CRICKET MATCH WINNER PREDICTOR

Krishna Advait Sripada
Student# 1007898884
krishna.sripada@mail.utoronto.ca

Lianne Choong
Student# 1007874230
l.choong@mail.utoronto.ca

Katarina Poffley
Student# 1005866412
katarina.poffley@mail.utoronto.ca

Snehal Sobti
Student# 1007675776
snehal.sobti@mail.utoronto.ca

ABSTRACT

A game that could last from a day to five and still ends in a draw stands out as one of its own kind like no other modern sport. Cricket! This paper aims to efficiently and effectively model the outcomes of past Cricket games using multi-label Artificial Neural Networks (ANNs). The final model is a 4-layered ANN that has an architecture of an input layer, 2 hidden layers, and an output layer, where the input takes in an array of 22 by 18 and outputs winner index - 0 if the first team wins, 1 if the second team wins, or 2 for a no-result. On the other hand, the baseline model is a Random Forest Classifier, which takes in a 22 by 18 array and outputs the winner index (0, 1, or 2). Both models went through a hyperparameter tuning process. For the final model, the best results yield from the following hyperparameters of a learning rate of 0.04, a number of epochs of 25, a batch size of 128, and a momentum of 0.9. In the case of the baseline model, optimal outcomes are achieved with the following hyperparameters: a number of estimators set to 125 and a maximum depth left as 'None'.

—-Total Pages: 9

1 INTRODUCTION

Growing up in an environment where Cricket has attained a status comparable to that of a religion, it has had a significant impact on the lives of our teammates, Krishna and Snehal. Observing the extensive analysis made by the fans, media journalists, commentators, and other enthusiasts prior to every cricket match sparked curiosity to determine if there was a way to predict various outcomes in a match. Katarina and Lianne, who relate to their obsession with a particular sport, have agreed to join them on this mission to develop a way to achieve these predictions.

Our primary objective is to predict the winner of an ODI match, a format of Cricket that goes on for a day. To accomplish this, we build and train a multi-label Artificial Neural Network to accurately predict the outcome of a cricket match when provided with 2 team names, 22 player names and the match year.

The journey is interesting as it allows us to gain a deeper understanding of the sport and the history, tactics, strategies, intricacies, and nuances associated with it. In addition, constructing such a model can captivate a diverse spectrum of enthusiasts. This encompasses fans, teams, broadcasters, and analysts, each deriving distinct benefits to fulfill their unique requirements.

We will leverage deep learning as it offers a powerful and sophisticated approach due to its ability to efficiently handle large and complex data sets. Its highly optimized algorithms can competently process and analyze vast amounts of data and discover valuable trends, patterns, and insights that might not be apparent through traditional methods.

2 ILLUSTRATION

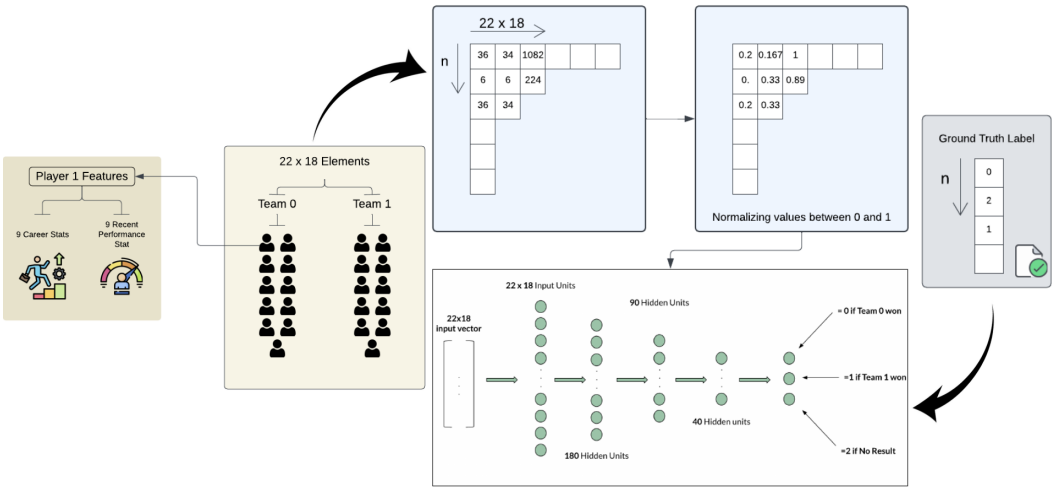

Figure 1: Overall Project Diagram

3 BACKGROUND AND RELATED WORK

3.1 ICC T20 CRICKET WORLD CUP 2020 WINNER PREDICTION USING MACHINE LEARNING TECHNIQUES

Basit et al. (2020)

This paper used ESPN Cricinfo to extract data and utilized various machine learning algorithms belonging to the decision tree algorithm family such as Decision trees (ID3, C4.5, and Extra Trees), and Random Forest Classifiers to build predictive models. Notably, the paper accurately predicted Australia as the winner of the T20 World Cup 2021, which was the actual outcome (the T20 World Cup 2020 was postponed to 2021 due to COVID-19).

3.2 "APPLICATIONS OF MODERN CLASSIFICATION TECHNIQUES TO PREDICT THE OUTCOME OF ODI CRICKET"

Pathak & Wadhwa (2016)

This paper accounted for several factors for the construction of the model, including home advantage, toss, physical fitness of teams, day/night matches, etc. Modern classification techniques such as Naive Bayes, SVMs, and Random Forest were applied, and a comparative analysis of their outcomes and performances was conducted.

3.3 "A STUDY ON MACHINE LEARNING APPROACHES FOR PLAYER PERFORMANCE AND MATCH RESULTS PREDICTION"

Mittal et al. (2021)

Data was collected from ESPN Cricinfo and Kaggle to perform a comprehensive analysis using various machine-learning approaches and algorithms. These included k-Nearest Neighbour (KNN), Logistic Regression, Neural Networks, Support Vector Machines (SVMs), Naïve Bayes, and Random Forest.

In the case of Neural Networks, an ensemble classification approach was utilized, training the model with data up to the 2011 World Cup to forecast the winner of the 2015 World Cup. Australia was

assigned a 25.1 percent win probability, the highest among all teams, and ultimately emerged as the actual winner.

3.4 Increased Prediction Accuracy in the Game of Cricket Using Machine Learning

Passi & Pandey (2018)

Data was collected from www.cricinfo.com using scraping tools, ParseHub, and import.io and the data mining algorithms used were Weka and Dataiku. Attributes and weights were calculated and learning algorithms such as Naïve Bayes, decision trees, random forest, and multiclass support vector were utilized where Random Forest produced the best accuracy for score predictions.

3.5 Cricket Match Analytics Using the Big Data Approach

Awan et al. (2021)

A machine learning linear regression model has been applied to predict the team scores using the data of one-day international matches from 2006 to 2017. Scikit Learn and Spark ML have been used for linear regression models. The data has been collected from Cricsheet, which contains ball-by-ball information for over 14000 matches.

4 DATA PROCESSING

4.1 SOURCES OF DATA

1. To acquire data from ESPNcricinfo's Statsguru tool, we filtered data by: ESP
   (a) ODI (One Day International) matches from 5th Jan. 1971 to 11th Jul. 2023
   (b) Batting, bowling, or team records
   (c) Sorting by start date, player name, or matches played
2. For web scraping, we used Python's pandas library, specifically the *pandas.read html()* function, to efficiently extract and manipulate data tables from web pages.
3. We used a master URL prefix: *'https://stats.espncricinfo.com/ci/engine/stats/index.html?class=2;'*
4. To retrieve specific data, we added suffixes to the URL. For instance,
   (a) Type: *'type=batting'* suffix for the batting records
   (b) View: *'view=innings'* suffix for the innings-by-innings list
   (c) Order By: *'orderby=start'* suffix for sorting by match start date
5. To handle data across multiple web pages of ESPNcricinfo, we employed a 'for' loop with URL prefix, suffix and page number. For example, to gather match results:
   (a) URL prefix = *'https://stats.espncricinfo.com/ci/engine/stats/index.html?class=2;'*
   (b) URL suffix = *'template=results;type=aggregate;view=results"'*
   (c) Using a loop, we iterated through all pages and formed URLs like: **URL = URL prefix + page number + URL suffix**

4.2 DATA ORGANIZATION

1. Data is fetched into pandas dataframes, then cleaned, processed, and saved as .xlsx files for future analysis.
2. Storing data in spreadsheets eliminates repeated web scraping, allowing easy access through Excel files as pandas dataframes.
3. We stored the following data for our final model:
   (a) Career Stats for batsmen and bowlers
   (b) Year-wise Stats for batsmen and bowlers
   (c) Match Results
   (d) Innings-by-innings list of batsmen and bowlers

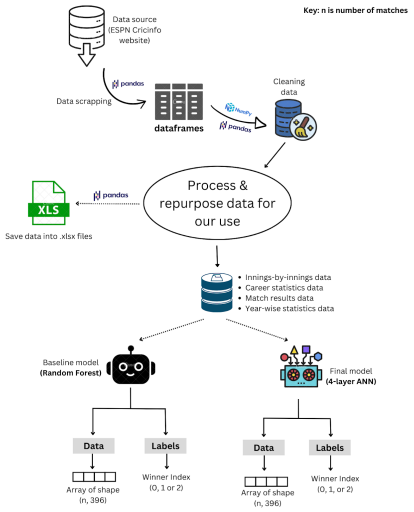

Figure 2: Summary of Data Processing Procedure

4.3 DATA PROCESSING AND RE-PURPOSING FOR FINAL MODEL

1. We need a 22 by 18 (396-element) array to represent each match data which represents the 22 players of the two teams and 9 features for each player's career stats (batting and bowling combined) and 9 features for each player's recent performance stats (batting and bowling combined).
2. We created a common key element called 'Match ID' that connects the two dataframes of innings-by-innings and match results data.
   (a) In this scenario, extensive data processing and cleaning were necessary due to the absence of Match IDs provided by ESPNcricinfo. To address this, we devised an algorithm that utilizes the Ground, Start Date, and Opposition Team data from the Innings List. This algorithm assigns a unique match ID to each match, enabling effective organization and analysis of the data.
   (b) We also assigned a Match ID to the Match Results Data
   (c) We excluded matches that lacked innings data, such as those that were abandoned, canceled, or where one of the teams won by a walkover (cases where the match did not even commence).
   (d) We did not discard the matches with 'tied' and 'no-result' outcomes as these contained data for partial innings. We modified the incomplete fields in such cases by zeros.
3. From innings data, we retrieved a 22-element array for each match ID, which contained the names of 22 players involved in the match
4. Then, we utilized the career stats and recent performance stats to generate a 396-element (22 by 18) array for each match
5. From the match results data, we retrieved the ground truth labels for each match. These labels indicated the winner index: 0 for Team 0, 1 for Team 1, and 2 for a no-result, tied, or partially completed match. Additionally, we utilized the match results data to map the match ID to the corresponding set of innings in the innings data.

5 ARCHITECTURE

The final model is a multi-class Artificial Neural Network (ANN) with four layers. The input layer comprises 396 neurons, aligned with a 22x18 vector input. Subsequently, the first hidden layer accommodates 180 neurons, followed by 90 neurons in the second hidden layer, 40 neurons in the third hidden layer, and culminating in three output neurons. These output neurons encapsulate the

Mat_C_Player_1	Inns_batting_C_Player_1	Runs_C_Player_1	Ave_batting_C_Player_1	SR_C_Player_1	Inns_bowling_C_Player_1	Wkts_C_Player_1	Ave_bowling_C_Player_1	Econ_C_Player_1
36	34	1082	36.06	53.56	6	5	21	3.75
6	6	224	37.33	57.88	2	3	18	4.2
36	34	1082	36.06	53.56	6	5	21	3.75
6	6	224	37.33	57.88	2	3	18	4.2
3	3	62	0	59.61	0	0	0	0
41	40	1598	47	68.05	1	0	0	0
115	98	1751	21.61	75.5	112	158	21.56	3.3
12	12	311	25.91	70.68	1	2	5	6
5	5	19	0	22.35	0	0	0	0
3	3	62	0	59.61	0	0	0	0
1	1	53	0	79.1	0	0	0	0
108	102	3092	35.13	62.26	4	1	25	7.5
108	102	3092	35.13	62.26	4	1	25	7.5

(a) Data

Winner_Index
1
1
1
1
0
1
2
1
1
1
0
1
1
1
1
1
2
2
0
0
0
1

(b) Labels

Figure 3: Structure of Data and Labels for Final Model

possible match outcomes: team 0's victory, team 1's victory, or an absence of results, encompassing scenarios like tied or incomplete (e.g., due to rain) matches.

ReLu serves as the activation function following the initial three linear layers. During the training process, the loss function employed is *Cross Entropy Loss*, while the optimizer utilized is *Stochastic Gradient Descent*.

The two data files used for training the final model are *final_model_data.xlsx* and *final_model_labels.xlsx*. The former, *final_model_data.xlsx*, encompasses a 396-element (22x18) array for around 4500 match samples. The latter, *final_model_labels.xlsx*, encompasses the ground truth label for each of those 4500 matches, designated with values of 0, 1, or 2, corresponding to match outcome, with 0 denoting team 0's win, 1 for team 1's win, and 2 indicating no result.

Each entry in the final model data file corresponds to the 9 career and 9 recent performance statistics for all 22 players involved in a specific match. Our concentrated attention lies on specific performance statistics, including Matches Played, Innings Played as a Batsman, Innings Played as a Bowler, Runs Scored, Batting Average, Strike Rate, Wickets Taken, Bowling Average, and Economy. Incorporating the match year as an input to the model signifies the usage of recent data for the training procedure to make a final prediction.

In order to get a prediction, the two arrays that contain the names of the players in team 0 and team 1 are passed into a function called *getWinner*, along with the match year. This function allows us to get predictions for matches that happen in the future. The function *getWinner* loads the pre-trained weights in the four-layered ANN and creates a prediction on which team is the winner.

We added historical data by creating a function called *get_features_dataset* where the arrays of 22 player names, and the match year are passed in. Two dictionaries are created; the first contains the combined batting and bowling career data for each player and the second dictionary has the year as the key and the value is another dictionary with the player name as the keys and an array of the 9 elements reflecting the recent performance of the player. The second dictionary, which takes into consideration the most recent performance features is done for the last two years that each player played (i.e., if the prediction year is 2011, two recent form values will be created for each player; one for 2010 and one for 2009, and will be added).

6 BASELINE MODEL

The baseline model adopts a heuristic approach by utilizing a straightforward Random Forest model. Data for the model is extracted from the *final_model_data.xlsx* data file, encompassing 18 features for all 22 players. Labels, drawn from *final_model_labels.xlsx* data file, are represented as values of 0, 1, or 2 similar to the final model's architecture.

Data and labels undergo conversion into a NumPy array before being inputted into the Random Forest Model. A 70:30 split ratio is applied, segregating the data into training and test sets.

To align with standard practices, the maximum features for node splitting are set to the square root of the total features. The model's *n_jobs* parameter is assigned a positive integer value of '1' to leverage all available CPU cores, expediting the training process through parallelization while fine-tuning the number of decision trees.

Upon conducting diverse hyperparameter tuning experiments, optimal outcomes emerge with the following settings:

- *n_estimators* = 125
- *max_depth* = None

The findings of these configurations are encapsulated in Figure 4. Accuracy reaches a plateau at around 100 trees, displaying minimal growth. Thus, this value is chosen. The baseline model attains a test accuracy of 72.247% for this value.

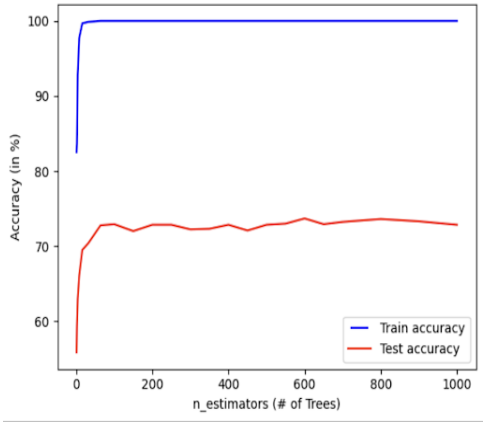

Figure 4: Training and Testing Accuracy of Random Forrest

7 Quantitative Results

After experimentation with various sets of hyperparameters, the best results were obtained with the following hyperparameters:

- Learning rate = 0.04
- Epochs = 25
- Batch size = 128
- Momentum = 0.9

The (best) final training accuracy of 85.934% and a final validation accuracy of 71.057% was achieved. And, the final testing accuracy of the final model was 72.519%. In comparing the final testing accuracy to the baseline model, our final model performs better by 0.272%.

Figure 5 below illustrates the training loss and the training and validation accuracy curves of the final model with the above-mentioned hyperparameters.

8 Qualitative Results

Figure 6 illustrates the functionality of the final model, along with the UI that we created for a match that occurred on July 27th, 2023 between India and the West Indies. The final model was trained on match data up until July 11th, 2023 so this match is considered never before seen data. The match performed well; the predicted winner was India and the actual winner was India.

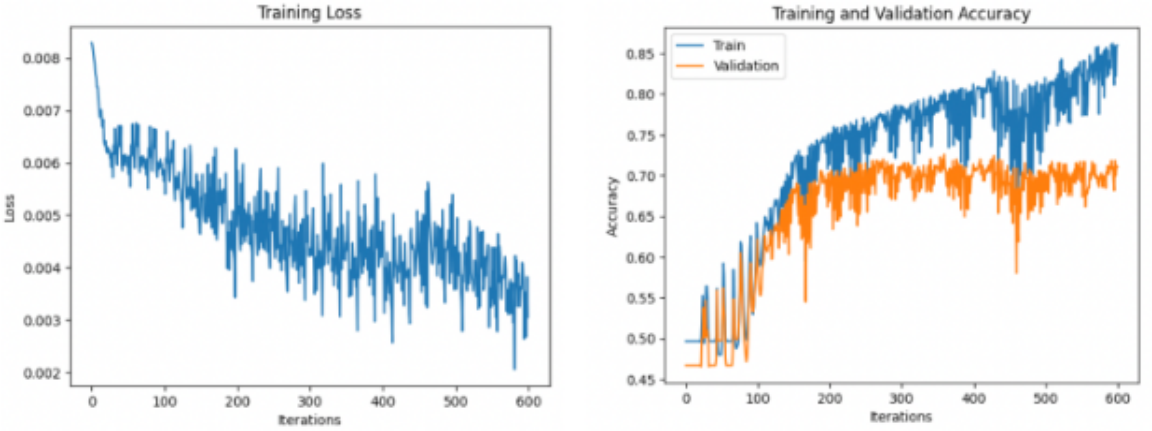

Figure 5: Loss and Accuracy Curves during Training

To further reflect our model's qualitative performance on never before seen data, we create imaginary test cases such as a strong team vs a weak team, an experienced vs an inexperienced team, and so on. We obtain accurate results on all of the imaginary test cases. We elaborate on these self-made testing samples in the next section.

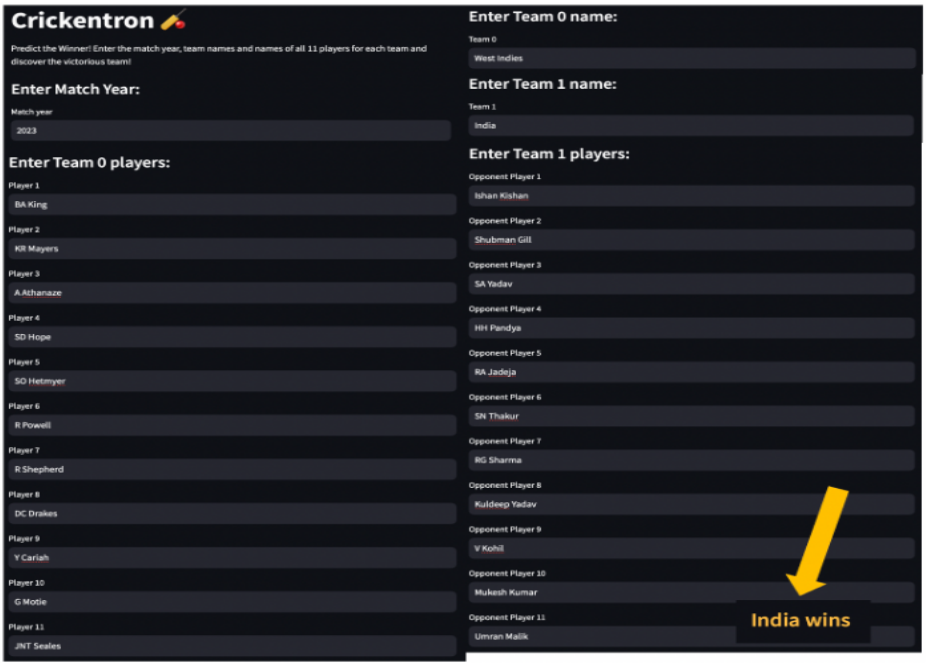

Figure 6: Example of Model's Functionality

9 EVALUATE MODEL ON NEW DATA

To ensure the results are a good representation of the model's performance, the model is trained with multiple sets of hyperparameters. Through the tuning of hyperparameters, the best set consists of a batch size of 128, 25 epochs, and a learning rate of 0.04. It yields a training accuracy of 85.934%, and a validation accuracy of 71.057%. Using the model with the tuned hyperparameters, the model is able to perform adequately well on never-before-seen data.

Data collected during data processing is split in a ratio of 70:15:15 for training, validation, and testing. The never-before-seen data for our model is taken from the 15% split, which is around 675 matches (15% of 4500), testing the model's performance. Our model yielded a testing accuracy of 72.519%. To add on, since data collected for the 4500 matches are ODI matches between Jan 5, 1971, to July 11, 2023, new data uploaded after July 11, 2023, from ESPN Cricinfo can also be collected as never-seen-before data as well. Other than these matches, as a team, we also built imaginary testing samples to test the sensitivity of our model and determine if our model is able to handle disruption.

Here are the details of how we augmented our existing data to create new data. On all of the following cases except the close encounter (5th in this list), our model predicted the expected results:

1. A team of experienced players vs a team of inexperienced players
   (a) A random year will be selected
   (b) Players will be sorted based on the number of matches they have played
   (c) Players are then sorted into an experienced or inexperienced team
2. A strong team vs a weak team
   (a) A random year will be selected
   (b) Batsmen will be sorted based on the number of runs they have scored
   (c) Bowlers will be sorted based on the number of wickets they have taken
   (d) Batsmen and Bowlers are then sorted into a strong and a weak team
3. A retired team vs. a new team in a future year. This will test just the dependence of our model on career statistics ignoring the recent performances.
   (a) Retired players are selected
      i. Based on Historical Data (before 2010)
      ii. Based on Number of Runs and Wickets
   (b) New team players are selected
      i. Based on Recent Years (closer to 2020)
      ii. Based on Number of Runs and Wickets
4. Substitution of a key player with another player in a match that has already happened. This will help us determine if our model can handle such disruptions.
   (a) A key player or a regular player will be selected based on cricket knowledge and existing statistics
   (b) Depending on the match, one key player of one of the teams will be substituted with a regular player
5. Choosing 2 similar teams in strength. This will test the sensitivity of your model to small differences.
   (a) A random year will be selected
   (b) 2 Teams with similar performance stats in that year will be selected but will keep minute differences

10 Discussion

We think that our model is performing well given the difficulty involved in a cricket match prediction. A cricket match is affected by innumerable factors such as pitch conditions, weather conditions, the kind of bowl being used, the venue of the match, players' individual performances, team performances, and much more. Moreover, cricket history is full of unusual results and upsets (an underdog beating a strong team) which deviated from the statistics and perhaps, our model would have failed to detect those because it is based on the career and recent performance stats of the players. Cricket is a strange game where a lower-ranked team can beat the higher-ranked team on any given day. On any given day, all the key players of a team may underperform which might lead to a not-so-expected result. Despite this difficulty, our final model was successful in achieving a testing accuracy of 72.519%. Moreover, it performs sufficiently well on the augmented data samples that we elaborated in the previous section.

But one interesting thing about our results is that the accuracy of the final model is just slightly higher than the baseline model which is a simple Random Forest model. And moreover, the final validation accuracy of our final model could never cross 73%. This again can be attributed to the randomness and the unpredictability of a cricket game, its players and the varying factors. Moreover, even after we have considered all the ODI matches that have ever happened (except the three matches that happened recently), the data is still limited to only 4500 matches. And even in those 4500 matches, the teams and players have evolved considerably. For instance, West-Indies won the first two ODI world cups and strangely, they could not qualify for the 2023 world cup.

One of the biggest lessons we learned was the importance of understanding the structure of data and how we will be processing it for our deep-learning model. Fetching, organizing and processing the data for our final model was the most crucial part of our project.

Deciding on what features should be selected and how data should be input to our final model to incorporate several factors, was the most critical roadblock during this project. But finally, we learnt about ways in which we could format our data and finalized on passing a 22x18 element array as input to our model.

11 Ethical Considerations

The most significant threat to using the cricket analysis system is its application for illegal activities such as gambling and match-fixing. Certain individuals may use the system for personal gain and exploitation of betting markets. The match results may also be influenced, which can have severe ethical implications as it sabotages the integrity of the sport.

There are many forms of bias in Machine Learning that can give rise to ethical concerns. One is a historical bias; there is a lack of availability of data for newly joined teams which makes it harder to predict results and fosters inaccuracy. We notice that there are some teams that have significantly more data than the others because they have been playing for a longer time period, as such, we did not create any all-encompasing team strength score.

12 Project Difficulty / Quality

The major difficulties of the project include the amount of data available, the randomness of the cricket game, and determining the input to our model. ESPNcricinfo has the data for all the ODI matches but only around 4500 ODI matches have happened till now. Though the number of matches is just 4500 but each match involves 22 players and we needed to scrape 22*4500 data entries which were organized in around 2000 web pages. So, the web scraping time would have been around 3-4 hours but to make the web scraping process 10 times faster, our team specifically utilized Python's multi-threading. However, this led to challenges in retrieving the return value of the threads as passing the same memory treads resulted in memory conflicts. To overcome this, an array or n-arrays are then employed to ensure that threads in the same memory location are not modified by the other threads.

Another major difficulty of the project is determining the input for our model. There are tens of factors that affect a cricket match. Firstly, not all the factors' data is available for the past matches. For instance, fetching the environmental factors such as pitch and weather conditions for each match in history is not feasible. Secondly, some factors cannot be accounted for, such as the player's performance on any given day can vary significantly from his/her career and recent performances. Moreover, a strong team losing to a weak team is also unpredictable. Given the dependence of a cricket match on so many factors, the complexity of this project increases manifolds. Nevertheless, we finalised on passing 18 features for each of the 22 players as the input to our model.

Despite all the difficulties involved regarding the lack of data, randomness and complexity involved in a cricket game and the ever-changing conditions, we were able to develop a fully-functioning deep learning model that can predict the winner of a cricket ODI match and it yielded 72.519% testing accuracy.

REFERENCES

URL https://www.espncricinfo.com/records.

Mazhar Javed Awan, Syed Arbaz Haider Gilani, Hamza Ramzan, Haitham Nobanee, Awais Yasin, Azlan Mohd Zain, and Rabia Javed. Cricket match analytics using the big data approach. *Electronics*, 10(19), 2021. ISSN 2079-9292. doi: 10.3390/electronics10192350. URL https://www.mdpi.com/2079-9292/10/19/2350.

Abdul Basit, Muhammad Bux Alvi, Fawwad Hassan Jaskani, Majdah Alvi, Kashif H. Memon, and Rehan Ali Shah. Icc t20 cricket world cup 2020 winner prediction using machine learning techniques. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pp. 1–6, 2020. doi: 10.1109/INMIC50486.2020.9318077.

Harsh Mittal, Deepak Rikhari, Jitendra Kumar, and Ashutosh Kumar Singh. A study on machine learning approaches for player performance and match results prediction, Aug 2021. URL https://arxiv.org/abs/2108.10125.

Kalpdrum Passi and Niravkumar Pandey. Increased prediction accuracy in the game of cricket using machine learning. *International Journal of Data Mining Knowledge Management Process*, 8: 19–36, 03 2018. doi: 10.5121/ijdkp.2018.8203.

Neeraj Pathak and Hardik Wadhwa. Applications of modern classification techniques to predict the outcome of odi cricket. *Procedia Computer Science*, 87:55–60, 2016. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2016.05.126. URL https://www.sciencedirect.com/science/article/pii/S1877050916304653. Fourth International Conference on Recent Trends in Computer Science Engineering (ICRTCSE 2016).