NBA Match Prediction using pure statistics and machine learning

Utsav Lal
ualal@ncsu.edu
North Carolina State University
Raleigh, NC, USA

Swaraj Kaondal
skaonda@ncsu.edu
North Carolina State University
Raleigh, NC, USA

ABSTRACT

This paper discusses the findings and observations derived from our project focused on developing a machine learning model to predict NBA game outcomes. We introduce innovative approaches for integrating datasets from two distinct sources and explore three different machine learning techniques for match winner prediction. Additionally, we formulate a hypothesis and investigate its validity within our study. Our analysis highlights the challenges and insights gained from applying these methods to NBA game prediction, contributing to the broader field of sports analytics and predictive modeling.

1 INTRODUCTION AND BACKGROUND

Basketball, a team-based game typically featuring five players per team, involves competing on a rectangular court. The main aim is to shoot a basketball through the opponent's hoop, which is approximately 9.4 inches (24 cm) in diameter, mounted 10 feet (3.048 m) high on a backboard at each end of the court. The opposing team tries to prevent this while aiming to score through their own hoop. It's a sport deeply ingrained in American culture, with extensive historical data available for analysis. In the era of data analytics, there has been significant focus on uncovering hidden patterns within this data to enhance player performance. This project aims to leverage historical player statistics to forecast the outcome of basketball matches.

1.1 Problem Statement

As stated earlier, bleeding edge techniques in data analysis has brought forward new ways of finding interesting patterns within data. National Basketball Association (NBA) hosts one of the oldest basketball competitions across the USA. Due to its historic nature a huge amount of player data has been collected in the process. With this project we aim to analyze this available data of player statistics and ultimately design a machine learning algorithm that would eventually predict a winner between two teams. At the midway stage, we clean the available dataset and transform the dataset for training. We then train a Naive Bayes model which sets the baseline for future neural network implementation.

1.2 Prior Research on topic

In [1] the authors used artificial neural networks to predict the eventual champion of an NBA season. They come up with a neural network with 3 hidden layers and use batch based training with a batch size of 32 and total of 50 epochs. They came up with 2 models for their use case which performed extremely well with very small overall loss. As a final step they predicted the champion of the 2022-2023 season. Their model was able to correctly predict the winner of the season. Another similar research is that of [5] where the authors tried to predict Soccer World Cup 2018 matches using Radial-basis neural networks. They achieved a correct percentage of win and loss of 83.7% and 72.7%. The authors used a 3 layer neural network with radial basis functions as their activation functions to achieve excellent results. They were also able to find out interesting tactics which led teams to win soccer games. Another interesting research is of [9] where a decision tree classifier was trained on football matches in the CFASL league. The authors have provided an algorithm for training the decision tree and built a model using the said algorithm. They are able to achieve a 57.7% accuracy in correctly predicting a match winner.

2 METHOD

2.1 Introduction

In the project, we want to propose a artificial neural network based machine learning technique to predict the winner of a match given players on each side. To measure the success of our proposed neural network architecture we also design a baseline Naive Bayes model for match predication. Naive Bayes is a classification algorithm which uses Bayes theorem [7] to calculate the probability of a match winner. Due to its simplicity it is a great machine learning algorithm tool for comparison to our actual proposed solution which will be based on an artificial neural network. To validate the hypothesis outlined in section 4, we also intend to develop a model that evaluates the L2 normalization values of attributes for both teams and the team with the higher value will be predicted as the winner.

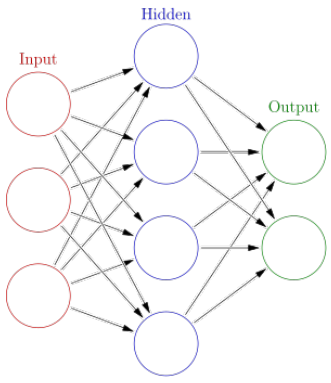

Figure 1: An artificial Neural network

2.2 Why Artificial Neural Network?

Our problem statement revolves around forecasting match winner. More specifically this translates into a classification problem. The

first algorithm which we thought of was logistic regression since our problem has a binary relation. But logistic regression doesn't deal well with complicated dataset like ours. Next came binary trees, but again since we have a lot of features in our dataset it decision trees might overfit and become overly complicated to correctly make a prediction. K-Nearest neighbors doesn't work well with large datasets. Hence in the end we decided to use Artificial neural networks (ANN) [10].

In recent years ANNs has emerged as one of the best techniques for solving classification problems. One of the main reasons for selecting neural networks for solving our problem is because of its success in solving similar problems as stated in the prior research section. As we will see later in the paper, Naive Bayes fails to capture the non linearity of the data. ANNs with its architecture of hidden layers becomes a great tool for capturing the non linearity within the data. It can also capture hidden features within the data which might not be obvious from just looking at the data. For these reasons ANNs emerge as an excellent tool for solving our problem.

2.3 Novelty

In our project, our objective is to predict the winner of a specific game using player statistics exclusively. Previous studies have typically concentrated on predicting either the winner of an entire tournament or outcomes within a single tournament using statistical methods. However, our focus is on forecasting the winners of individual matches. Although this approach may result in somewhat less precise outcomes, it offers valuable insights into the complexities of match outcome prediction.

Additionally, a unique aspect of our project is the integration of datasets from two different sources. This introduces additional challenges, as previous research has not explored the merging of two distinct datasets in this context.

2.4 Approach

(1) Naive Bayes: For Naive Bayes, the approach will be simple. For Naive Bayes we assume that each of the feature within the dataset is conditionally independent of each other given we have an outcome of the match. It might not be true for our dataset but it helps us create a baseline model. In our case since the player attributes are continues we will use the Guassian Naive Bayes algorithm. Probability of a feature given y will be defined as:

$$P(x_i/y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} * e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}} \tag{1}$$

And the probability of an outcome is given by

$$P(y/x_1x_2...x_n) = \arg\max P(Y=y) \prod P(x_i/Y=y]) \tag{2}$$

With this we can predict the winner of a match with players of given statistics. Each stat in this case will be a feature.

(2) L2-Norm comparator: In this model we create an array of all the attributes of each team and the calculate the L2-normalization value of this feature array. The team with a higher L2-norm value is predicted to be the winner. The idea behind this model is that a team with better performance will have higher attribute values and hence should be able to win a match against a team with lower attribute values. The algorithm is as follows,

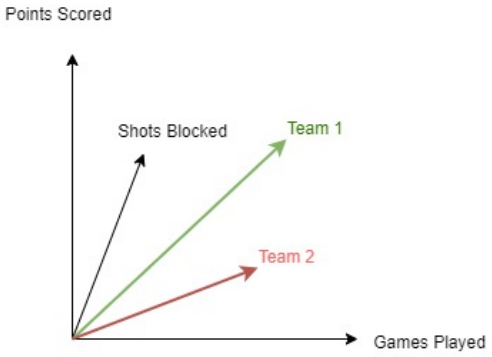

Figure 2: L2-norm comparison

- Create an array of all the features for each team.

$$T1 = [GP_1, Minutes_1, Points_1, Steal_1, ...] \tag{3}$$

$$T2 = [GP_2, Minutes_2, Points_2, Steal_2, ...] \tag{4}$$

- Calculate the L2-norm value for each team.

$$|T1| = \sqrt{GP_1^2 + Minutes_1^2 + Points_1^2 + Steal_1^2 + ...} \tag{5}$$

$$|T2| = \sqrt{GP_2^2 + Minutes_2^2 + Points_2^2 + Steal_2^2 + ...} \tag{6}$$

- The team with a higher L2-norm value is the winner.

$$Winner = max(|T1|, |T2|) \tag{7}$$

(3) Artificial Neural Network: An artificial neural network at its core has two modes of operations forward propagation and backward propagation.

- During forward propagation, input data is passed through the network, layer by layer, from the input layer to the output layer. Each neuron receives inputs from the previous layer, applies a weighted sum operation, adds a bias term, and passes the result through an activation function to produce an output. This process continues until the output layer produces the final prediction or output.
- Backpropagation is the process by which the network learns from the errors in its predictions. It involves calculating the gradients of the loss function with respect to the network's weights and biases. These gradients indicate how much each weight and bias contributed to the error, allowing the network to adjust its parameters to minimize the error. This process is typically performed using optimization algorithms such as gradient descent.

A major aspect of neural network training is adjusting hyper parameters like number of hidden layers, number of neurons in the layer, choice of activation function, learning rate, etc. To find out the best hyperparameters we will employ a grid search strategy where we will try a bunch of different combinations. We predict that artificial neural network should work the best among the other techniques

because as shown by research [4] [6] [3], it has a proven track record of providing good results in classifying tasks.

3 PLAN AND EXPERIMENT

3.1 Datasets

We have access to data spanning from 1946 to 2004 for both the NBA and ABA basketball leagues. This dataset encompasses various attributes of players across three different types of basketball matches within these leagues, which are as follows:

- Regular Season: The regular season in basketball consists of a series of scheduled games where teams compete to accumulate wins and secure a spot in the playoffs.
- Playoffs: The playoffs are a series of elimination games following the regular season, where the top teams from each conference compete in a tournament-style format to determine the league champion.
- All-Star Game: The All-Star game is an exhibition match held midway through the season, featuring the league's top players as selected by fan, player, and media voting, providing a showcase of talent and entertainment for fans.

3.2 Data Source and Description

3.2.1 NBA Statistics data, Carnegie Mellon University. [2]

(1) Players table - The master list of all players.
(2) Player regular season table: The stats of each player for each season.
(3) Player regular season career table: Cumulative stats of each player during their career.
(4) Player playoffs table: The stats of players who have played in playoffs
(5) Player playoffs career table: The cumulative stats of players who have played in playoffs.
(6) Player allstar table: The stats of players for all star matches
(7) Teams table: Master list of all teams in NBA and ABA
(8) Team season table: The stats of whole team over a season for each season
(9) Draft table: The list of players drafted
(10) Coaches season table: The stats of coaches for a season for all season
(11) Coaches career table: The stats of coaches aggregated for all season in their career.

3.2.2 NBA Dataset, Kaggle. [8] The dataset includes information on individual games spanning from 1946 to 2023. The data is organized at the team level and includes attributes such as attempted and made goals, assists, steals, and blocks. Each player, team, and coach is assigned a unique ID that is shared across all entities.

4 HYPOTHESIS

A team with higher sum of normalized statistics should always win the match regardless of other external factors. We want to find out with our proposed solution if this hypothesis always holds true or not. To verify this hypothesis we will train an artificial neural network on the dataset and ask the model to predict the winner for matches which it had not seen during training. We will also employ the L2 - Norm method as discussed above to further validate the result.

5 EXPERIMENTS

5.1 Data Pre-processing

Due to the lack of match-specific details in the NBA Statistics data, we aim to merge the Kaggle dataset, which includes match outcomes, with CMU's dataset, containing player statistics for those matches. The objective is to develop a model capable of establishing a relationship between a set of player statistics and the likelihood of winning by training on historical match data. Initially, the primary challenge lies in preprocessing the data to ensure its compatibility with the model. This includes aligning the Kaggle dataset with CMU's dataset, as they have varying naming conventions and attributes. Ensuring data authenticity involves verifying values from both datasets through cross-referencing for validation purposes.

Our objective is to predict the winner of a match given a list of players and their attributes. To train a model for such predictions, we require the attributes of each player who played a match and the outcome of that match. Unfortunately, we couldn't find a data source that could provide such data directly.

We explored the CMU dataset, which has all the attributes of players for each season but lacks match-specific data. On the other hand, the Kaggle dataset includes the list of players who played a match but lacks player attribute data. Thus, our first task was to map the player IDs from the Kaggle dataset to the CMU dataset.

5.1.1 Player Roster Extraction For Each Match. Although the Kaggle dataset contains match-specific data, it is not directly available. Kaggle's play-by-play data indicates each point scored, pass, block, or steal and the player who made the play. We parsed this data to identify the list of players for each team in a specific game and consolidated it into one table. However, even after this consolidation, one piece of information was missing: the winner of the match. To determine the winner, we joined the new table with another table from the Kaggle data that provided this information.

5.1.2 Player ID Mapping: Linking CMU and Kaggle Datasets. Our next task was to map the player IDs in the CMU data to the Kaggle data. This mapping was necessary because Kaggle had game-specific data, including the list of players present in the game but not their attributes, while the CMU data had player attributes but not the list of players present in a specific game. To perform this mapping, we joined the two-player datasets based on their first name, last name, and birth year. Initially, there were 3572 unique players in the CMU data and 4171 unique players in the Kaggle dataset. After the join, we were left with 2717 unique players, indicating that only these players had matching first names, last names, and birth years in both datasets. This suggests some discrepancies in the names of players or the absence of player data in either of the datasets.

5.1.3 Merging Game Data with Player Information. Using the game data created in step 1 and the player mapping created in step 2, we were able to map the game data with the players present in each game using the Kaggle player IDs. Before this mapping, we had data for around 20,000 games. After the mapping, the number

reduced to around 8,000 games because we were not able to map the IDs of many players.

5.1.4 Player Statistic Normalization and Averaging by Season. The player statistics data provides the overall performance of the player for each attribute. For example, if a player has played 80 games in a season, then the dataset has the sum of all the points scored by that player in those matches. Using this raw data for predictions could disadvantage players who have played fewer games, as their overall scores for points scored, blocks, and steals would be lower compared to those who have played more games. To address this, we normalized the data by dividing each attribute by the number of games played. For predicting the winner of a match given the player attributes, we needed to determine which season's attributes to consider. We had two options: consider the attributes of a player from the last season or take the average of the attributes from all the previous seasons. We chose to take the average of attributes from all the previous seasons and calculated this for every season.

5.1.5 Calculating team statistics. With the normalized and averaged attributes of all previous seasons for all players and the game data containing the list of players in each game and the game-winner, we merged this data and took the average of each player's attribute to get the attributes of the team. This data now included the attributes of two teams and the winner of each game, making it ready for training and predictions.

5.2 Feature Selection

As each table contains a wide range of attributes related to teams and players, a crucial step is to choose appropriate features for training the model. To accomplish this, we have employed Pearson Correlation. The correlation of each feature with every other feature and with the target variable is depicted in Figure 3. The heat map reveals that there is no linear correlation between the target variable "winner" and the features.

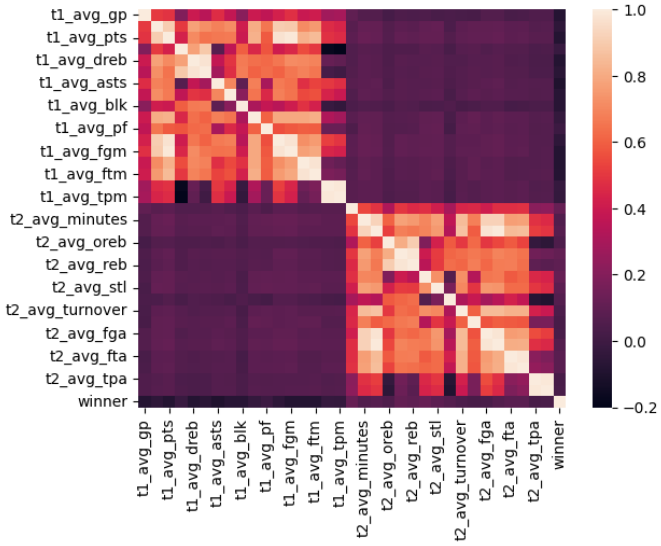

Figure 3: Feature Correlation

The correlation matrix suggests that the previously stated hypothesis may not hold true, as there appears to be minimal linear correlation between player statistics and match outcomes. However, artificial neural networks offer the potential to uncover nonlinear relationships between features. By employing ANNs in our analysis, we aim to gain a clearer understanding of the hypothesis and its validity.

After establishing the baseline model, a deeper analysis of NBA games revealed that certain features could be either omitted or amalgamated to generate a new feature that provides richer insights into player performance. For instance, the dataset contained separate features for defensive and offensive rebounds, as well as a total rebound feature representing their sum. Consequently, the total rebound feature was deemed redundant and removed. Similarly, individual counts of goals attempted and goals made were present. Instead of retaining these separate features, they were consolidated into a single feature representing the ratio of goals made to goals attempted. This restructuring enabled the creation of a more suitable dataset for training purposes.

5.3 Modelling

5.3.1 Baseline model - Naive Bayes. As per our proposal, we initially adopted Naive Bayes as the baseline model. However, the analysis of feature selection indicates no linear correlation between the features and the target variable. Consequently, we anticipate that the performance of the Naive Bayes model may be sub-optimal. To address this, we intend to utilize a neural network to capture any complex relationships that exist between the features.

For training the Naive Bayes model, we have data from 8777 games, with each game having 34 continuous features (17 for each team) and a discrete target variable that indicates the winning team with a binary value.

5.3.2 L2-Norm comparator. To predict the winner using the L2-norm comparator, we must ensure that a higher value for certain attributes corresponds to better player performance. For instance, a greater number of games played reflects a player's experience, which could enhance a team's chances of winning. Conversely, attributes such as the number of fouls received are better when lower. Therefore, we invert the values of features that indicate poor performance for higher values before feeding them into the model for prediction.

We have used 13 features for predicting the results of 8777 games all having continuous values.

5.3.3 Neural network. The first two models are effective only when there is a linear correlation between the attributes and the target variable. To capture the nonlinear relationship, we utilized a neural network. The neural network's input layer consists of 34 neurons, each representing one of the 34 features (17 from each team). The output layer has 1 neuron with a sigmoid activation function. If the sigmoid function's value exceeds 0.5, team 2 is predicted as the winner; otherwise, team 1 is predicted as the winner. We conducted a grid search to determine the optimal number of neurons, hidden layers, activation functions, and optimizers for the hidden layers. We experimented with 34, 68, and 102 neurons in hidden layers ranging from 1 to 4, using relu, tanh, and leaky relu activation

functions. Additionally, we tested three different optimizers - adam, rmsprop, and sgd. After evaluating these combinations of hyperparameters for 10 epochs each, we identified the best configuration as follows: a first layer with 68 neurons and relu activation, followed by a layer with 34 neurons and relu activation, and finally an output layer with 1 neuron and sigmoid activation.

Table 1: Sequential Model Summary

Layer (type)	Output Shape	Param #
Dense	(None, 68)	2380
Dense	(None, 34)	2346
Dense	(None, 1)	35
Total params:	4761 (18.60 KB)	
Trainable params:	4761 (18.60 KB)	
Non-trainable params:	0 (0.00 Byte)	

After finalizing the model, we experimented with different numbers of epochs to determine the optimal fit for the given data, refer fig.4 and fig.5

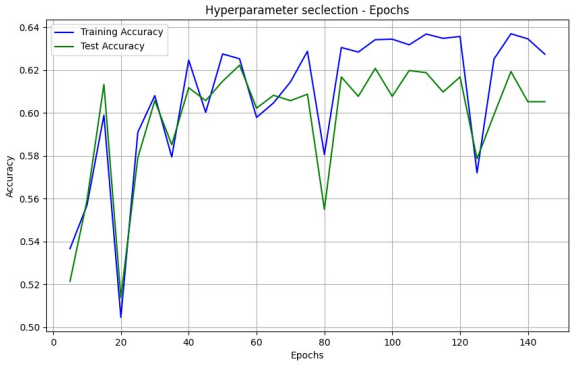

Figure 4: Accuracy for different epochs

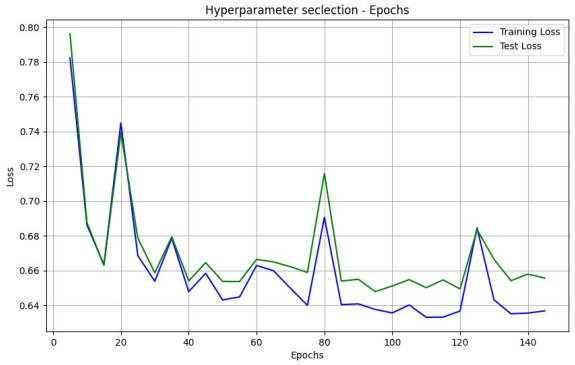

Figure 5: Loss for different epochs

After reviewing the results of hyper-parameter tuning, we decided to set the number of epochs to 40.

5.4 Results

5.4.1 Naive Bayes. We trained a Gaussian Naive Bayes model using 80% of matches as training data. We used the remaining 20% data for our analysis of the classifier. As anticipated, the model's performance was poor, achieving an accuracy of approximately 56%, which is comparable to random guessing.

Table 2: Confusion Matrix

	Predicted	
Actual	**Team 1 Win**	**Team 2 Win**
Team 1 Win	549	401
Team 2 Win	380	426

Table 3: Classification Report

Class	Precision	Recall	F1-Score	Support
Team 1 Win	0.59	0.58	0.58	950
Team 2 Win	0.52	0.53	0.52	806
Accuracy			0.56	1756
Macro Avg	0.55	0.55	0.55	1756
Weighted Avg	0.56	0.56	0.56	1756

Due to the many number of features in our dataset, Naive Bayes as predicted earlier performs pretty poorly for our data. But this gives us a good baseline model to build upon.

5.4.2 L2-Norm comparator. The L2-norm comparator does not necessitate training; it simply compares the L2-norm values of the features of two teams and predicts the winner to be the team with the higher value. There was a marginal increase in accuracy for this model as compared to Naive Bayes.

Table 4: Confusion Matrix

	Predicted	
Actual	**Team 1 Win**	**Team 2 Win**
Team 1 Win	2987	2307
Team 2 Win	1941	2719

5.4.3 Artificial Neural Network. We expected the neural network to perform exceptionally well compared to the other two models. However, this was not the case. After fitting the train data to the network defined in table no.1 we got the following results,

To check our hypothesis, we sampled 50 examples from the dataset where the statistics of team 1 were better than team 2 and we sampled 10 more examples where statistics of team 2 were better than team 1. We ran through the neural network and this was the result.

Table 5: Classification Report

Class	Precision	Recall	F1-Score	Support
Team 1 win	0.61	0.56	0.58	5294
Team 2 win	0.54	0.58	0.56	4660
Accuracy			0.57	9954
Macro Avg	0.57	0.57	0.57	9954
Weighted Avg	0.58	0.57	0.57	9954

Table 6: Confusion Matrix

	Predicted	
Actual	**Team 1 Win**	**Team 2 Win**
Team 1 Win	965	100
Team 2 Win	696	230

Table 7: Classification Report

Class	Precision	Recall	F1-Score	Support
Team 1 Win	0.58	0.91	0.71	1065
Team 2 Win	0.70	0.25	0.37	926
Accuracy			0.60	1991
Macro Avg	0.64	0.58	0.54	1991
Weighted Avg	0.63	0.60	0.55	1991

Table 8: Confusion Matrix where T1 has better stats

	Predicted	
Actual	**Team 1 Win**	**Team 2 Win**
Team 1 Win	15	25
Team 2 Win	5	5

6 DISCUSSION

6.1 Hypothesis

As stated earlier our hypothesis was that a team having a greater overall sum of statistic value should ideally win more matches. But this was not the case. As seen from the result of the L2 Norm classification 5 we only get predict 57% correctly using purely sum of statistics.

What is more interesting is the results of the neural network for this task. As seen from the results of tables 8 and 9, the neural network does a good job of predicting the winner in the case where team 2 had better stats. The accuracy coming out to be close to 80% in this case. For the other case, where Team 1 has better statistics, the neural network does this poorly. It only predicts correctly around 40% of time. This might give us an indication that the neural network is doing an overall good job of predicting team 2 wins when their stats are higher.

These results also gives us the bigger picture of the overall hypothesis. It seems that a team made entirely of statistically better players will not be always the winner of an NBA match. This highlights the unpredictability in the game. This also gives us an insight of the competitiveness in the league.

Table 9: Confusion Matrix where T2 has better stats

	Predicted	
Actual	**Team 1 Win**	**Team 2 Win**
Team 1 Win	10	5
Team 2 Win	5	30

6.2 Other inferences

As previously mentioned, we expected the neural network to outperform other models, but based on the results, we cannot definitively conclude that the neural network was clearly superior. We believe the following factors may have contributed to this outcome:

- Insufficient data volume: Neural networks typically excel with large datasets. Despite having approximately 9000 games in our training sample, the network may not have been able to effectively learn the features and predict match outcomes due to the relatively limited dataset size.
- Data discrepancies: Despite our efforts to verify the data, there could have been inconsistencies introduced by merging two datasets. While this scenario is unlikely, it remains a potential explanation for the model's suboptimal performance.
- Inherent unpredictability of NBA: NBA games are highly competitive and unpredictable. Human experts find it challenging to predict game outcomes, making this task equally difficult for machine learning models.

7 CONCLUSION

In this project, our goal was to develop an effective approach for predicting NBA game outcomes. We tackled challenges associated with merging datasets from disparate sources and utilized popular machine learning techniques such as Gaussian Naive Bayes and Artificial Neural Networks to construct a classification model. Our findings and insights presented in this paper establish a foundation for future research. While our results were not optimal, we derived meaningful conclusions from our observations. Moving forward, future studies can delve into uncovering hidden dataset features such as ball possession rates, substitution frequencies, and the impact of outlier players, among other potential areas of exploration. In conclusion, our project lays the groundwork for further investigation into nuanced features within NBA datasets, offering pathways for refining predictive models and enhancing our understanding of game dynamics.

REFERENCES

[1] Bruin Sports Analytics. [n. d.]. Decoding the Game: Forecasting NBA Champions with Neural Network Algorithms — bruinsportsanalytics.com. https://www.bruinsportsanalytics.com/post/nba-champs-neural-network. [Accessed 03-03-2024].

[2] School of Computer Science Carnegie Mellon University. [n. d.]. *NBA statistics data.* http://www.cs.cmu.edu/~awm/10701/project/databasebasketball2.0.zip

[3] B Dębska and B Guzowska-Świder. 2011. Application of artificial neural network in food classification. *Analytica Chimica Acta* 705, 1-2 (2011), 283–291.

[4] Stephan Dreiseitl and Lucila Ohno-Machado. 2002. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics* 35, 5-6 (2002), 352–359.
[5] Amr Hassan, Abdel-Rahman Akl, Ibrahim Hassan, and Caroline Sunderland. 2020. Predicting wins, losses and attributes' sensitivities in the soccer world cup 2018 using neural network analysis. *Sensors* 20, 11 (2020), 3213.
[6] Georgef Hepner, Thomas Logan, Niles Ritter, and Nevin Bryant. 1990. Artificial neural network classification using a minimal training set- Comparison to conventional supervised classification. *Photogrammetric Engineering and Remote Sensing* 56, 4 (1990), 469–473.
[7] James Joyce. 2003. Bayes' theorem. (2003).
[8] Kaggle. [n. d.]. *NBA dataset.* https://www.kaggle.com/datasets/nathanlauga/nba-games
[9] Xiaohu Tang, Zhifeng Liu, Taizhao Li, Wenbin Wu, and Zhenhua Wei. 2018. The application of decision tree in the prediction of winning team. In *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*. IEEE, 239–242.
[10] Bayya Yegnanarayana. 2009. *Artificial neural networks.* PHI Learning Pvt. Ltd.