Evaluating Existing Graph Convolutional Network Algorithms for NBA Season Predictions against Chalk Betting

Rishabh Chhabra and Nikhil Nair

5/19/2024

Yilmaz - Period 7

Semester 2 Project

Abstract

This study aims to enhance existing Graph Convolutional Network (GCN) algorithms for predicting NBA game outcomes. Our methodology incorporates feature extraction techniques such as Principal Component Analysis (PCA), LASSO, and Random Forest (RF) to refine model inputs. Our results demonstrate that GCNs, while an expansive way to model individual data within an NBA season are worse than standard machine learning models, like Decision Trees, SVMs, and even Linear Regressions at predicting game results with a large number of relevant statistics. To further assess the success of our GCN model, we conducted an additional experiment simply choosing the favorite team of each matchup given the end-of-season standings. This model resulted in a higher accuracy than that of our GCNs. These findings highlight the importance of model selection and the potential limitations of GCNs in this specific application

Introduction

Accurate prediction of NBA game outcomes is essential for teams, coaches, analysts, and bettors. These predictions can inform strategic decisions, enhance performance analysis, and improve betting accuracy. However, traditional machine learning models used for these predictions frequently fail to capture the intricate interactions between teams and the spatial-temporal aspects of the game. This study seeks to address this challenge by leveraging Graph Convolutional Networks (GCNs) to predict NBA game outcomes, with a particular focus on evaluating existing GCN algorithms against a chalk betting model, which picks the heavily favoured side.

The primary objective of this research is to utilize GCN algorithms to predict NBA game outcomes and compare the performance of these models with a traditional chalk betting approach. By recreating and expanding upon models from previous studies, we aim to provide a more accurate and adaptable predictive framework. Our approach involves integrating advanced feature extraction techniques to refine the input data, thereby improving the overall performance of the GCN model.

Our study utilizes two distinct datasets to achieve our objectives. The first dataset, comprising NBA game statistics from the 2012-2018 seasons sourced from Kaggle, includes comprehensive game statistics such as points per game, shooting percentages, rebounds, assists, and turnovers. This dataset is used to recreate the results from previous research, allowing us to validate our approach and ensure the robustness of our methods.The second dataset, also covering the 2012-2018 seasons, sorts teams by end-of-season records to construct a chalk betting model. We preprocess the data to ensure consistency and apply feature extraction techniques like Principal Component Analysis (PCA), LASSO, and Random Forest (RF) to identify the most significant predictors. By applying these methods, we can replicate and verify the effectiveness of the original GCN model.

Related Works

Early research in NBA game outcome prediction predominantly utilized traditional machine learning models. Loeffelholz et al. (2009) employed feed-forward neural networks (FFNN) to model the NBA 2007-2008 season, achieving a success rate of 74.33%. Similarly, Zdravevski and Kulakov (2009) used logistic regression to predict two consecutive NBA seasons, with a success rate of 72.78%. Miljkovic et al. (2010) applied naive Bayes to the NBA 2009-2010 season, resulting in a success rate of 67%. These studies demonstrated the potential of machine learning in sports predictions but also highlighted the limitations of traditional models in capturing the complex interactions between teams.

Graph Neural Networks (GNNs) have emerged as a powerful tool for handling structured data, where relationships between entities are crucial. Liu et al. (2019) applied GNNs to predict football game outcomes, achieving a significant improvement over traditional methods. This study demonstrated the potential of GNNs in sports analytics, particularly in capturing the relational aspects of the data. Similarly, Li et al. (2020) used GNNs to predict outcomes in NFL games, achieving significant reductions in test set losses. These studies underline the versatility of GNNs in sports predictions but also reveal the need for further improvements in accuracy and generalization.

Several researchers have applied machine learning models to predict NBA game outcomes, focusing on different seasons and datasets. Cao (2012) used simple logistic regression to predict NBA games, achieving a success rate of 69.67%. Tran (2016) applied matrix factorization to predict NBA outcomes, achieving an accuracy of 72.1%. These studies provide a foundation for NBA game prediction but often rely on complex models that may be less effective.

Dataset and Features

The first dataset we employed comprises NBA game statistics from the 2012-2018 seasons, sourced from Kaggle. This dataset includes comprehensive game statistics such as points per game, shooting percentages, rebounds, assists, and turnovers for each team. This dataset serves as the basis for recreating the results from previous research, allowing us to validate our approach and ensure the robustness of our methods.

Data preprocessing and feature selection were conducted meticulously to prepare this dataset for analysis. Initially, we removed any rows with missing values to ensure the dataset is complete and consistent. We then applied the StandardScaler from scikit-learn to normalize the data, ensuring that all features have a mean of zero and a standard deviation of one. This step is crucial for the performance of many machine learning algorithms. Additionally, we generated advanced metrics such as player efficiency rating (PER), true shooting percentage (TS%), and offensive and defensive ratings to provide more insights into game outcomes.

For feature selection, we employed three techniques: Principal Component Analysis (PCA), LASSO (Least Absolute Shrinkage and Selection Operator), and Random Forest (RF).

PCA was used to reduce the dimensionality of the dataset by transforming it into a set of orthogonal components that explain 95% of the variance. LASSO, with an alpha value of 0.1, applied a penalty to regression coefficients, effectively setting some of them to zero and selecting only the most significant features. RF evaluated feature importance based on their contribution to model accuracy, selecting the top features that contributed the most to the predictive power of the model, with a cutoff of 0.1.

For our own experiments, we focused on the end-of-season records of the 32 teams in the NBA to model “Chalk Betting.” As this dataset was created by us based off of official NBA records, there was no need for cleaning/preprocessing of the data because it was already optimized for our purposes.

Methods

5.1 GCN Architecture

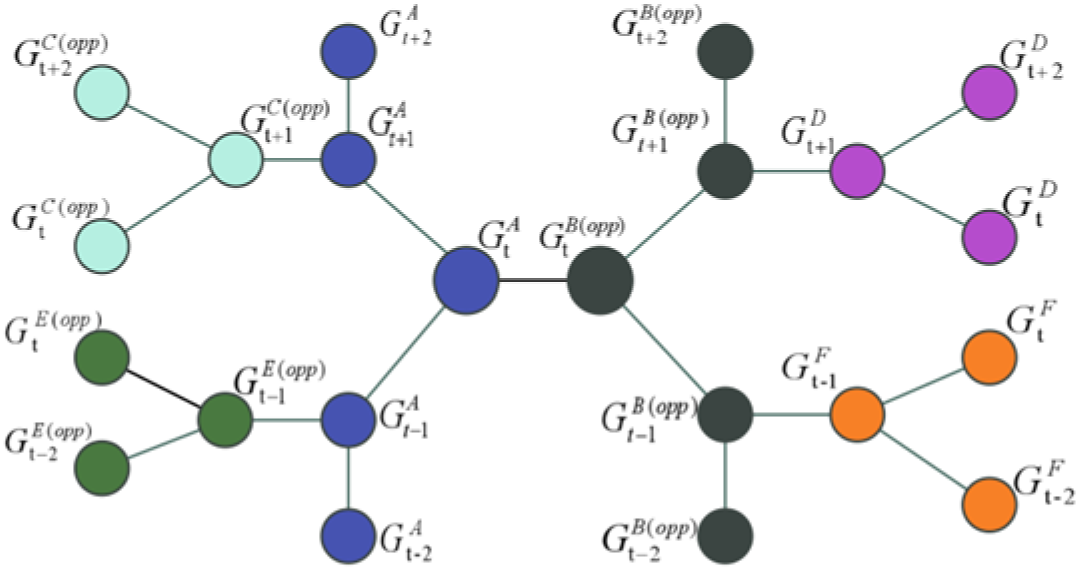
The architecture of a GCN typically involves several layers of graph convolution operations. Each layer aggregates information from a node’s neighbors to update its representation. Mathematically, the update rule for a GCN layer can be expressed as:

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)} \right)$$

where $h_i^{(l+1)}$ is the representation of node i at the layer $l+1$, $N(i)$ denotes the set of neighbors of node i, c_{ij} is a normalization constant, $W^{(l)}$ is the weight matrix of layer l, and σ is a non-linear activation function such as ReLU.

In our model, we use a multi-layer GCN where each layer is followed by a ReLU activation and dropout regularization to prevent overfitting. The final layer uses a softmax function to output probabilities for each possible game outcome.

5.2 Construction of Homogeneous Graphs



5.2.1 Nodes and Edges

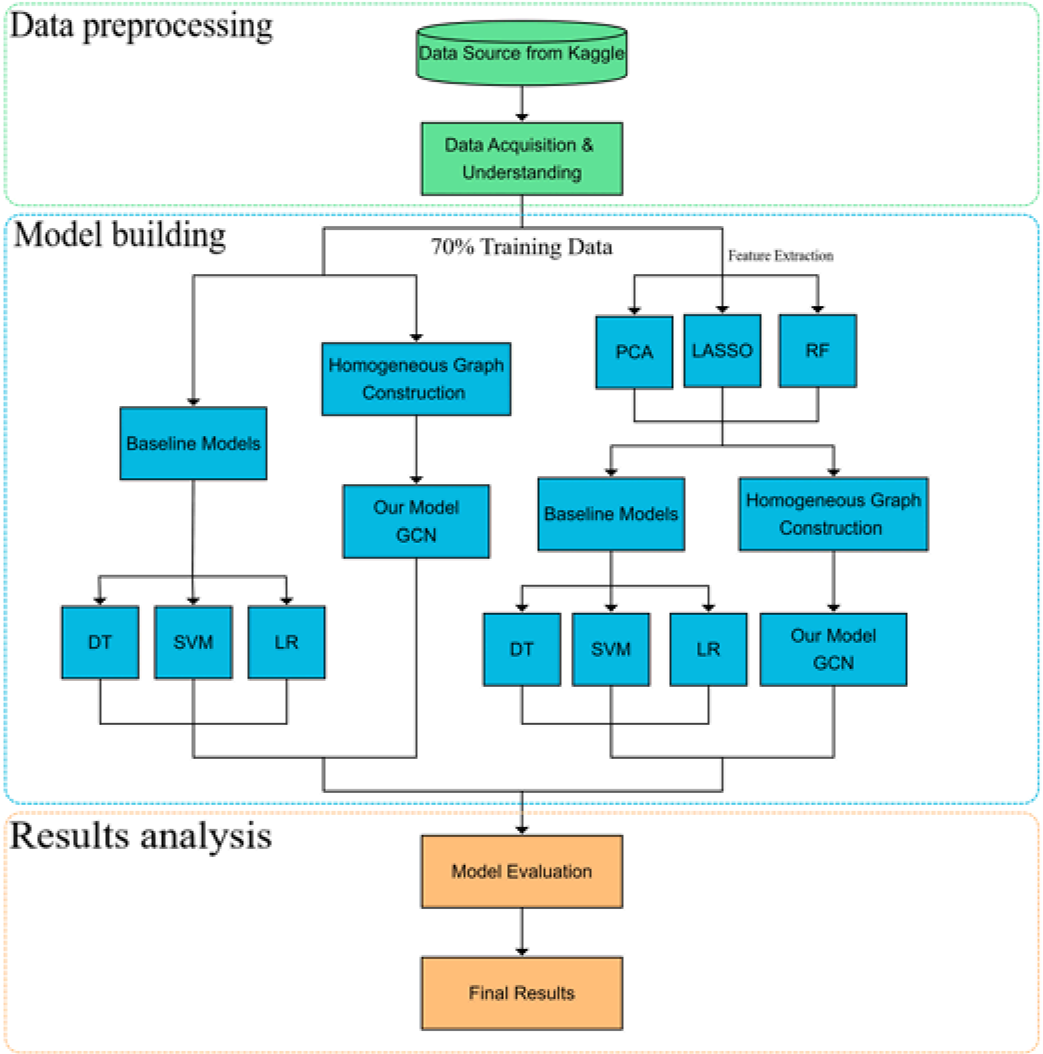
The nodes in our graph represent NBA teams. Each node is associated with a feature vector that encapsulates the team's statistics for a particular season, including points per game, shooting percentages, rebounds, assists, turnovers, player efficiency rating (PER), true shooting percentage (TS%), offensive rating, defensive rating, and other derived statistics.

Edges between nodes represent the games played between teams. Each edge carries attributes such as the outcome of the game, the date of the game, and the location (home or away). These attributes are used to enhance the feature vectors associated with each node.

5.2.2 Adjacency Matrix

The relationships between nodes are captured in an adjacency matrix A. The adjacency matrix is a square matrix where Aij is non-zero if there is an edge between nodes and j. In our case, Aij = 1 if team i played a game against team j, and Aij=0 otherwise. The adjacency matrix is symmetric because the games are undirected relationships between teams.

5.3 Flow of Experiment

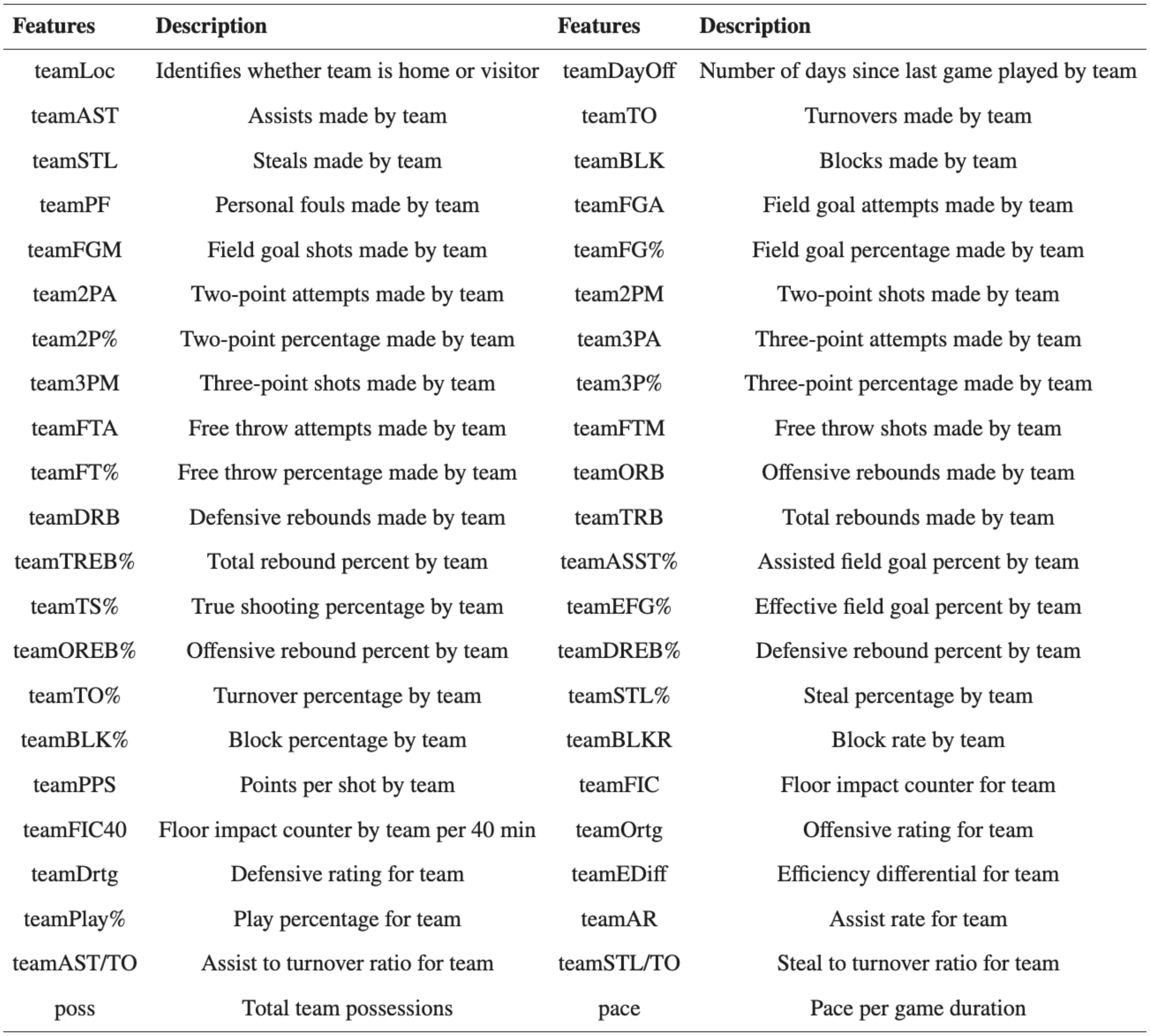


After splitting the data into a 70-20-10 training, testing, and validation set ratio, the data undergoes four individual feature selection processes (None, PCA, LASSO, RF). Within each of these feature selection processes, three baseline models are trained and tested (Decision Tree, Support Vector Machine, and Logarithmic Regression) whose results are compared to the GCN algorithm.

Experiments and Results

6.1 Dataset Breakdown

Our first step was to split the original dataset up by season. We first had to cut our dataset down to the 44 features as defined by the original paper. These are the features that were selected:



Features	Description	Features	Description
teamLoc	Identifies whether team is home or visitor	teamDayOff	Number of days since last game played by team
teamAST	Assists made by team	teamTO	Turnovers made by team
teamSTL	Steals made by team	teamBLK	Blocks made by team
teamPF	Personal fouls made by team	teamFGA	Field goal attempts made by team
teamFGM	Field goal shots made by team	teamFG%	Field goal percentage made by team
team2PA	Two-point attempts made by team	team2PM	Two-point shots made by team
team2P%	Two-point percentage made by team	team3PA	Three-point attempts made by team
team3PM	Three-point shots made by team	team3P%	Three-point percentage made by team
teamFTA	Free throw attempts made by team	teamFTM	Free throw shots made by team
teamFT%	Free throw percentage made by team	teamORB	Offensive rebounds made by team
teamDRB	Defensive rebounds made by team	teamTRB	Total rebounds made by team
teamTREB%	Total rebound percent by team	teamASST%	Assisted field goal percent by team
teamTS%	True shooting percentage by team	teamEFG%	Effective field goal percent by team
teamOREB%	Offensive rebound percent by team	teamDREB%	Defensive rebound percent by team
teamTO%	Turnover percentage by team	teamSTL%	Steal percentage by team
teamBLK%	Block percentage by team	teamBLKR	Block rate by team
teamPPS	Points per shot by team	teamFIC	Floor impact counter for team
teamFIC40	Floor impact counter by team per 40 min	teamOrtg	Offensive rating for team
teamDrtg	Defensive rating for team	teamEDiff	Efficiency differential for team
teamPlay%	Play percentage for team	teamAR	Assist rate for team
teamAST/TO	Assist to turnover ratio for team	teamSTL/TO	Steal to turnover ratio for team
poss	Total team possessions	pace	Pace per game duration

From there, we used a method to split the large dataset up into smaller datasets for each season. After that, each of the smaller datasets was divided into features and labels and stored in allBoxScores and allResults respectively.

6.2 Feature Selection

We then ran the three aforementioned feature selection strategies (PCA, LASSO, and Random Forest) to create new datasets for each season. The feature selection strategies were done on the 2013-14 dataset. For PCA, as specified in the other paper, we used a contribution rate of 0.95 and then selected the 7 best features which were:

teamASST%, teamEDiff, teamOREB%, teamSTL/TO, teamDREB%, team2PA, teamDrtg

For LASSO, we used an alpha of 0.1 and found our best features to be (9 in total):

teamPF, teamFGA, team3PA, teamFTA, teamDREB%, teamFIC, teamDrtg, teamEDiff, teamSTL/TO

We used a similar principle for Random Forest, but unlike the other two strategies, Random Forest was implemented as a classifier and then feature importances were calculated. As per the other paper, we used a threshold of 0.1 and found the two resulting features to be:

teamDrtg, teamEDiff

As we can see, teamDrtg and teamEDiff tend to be the best features for prediction.

6.3 Graph Construction and Model Setup

For our GCN, we needed to make the homogenous graphs for each individual season. We first created nodes using a custom "TeamNode" class that stores necessary information, such as team name, game date, team day off, and opponent name to connect nodes correctly, as well as the specific feature data for each node in a list titled additionalData. Per the paper's instructions, we connected each node with its previous game and its opponent node for that game, eventually constructing 6 graphs across each of the seasons. For all of our models, we used accuracy as our primary metric as per the other paper.

6.4 Results - No Feature Selection

Year	Decision Tree	SVM	Linear Regression	GCN Model
2012-13	1.000	0.983740	0.981030	0.760649
2013-14	1.000	0.978320	0.967480	0.548780
2014-15	1.000	0.981030	0.956640	0.520325
2015-16	1.000	0.982385	0.966125	0.487805
2016-17	1.000	0.972900	0.975610	0.520325
2017-18	1.000	0.987805	0.976965	0.485772
Average	1.000	0.981030	0.970641	0.553943

6.5 Results - Principal Component Analysis

Year	Decision Tree	SVM	Linear Regression	GCN Model

2012-13	1.000	0.995935	1.000	0.768763
2013-14	1.000	0.995935	0.991870	0.522358
2014-15	1.000	0.994580	0.990515	0.516260
2015-16	1.000	0.994580	1.000	0.487805
2016-17	1.000	0.993225	1.000	0.518293
2017-18	1.000	0.994580	0.998645	0.502033
Average	1.000	0.994806	0.996838	0.552585

6.6 Results - LASSO

Year	Decision Tree	SVM	Linear Regression	GCN Model
2012-13	1.000	0.990515	0.981030	0.784990
2013-14	1.000	0.979675	0.978320	0.489837
2014-15	1.000	0.982385	0.966125	0.497967
2015-16	1.000	0.982385	0.983740	0.506098
2016-17	1.000	0.976965	0.974255	0.481707
2017-18	1.000	0.987805	0.979675	0.500000
Average	1.000	0.983288	0.977191	0.543433

6.7 Results - Random Forest

Year	Decision Tree	SVM	Linear Regression	GCN Model
2012-13	1.000	0.998645	1.000	0.760649
2013-14	1.000	1.000	1.000	0.500000
2014-15	1.000	1.000	1.000	0.493902
2015-16	1.000	1.000	1.000	0.483740

2016-17	1.000	1.000	1.000	0.479675
2017-18	1.000	0.998645	1.000	0.489837
Average	1.000	0.999548	1.000	0.534634

6.8 Additional Experiment

Since our GCN model was fairly unsuccessful relative to our other models and the findings of the other paper, we wanted our experiment to look at the simplest statistical measure used to predict a matchup winner - the teams' rankings. In ranked tournaments, teams with a higher seed are usually favored to win a matchup against teams of a lower seed, so we looked at the end-of-season rankings for each team and used those as our only variable. If a team had a better total record than its opponent, our model predicts them to win. Below is an example list of standings from the 2012-13 season:

['MIA','OKC','SA','DEN','LAC','MEM','NY','IND','BKN','GS','CHI','LAL','HOU','ATL','UTA','BOS','DAL','MIL','PHI','TOR','POR','MIN','DET','WAS','SAC','NO','PHO','CLE','CHA','ORL']

As we can see, the 66-16 Miami Heat were the number one ranked team and our model would predict them to go undefeated. In comparison, the 20-62 Orlando Magic were the worst team in the league, and our model would predict them to go winless.

Here are our model accuracies for the six seasons tested previously.

Year	By Ranking
2012-13	0.685110
2013-14	0.678862
2014-15	0.702439
2015-16	0.704065
2016-17	0.647967
2017-18	0.676423
Average	0.682478

6.9 Discussion

Our analysis compared the performance of several models - Decision Tree, SVM, Linear Regression, and GCN - using different feature selection methods. Without feature selection, the Decision Tree consistently achieved perfect accuracy, outperforming SVM and Linear Regression, which also performed well but not as consistently. The GCN Model, however, struggled significantly, showing much lower accuracy. When feature selection methods like PCA, LASSO, and Random Forest were applied, most models saw improvements. Decision Tree maintained perfect accuracy across all methods. SVM and Linear Regression also showed high accuracy, particularly with PCA and Random Forest, indicating these methods effectively reduced noise and enhanced prediction accuracy. The GCN Model showed slight improvements but continued to lag behind the other models, suggesting persistent challenges in its predictive capabilities. An additional experiment using team rankings as the sole variable showed lower accuracy compared to the sophisticated models but still provided a reasonable baseline, and was better than that of the GCN model. This highlights that while advanced models generally offer higher accuracy, simpler statistical measures can also yield valuable predictions. Overall, the Decision Tree emerged as the most robust model, with SVM and Linear Regression also performing strongly, while the GCN Model requires further refinement.

Conclusion

In our study, we aimed to enhance existing Graph Convolutional Network (GCN) algorithms for predicting NBA game outcomes. By recreating and expanding upon models from prior research, we applied our methods to both details and summative datasets to assess performance shifts and accuracy improvements. Our methodology incorporated feature extraction techniques such as Principal Component Analysis (PCA), LASSO, and Random Forest (RF) to refine model inputs.

Our results demonstrated that GCNs, while an expansive way to model individual data within an NBA season are less effective than standard machine learning models, such as Decision Trees, SVMs, and Linear Regressions, at predicting game results with a large number of relevant statistics. The findings suggest that traditional machine learning models, with appropriate feature selection and extraction techniques, consistently outperformed GCNs in terms of prediction accuracy. This indicates that while GCNs offer a novel approach to modeling sports data, the added complexity does not necessarily equate to better performance for this particular task. Moreover, the ease of implementation and interpretability of models like Decision Trees and SVMs make them attractive alternatives for similar predictive tasks.

Our experiment compared our GCN model to a layman's "Chalk Betting" strategy. Our results indicated that on average, betting with the favoured odds had a better accuracy at predicting the winning team when compared to our GCN model.

Future work should focus on further refining GCN models to better capture the nuances of basketball games. This could involve exploring hybrid models that combine the strengths of GCNs with traditional machine-learning approaches or investigating alternative graph-based methodologies. Additionally, it would be beneficial to examine the impact of different feature engineering techniques and incorporate more sophisticated temporal dynamics to improve predictive performance.

Contributions

Throughout the research process, work was divided evenly depending on both project member's strong suits. Nikhil implemented the feature selection and the three baseline models. Rishabh then applied these methods to the complex GCN. The paper was also written equitably as both partners picked up sections they were willing to write, and then the other would proofread and add information they felt was missing.

References

Zhao, K., Du, C., & Tan, G. (2023). Enhancing Basketball Game Outcome Prediction through Fused Graph Convolutional Networks and Random Forest Algorithm. Entropy (Basel, Switzerland), 25(5), 765. https://doi.org/10.3390/e25050765
https://www.mdpi.com/1099-4300/25/5/765

Loeffelholz, B., Bednar, E., & Bauer, K. W. (2009). Predicting NBA games using neural networks. Journal of Quantitative Analysis in Sports, 5(1).
https://www.degruyter.com/document/doi/10.2202/1559-0410.1156/html

Zdravevski, E., & Kulakov, A. (2009). System for prediction of the winner in a sports game. Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing (CompSysTech'09), 1-6.
https://www.researchgate.net/publication/226597761_System_for_Prediction_of_the_Winner_in_a_Sports_Game

Miljkovic, D., Gajic, L., Gajic, D., & Kovačević, A. (2010). The use of data mining for basketball matches outcomes prediction. 2010 International Symposium on Intelligent Systems and Informatics.
https://www.researchgate.net/publication/261501109_The_use_of_data_mining_for_basketball_matches_outcomes_prediction

Liu, Z., Yang, W., & Wu, X. (2019). Football match result prediction with Graph Neural Networks. arXiv preprint arXiv:1912.01755. https://arxiv.org/pdf/2207.14124

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2020). Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. Proceedings of the International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1707.01926

Chen, Y., Zhao, X., & Zong, W. (2021). Traffic Flow Prediction with Feature Selection and Graph Convolutional Networks. IEEE Transactions on Intelligent Transportation Systems. https://www.researchgate.net/publication/346115754_Traffic_Flow_Prediction_Based_on_Deep_Learning_in_Internet_of_Vehicles

Cao, C. (2012). Sports data mining technology used in basketball outcome prediction. Applied Mechanics and Materials, 263-266, 1937-1941.
https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1040&context=scschcomdis

Tran, D. (2016). Matrix factorization-based approaches for predicting basketball outcomes. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
https://dspace.mit.edu/bitstream/handle/1721.1/106385/967660262-MIT.pdf?sequence=1