL-9," *Int. J. Comput. Appl.*, vol. 152, no. 6, pp. 6–11, Oct. 2016, doi: 10.5120/ijca2016911875.

[7] P. Somaskandhan, G. Wijesinghe, L. B. Wijegunawardana, A. Bandaranayake, and S. Deegalla, "Identifying the optimal set of attributes that impose high impact on the end results of a cricket match using machine learning," in *Proc. IEEE Int. Conf. Ind. Inf. Syst. (ICIIS)*, Dec. 2017, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8300399/

[8] S. Shah, P. J. Hazarika, and J. Hazarika, "A study on performance of cricket players using factor analysis approach," *Int. J. Adv. Res. Comput. Sci.*, vol. 21, no. 3, pp. 656–660, 2019. [Online]. Available: https://www.researchgate.net/profile/Sricharan-Shah-2/publication/333669213_A_Study_on_Performance_of_Crick-et_Players_using_Factor_Analysis_Approach/links/5cfd073ba6fdccd1308e109c/A-Study-on-Performance-of-Cricket-Players-using-Factor-Analysis-Approach.pdf

[9] M. J. Hossain, M. A. Kashem, M. S. Islam, and M. E-Jannat, "Bangladesh cricket squad prediction using statistical data and genetic algorithm," in *Proc. 4th Int. Conf. Electr. Eng. Inf. Commun. Technol. (iCEEiCT)*, Sep. 2018, pp. 178–181. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8628076/

[10] M. Shetty, S. Rane, C. Pandita, and S. Salvi, "Machine learning-based selection of optimal sports team based on the players performance," in *Proc. 5th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jun. 2020, pp. 1267–1272. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9137891/

[11] R. Jadhav, B. Pawar, N. Bhat, S. Kawale, and A. Gawai, "Predicting optimal cricket team using data analysis," in *Proc. Int. Conf. Emerg. Smart Comput. Informat. (ESCI)*, Mar. 2021, pp. 269–274. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9396861/

[12] K. Passi and N. Pandey, "Increased prediction accuracy in the game of cricket using machine learning," 2018, *arXiv:1804.04226*.

[13] A. I. Anik, S. Yeaser, A. G. M. I. Hossain, and A. Chakrabarty, ''Player's performance prediction in ODI cricket using machine learning algorithms,'' in *Proc. 4th Int. Conf. Electr. Eng. Inf. Commun. Technol. (iCEEiCT)*, Sep. 2018, pp. 500–505. Accessed: Feb. 21, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8628118/

[14] K. P. Jayalath, ''A machine learning approach to analyze ODI cricket predictors,'' *J. Sports Anal.*, vol. 4, no. 1, pp. 73–84, 2018, doi: 10.3233/JSA-17175.

[15] S. Irvine and R. Kennedy, ''Analysis of performance indicators that most significantly affect international Twenty20 cricket,'' *Int. J. Perform. Anal. Sport*, vol. 17, no. 3, pp. 350–359, May 2017, doi: 10.1080/24748668.2017.1343989.

[16] V. S. Vetukuri, N. Sethi, and R. Rajender, ''Generic model for automated player selection for cricket teams using recurrent neural networks,'' *Evol. Intell.*, vol. 14, no. 2, pp. 971–978, Jun. 2021, doi: 10.1007/s12065-020-00488-4.

[17] H. Saikia, D. Bhattacharjee, and H. H. Lemmer, ''Predicting the performance of bowlers in IPL: An application of artificial neural network,'' *Int. J. Perform. Anal. Sport*, vol. 12, no. 1, pp. 75–89, Apr. 2012, doi: 10.1080/24748668.2012.11868584.

[18] M. Vohra and G. S. D. Gordon, ''Markov cricket: Using forward and inverse reinforcement learning to model, predict and optimize batting performance in one-day international cricket,'' 2021, *arXiv:2103.04349*.

[19] M. T. Shahriar, Y. Islam, and M. N. Amin, ''Player classification technique based on performance for a soccer team using machine learning algorithms,'' in *Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Feb. 2019, pp. 1–4. Accessed: Feb. 21, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8868989/

[20] T. S. N. Peiris and R. M. Silva, ''Player ranking in taekwondo: A Bayesian Elo rating system,'' in *Proc. From Innov. to Impact (FITI)*, vol. 1, Dec. 2020, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9424891/

[21] L. Pappalardo, P. Cintia, P. Ferragina, E. Massucco, D. Pedreschi, and F. Giannotti, ''PlayeRank,'' *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 5, pp. 1–27, Sep. 2019, doi: 10.1145/3343172.

[22] A. E. Aydemir, T. T. Temizel, A. Temizel, K. Preshlenov, and D. M. Strahinov, ''A dimension reduction approach to player rankings in European football,'' *IEEE Access*, vol. 9, pp. 119503–119519, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9521890/

[23] O. Uzochukwu and P. Enyindah, ''A machine learning application for football players' selection,'' *Int. J. Eng. Res. Technol. (IJERT)*, vol. 4, no. 10, pp. 459–465, 2015. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/63303740/a-machine-learning-application-IJERTV4IS10032320200513-85937-1vlho2j-libre.pdf?1589434090=&response-content-disposition=inline%3B+filename%3DIJERT_A_Machine_Learning_Application_for.pdf&Expires=1732640712&Signature=QM9l9YreRB4woumcdJioY0Sov9D3wajiVyF6AkKMWH3jlbe9sCB8JToOwNZg8rUZm9WXtFv5q2Uq-JF87oABbUp xwriDOdRKEzrk5zaUBkRdzJGpkBwXJfGJI5mJeZkOo3PbFlxhL1LPKqj8Up3BQtDyGpdDWW24W9SpOyj9AL1UV8a1cKZj 3x0NA8iN2T2f8Q1rABd6whkWEVdTej6e9yEVBflasfEmJO6idQWV3wr5oadHSRF29QuZNBEzUtpO 2ttL8n8XYNZio0QveGijlCqhtJanX9H9dd6l4EbTvt-MaWPrghKW7bn5OjVQnvbl0nUWYgSTXbq9l-Q-FFQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[24] R. Al-Shboul, T. Syed, J. Memon, and F. Khan, ''Automated player selection for sports team using competitive neural networks,'' *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 8, pp. 457–460, 2017. [Online]. Available: https://www.proquest.com/docview/2656437591?pq-origsite=gscholar&fromopenview=true&sourcetype=Scholarly%20Journals

[25] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, ''Learning to rank using gradient descent,'' in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 89–96, doi: 10.1145/1102351.1102363.

[26] M. Haldar, M. Abdool, P. Ramanathan, T. Xu, S. Yang, H. Duan, Q. Zhang, N. Barrow-Williams, B. C. Turnbull, B. M. Collins, and T. Legrand, ''Applying deep learning to airbnb search,'' in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1927–1935, doi: 10.1145/3292500.3330658.

[27] H. Akiyama, M. Tsuji, and S. Aramaki, ''Learning evaluation function for decision making of soccer agents using learning to rank,'' in *Proc. Joint 8th Int. Conf. Soft Comput. Intell. Syst. (SCIS) 17th Int. Symp. Adv. Intell. Syst. (ISIS)*, Aug. 2016, pp. 239–242. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7801646/

[28] A. Anuraj, G. S. Boparai, C. K. Leung, E. W. R. Madill, D. A. Pandhi, A. D. Patel, and R. K. Vyas, ''Sports data mining for cricket match prediction,'' in *Proc. Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, Mar. 2023, pp. 668–680.

[29] L. Lin, P. Li, Q. Wang, B. Bai, R. Cui, Z. Yu, D. Gao, and Y. Zhang, ''An EEG-based cross-subject interpretable CNN for game player expertise level classification,'' *Exp. Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121658, doi: 10.1016/j.eswa.2023.121658.

[30] M. Robel, M. A. R. Khan, I. Ahammad, M. M. Alam, and K. Hasan, ''Cricket players selection for national team and franchise league using machine learning algorithms,'' *Cloud Comput. Data Sci.*, vol. 5, no. 1, pp. 108–139, Dec. 2023, doi: 10.37256/ccds.5120243741. [Online]. Available: https://ojs.wiserpub.com/index.php/CCDS/article/view/3741

[31] *Official International Cricket Council Website*. Accessed: Feb. 21, 2024. [Online]. Available: https://www.icc-cricket.com/

[32] *T20I Matches | Batting Records | Most Runs in an Innings | ESPNcricinfo*. Accessed: Feb. 21, 2024. [Online]. Available: https://www.espncricinfo.com/records/most-runs-in-an-innings-284262

[33] *HowSTAT! The Cricket Statisticians—Home Page*. Accessed: Feb. 21, 2024. [Online]. Available: http://www.howstat.com/cricket/home.asp

[34] Q. Ai, X. Wang, S. Bruch, N. Golbandi, M. Bendersky, and M. Najork, ''Learning groupwise multivariate scoring functions using deep neural networks,'' in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retr.*, Sep. 2019, pp. 85–92, doi: 10.1145/3341981.3344218.

[35] T. Qin, T.-Y. Liu, and H. Li, ''A general approximation framework for direct optimization of information retrieval measures,'' *Inf. Retr.*, vol. 13, no. 4, pp. 375–397, Aug. 2010, doi: 10.1007/s10791-009-9124-x.

[36] S. Bruch, M. Zoghi, M. Bendersky, and M. Najork, ''Revisiting approximate metric optimization in the age of deep neural networks,'' in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 1241–1244, doi: 10.1145/3331184.3331347.

[37] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, ''Listwise approach to learning to rank: Theory and algorithm,'' in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1192–1199, doi: 10.1145/1390156.1390306.

[38] S. Ioffe, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1–11. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

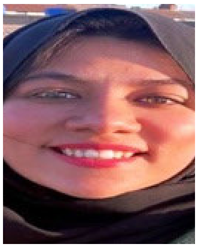
[39] A. Zhang, Z. C. Lipton, M. U. Li, and A. J. Smola, *Dive Into Deep Learning*. Cambridge Univ. Press, 2023. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=vfDiEAAAQBAJ&oi=fnd&pg=PR17&dq=Zhang,+A.,+Lipton,+Z.,+Li,+M.+%26+Smola,+A.,+n.d.+Dive+into+Deep+Learning.+arXiv+preprint+arXiv:2106.11342.&ots=ZQbiIoDzrk&sig=nGaG2Sgd5vYTdCxpRaG84ysbEY8

[40] A. Liu, R. P. Mahapatra, and A. V. R. Mayuri, ''Hybrid design for sports data visualization using AI and big data analytics,'' *Complex Intell. Syst.*, vol. 9, no. 3, pp. 2969–2980, Jun. 2023.

[41] C. Kapadiya, A. Shah, K. Adhvaryu, and P. Barot, ''Intelligent cricket team selection by predicting individual players' performance using efficient machine learning technique,'' *Int. J. Eng. Adv. Technol.*, vol. 9, no. 3, pp. 3406–3409, Feb. 2020.

BILAL HASSAN is pursuing the Ph.D. degree in computer vision from Queen Mary University of London, U.K., and working as a Senior Lecturer in computer vision and AI. He is an active researcher in biometrics and human-centred computing with over 40 scholarly articles in top-tier journals and conferences on his credit beside winner of several research grants from bodies like ICT Research and Development Pak, UMT PAK, Northumbria University, U.K., and British Council. He also published several book chapters and technical reports for the European Commission. He has several professional affiliations, such as IEEE, BMVA, ACM, IET, and BCS. He is an Associate Editor of a *few known journals* and technical committee member of renowned conferences with a few special issues and technical sessions on his record.

CLARE CLOUGH received the M.Sc. degree from the Queen Mary University of London, U.K. For the M.Sc. thesis, she developed for the development of a player ranking system in cricket using the largest dataset of player performance parameters collected using custom developed scripts, under the supervision of Dr. Bilal Hassan. She is a Researcher and a Data Scientist with the Queen Mary University of London. Her professional and research interests span over several application domains of AI, including healthcare, supply chain, and educational technologies.



YUSRA SIDDIQI received the master's degree in computing and technology from Northumbria University, London Campus, U.K. She is an Active Researcher in sports sciences with a passion for cricket. She has several good publications in sports sciences on her record. Her broad research interests include but not limited to sports situation analysis, player performance analysis, and ramification of illegalities in sports, especially in cricket.

RAO FAIZAN ALI (Member, IEEE) received the bachelor's degree in computer science from COMSATS University Islamabad, Pakistan, the M.Phil. degree in computer science from the University of Management and Technology, Lahore, Pakistan, and the Ph.D. degree from the University Technology PETRONAS, Malaysia. He has ten years of experience in teaching and research. He is currently a Lecture with the School of Computing, University of Kent, U.K. He has been a consultant with various computer science positions in financial, consulting, academia, and government sectors.

MUHAMMAD ASAD ARSHED received the M.S. degree in computer science from the Islamia University of Bahawalpur (IUB), Pakistan. He has an experience in bioinformatic research as a Research Officer with KICS, University of Engineering and Technology, Lahore, Pakistan. Currently, he is a Faculty Member with the School of Systems and Technology, University of Management and Technology, Lahore. His research interests include medical imaging, bioinformatics, machine learning, deep learning, and NLP.