NBA Outstanding Player Detection and Game Winner Predictor

Ahmad Jawaad Shah[1]
School of Mathematics, Statistics and Computer Science
University of KwaZulu-Natal, Westville Campus, Durban, 4000 RSA
[1]218029400@stu.ukzn.ac.za

Abstract. The sporting domain is one of the most widely followed domains in the world. Amongst spectators, the hype before new seasons and global tournaments are unparalleled and leads to deeper analysis into teams, their players as well as their game day strategy. Furthermore, through the advent of social media, it has become increasingly popular for spectators to publicly state their MVP of the season/tournament as well as provide game day predictions. However, when it comes to punters, there is money on the line, and they would want to make informed decisions before placing any bets. Thus, correctly predicting the winning team as well as any outstanding performing players from a season becomes an increasingly difficult problem. Thus, to fill this gap, machine learning techniques can be employed to find complex relationships within the data, which can then be used to make informed decisions. In this paper, we provide two architectures, one for outstanding player detection and the other for game predictions on NBA/ABA data. In particular, to identify outstanding players, we conduct clustering on player data and thereafter select a list of top 10 players through a principled ranking approach. Furthermore, for game predictions, we propose an Attack Defence Win (ADW) Network, which utilises Multi-Layered Perceptron (MLP) and Ensemble techniques to predict the outcome of a game of Basketball from the 2004-2005 NBA season. The proposed models provide adequate results when compared to other models (predictive model) and historical data.

Keywords: Machine Learning ◦ Clustering ◦ Multi-Layered Perceptron ◦ Ensemble Techniques

1 Introduction

Sport is a widely followed domain in which millions of spectators come together for the passion of the game as well as continuously push their team's agenda and bragging rights. Throughout history, sporting events have increased in size as well as brough in a large following of new spectators. With the advent of social media, it has become increasingly popular for sporting events to trend on search pages as well as provides the platform where users and pundits share their opinions. Now, more than ever, with the abundance of statistical data, spectators and the likes can make informed predictions on their respective teams. Thus, through machine learning, there is a gap wherein we can apply statistical methodologies as well as appropriate algorithms to learn relationships and trends.

Given the large amounts of data available, a substantial number of features can be collected including the historical performance of the teams, results of matches, and data on players, to help different stakeholders understand the odds of winning or losing forthcoming matches [1]. The automation of this process becomes of paramount importance to the teams, important stakeholders, bookmakers and potential bidders, the latter who are interested in approximating the odds of a game beforehand to make informed betting decisions on the obtained odds [2].

In this paper, we are interested in exploring NBA statistics for outstanding player detection and game day match predictions. In this regard, we are presented with two machine learning techniques that take precedence in this sort of problem. Namely, we encounter supervised and unsupervised learning algorithms. Through the use of supervised algorithms, the general idea is to develop a predictive model based on input and output data in order to facilitate classification or regression. Contrary to this, in unsupervised algorithms, we group the input data based on their features, known as clustering, to identify different regions of interest. Alternatively, unsupervised learning can be facilitated through association which is useful for finding relationships between variables. Figure 1 depicts the options available for model development.

Taking the aforementioned into consideration, we seek to explore avenues in which we utilise statistical strategies and clustering techniques to find the outstanding players in an ABA/NBA season. Furthermore, we also provide a principled approach in the form of an Attack Defence Win (ADW) Network coupled with ensemble techniques to provide match day predictions for the 2004-2005 NBA season. The approach used for identifying outstanding players involves data preparation, in the form of pre-processing, clustering and a further ranking mechanism to filter the top players in order to select the top 10 influential players. Thus, the approach used for outstanding player detection is unsupervised.

Regarding the ADW Model, we propose to use two Multi-Layered Perceptron (MLP) neural networks for regression. Moreover, we train the MLP models on the attacking and defensive data to obtain attacking and defensive networks. Thereafter, we train an ensemble model, an AdaBoost regressor, to predict the win probability given the output from the attacking and defensive networks. This is done for both teams that will be playing and the team that has the highest winning confidence will be selected at the winner. This approach is intuitive as we seek to learn the average points that will be scored and conceded by a team. Thus, the assumption is that teams with a strong attack and weak defence will not win every game, as various teams have strengths in different portions of the game, which will ultimately influence output. Thus, the approach for game winner prediction falls under the supervised category as we aim to predict novelty scores (regression) which ultimately identifies our winner through a predicted confidence rating.

The remainder of the paper is structured as follows. We follow up the introduction, presented in this section, with a discussion and analysis of prior work in this field, in Section 2. Thereafter, in Section 3, we discuss, in-depth, the

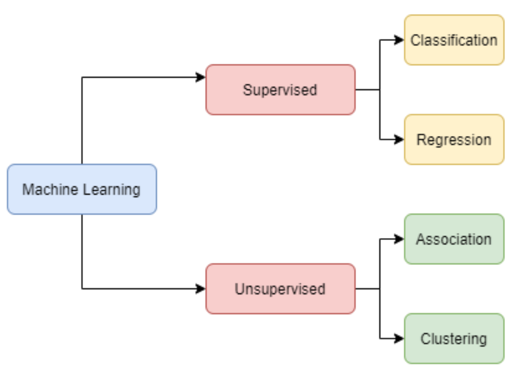


Fig 1. Machine Learning Techniques

proposed architecture for each model as well as outlines the processes and techniques used to design an effective model. In section 4, we provide an insight into the performance of the model according to some predefined metrics and benchmarks. Thereafter, the paper is concluded with directions for future work.

2 Literature Review and Related Work

In the sporting domain, there are many statistics available through which we can gauge the performance of a player or team. However, the problem arises on which statistics to use when carrying out such evaluation. Goals and Assists in many sports will provide a high-level conceptualisation of a player's performance; however, this inherently favours attacking players over defensive players [9].

In [9], a method is proposed to use machine learning techniques to identify the MVPs for the game of handball. Furthermore, their approach utilises five critical steps including Information Gathering for data acquisition, Indicator Selection for feature selection, 2D/3D Projection for visualization, Outlier Detection for detection of players with different performances and Expert Interpretation.

Various approaches have been used to conduct operations on player data such the work done in [11] which proposes a Bayesian regression model to estimate the performance of football (soccer) players. Furthermore, in [12], an approach is presented for player ranking based on the number of passes completed. A reward-based mechanism is used in [13] where players are assigned a reward for doing certain actions. For the task at hand, outlier detection plays a critical role in finding the outstanding players. Within the field of outlier detection, there are many different methodologies and taxonomies such as univariate, multivariate, parametric, and non-parametric techniques to choose from [10].

Previous work in match predictions is documented in [7], [14], [15] and [16]. These each explore methods for predicting the winner of NBA games. In particular, [7] explores the maximum entropy principle for NBA game predictions. This process entails creating feature vectors, thereafter the model is created using the Maximum Entropy principle whereby they count the games with the same features and the same outcome in the training dataset, and then divide them by the training dataset size [7].

[14] explores statistical methods for game day predictions. In this model, the use of context is applied whereby a consideration for the home advantage, teams playing back-to-back games i.e., on two days and the relative strength of each team is used.

Particularly interesting is the work conducted in [16] whereby contemporary statistical modelling techniques used are combined with Neural Networks. In this regard, they proposed four different neural networks, the first being a feed forward neural network with a sigmoid activation function. This neural network is first used for feature pre-processing thereafter feature reduction is conducted through Signal to Noise Ratio (SNR) technique. The SNR method examines the lower-level weights of feed-forward neural network and computes a saliency measure for each feature [16]. The concept of fusion is also explored where other neural networks complement each other.

From the above, it is evident that there have been various techniques proposed for player detection and match day predictions. Each involve finding relationships between the statistics and finding an appropriate technique to leverage those relationships. Regarding this paper's contribution towards outstanding player detection, we employ the use of a non-parametric, multivariate algorithm to facilitate outstanding player detection. Furthermore, for match outcome prediction, we propose a three-fold ADW model that utilises MLPs and ensemble techniques.

3 Methods and Techniques

In this section we highlight the key features for each of the proposed models. In particular, we begin by conducting exploratory data analysis, which forms the base for an effective model. Thereafter, the remainder of the techniques used in each model are explained.

3.1 Overview of the Proposed Framework – Outstanding Player Detection

This subsection details the processes used to define the outstanding player detection model. In particular, the idea behind this model is to cluster players into different groups. Thereafter, a group of top players for the season are selected and further passed through a ranking mechanism where points are allocated based on where they rank against their fellow NBA stars. From the resultant ranking, the top 10 players with the highest score are selected and presented as outstanding players.

Data Pre-Processing. Data pre-processing formed the first step of the outstanding player detection framework. In this regard, we were presented with 23 features, of which, 17 were basketball statistics from the years 1946 to 2004. Table 1 summarises the 17 basketball stats that are present for each player.

The first noticeable anomaly with the data is that the earlier years don't have data for every attribute. Thus, to solve this we propose to find outstanding players per season rather than of all-time. Furthermore, there are strong correlations between features such as OREB, DREB and REB, FGA and FGM, FTA and FTM and lastly, TPA and TPM. Thus, to solve this, we propose to get an average for those features rather than having both present. Figures 2a and 2b depict the transformation in terms of correlation for the data.

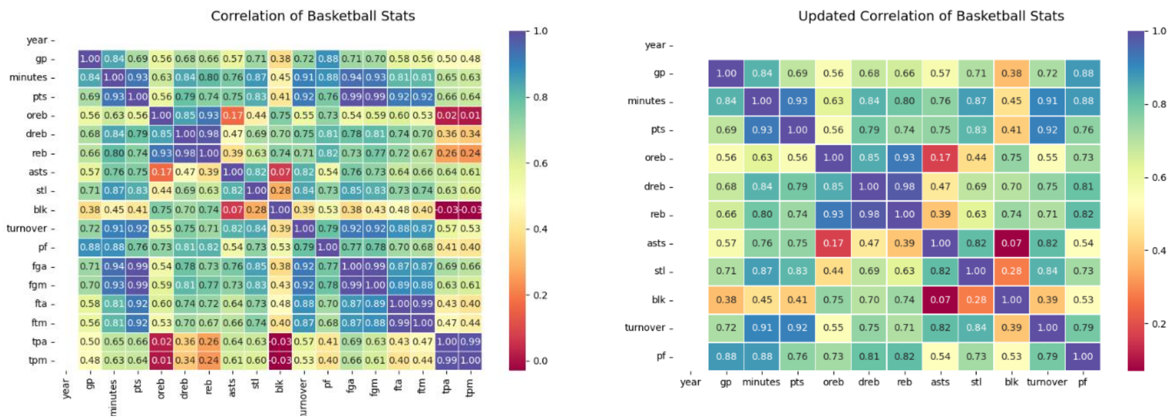
Furthermore, to get a representative value of how well each player performed, the data was further pre-processed, by dividing some features by the number of games the player had played. The assumption behind this step of transformation is to ensure that the player's statistics are representative of how the player played per game. Thus, players that had played fewer games are not disadvantaged. Mathematically, this can be defined as

$$f_{ij} = \frac{f_{ij}}{gp_j} \qquad f_i \in F \tag{1}$$

where f_{ij} is the feature i from the set of features F for a player j and gp_j is the number of games played by player j. Thereafter, Min Max Scaling was applied on

Table 1. Basketball Features Available

Statistic	Description	Statistic	Description	Statistic	Description
GP	Games Played	**ASTS**	Assists	**FGM**	Field Goals Made
MINUTES	Total Minutes Played	**STL**	Steals	**FTA**	Free Throws Attempted
PTS	Points scored	**BLK**	Blocks	**FTM**	Free Throws Made
OREB	Offensive Rebounds	**TURNOVER**	Turnovers	**TPA**	Three Points Attempted
DREB	Defensive Rebounds	**PF**	Personal Fouls	**TPM**	Three Points Made
REB	Rebounds	**FGA**	Field Goals Attempted		



(a) Before Averaging **(b) After Averaging**

Fig 2. Correlation of Features

all the data, so that no single feature is more dominant over the rest, as well as to have normalised input data.

Isolation Forest. The isolation forest [3] ensemble technique [4] was the algorithm used to identify anomalies in the data. The word anomaly in this case, represents our outstanding players. Continuing from our previous sub-topic, we then utilise the normalized data as input into the Isolation Forest.

The isolation forest algorithm works by constructing an ensemble of iTrees for the input data. Thereafter, once constructed, anomalous data is identified by having short average path lengths amongst the trees. Furthermore, the performance of this algorithm is desired as it converges quickly, even with a small number of trees. Moreover, although it converges quickly, it is still able to achieve a high detection rate and high efficiency [3].

Regarding this algorithm, we explored its anomaly detection mechanism by modifying the number of estimators (trees) used (0, 50], the contamination value (0, 0.5], which defines the proportion of outliers in the data. These results are presented in Section 4.

The output from the Isolation Forest provided us with a subset of NBA players whose season statistics were not average. Since the output was fairly large, we provided a secondary ranking algorithm approach to filter those with outstanding statistics and those with weaker statistics.

Principal Component Analysis. Principal Component Analysis (PCA) is a technique used for dimensionality reduction on large datasets. In particular, the large number of variables present in these datasets are reduced to a smaller set of variables, whilst still managing to capture the important information [5].

In this paper, we adopt PCA to support the results obtained from anomaly detection. Thus, the role provided by PCA for us in this outstanding player detection solution is to convert the 17 input features into a 2-dimensional space. This is done so that we can visualise the datapoints present i.e., the players characterised by their statistics.

Furthermore, since each player can be characterised by their stats through PCA, and plotted on a 2-dimensional plane, we then combine this with the anomaly detection predictions from the Isolation Forest, to create a visualization of the data. Figure 3 is the result of combining the anomaly prediction and PCA outputs.

Ranking Mechanism and Top 10 Selection.
From Figure 3, we notice that there is a large number of players who have been classified as anomalous. Thus, there is a need to filter these results so that only those who have performed well remain. In this regard, we propose to rank the obtained players based on 7 statistics, thereafter, selecting only the top 10.

The chosen statistics used to rank the players are Total Points Scored, Assists, Offensive Rebounds, Defensive Rebounds, Steals, Blocks and Turnovers. These strike a balance between attacking and defensive minded players as three are offensive statistics and four are defensive.

To facilitate ranking, we let n represent the number of players predicted to be anomalous. Thereafter, for each statistic, we award 1 point for the person with the lowest value, 2 points for the second-worst player and so on until we reach the best player, who receives n points. Thus, each player accumulates a score, the maximum of which is n x 7. Mathematically, we can represent the score accumulated by a player p_i as,

$$\mathrm{score}_i = \sum_{j=0}^{6} \mathrm{rank}(\mathrm{stat}(j, p_i)) \tag{2}$$

where j is one of the selected features, score_i is the accumulated score of player p_i, $\mathrm{stat}(j, p_i)$ is the stat of feature j for p_i, and $\mathrm{rank}(\mathrm{stat}(j, p_i))$ is the rank in the range [1, n] for points to be awarded. Following the ranking process, we select the top 10 players with the highest accumulated score. We can then update the visualization, and the result is shown in Figure 4.

3.2 Overview of the Proposed Framework – Match Prediction

In this subsection, we discuss the intuition and techniques employed in developing the Attack Defence Win model.

Data Pre-Processing. Similar to the previous model, the input data needs to be pre-processed for the given task. The dataset used provides overall data on the season for each team. In this regard, we split the data into different years and convert the season statistics into per game statistics.

From the provided data, we select 15 offensive, 15 defensive features. Furthermore, for each of the offensive features, there is a defensive counterpart. Thus, we can consider the features to be symmetric and can be used to create our Attack and Defence models, respectively.

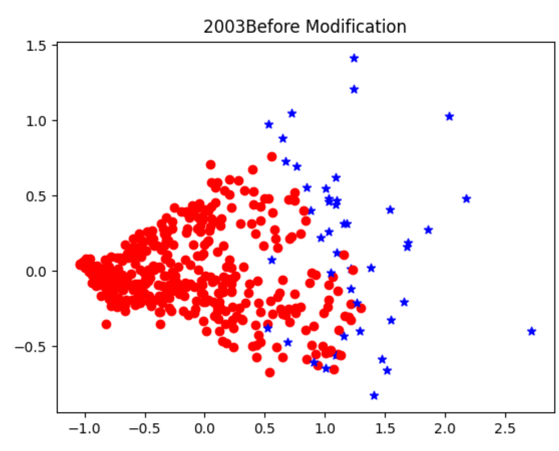


Fig 3. 2003 NBA Players Detected as Outliers

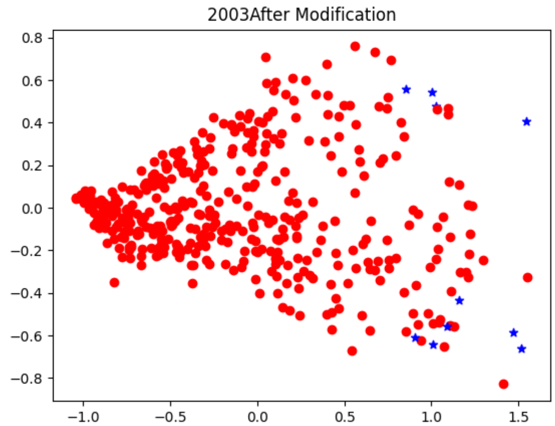


Fig 4. Top 10 Selected NBA Players for the 2003 Season

We also choose to use the win and loss ratios; however, this is for reference as only the win ratio is used. Table 2 shows the attacking and defensive attributes alongside each other.

Attack and Defence Network. The attack and defence networks are two MLP neural networks trained with the same configurations. The attacking network is trained on the per game attacking data with the points scored (o_pts) being the target variable, whereas the defence network is trained on the defensive per game data with the points conceded (d_pts) as the target variable.

The intuition behind the two proposed MLP models stems from the following logic. In the attacking network, during training, we seek to learn a way to predict an output score given some attacking statistics. In the defensive network, a similar approach is taken, as during training we seek to learn a way to predict a score that will represent the number of points we will concede. The models are given sufficient epochs to train on as well as provides a mechanism for early stoppage. Thus, we ensure that given our training data and validation portion, the MLP will converge on an adequate loss function representation with optimum weights.

Regarding the main focus of this model, an unorthodox approach is taken. We let α represent the points that the home team will score, β the points that the away team will score, δ the points that the home team will concede and ε the points that the away team will concede. For α, we pass in the away team's defensive statistics into the attacking MLP. This represents the points that the home team will score, given that the away teams defensive statistics are set up in a particular away. Thus, the home teams points scored is what the away team will possibly concede when going up against an attack. The same procedure is done for β, where the home teams defensive statistics are given to the attacking network. For δ, we pass the away team's attacking statistics into the defensive network. This represents the

Table 2. Offensive and Defensive Team Stats

Attack	Defence	Attack	Defence	Attack	Defence
o_fgm	d_fgm	o_dreb	d_dreb	o_to	d_to
o_fga	d_fga	o_reb	d_reb	o_blk	d_blk
o_ftm	d_ftm	o_asts	d_asts	o_3pm	d_3pm
o_fta	d_fta	o_pf	d_pf	o_3pa	d_3pa
o_oreb	d_oreb	o_stl	d_stl	o_pts	d_pts

points that the home team will concede given the away team's attacking performance on a defence. This is done the other way around to obtain ε. Once we receive the values from the respective MLP's, match scenarios are setup as follows.

$$\text{Home Team} = [\alpha, \delta] \tag{3}$$

$$\text{Away Team} = [\beta, \varepsilon] \tag{4}$$

These match scenarios are passed onto a subsequent Win Confidence model, that will provide a confidence rating given on the match scenario at hand. This model is discussed below.

Win Confidence Model. In this model we use techniques to learn a way of predicting a win given points scored and points conceded. In particular, from Table 2, we are interested in the points scored (o_pts) and points conceded (d_pts) features. Furthermore, we set aside the team win ratio to be the target variable. Since we normalised the data to represent per game statistics, the model created will be able to identify the win percentage given a game score. We obtain the game score from MLP trained in the previous section.

To the foregoing, the proposed model utilises ensemble techniques to facilitate confidence prediction. An AdaBoost regression model [6] is used with the base regression models being MLP's. In this regard, we experiment with the number of base estimators and base epochs used, the results of which are presented in Section 4.

The intuition for this model stems from the following. Each team play m games per season, where they win x% and lose (1-x) %. Given this, each team also scores an average of y points per match and concedes z points per match. For weak teams, where the loss percentage is high, they are more likely to have a lower y and higher z. Conversely, we may assume the opposite for a strong team. Thus, through the AdaBoost ensemble technique we train multiple base models to learn how to predict the x% given the y and z values presented. These y and z values will be well presented as they are obtained from the attack and defence networks. From (3) and (4) given earlier, we present these to the model and obtain a confidence ranking for the home and away team. The team with the higher win confidence will be predicted as the winner.

$$\text{Home Confidence} = \text{Confidence}([\alpha, \delta]) \tag{7}$$

$$\text{Away Confidence} = \text{Confidence}([\beta, \varepsilon]) \tag{6}$$

$$\text{Winner} = \begin{cases} \text{Home Team} & \text{if Home Confidence} \geq \text{Away Confidence} \\ & \text{Away Team} \end{cases} \tag{5}$$

4 Results and Discussion

In this section, we evaluate both of the proposed models through appropriate techniques. We begin by defining the experimental settings and techniques used, thereafter a critical evaluation is done.

4.1 Experimental Settings

In this subsection, we briefly discuss the settings under which the models were evaluated. In particular, we formalise the dataset used thereafter, provide an overview of the metrics used to measure the performance of the unsupervised model and provide the benchmark for the match prediction model.

Dataset. The dataset used for outstanding player detection contained a vast amount of data and their overall season performance. It would be unfair to compare players from different eras and seasons, as there are many more players now than there were many years ago, which would give an unfair bias towards this generation's players. Thus, we proposed to use the data in a per season format. For this model, we also utilise a dataset containing NBA all-stars for each season. Figure 5 shows the increase in player numbers over the years.

Regarding match predictions, teams gradually started to enter the sport over time. Thus, historical data for all teams is not equally present. In this regard, we propose to conduct match predictions on the 2004-2005 NBA season only. Furthermore, 2004-2005 NBA match data was scraped from Basketball Reference [1]. Figure 6 shows the influx of teams over time

Evaluation Metrics and Benchmarks. Since the outstanding player detection model is unsupervised, we measure its performance against the All-Stars for that particular season. There may be a slight discrepancy in identifications as it was noted that the IDs of players in the different dataset occasionally varied by a character which caused detection issues. One such case was the basketball ID of Shaquille O'Neal.

For match prediction, we measure our model against the real-world scores. Thus, we measure and obtain an accuracy for our predictive model. Furthermore, we extend the model to an SVM implementation, which we utilise as a benchmark.

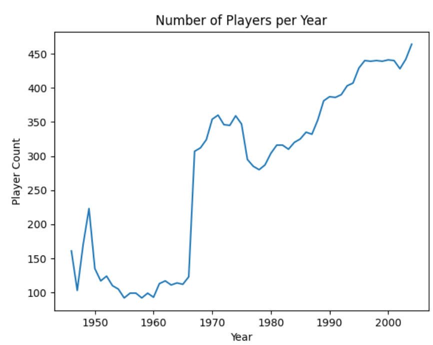


Fig 5. Increase in players

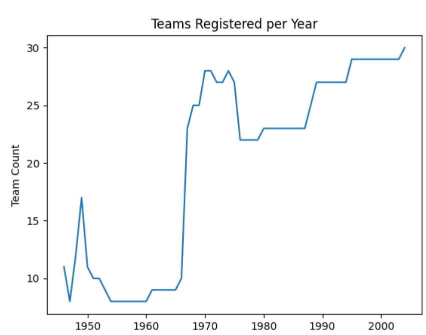


Fig 6. Influx of Teams over the years

[1] https://www.basketball-reference.com/leagues/NBA_2005_games.html

4.2 Model Analysis – Outstanding Player Detection

In this subsection, we explore the results obtained from our outstanding player detector against the set of All-Stars for each season. In particular, we are interested in the seasons 2000 to 2004.

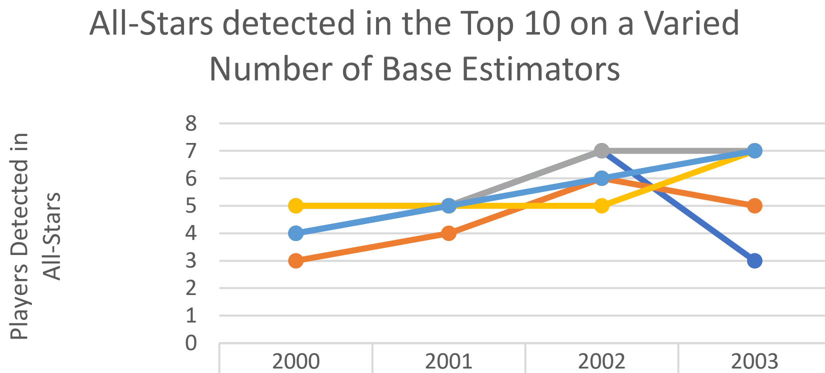
To measure the model, we varied model parameters to find a good set of parameters that will provide us with a suitable detector. In this regard, the number of base estimators for the model was tweaked, the results of which are presented in Figure 7. Furthermore, the contamination value varied, which defines the proportion of the data that will be considered as outliers. These results are shown in Figure 8.

The crux of this evaluation was comparing the selected players to the NBA All-Stars list of a particular season. An NBA All-Star is a star player of the given season [8]. As depicted in Figures 7 and 8, our best results are when the number of estimators was set to 30 and contamination value to 0.1. The low contamination value indicates that we are able to select a better top 10 when we are stricter with regards to what is considered an outlier.

Upon inspection into the results obtained, for the NBA season of 2003, we have 7 players considered outstanding who are part of the All-Stars. A player who was not part of the All-Stars for that season was LeBron James; however, we picked him to be an outstanding player. During the 2003 season, LeBron James was awarded the title of "Rookie of The Year". This is according to Basketball Reference[2], a website containing basketball statistics. Thus, we see that our model is able to select players who have performed well but not made it into the All-Stas for some reason.

4.3 Model Analysis – Match Prediction

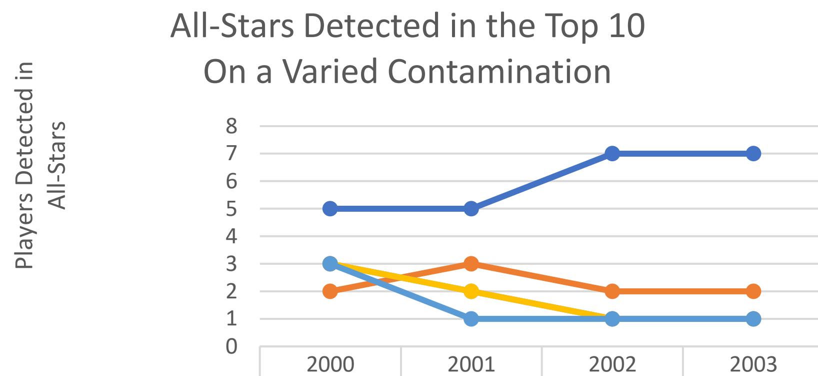
As stated earlier, to facilitate evaluation of the predictive model, we measure our results against the actual results obtained in the NBA basketball league. We also created a second regression model using SVMs to be a benchmark model.



	2000	2001	2002	2003
10 Estimators	4	5	7	3
20 Estimators	3	4	6	5
30 Estimators	5	5	7	7
40 Estimators	5	5	5	7
50 Estimators	4	5	6	7

Fig 7. The number of players from the top 10 that were part of the All-Stars obtained through a varied number of base estimators

[2] https://www.basketball-reference.com/leagues/



	2000	2001	2002	2003
0.1 Contamination	5	5	7	7
0.2 Contamination	2	3	2	2
0.3 Contamination	3	2	1	1
0.4 Contamination	3	2	1	1
0.5 Contamination	3	1	1	1

Fig 8. The number of players from the top 10 that were part of the All-Stars obtained through a varied contamination rate

Regarding the MLP model, it was only in the unoptimized case i.e., the MLP did not converge and the MLP was using default parameters that the accuracy value was above 50% but below 60%. However, through parameter tuning, the accuracy value varied from 60% onwards; but the model did not stray below 60%. Furthermore, we are able to obtain a maximum accuracy of 68%. It is worth noting that the scaped NBA game data file had 1314 games played of which 68% is 893 (Rounded down) games correctly predicted. This is a fairly large number of correct predictions given that the data that the model was trained on had no prior match day experience, but rather relied on averaged season stats to obtain game stats.

Nonetheless, to obtain the above results, we varied the number of epochs used in training the base regressors for the AdaBoost model. Furthermore, we also varied the number of estimators used to provide the final win percentage prediction. Regarding the MLP, the results were fairly high and constantly in the 60's region, whereas the SVM model did not change regardless of the different parameters. Furthermore, we see note that MLP base model with higher epochs generally provide us with better accuracy than those trained on lower epochs. The graph in Figure 9 summarises the MLP's results.

Regarding the SVM, the model maintains 58% accuracy throughout. Modifications to the AdaBoost parameters have no effect on the SVM base model and as such remain at a 58% accuracy. In this regard, we can conclude that for the given experiments, the MLP provides us with favourable results and is more likely to be trusted by punters.

According to [7], which explored match prediction on NBA data for the season 2007 onwards, the models such as Naïve Bayes, Logistic Regression, BP Neural Networks and Random Forests generally achieved a lower accuracy than our model. Furthermore, these we only applied on playoff games which is a significantly smaller number of games. However, this is not a good benchmark for us since their models were applied on different data.

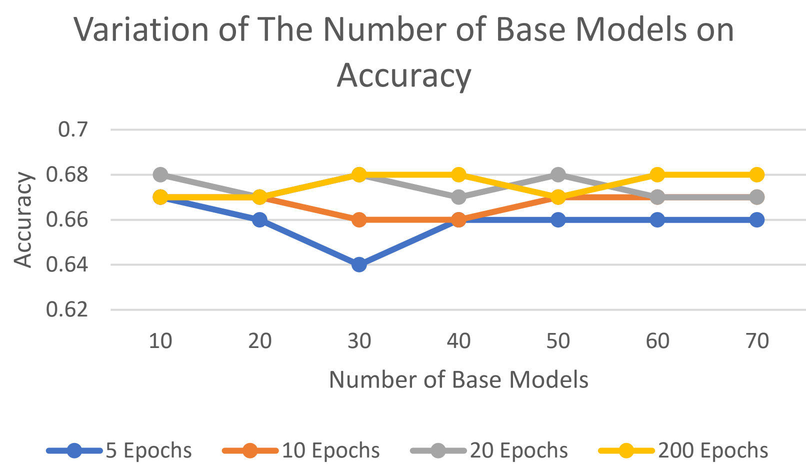


Fig 9. The variation in accuracy as the number of base models increased

5 Conclusion and Future Work

In this paper we described techniques for outstanding player detection and match predictions all on NBA season statistics. The results presented are fairly mediocre but are acceptable since there was a lack of context data used for in-depth match modelling. Thus, regarding outstanding player detection, future works in this field could incorporate player history to gauge whether or not the player has an off-season or a good one, and thereafter proceed to identify the top players. Regarding future work in match prediction, these techniques could be extended to incorporate previous match data and encounters with that specific opponent. Furthermore, more advanced statistical techniques could be applied on the features during pre-processing.

6 References

[1] R. P. Bunker and F. Thabtah, "A machine learning framework for sport result prediction," *Applied Computing and Informatics*, vol. 15, no. 1, pp. 27–33, Jan. 2019, doi: 10.1016/j.aci.2017.09.005.

[2] M. Fernandez, B. Ulmer, Predicting Soccer Match Results in the English Premier League, 2014.

[3] T. Fei, K. Liu, Ting, and Z.-H. Zhou, "Isolation Forest." [Online]. Available: https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf.

[4] "sklearn.ensemble.IsolationForest — scikit-learn 0.23.1 documentation," *scikit-learn.org*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html.

[5] Z. Jaadi, "A Step by Step Explanation of Principal Component Analysis," *Built In*, Sep. 04, 2019. https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[6] “sklearn.ensemble.AdaBoostRegressor — scikit-learn 0.23.2 documentation,” *scikit-learn.org*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html.

[7] G. Cheng, Z. Zhang, M. Kyebambe, and N. Kimbugwe, “Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle,” *Entropy*, vol. 18, no. 12, p. 450, Dec. 2016, doi: 10.3390/e18120450.

[8] F. N. com Staff, “NBA All-Star Draft: How it all works,” *NBA.com*. https://www.nba.com/nba-all-star-draft-explainer.

[9] F. Romero and E. Angulo, “Finding outstanding performance in handball players based on statistical analysis.” Accessed: Nov. 11, 2021. [Online].

[10] I. Ben-Gal, “OUTLIER DETECTION.” Accessed: Nov. 11, 2021. [Online].

[11] Saez Castillo, A., Rodriguez Avi, J., and Perez Sanchez, J. M. (2013). Expected number of goals depending on intrinsic and extrinsic factors of a football player. an application to professional Spanish football league. European Journal of Sport Science, 13(2):127-138

[12] Brooks, J., Kerr, M., and Guttag, J. (2016). Developing a data-driven player ranking in soccer using predictive model weights. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 49-55. ACM.

[13] Kharrat, T., Pena, J. L., McHale, I. (2017). Plus-minus player ratings for soccer. arXiv preprint arXiv:1706.04943.

[14] H. Manner, “Modeling and forecasting the outcomes of NBA basketball games,” *Journal of Quantitative Analysis in Sports*, vol. 12, no. 1, Jan. 2016, doi: 10.1515/jqas-2015-0088.

[15] E. Jones, “PREDICTING OUTCOMES OF NBA BASKETBALL GAMES,” 2016. Accessed: Nov. 11, 2021. [Online].

[16] B. Loeffelholz, E. Bednar, and K. W. Bauer, “Predicting NBA Games Using Neural Networks,” *Journal of Quantitative Analysis in Sports*, vol. 5, no. 1, Jan. 2009, doi: 10.2202/1559-0410.1156.