Evaluating Machine Learning Varieties for NBA Players′ Winning Contribution

Po-Han Hsu, Sainzaya Galsanbadam, Jr-Syu Yang and Chan-Yun Yang, *Member, IEEE*

***Abstract* — The NBA brand has been spreading all over the world in recent years. As the league concerns a lot of money and fans, several of researches have been trying to predict its results and winning teams. Through its history, a lot of data and statistics have been collected for every player and going to enrich more details because NBA is developing more than ever. Hence, a big data issue for this kind of predictions deserves to have attention and effort on it. With the availability of such enormous data, it is still complicated to analyze and predict the outcome of match. The study chooses different approaches such as support vector machine regression, polynomial regression and random forest regression for a comparative analysis to discover how individual player's performance influences the team winning rate. The comparison results show that the learning techniques we have adopted exhibit competitive capability in prediction, and give quite consistent performance regardless of complexity in input data features.**

I. Introductions

National Basketball Association known as NBA is a sports league that has spread its influence worldwide and attract millions of fans from every corner of the world. It has 30 teams divided equally in west and east conference. Every team competes against each other at least one time in the 82 games in regular season in order to earn a position for playoffs, which is a post-season for championship. Only the best 8 teams in both conferences have qualification to attend playoff season.

Since the league is popular over the world, predicting the outcome of the match has become an essential task among fans, coaches and specialists [1-3]. Basketball has many features such as steals, assist, block, etc. Our goal is to predict the best sixteen teams in the league by applying machine learning methods based on players' features. In other words, our model will calculate players' winning contribution for their team. We implemented many varieties of learning methods such as polynomial regression, random forest regression and support vector regression, and reported their corresponding performance for evaluation. Also, player efficiency rating (PER) is installed as another method to compare with learning methods. After training the prediction models, the winning scores of each team will be assessed and be compared with the real ranking to check the accuracy. One common problem for machine learning models is that you will not get to know how well a model performs until you test its performance on an independent dataset. Thus, eight fold cross validation experiments [4] are implemented to estimate the performance of the models.

As for data preparation, many websites and source references have available wide range of supports of accessible data. For example, http://www.basketball-reference.com [5] provided player attributes data per game which is used for predicting players' winning contribution, while https://stats.nba.com/ [6] contributed data for PER that is essential for comparing our model with actual result. For each case, the included data for every season are from 2011 to 2018.

II. The Game of National Basketball Association

NBA teams are divided into two conferences that consist of three distributions, and each includes five teams totaling of thirty teams [7]. Conventional abbreviation titles of the teams are used in this paper, for example, GSW is for Golden State Warrior, TOR for Toronto Raptors, HOU for Houston Rockets, etc. The whole set of abbreviations can be referred from [8].

TABLE I. Divisions from NBA teams

Eastern Conference			Western Conference		
Atlantic	Central	Southeast	Northwest	Pacific	Southwest
5 teams	5 teams	5 teams	5 teams	5 teams	5 teams

Each team has to play a total of 82 games in a regular season, dividing into 41 games at their own stadiums and 41 games at away stadiums. Each team plays against their division team totaling of 16 games in a year. As for other divisions in the regarding conference, each team has to play a total of 24 games. Finally, all the teams face each team from another conference twice a year totaling 30 games.

NBA playoffs for the championship start in late April, with best of 8 teams from both conferences. Those 8 teams consists of winners from each division, a team that has the best records from the conference just after division winners and the teams in lower four seeds. NBA playoffs are held in a tournament fashion. Every team plays in a format of best of seven series, it means that the first team achieving four wins can go forward for the next round, meanwhile the losing team will be eliminated from the tournament. This process is repeated in the following rounds among the teams in the same conference until the only one team remained. Finally, the playoff is concluded by two teams from each conference facing against each other in the best of seven series. This match is called NBA finals, which is held traditionally in June of every year.

Research supported by Ministry of Science and Technology, Taiwan.

G. Sainzaya and P.-H. Hsu are with the Electrical Engineering Department, National Taipei University, New Taipei City, Taiwan (e-mail: sainaa425@yahoo.com, shupohan@gmail.com, respectively).

J.-S. Yang with Department of Mechanical and Electro-Mechanical Engineering, Tamkang University, Tamsui, Taipei, Taiwan (e-mail: 096034@mail.tku.edu.tw).

C.-Y. Yang is with Department of Electrical Engineering, National Taipei University, No. 151, University Rd., San Shia District, New Taipei City, 23741 Taiwan (Corresponding author, phone: +886-2-86741111 # 66737; e-mail: cyyang@mail.ntpu.edu.tw).

III. Related Works

Many researchers have done works regarding predicting the outcome of the NBA games, each trying to produce the best model for providing the best features for analysis or simulating the game. In order to find similar works related to our problem, various researches have been done with vast source of statistics. Two papers from the researches were selected for discussion here, which also attempted to predict the outcomes of NBA games.

The first paper is "Prediction of NBA games based on machine learning methods," written by Torres [9], who used team statistics to predict the outcome of the games. His main focus is scoring the winning percentage and mean value of scoring margin. Instead of trying to predict the points for scoring, he implemented the model to predict the winning and loss rate. Similar to our research, he used linear regression and support vector machines. However, he applied principal component analysis over linear regression and support vector machines because he expected his features to be highly uncorrelated. According to Torres' results, linear regression was his best predictor of winners and losers with a prediction rate of 70.09%.

The second paper is "The use of data mining for basketball matches," written by Miljković [10], who used linear and logistic regression, naive Bayes, neural networks, and support vector machines as their classifiers. He also used box score statistics as features for the loss function. As for his prediction of spread, the system has provided predictions for 78 of 778 matches approximately to ten percent, which was his desired result. Because he implemented it is very hard to predict the exact difference in which basketball game ends and the goal of his attempt was to provide only the approximate information on the possible difference.

IV. Methodology for Evaluation

First of all, the dataset should be extremely carefully studied, selected and organized. Main analysis often done by using just regular season data because players would be signed and traded to different teams from one year to another. Even though there are large varieties of useful data, most of them are not considered as important aspects of the features. It is crucial to choose the significant features carefully. Based on our experience on long period experiments, statistics and analytics feature aspects which have gained most effectiveness were chosen as those shown in Table Ⅱ.

Secondly, the machine learning methods will be implemented and prepared for simulation. As mentioned in the previous section, support vector regression has been suggested as a good way to be applied in the study. Because it has good generalization performance with a strong fitting robustness to its regression data. However, depending only on one method without comparison is not an ideal way to get convincingly desired result. Thus, the same tasks will be performed on random forest regression and polynomial regression to obtain reliably competitive results for comparison.

Finally, the prediction results have to be compared with the actual results and PER to qualify the performance of our model. Hollinger's player efficient rating abbreviated by PER [11] sums up all a player's positive accomplishments, subtracts the negative accomplishments, and returns a per-minute rating of a player's performance. The PER is used for predicting players win rate by basketball experts. The PER also provides a strong reference for fans around the world to bet money for their favorite team and even team coaches use it for making decision for their forthcoming contract with players.

TABLE II. Features used for our simulation

Abbreviation	Description
Age	Age
PTS	Points
FGM	Field Goals Made
3PM	3 Pointers Made
FTM	Free Throws Made
OREB	Offensive Rebounds
DREB	Defensives Rebounds
AST	Assist
STL	Steal
BLK	Block
TOV	Turnovers
PF	Personal Fouls
+/-	Plus/Minus Scores
FG_MISS	Field Goal Miss
FT_MISS	Free Throws Miss

A. *Support Vector Regression*

Support vector regression [12] is a learning principal focused on statistical learning that implements structural risk minimization principles. It uses training data to construct its model of the prediction. This is a very popular method among solving the linear and non-linear regression problems. In case of training data are non-linear, support vector regression converts implicitly the data into a high dimensional feature space by a non-linear mapping function. In the high dimensional space, the samples would be able to be linearly approximated, and the naïve linear edition of optimal regression could be applied accordingly. The non-linear edition of the support vector regression use a kernel operation, $<\cdot, \cdot>$, instead of the inner product for the linear edition, to calculate the similarity among input vectors. The kernel function in fact is a key to have the implicit high-dimensional similarity without an explicit real mapping. The direct mapping is actually difficult to manipulate [13]. In general, the non-linear support vector regression could be expressed mathematically as an optimization problem below:

$$\frac{1}{2}||\omega||^2 + C\sum_{i=1}^{\varepsilon}(\varepsilon_i + \varepsilon_i^*) \tag{1}$$

$$\begin{cases} y_i - < \omega, x_i > -b \le \varepsilon + \varepsilon_i \\ < \omega, x_i > +b - y_i \le \varepsilon + \varepsilon_i^* \\ \varepsilon, \varepsilon_i^* \ge 0 \end{cases} \tag{2}$$

where ε denotes the width of a surrogate loss function ε^* insensitive loss function which could tolerate the input uncertainty with the so called ε-tube, C denotes a trade-off ratio to gain as much the conceptual flatness which would be good for the model generalization of the regression, ω denotes a normal vector to the regression hyperplane, and b denotes a bias. Both ω and b could be calculated from a set of optimized

Lagrange multipliers α_i and α_i^* which would derived first from a dual convex constrained quadratic optimization problem which has been evolved from its primal form in (1)-(2). With the optimized ω and b, we can describe the non-linear input-output relationship as:

$$f(x) =< \omega, x > +b \text{ with } x \in X \tag{3}$$

where X denotes the space of the input patterns. It finds the linear problems in higher dimensions so that our desired distance between points and line minimized. After having fit this line, the algorithm predicts the outcome of an unseen data point by plugging in the point's features to the line equation. This generalization error bound is the combination of the training error and a regularization term that controls the complexity of the hypothesis space.

First study selection was radial basis function because it is possible kernel features will not have an optimal separating hyperplane. However, after testing on two different kernel functions found out linear regression can find the hyperplane and the result is better than radial basis function. Thus, implementation is done by using kernel with polynomial and sigmoid, the result in sigmoid was not an ideal solution since it could not give a good prediction rating. The kernel in polynomial was even worse, because it could not converge to find a hyperplane.

B. Polynomial Regression

A polynomial regression is a way used for changing a linear regression model into a curve. It qualifies as a linear regression, which makes it a nice and straight way to model curved non-linear model. To fit in the model it becomes a problem of finding a polynomial that will pass as close as possible to the input points. This kind of problems have been studied for a long time with multiple applications [14], including statistics [15], machine learning [16], and computer aided system [17]. As with linear regression, the classic solution to polynomial regression is least squares, but this is not robust to outliers, which can perturb the least squares solution by an arbitrary amount.

The model is simply a general linear regression model with k predictors raised to the power of i where $i = 1$ to k. A second order ($k = 2$) polynomial forms a quadratic expression (parabolic curve), a third order ($k = 3$) polynomial forms a cubic expression and a fourth order ($k = 4$) polynomial forms a quartic expression. In general, the greater k will provide more flexibility to the fitting curve. For regression model, we used function that adequately approximates our data.

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_k x^k \tag{4}$$

where $\beta_i = 1, 2, \ldots, k$ are the coefficients of the function and β_0 is called intercept term. Each model is reliant on the coefficient of determination that shows how well data fits in the statistical model. While a greater order k generate more flexible approximation, it also increases possibly unnecessary model complexity. In general, a sufficient large number of k is adapted, instead a further bigger number for polynomial regression.

The least square approach applied for minimizing the sum of squared residuals produced by approximation of regression.

$$X_{ij} = \frac{\partial f(x_i\beta)}{\partial \beta_j} = \phi_j(x_i) \tag{5}$$

where ϕ_j is a function of x and given by β least square estimator is represented by

$$\beta = (X^T X)^{-1} X^T Y \tag{6}$$

Least square analysis can be used as multiple regression technique using $x_1, x_2, \ldots$ as being distinct because the power of x would be square, cubic, or higher, it's no longer a direct proportion which is independent variables in a multiple regression model.

C. Random Forest Regression

Multiple decision trees form a random forest regression. Every single decision trees can represent the training data, which split into subset of population based on their differentiating variables. The reason why it can handle outliers and avoid overfitting during the learning process is each tree made on a different subset of the training observations. Our data has a plenty of features and a random forest can help unravel complex unknown interactions between predictor variables [18].

First testing is done at number of tree from the state thirty and the prediction was only 75.0%. As the tree number increases, it can reach 81.25% when tree number gets eighty as shown in Figure 6.

As for regression model combination of decisions from a sequence base model was used.

$$c(x) = f_0(x) + f_1(x) + \cdots + f_n(x) \tag{7}$$

where the final model c is the sum of simple base models $f_i(\cdot)$. Here, each base classifier is a simple decision tree, each forests are constructed independently using a different subsample of the data. It has three processing steps [19].

1) Training

Random forest uses random decision trees for training. Trees are trained by random features of random data sets. Thus, the learning sets can be either in classification or regression problems. Next step in each node is selection of random features. Final result is decided on the votes of trees. With previous steps, forest generates trees with lowest possible correlation.

2) Error Estimation

In random forest, trees are continuously fitted to subsets of observations. It is very similar to cross validation. Each tree uses two third as observations and the remaining sets used for validation. Error estimation displayed in equation (8).

$$E^c - E^s \geq \mu \times E^r \tag{8}$$

where meaning E^c current error, E^s splitting error, E^r root error and μ is the tuning parameter (0,1).

3) Split Selection

The splitting process performed by the degree of impurity in each node. It can use Gini index to evaluate the features that chosen by impurity degree.

Random forest model is very good at handling tabular data with numerical features, or categorical features with fewer than hundreds of categories. Unlike linear models, random forests are able to capture non-linear interaction between the features and the target. By using these interactions, it is possible to increase the accuracy of prediction result. As for splitting criteria, the following equation can represent model

$$RSS = \sum_{left}(y_i - y_l^*)^2 + \sum_{right}(y_i - y_r^*)^2 \qquad (9)$$

where RSS meaning residual sum of squares and y_l^* denotes mean y value for the left node while y_r^* denotes mean y value for the right node. Simple splitting process of the random forest regression is shown in Figure 1.

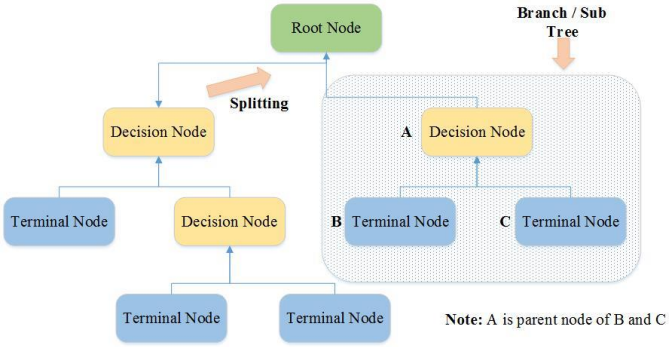

Figure 1. Splitting process of random forest regression.

V. SIMULATION RESULTS

Three different machine learning modules are implemented to predict the outcome of the game by using players' individual stats across the multiple seasons of the game. Moreover, the simulation performances of our model are utilized to see how well they match the actual rankings of the NBA teams. The full processing steps are shown in Figure 2.

Figure 2. Step by step simulation process.

The learning models calculate the players' winning contribution based on the features from each individual and give them corresponding scores. These scores indicate how many games a player can win in a regular season. We sum up all the players' winning contribution in a team and divided by the number of total members on the team. The height of this winning mean value represents team's ranking in a regular season.

The winning mean values of different methods in 2017-2018 season are shown in Figure 3, Figure 4, Figure 5 and a combination is shown in Figure 6.

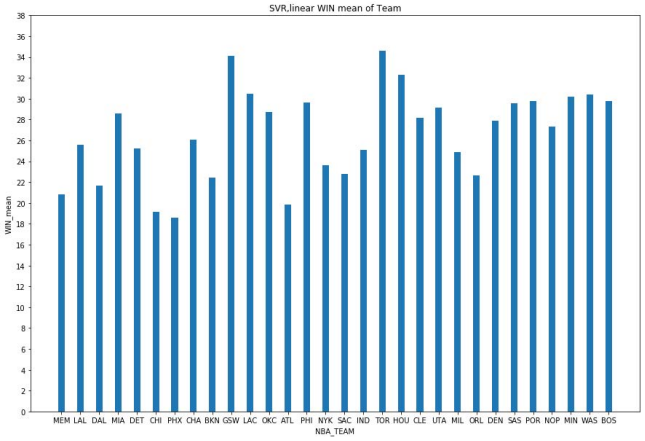

Figure 3. Winning mean value of support vector regression.

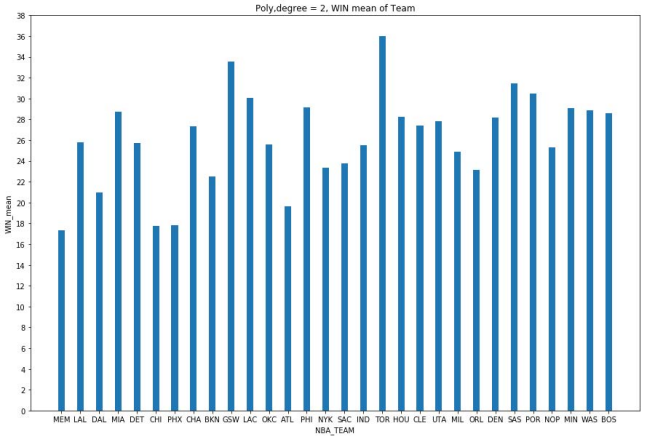

Figure 4. Winnig mean value of polynomial regression.

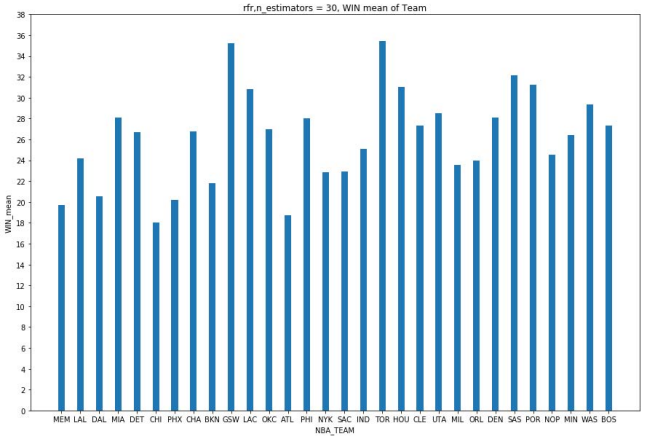

Figure 5. Winnig mean value of random forest regression.

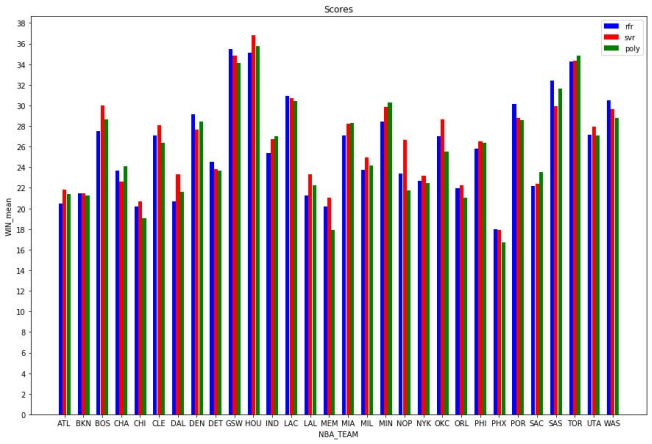

Figure 6. Combination of winning mean value.

Polynomial regression and random forest tree regression have the best two teams while support vector regression have the same best three teams in the league from the actual ranking even though their rankings may be placed slightly different in position as shown in Table III.

TABLE III. PREDICTED RANKINGS OF 2017-2018 SEASON

SVR Ranking		Poly Ranking		RFR Ranking	
Team	Rank	Team	Rank	Team	Rank
TOR	1	TOR	1	GSW	1
GSW	2	GSW	2	TOR	2
HOU	3	SAS	3	SAS	3
LAC	4	POR	4	POR	4
WAS	5	LAC	5	HOU	5
MIN	6	PHI	6	LAC	6
BOS	7	MIN	7	WAS	7
POR	8	WAS	8	UTA	8
PHI	9	MIA	9	DEN	9
SAS	10	BOS	10	PHI	10
UTA	11	HOU	11	MIA	11
OKC	12	DEN	12	CLE	12
MIA	13	UTA	13	BOS	13
CLE	14	CLE	14	OKC	14
DEN	15	CHA	15	MIN	15
NOP	16	LAL	16	DET	16
CHA	17	DET	17	CHA	17
LAL	18	OKC	18	NOP	18
DET	19	IND	19	IND	19
IND	20	NOP	20	LAL	20
MIL	21	MIL	21	ORL	21
NYK	22	SAC	22	MIL	22
SAC	23	NYK	23	SAC	23
ORL	24	ORL	24	NYK	24
BKN	25	BKN	25	BKN	25
DAL	26	DAL	26	PHX	26
MEM	27	ATL	27	DAL	27
ATL	28	PHX	28	MEM	28
CHI	29	CHI	29	ATL	29
PHX	30	MEM	30	CHI	30

As for the PER rankings, the winning contribution of players is replaced by PER of players. By comparing PER prediction with actual rankings, it can be concluded that fixed formula is not accurate enough for scoring player's performance. Thus, it is essential that learning method should be implemented to get more accurate result. Full description of PER and actual rankings are shown in Table IV.

TABLE IV. PER AND ACTUAL RANKINGS OF 2017-2018 SEASON

PER Rankings		Real Rankings	
Team	Rank	Team	Rank
UTA	1	HOU	1
POR	2	TOR	2
MIN	3	GSW	3
SAS	4	BOS	4
TOR	5	PHI	5
LAC	6	CLE	6
WAS	7	POR	7
IND	8	IND	8
CHA	9	OKC	9
DEN	10	NOP	10
MIL	11	UTA	11
CLE	12	SAS	12
MEM	13	MIN	13
GSW	14	DEN	14
SAC	15	MIA	15
PHI	16	MIL	16
BOS	17	WAS	17
OKC	18	LAC	18
BKN	19	DET	19
ORL	20	CHA	20
PHX	21	LAL	21
MIA	22	NYK	22
NOP	23	BKN	23
DAL	24	SAC	24
NYK	25	CHI	25
LAL	26	ORL	26
HOU	27	DAL	27
DET	28	ATL	28
ATL	29	MEM	29
CHI	30	PHX	30

We could find out that standard deviations (STD) in learning methods were high enough to classify teams' winning mean values. However, standard deviation in PER was not high enough, meaning that the PER in each team was nearly the same ratings to be distinguished. This is a reason why learning methods have a better prediction than PER. Full details are shown in Table V.

TABLE V. STATISTIC NUMBER FOR WINNING MEAN VALUE

Poly		SVR		RFR		PER	
Mean	26.13	Mean	26.63	Mean	26.24	Mean	13.02
STD	4.44	STD	4.24	STD	4.47	STD	1.96
Max	35.98	Max	35.58	Max	35.90	Max	19.37
Min	17.34	Min	18.59	Min	18.14	Min	10.52

Final prediction results are shown in Figure 7. It demonstrates that our learning methods have very high prediction accuracy on predicting the best sixteen teams in the season. On the other hand, the traditional scoring of PER greatly differed from the actual rankings with not being able to correctly predict any of the top three teams.

Random forest regression prediction accuracy = 81.25%
Support vector regression prediction accuracy = 81.25%
Polynomial regression prediction accuracy = 75.0%
PER prediction accuracy = 56.25%

Figure 7. Prediction accuracy of 2017-2018 season.

Eight fold cross validation is implemented to evaluate the accuracy of model predictions by partitioning our data into eight different seasons. Prediction validation is calculated by using one season as testing set while the others as training set. The eight fold cross validation results are shown in Figure 8.

Random forest regression prediction accuracy = 81.25%
Support vector regression prediction accuracy =76.56%
Polynomial regression prediction accuracy = 75.0%
PER prediction accuracy = 56.25%

Figure 8. Eight fold cross validation accuracy

VI. CONCLUSIONS

Statistics are more and more important than ever in the sports now. Every field such as tennis, football, baseball etc. are all applying statistical method to make their teams or players better. This research focused on one of the hottest trends in the world regarding machine learning modules.

The main topic is based on regression method rather than classification method because our goal of the research is to predict team rankings instead of classifying their rankings. During our data selection processing, we did not choose players' minutes on the court as the feature aspect in order to avoid overfitting. Reason behind while sorting data as significance level it did not contribute useful value. Thus concluded during NBA games, some players values cannot be determined by time spent on the court.

Three regressions models installed in our research, support vector machine, random forest regression, and polynomial regression. These methods determine each player's winning contribution which is then used for predicting the outcome of the rankings. To distinguish our performance, PER scores are applied for testing which in result, random forest regression, polynomial regression and support vector regression gave better prediction accuracy.

VII. FUTURE WORK

In summary, regression models are astounding for predicting team rankings based on player performance stats.

Our next step is to make a progress in playoffs season, which is a bigger stage than regular season. We would like to implement other learning methods such as classification techniques to test difference between results achieved from previous work.

Additionally, future work will focus on getting results that are more accurate in even complicated situations such as considering player transfer to other team during season play, and which players would be more efficient to play against certain opponents that can lead to victory.

ACKNOWLEDGMENT

The corresponding authors gratefully acknowledges the financial support of the Ministry of Science and Technology of Taiwan through its grants MOST 105-2221-E-305-003 and MOST 106-2221-E-305-001

REFERENCES

[1] B. Ji and J. Li, "NBA All-Star lineup prediction based on neural networks," *in Proceedings of Information Science and Cloud Computing Companion*, Guangzhou, China, 2013.

[2] M. Atlas and Y.Q. Zhang, "Fuzzy neural agents for online NBA scouting," *in Proceedings of IEEE/WIC/ACM International Conference*, Beijing, China, 2004.

[3] R. Nagarajan and L. Li, "Optimizing NBA player selection strategies based on salary and statistics analysis," *in Proceedings of Intelligence and Computing and Cyber Science and Technology Congress*, Orlando, USA, 2017.

[4] M. Vasinek, J. Plato and V. Snasel, "Limitations on low variance k-fold cross validation in learning set of rules inducers," *in Proceedings of Intelligent Networking and Collaborative Systems 2016 International Conference*, Ostrawva, Czech Republic, 2016.

[5] *S*ports Reference LLC. *2017-18 NBA player stats: advanced*. [On-line]. Available: https://www.basketball-reference.com/leagues/NBA_2018_advanced.html [Apr 17, 2018].

[6] NBA Media Ventures LLC. *NBA Advanced Stats*. [On-line]. Available: https://stats.nba.com/players/traditional/?sort=PTS&dir=-1 [Apr 17, 2018].

[7] Wikimedia Foundation Inc. *National basketball association*. [On-line]. Available: https://en.wikipedia.org/wiki/National_Basketball_Association [Apr 17, 2018].

[8] 2018 XML Team Solutions LLC. *Abbreviations*. [On-line]. Available: http://sportsforecaster.com/nba/abbreviations [Apr 17, 2018].

[9] R. A. Torres, "Prediction of NBA games based on machine learning methods," *in Proceedings of Computer-aided engineering department*, *University of Wisconsin*, Wisconsin, United States, 2013.

[10] D. Miljkovic, L. Gajic, A. Kovacevic and Z. Konjovic "The use of data mining for basketball," *in Proceedings of 2010 IEEE 8th International Symposium on Intelligent Systems and Informatics*, Subotica, Serbia, 2010.

[11] Sports Reference LLC. *Calculating PER*. [On-line]. Available: https://www.basketball-reference.com/about/per.html [Apr 17, 2018].

[12] D. Basak, S. Pal and D. C. Patranabis, "Support vector regression," *in Proceedings of Neural Inf. Process. Lett. Rev*, vol. 11, pp. 203–224, India, 2007.

[13] K. Shi and M. Z. Zhang, "Support vector regression based indoor location in IEEE 802.11 environments", *in Proceedings of Mobile Information Systems*, vol. 2015, pp. 14, Wuhan, China, 2015.

[14] J. Gergonne, "The application of the method of least squares to the interpolation of sequences," *in Proceedings of Historia Mathematica*, vol. 1, no. 4, pp. 439–447, Ottawa, Canada, 1974.

[15] I. B. MacNeill, "Properties of sequences of partial sums of polynomial regression residuals with applications to tests for change of regression at unknown times," *in Proceedings of The Annals of Statistics*, pp. 422–433, London, Ontario, Canada, 1978.

[16] A. T. Kalai, A. R. Klivans, Y. Mansour, and R. A. Servedio, "Agnostically learning halfspaces," *in Proceedings of SIAM Journal on Computing*, vol. 37, no. 6, pp. 1777–1805, Colombia, 2008.

[17] V. Pratt, "Direct least-squares fitting of algebraic surfaces," *in Proceedings of SIGGRAPH Comput. Graph*, vol. 21, no. 4, pp. 145–152, Stanford, USA, 1987.

[18] B. Aydin, A. A. Kindiroglu, O. Aran and L. Akarun. "Automatic Personality Prediction from Audiovisual Data using Random Forest Regression," *in Proceedings of 23rd International Conference on Pattern Recognition*, Cancún, México, 2016.

[19] A. Hernández, H. Fabelo, S. Ortega, A. Báez, G. M. Callicó and R. Sarmiento, "Random forest training stage acceleration using graphics processing units," *in Proceedings of University of Las Palmas de Gran Canaria*, Spain, 2017.