Machine Learning in Predicting NBA Team Win Rates

Nicholas Low
School of Engineering and Computer Science
University of the Pacific
Stockton, California

Abstract— **Predicting NBA teams' win rates is growing in popularity due to the new commissioner, Adam Silver, being an advocate of sports betting. A method of predicting win rates is by utilizing machine learning. There are various algorithms that can get a prediction; by comparing two algorithms: neural networks and linear regression, a method is evaluated to be the better model in predicting NBA win rates. This model was evaluated based on a dataset of 445 samples that included 18 features. The model best for predicting NBA wins is evaluated to be a neural network with two hidden layers.**

Keywords—basketball; machine learning; predicting; win rates

I. Introduction

With sports betting on the verge of becoming legal, the importance of being able to predict stats in the NBA (National Basketball Association) is becoming increasingly important. One of the most important stats to know when it comes to sports betting is a team's win rate. By knowing the team's win rate, people can determine whether an individual NBA player will have a solid win rate or have a statistically great season. These statistics are related as the basketball is arguably one of the most team oriented sports. Although there are many star players that make the headlines frequently, it has been proven constantly that a player cannot win consistently without a great team. A modern comparison of this statement is the difference between the Cleveland Cavaliers and the San Antonio Spurs. The Cleveland Cavaliers have a better roster that boasts the current best player in the league, Lebron James, superstars Kevin Love and Kyrie Irving, and a candidate for sixth man of the year, Dion Waiters. However, the Cleveland Cavaliers have barely begun to rise above .500 in the 2014-2015 season. Although the Cavaliers are expected to do better as their teamwork solidifies, the San Antonio Spurs are already proving to be the better team with a .750 win rate and a championship from last season. This paper seeks to present an evaluation of a couple machine learning algorithms that have the ability to predict NBA team win rates. These methods include: neural networks and linear regression.

This paper will be divided into three major sections. The first section will discuss related works and what methods or ideas can be gained from the article. The second section will discuss the methodology which will include the problem statement and the solution to this problem. Finally, the last major section will discuss the results and other important thoughts that arise due to the results of the system.

II. Related Work

There have been quite a few articles that deal with machine learning or other methods of predicting in sports. Predominantly, American football is the sport that is studied and evaluated the most. There are a couple major factors that contribute to the lack of predicting in the NBA. First, the amount of samples is too low to predict accurately. The NBA rules have changed frequently over the past decades and as a result, the statistics gained are inaccurate since they no longer represent the current rules of basketball. Second, the need for predictions is more prevalent in American football than in the NBA as football is more popular in large sports betting areas.

A common method of predicting in the NBA is through the use of the Naïve Bayes classification algorithm [5]. This classification algorithm is largely used to make the determination of whether a team will win or lose a match. According to article [5], the prediction model would predict the outcome correctly approximately 67% of the time. This prediction model was based on 778 samples that came from 2009-2010 season. This method also involved using a multivariable linear regression algorithm to determine the point differential that would occur. This article found its inspiration from NFL prediction that used Naïve Bayes classification to determine whether an NFL team would win or not.

Another method of predicting data in the NBA is by using neural networks. Neural networks have been used to predict large sports data sets in football constantly and seem to be commonly the best at predicting data for win rates. In the articles [2] and [3], neural networks are largely used to determine individual player worth and analyze a single basketball game where there is no classification. Therefore, the results are given in numerical values where the individual player is given a number that demonstrates their worth compared to another player.

Other methods that are common in the NBA are largely fuzzification or clustering algorithms to determine different styles in the NBA. These methods also take in data from video clips of different sports and make conclusions based on this data. These conclusions were solely focused on what kind of sport is being played in the video. Another popular focus with fuzzification is to determine player worth for NBA scouting. This system serves to be an assist tool in a NBA scouts job of determining whether players are worth the time, effort, and money an NBA team might be willing to give to develop a player.

III. Methodology

This section of the paper will focus on describing how the created system is going to work. This involves presenting the problem, determining the solution, and then implementing the solution.

A. *Problem Statement and Solution*

The problem intended to be solved by the system designed in this paper is to determine the best model for predicting NBA teams win rates based on their in-season team statistics. These statistics will be:

Table I:
Dataset Features

18 Features (Statistics)	
Features	***Definitions***
MOV	Average Margin of Victory
SOS	Strength of Schedule
ORtg	Offensive Rating
DRtg	Defensive Rating
Pace	Offensive Pace
Ftr	Free Throw Rate
3PAr	3 Point Attempt Rate
TS%	True Shooting %
Attendance	Attendance at Home Stadium
OeFG%	Offensive Effective Field Goal %
OTOV%	Offensive Turnover %
ORB%	Offensive Rebounding %
OFT/FGA	Offensive Free Throw per Field Goal Attempt
DeFG%	Defensive Effective Field Goal %
DTOV%	Defensive Turnover %
DRB%	Defensive Rebounding %
DFT/FGA	Defensive Free Throw per Field Goal Attempt

These statistics are regularly used in the NBA to note a team's play style in numbers. These numbers also attempt to give more weight to stats that give more points such as making a three-point shot.

As the results from the model will be an actual numerical determination of a win rate ranging from 0-100%, classification is not an appropriate tool. Therefore, the system will not take into consideration the Naïve Bayes classification algorithm that has been popular in related work. Aside from the results, the system needs to determine the best model. The system will be based on two of the more common models from other articles: neural networks and linear regression. As there can be some discrepancies based on the number of hidden neurons in a neural network, the system will take 2-4 hidden neurons and test those three along with a linear regression model. An example of a neural network can be seen in Figure 1.

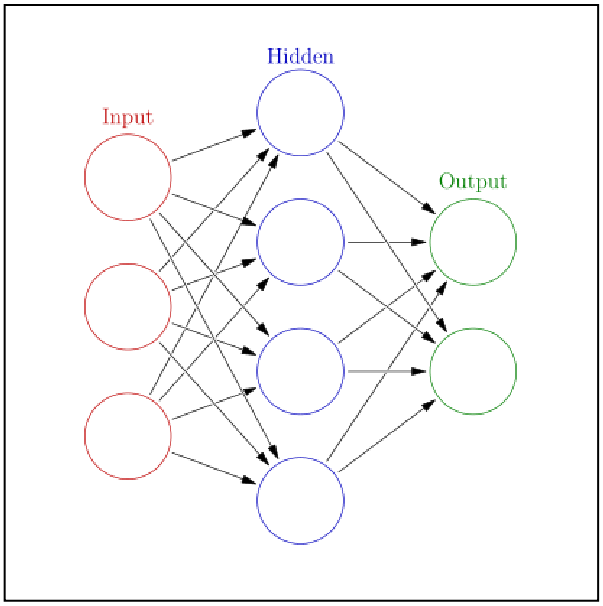

Figure 1: Neural Network Example

In order to completely test a dataset out, the system will need to determine its training points, validation points, and test points. The total amount of samples in the dataset is equivalent to 445 which is the 1999-2000 season to the 2013-2014 season. This particular data set was selected due to rule changes. The 1998-1999 season was the last season to have major rule changes specifically to how the three point line works. During the 1999-2000 season hand-checking was implemented; however, this changed how physical the game of basketball is played which cannot be described by the statistics. As a result, any data before the 1999-2000 season was not used. The system will take 70% of the total 445 samples and use them as training points. This number is a common percentage of data samples taken as training points. However, issues will arise with training if the system has too many training points. By determining how many weights the neural network would create, the upper bound was found to be 390 samples, 70% of 445 or 311 samples are under the upper bound. The training will be trained based on a set of target values which are the corresponding win rates for each team over all the samples. Table 2 gives an example of what the data for all the target values will look like.

TABLE II:
Target Function Example

Team	Win Rate
Los Angeles Lakers	0.765
Golden State Warriors	0.685
Boston Celtics	0.732
San Antonio Spurs	0.700
Toronto Raptors	0.532
Sacramento Kings	0.492
Oklahoma City Thunder	0.632
Miami Heat	0.542
Charlotte Hornets	0.329
Brooklyn Nets	0.478
New York Knicks	0.321
Memphis Grizzlies	0.476
Houston Rockets	0.598
Milwaukee Bucks	0.268
Denver Nuggets	0.489

The remaining 30% will be used for validation and test points. The validation set is used to determine at what epoch is the best model for neural network. By plugging the validation set while the training is occurring, the validation error is determined; when the validation error increases over six times, the training stops. By comparing validation errors, the system can approximate which model is the best model. The validation error curve can be seen in Figure 2.

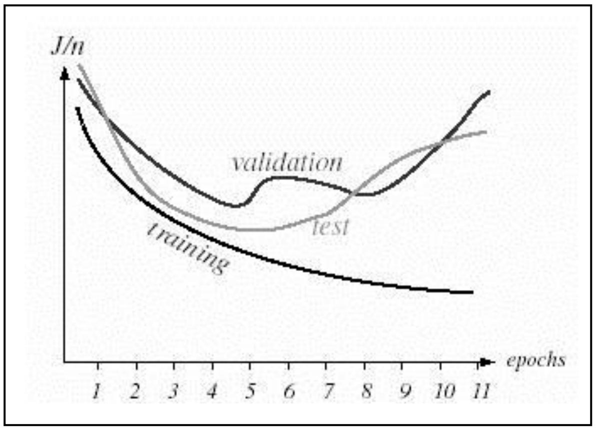

Figure 2: Performance Curve Example

The test sample set will be used to determine how accurate the model will be with points that were not used to train the data. This sample set will consist of 15% of the total data set. The linear regression model will be implemented in a similar method to provide as much accuracy as possible. After dividing the dataset into the same amount of training points, validation points, and test points as the neural networks; the linear regression algorithm will be used to train the data set and determine what the average validation error will be. By comparing the four models, the system will prove that the selected model is at least one of the better choices. An example of a linear regression product is shown in Figure 4.

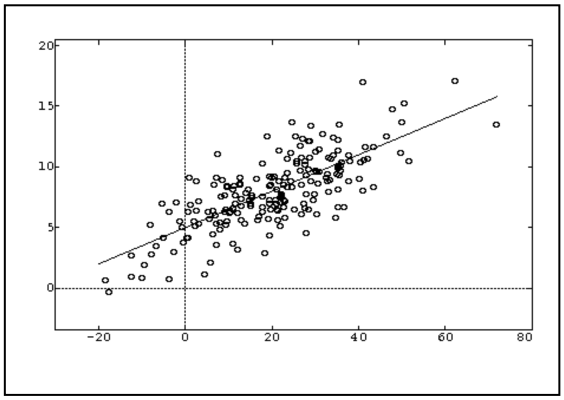

Figure 3: Linear Regression Example

After selecting the best model, the system will use the model to create a prediction as to what the 2014-2015 season win rates could be.

B. *Software Implementation*

To implement the solution to the problem on software, the system utilized MATLAB. MATLAB is a useful tool when applying machine learning techniques as it provides graphs and ease of access to results for the user to analyze.

The MATLAB code breaks the system down into six major steps. The first step is to initialize and grab the data set. By using the xlsread function, MATLAB can access the datasets and pull them into a matrix whose name is specified by the user. In the case of this system, X is the data set and Y is the target values. The size of X is 18x445 and the size of Y is 1x445. After grabbing the data, the system sets the first neural network up by calling fitnet(2). This allows the code to set up a neural network with two hidden neurons as depicted in Figure 4.

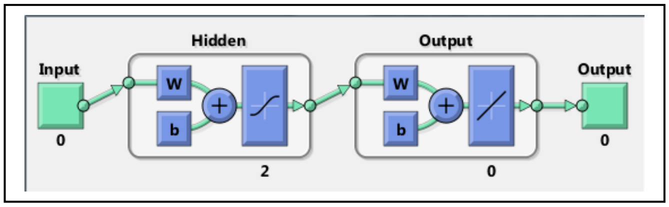

Figure 4: Neural Network with Set Hidden Neurons

After dividing the data set and target values into a training set, a validation set, and a test set; the code trains the data set and updates a set of weights until the stop condition based on the validation error occurs. In Figure 5, the stop condition occurs have thirteen epochs have occurred. The reason is that after seven epochs the validation error increased six times in a row.

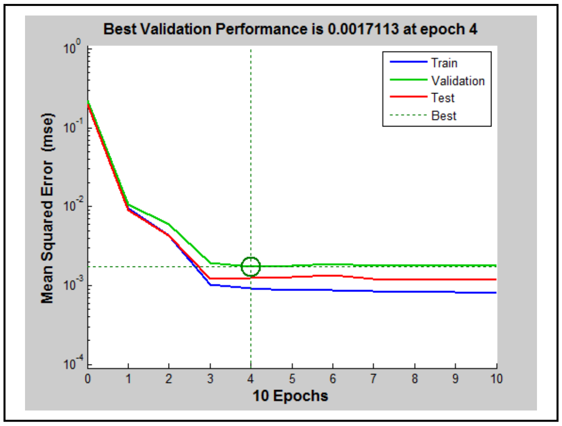

Figure 5: Performance Curve

The neural net toolbox automatically randomizes weights, training samples, validation samples, and test samples when the neural network is called. After taking 100 iterations of the training, the system takes the average of all the minimum validation errors. This same process is used to set up and utilize the neural networks with three hidden neurons and the neural network with four hidden neurons.

The linear regression model required more code to implement. MATLAB does not have the ability to automate randomizing weights, training samples, validation samples, and test samples for linear regression. Therefore, for each set of samples the system randomizes a set of indexes which are used to determine which of the 445 samples are used for each set of data. After determining the indexes that will be used for training, the system removes the indexes from the total set and randomizes another set of indexes based on the leftover indexes. These indexes would then be used to determine the validation set.

The same process would be repeated to determine the testing set. After determining these sets, the system uses the training set with its corresponding target values to train the model. This is done by using the linear regression algorithm which involves taking the pseudo-inverse of the input X to find the weights. Since the linear regression algorithm is inherently different than that of a neural network, where neural networks involve multiple iterations and a stop condition, the validation set is mainly used to determine whether the model should utilize the weights or not. If the validation error is larger than a previous iteration, the system will not take into consideration the weights as they will be more inaccurate. The system averages these validation errors after 100 iterations of training to determine the approximate worth of the model.

With the average validation error for the four models, the system needs to determine which model is the best for predicting NBA win rates. The code compares the average validation errors received from each of the models to determine which has the least validation error. The model associated with the least amount of validation error is then set to a variable that is printed out to show the user which model is the best. The names of the model will be: 'NN2', 'NN3', 'NN4', and 'LR'. Respectively, these represent the neural networks with two, three, and four hidden neurons along with the linear regression model. To further prove that the selected model is correct, the code uses the randomly generated test sets over 100 iterations to show which has the best average test error.

C. *Limitations*

During the design of this system, there were several limitations that were found. The first limitation was that getting a sample size that was great enough to test neural networks was very difficult. The reason the sample size is difficult to increase is because unlike college basketball, there is a lot less teams. Currently there are only 30 teams in the league. As the system had eighteen features, the neural network required approximately 390 samples. Although this value is an upper bound, it serves as a method to determining the correct amount of hidden neurons in the system. Another limitation that was found while determining how the system was going to work is that there were only 29 teams ten years ago. This reduces the amount of samples by a small amount more. Due to the inability to gain larger sample sizes, the neural network model could not be more complex which may or may not lead to more accurate results.

Another major limitation that the design of the system has is that the inputs of the system are real-time data. Every game that is played during an NBA season adds new input data. In the early season is when predictions are most important. As a result, the actual accuracy of the prediction model can only be determined after this season ends.

IV. Discussion/Results

The results of the system are that a neural network with two hidden neurons is best suited to predict NBA win rates as of the 2014-2015 NBA season.

A. *Neural Networks*

After implementing the three neural networks, the system outputted the results. With the neural networks there were a few important outputs to take note of. An example of an error histogram depicted in Figure 7, seems to have more errors deviating from zero than there should be. This may be due to a lack of a larger amount of samples.

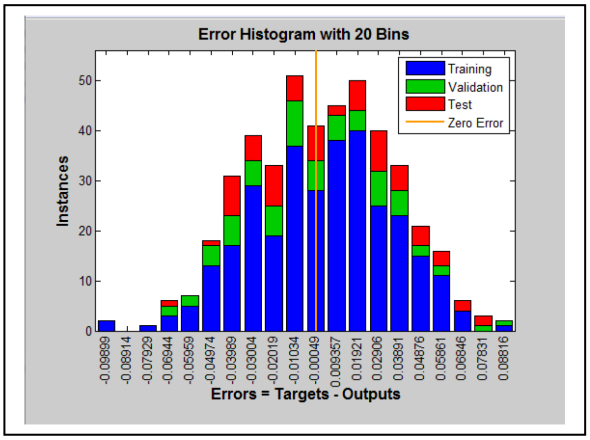

Figure 7: Error Histogram Results

If the test errors for each neural network are more closely inspected in Figures 8, 9, and 10; there is a fair amount of fluctuation. However, the fluctuations are very minimal. The result of these graphs shows that the neural network with two hidden neurons has the least amount of error consistently.

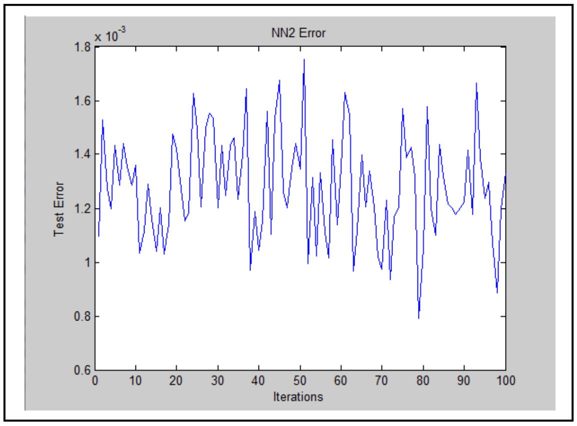

Figure 8: NN2 Test Error

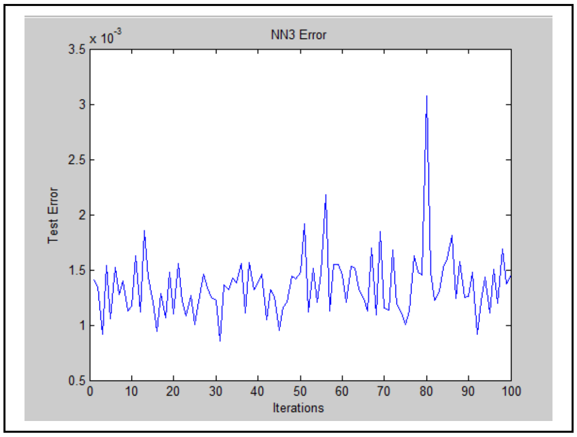

Figure 9: NN3 Test Error

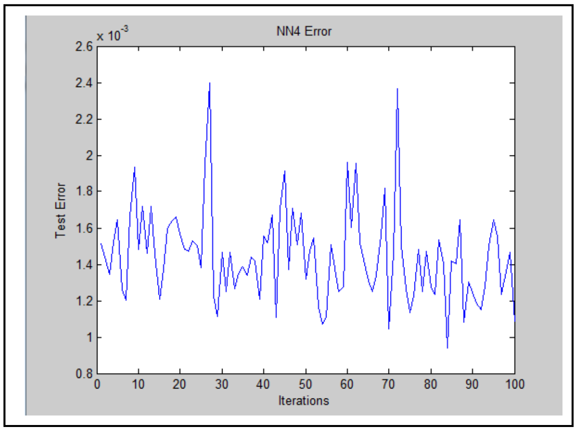

Figure 10: NN4 Test Error

In Table 3, the average test errors for the neural networks are shown. The errors are extremely low which supplies evidence on how the prediction is accurate only to a certain extent.

TABLE III:
Target Function Example

Neural Network	Average Test Error
Two Hidden Neurons	0.0013
Three Hidden Neurons	0.0016
Four Hidden Neurons	0.0016

An extremely important note is that these errors represent one possibility due to the limitations of the data received. Over the course of an NBA season, the statistics stabilize more and more creating more accurate results. What the current data in Table 3 represents is the accuracy of the results if the team's statistics stay similar to what was inputted into the system. Therefore, this prediction model can be very useful for situations like betting on playoff seeds. In these types of situations where prediction is required later in the season, the model can predict playoff seeds with a very small error.

Taking the situational usage of this prediction into account, the neural network with two hidden neurons is the most accurate. The other neural networks have more error most likely due to over fitting. According to a test for over fitting, if the difference between the two test errors is a negative number, .0013 - .0016, then over fitting has occurred. The reason for over-fitting is that the model's complexity is too high for the given amount of samples and features inputted into the system.

B. *Linear Regression*

The results for the linear regression can be seen in Table 4. The average test error is approximately 3%. Figure 11 shows the test errors over 100 iterations. This is most likely due to the simplicity of linear regression. More specifically, it does not optimize its weights and has no way of determining the best weights when making a model. The weights are determined by a single equation and make no corrections.

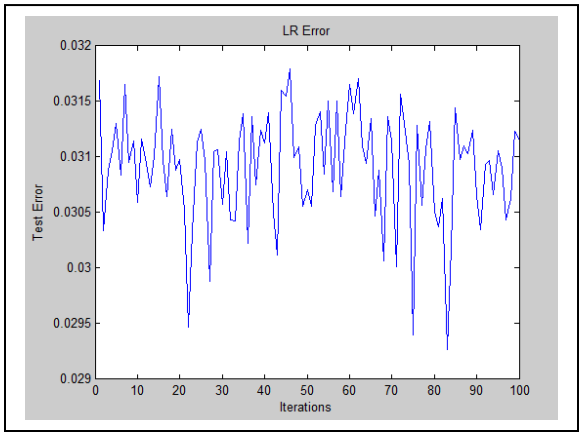

Figure 11: Linear Regression Test Error

TABLE IV:
Target Function Example

Neural Network	Average Test Error
Linear Regression	0.0308

C. *Comparison between Models*

The neural networks are significantly better at getting more accurate results than linear regression. A comparison between Table 3 and 4 shows the average test errors that were received from the system is heavily in the favor of neural networks. These results demonstrate that win rates are not necessarily linear and require a small amount of complexity. Specifically, the code always determines that a two hidden neuron neural network is best suited for predicting NBA team win rates. In Table 5, there is a small comparison of what the test outputs were. The neural networks are much closer to the actual values compared to linear regression.

TABLE V:
Comparison of Algorithms

Teams	Actual	NN2	NN3	NN4	LR
San Antonio Spurs	0.716	0.717	0.719	0.699	0.745
Los Angeles Clippers	0.710	0.712	0.704	0.701	0.716.
Oklahoma City Thunder	0.699	0.700	0.704	0.693	0.689
Houston Rockets	0.649	0.652	0.641	0.632	0.649

The system consistently determines that a neural network with two hidden neurons is the best model for predicting NBA win rates. As a result, the prediction that the system outputs for the 2014-2015 season is produced and the results are shown in Table 6. These results are very theoretical as there is no current way to test them. Also, if the inputs are adjusted to be what they will become in the middle of season, the win rates will become a lot closer to the actual values. Since this system was created in the beginnings of the 2014-2015 season there is bound to be errors compared to the win rates that will be seen near playoff time.

A system that attempts to predict perfectly how often a team will win in the NBA will always be impossible considering the amount of significant changes that occur in the NBA every season. These significant changes that can include major free agents coming to a team can complete change how the team plays. If a team completely changes, the prediction will be off. There can be a multitude of other factors that can alter predictions based on the emotions of each player and the drama occurring within the league. All these characteristics of the NBA make the association extremely entertaining for the world to view.

D. *Future Work*

Making predictions on a system that is extremely volatile requires a constant commitment. For future work, compiling more data to be able to make more complex models will be extremely important. Determining features that can give numerical value to other important factors to NBA team play and adding them to the system will also be necessary to create a more accurate model.

The system could also use a few added results. Being able to determine what stats are more important to winning basketball games will help expand on how this model can be used in the NBA infrastructure as coaches, players, and owners would be able to see how they can improve.

TABLE VI:
2014-2015 Season Predictions

Teams	Win Rates
San Antonio Spurs	0.752036845
Los Angeles Clippers	0.747678515
Oklahoma City Thunder	0.707070287
Golden State Warriors	0.736567051
Houston Rockets	0.707322384
Portland Trailblazers	0.665659652
Miami Heat	0.526974201
Indiana Pacers	0.625716673
Minnesota Timberwolves	0.589446745
Phoenix Suns	0.5560464
Dallas Mavericks	0.611396834
Toronto Raptors	0.515468493
Memphis Grizzlies	0.472959681
Chicago Bulls	0.533371255
Washington Wizards	0.511216572
Atlanta Hawks	0.465753468
Charlotte Bobcats	0.37724341
Denver Nuggets	0.383312142
New York Knicks	0.441248936
Brooklyn Nets	0.401540735
New Orleans Pelicans	0.444540906
Sacramento Kings	0.313391379
Cleveland Cavaliers	0.415678919
Detroit Pistons	0.293308371
Boston Celtics	0.226357319
Los Angeles Lakers	0.271400199
Orlando Magic	0.269510144
Utah Jazz	0.283973374
Milwaukee Bucks	0.212750328
Philadelphia 76ers	0.159515759

V. Conclusion

As the new commissioner, Adam Silver, attempts to legalize sports betting for the NBA, the entertainment value of the NBA will increase. Therefore, more people will be watching the games and more people will be betting on the games. Being able to predict how NBA teams will play and how much they win will be an ability that many people will want to have. Currently, a multi-variable neural network with two hidden neurons is the best at predicting the win rates. However, the NBA has just started to have enough data to predict accurately. In the next few decades, as the NBA attempts to become more entertaining and increase its viewer count, the amount of samples people can use will increase. This will allow for more complex models that may predict win rates better. It is important to keep in mind that a system that attempts to perfectly predict how often a team will win in the NBA will always be impossible considering the amount of significant changes that occur in the NBA every season. These significant changes that can include major free agents coming to a team can complete change how the team plays. If a team completely changes, the prediction will be off. This is what makes the NBA so entertaining to watch.

References

[1] K. Wheeler, Predicting NBA player performance[Online]. Available: http://cs229.stanford.edu/proj2012/Wheeler-PredictingNBAPlayerPerformance.pdf\

[2] Ivanković, Z., Racković, M., Markoski, B., Radosav, D., & Ivković, M. (2010). Appliance of neural networks in basketball scouting. *Acta Polytechnica Hungarica*, *7*(4).

[3] Ivankovic, Z., Rackovic, M., Markoski, B., Radosav, D., & Ivkovic, M. (2010, November). Analysis of basketball games using neural networks. In*Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on* (pp. 251-256). IEEE.

[4] Pena, J.M.; Menasalvas, E.; Muelas, S.; LaTorre, A.; Pena, L.; Ossowski, S., "Soft computing for content generation: Trading market in a basketball management video game," *Computational Intelligence in Games (CIG), 2013 IEEE Conference on* , vol., no., pp.1,8, 11-13 Aug. 2013

[5] Miljković, D.; Gajić, L.; Kovacevic, A.; Konjović, Z., "The use of data mining for basketball matches outcomes prediction," *Intelligent Systems and Informatics (SISY), 2010 8th International Symposium on* , vol., no., pp.309,312, 10-11 Sept. 2010