Data Mining Methods for Sports Prediction

Jacob Mitchell
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
mitc0380@morris.umn.edu

ABSTRACT

Historically, predicting sporting events has been handled by experts with in-depth knowledge of the sport, as well as a wealth of statistics to aid and back up their predictions. As statistics become more readily available and more advanced, the sports world is increasingly embracing mathematical and statistical methods of predicting results. Recently, data mining and machine learning algorithms have been implemented to predict results in various sports. With the wealth of data that can be analyzed to find connections between different statistics and success, algorithms seem like a natural option moving forward as the data becomes more complex. This paper focuses on two of these methods, Random Forest and Neural Networks. We compare the methods in different examples of data sets and draw conclusions on the possibility of these algorithms being used more in the future of sports prediction.

Keywords

Random Forest, Neural Networks, Sports Prediction, Machine Learning, Artificial Intelligence

1. INTRODUCTION

The world of sports prediction, which was long dominated by human odds-makers usually working out of Las Vegas, has gone through somewhat of a renaissance since the advent of computers. Computers brought an increase in data and statistics available for analyzing match-ups, which meant that odds-makers, sportswriters, and others involved in predicting games were forced to modify their approach from a hybrid of "going with their gut" and basic analysis of a team's statistics. The internet also made this data more available to the average consumer of sporting events, which meant the experts needed to come up with a more advanced approach to stay ahead of the average fan.

There are now numerous mathematical methods used by odds-makers to make more accurate predictions. Baseball was the first sport to embrace these new methods by using them to build teams around certain statistics that proved more valuable than others. Players have everything they do on and off the field analyzed to find any statistical correlation to success on the field. In recent years, there has been more research into the use of data mining and machine learning methods of predicting results. We go over two of these methods. The first is the *random forest*, which makes predictions based on a majority vote amongst the decision trees in the forest. This method is discussed in Section 2.3 The second method is *neural networks* which are described in Section 2.4. Neural networks are meant to mimic how the human mind makes decision by creating nodes based on inputs and outputting a prediction.

The paper is structured as follows. Section 2 presents the background required to understand the derivation of both the random forest and neural network methods, as well as some general terms and ideas required to understand machine learning and the prediction systems. Section 3 and 4 describe the random forest and neural network methods in detail, including the data used for testing, how they are implemented, and their results when compared against other prediction methods. Section 5 discusses the results of the tests and compares the two methods directly in detail and draws conclusions based on the comparison results as well as possible future implementation of machine learning algorithms into prediction systems.

UMM CSci Senior Seminar Conference, April 2017 Morris, MN.

2. BACKGROUND

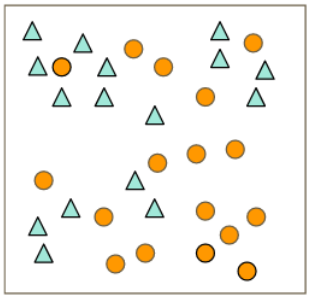
In this section, we describe some general terms and methods needed for understanding machine learning. We then present introductory information on random forests and neural networks and how they are constructed.

2.1 Machine Learning and Prediction

Machine Learning is a type of algorithm construction that is used to solve optimization problems, which in the case of this paper, is the best way to predict sporting events. Prediction involves giving an algorithm sets of input data and finding their relation to the optimal result. The algorithms analyze patterns of the input data that produce optimal results, and uses these patterns to better predict future results when given similar sets of data. Repeatedly running input data through the algorithm is called *training*, which is used to teach the algorithm how the data should be interpreted and placed into categories based on the relation of desired output to actual output. There are generally three types of learning for these algorithms, *supervised*, *reinforced*, and *unsupervised*. Each of the examples detailed in this paper will use supervised learning. Supervised learning involves giving the algorithm a set of categories to divide the input data into, based on desired output data.

Features are another important component of machine

Figure 1: A two-dimensional, two-variable depiction of an input space that is to be divided into partitions to create a decision tree.



learning algorithms. Features are aspects of the data that is chosen for the algorithms to focus on. They can be basic stats such as wins/losses, home/away, or previous game results. They can also be much more complex and/or obtuse such as who the referee of the match was or what the temperature was at game-time. Feature selection is an important component of *training* the algorithm. Training an algorithm involves giving the algorithm one or more sets of data to find features that are potentially linked to a certain outcome. The more training an algorithm is given, the more concrete these links become. If too few features are chosen, the algorithm will have similar chances of predicting a game as the average human expert. If too many features are given, it can become difficult to discern which affect the final outcome the most. The most important part of feature selection is allowing the algorithm enough training with a set of features to figure out which combination predicts the desired outcome most frequently.

2.2 Sports Gambling Terms and Ideas

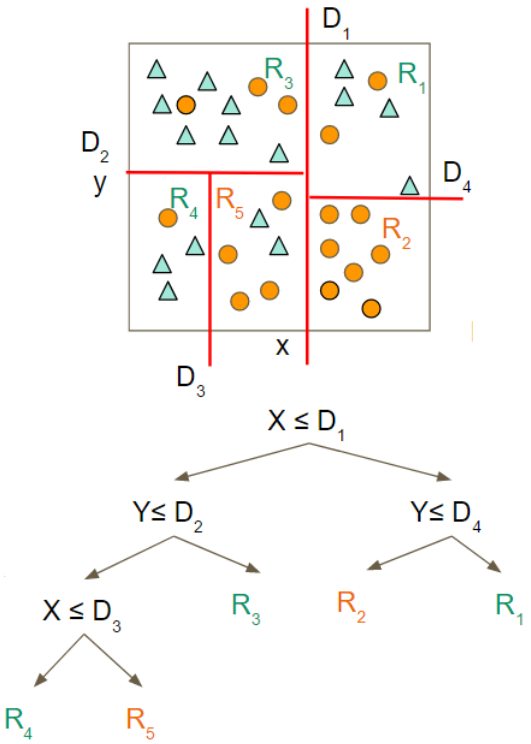
With betting being an important application for these prediction methods, this section goes over some basic background on the terms and ideas involved in betting.

The basic goal of betting is to make more than your initial investment. Profit measures the total number of units that have been gained over the initial investment. Bets are usually made against a betting line set by whoever is collecting the bets. A decimal line is the most typical especially with betting websites. They are called decimal lines because the return is some fraction of what the bettor invested. An example of a decimal line is 7/4. This means if you bet 4 unit, you can win 7 units plus your initial investment back. The examples used in Sections 3 and 4 are only trying to predict the winner or loser of a game. The profit of the prediction they make is based upon a given betting line. There are many other different facets of a match that can be bet on, which means there are many more applications to the algorithms than just predicting wins and losses.

2.3 Random Forests

The random forest is a machine learning method that constructs *decision trees* based on a data input at training time and outputs the best tree, or prediction, based on the mode of the data. Leo Breiman and Adele Cutler are generally credited with the creation of the algorithm [2]. Their implementation is used in both examples of random forests tested in Section 3.1 and Section 3.2.

Figure 2: The input space from Figure 1 after the partitions have been made. The resulting tree is shown below and demonstrated how each region is derived.



Decision trees are developed by splitting a set containing a collection of multidimensional input variables into exclusive, or close to exclusive, groups called classes. As an example, picture a fenced in area with n dogs and cats. The goal is to divide the dogs and cats into groups based on their position in the fenced in area. The initial input, or fenced in area, is shown in Figure 1. Imagine the triangles are cats and the circles are dogs.

To begin, we have to divide the fenced area into two rectangles with a reasonable separation between variables. This initial division is made at D_1 in Figure 2. We will continue to divide each rectangle into partitions until they are mostly exclusive to either dogs or cats. For the rectangle formed by points $\leq D_1$, this division is made at D_2. For the region $\leq D_2$, the division is made at D_3. For the rectangle formed by points $\geq D_1$, the division is made at D_4. Since R_1 and R_2 are reasonably exclusive to cats and dogs, no additional division are needed. After making the division at D_4, we are left with five regions mostly exclusive to either dogs or cats. Each division of the rectangles makes the subsequent divisions children of the previous large group, which becomes a parent node. These divisions become a part of a decision process that takes the form of a tree to determine whether a point is most likely a dog or a cat. For example, if we want to find out if a point in R_2 is a dog or a cat, we first check if its x-coordinate is $\leq D_1$. It is not so we then check if its y-coordinate is $\leq D_4$, and it is. From here we can see that all points in R_2 are dogs, so that means that our current point is also most likely a dog. if The tree structure is also shown in Figure 2.

This is how individual trees are created for the random forest. In Breiman's approach, each tree in the forest is formed by first selecting at random, a group of input features to split on. This group is usually a random percentage of the input data. For example, if there is 100 input variables, one tree could receive the first 50, while another tree could receive the last 50. This process is known as *feature bagging*. To build the forest, multiple decision trees are used as *base learners*, and combined to produce a better performing *ensemble classifier* [5]. An initial training is done to the algorithm to reveal which trees are optimal to look for when making a prediction. Many trees can become quite deep based on different sets of training data and become more unpredictable as a result. The training can find relationships between the deeper trees and average them out to find their most important features. After training is finished the forest is formed using the best decision trees. The output of the forest is the majority output of the trees.

2.4 Neural Networks

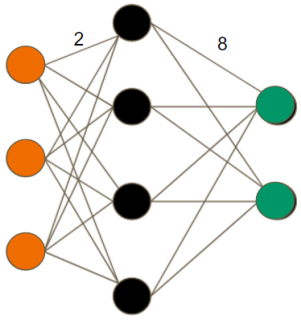
Neural networks get their name based on being modeled after the way our brain connects neurons. A neural network generally consists of three separate layers. The first is the *input layer*. The input layer receives the data and sends it to the *hidden layer* based on previous training and connections that have been made with that data type. The hidden layer consists of the bulk of the network and nodes. There can be multiple hidden layers, but each of the hidden layers still has a similar purpose. The hidden layer draws connections to the *output layer*. The output layer compares the output from the hidden layer to the desired output [6]. A diagram of these layers is shown in Figure 3. The layers are connected by edges that are given *weights*. Weights are calculated and updated during training and determine the correlation between the two nodes it connects to the desired result. Examples of weights are shown in Figure 3 connecting the top three nodes. Training a neural network is done by running the algorithm continuously until the weights have been updated to have optimal values that show their correlation to the desired result.

Each node has two functions attached to it. The first is the *sum* function. This function adds all of the inputs of the node multiplied by their weight. The second function is the *activation* function. This function uses the result of the sum function as an input, and outputs a value that becomes the output of the node [3]. This output is used as a reference for the node's role in the final output of the network.

Both examples of neural networks in Sections 4.1 and 4.2 are *Multilayer Perceptrons*(MLP) that use backpropagation as the method to update the weights. MLPs are neural networks in which each node is fully connected to every node in the next layer. They typically use backpropagation as the method to update the weights between layers. Backpropagation starts with random weights between each node and updates its weights based on the delta rule, which is shown in the code below [7].

```
initialize network weights
do
   forEach training example named x
      prediction = neural-net-output(network, x)
      actual = teacher-output(x)
      compute error (prediction - actual)
      compute change in wh
      compute change in wi
      update network weights
until classified correctly or other stopping point
return the network
```

Figure 3: This is a simple example of a neural network. The three dots on the left side are the input layer, the four dots in the middle are the hidden layer, and the two dots on the right are the output layer. The numbers on the connections represent the weights between nodes (not all weights are shown).



For every training example, x, the difference between predicted output and actual output is calculated. This is followed by a computing the desired change in the weights between the hidden layer and output layer w_h and the input layer and hidden layer w_i. This is where the term backpropagation comes from: for every time through the loop the weights are updated starting from the output layer of the network moving back toward the input layer. These changes are updated and the loop continues until all of x are classified correctly or the algorithm reaches another stopping criterion is satisfied. A stopping criterion is used to stop an issue known as *overfitting*. Overfitting occurs when a neural network essentially memorizes the training data, and becomes unable to generalize new data, making the algorithm have no ability to learn and update based on the new incoming data. Overfitting can be caused by the error crossing below a certain threshold, so a stopping condition is usually added to keep the error above whatever threshold has been chosen.

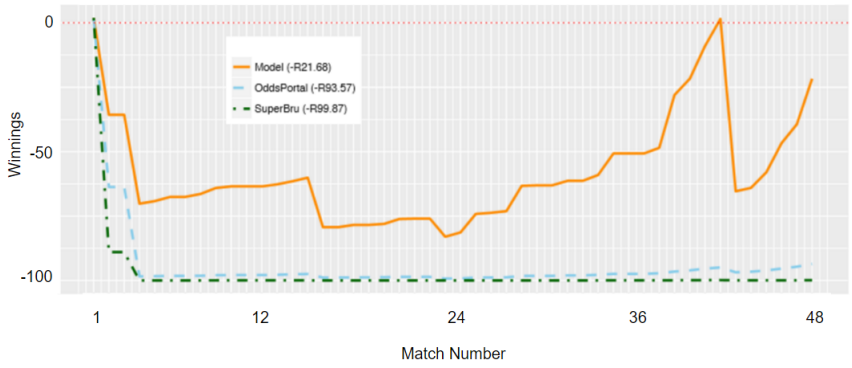
Most neural networks work in a similar fashion. Variations include a higher number of hidden layers, or setting the weights before the initial training is done.

Now that we have established some background on random forests and neural networks we can show how they predict given data from the sports world. We compare across multiple sports and data sets to show both the versatility and practicality of the given methods. The results are compared with popular methods currently being used by oddsmakers.

3. RANDOM FORESTS

In this section we cover two applications of random forests on data taken directly from two professional sports leagues. The two methods are compared against various other methods used for prediction including popular gambling sites that use aggregate data from its users, as well as other machine learning algorithms.

Figure 4: This is a graph showing the monetary winnings of the random forest model used by Pretorius[5] during the Rugby World Cup. The top and bottom x-axes show the matchup, with the current winnings shown on the y-axis. The solid line represents the random forest model, while the dotted lines represent the human prediction methods.



3.1 Rugby World Cup

The first example of a random forest implementation involves testing it on data collected from the Rugby World Cup. The training data comes from past World Cups. Many random forests were created and tested on the data. The best based on running time and test error were chosen before the start of the tournament [5]. Breiman's *Forest-RI*, which is a variation on Breiman's original method is used. *Forest-RI* builds an ensemble of randomized trees by sampling replacement from the training data and selects a random subset of input variables as candidates for splitting [5].

After each match, the model was updated with the latest data to assure it was as up-to-date as possible. The comparison made by the authors was between their method, *SuperBru*, and *OddsPortal*, which are aggregate prediction models based on inputs from the site's human users. *OddsPortal* is specifically designed with betting in mind, while *SuperBru* is based on competition with its users.

The authors used the null hypothesis that humans would be superior to the random forest method to show that random forests were at least as effective as humans. During the 2015 Rugby World Cup(RWC), *OddsPortal* and *SuperBru* predicted 85.42% of matches correctly, while the random forest predicted 89.58% correct. Given the small sample size of 48 matches, the authors could only concluded that there was no evidence to show humans were more effective at predicting than random forests [5]. The authors also compared the potential monetary winnings that would have come from each source and concluded that the random forest was superior in that regard. The results of this test are seen in Figure 4. It is clear from the beginning of the test that random forests were more effective than the human methods, although none of the methods were profitable in the end.

3.2 English Premier League

The next data tested using random forests is from soccer matches in the English Premier League. The goal of the experiment was to test machine learning methods against linear algebra methods including the Massey Method, Colley Method, and mHITS (Offense-Defense Model). The Massey Method uses linear squares regression to solve a system of equations. The Colley Method is a variation of the simple method

$$r_i = w_i/t_i$$

where r_i is the rating of a team i, w_i is the total wins of that team, and t_i is the total games played. mHITS is similar to the Colley method, except for instead of using the ratio of wins to games played, it uses the ratio of offense strength o_i to defensive strength, d_i.

Each method was run through three tests of accuracy. The first was a hindsight test. The hindsight test used data from games that had already been completed in the previous season. The algorithms were asked to predict a chosen game from the middle of the season based on all data from the season. As an example, if a team had won three games directly before the chosen game and the three games directly after, it is more likely the team also won the game in question given the trends around that game. Consequently all of the algorithms performed better than a typical foresight prediction in the trial. Foresight is the more typical of the two prediction types. The algorithm is asked to take all results up to the game in question and predict the outcome. This is what we would think of sports bettors using. They are asked to look at all info up to a given game that has yet to be played and predict the final result.

The data included all data available from www.football-data.co.uk [3]. The training set for the algorithms included the number of games played to that point in the season, so the more games that were played, the more data became available to the algorithms. Random forests performed especially well at hindsight prediction, predicting the winners of the match with a minimum accuracy of 94.74% over the five seasons tested between 2008 and 2013 [3]. In comparison to the other two machine learning methods tested, which were individual decision trees and neural networks, random forests showed a distinct advantage in hindsight prediction. It is interesting to see how much better random forests predicted compared to decision trees, which is the base structure of the forest. It follows that given many decision trees voting on a result that a random forest should be at least as accurate, but it amazing that there is at least a 26% difference in accuracy between the methods with the final season seeing a 40% difference in accuracy. The full results of this test are shown in Figure 5.

The second set of tests given with this data was fore-

Figure 5: Hindsight prediction results of the three methods used on English Premier League matches.

Season	Decision Trees	Random Forest	Neural Network
2008/2009	71.32%	97.11%	52.63%
2009/2010	72.11%	94.74%	56.32%
2010/2011	60.79%	94.74%	51.32%
2011/2012	58.42%	96.32%	50.53%
2012/2013	55.00%	95.79%	45.79%

Figure 6: Foresight prediction results of the three methods used on English Premier League matches.

Season	Decision Trees	Random Forest	Neural Network
2008/2009	49.47%	50.00%	56.84%
2009/2010	52.63%	38.42%	50.00%
2010/2011	46.32%	41.58%	46.32%
2011/2012	46.84%	37.89%	46.84%
2012/2013	48.74%	48.42%	50.53%

sight prediction. The algorithms were given the same data types, but for this test there was only games completed to that point in the season available. With smaller data sets, the algorithms were predictably less accurate, with neural networks slightly outperforming both the decision trees and random forests. All three methods hovered between 38% to 57% accuracy. Random forest was the weakest of the three methods, with two seasons under 40% [3]. This is an interesting result given the large difference between random forests and decision trees in the hindsight tests. The full results are shown in Figure 6.

The final set of tests involved foresight prediction with a focus on betting profitably. The initial investment made was 1000 units. The random forest performed sporadically. Three of the five season were profitable, with initial investment nearly doubled in 2010/2011 and 2012/2013. But, in 2009/2010 and 2011/2012 nearly all of the initial investment was lost. The full results are shown in Figure 7.

4. NEURAL NETWORKS

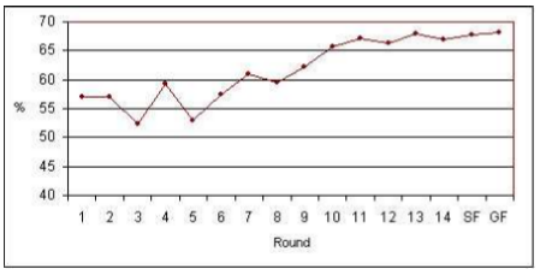
In this section we present the results of two neural network methods tested on professional sports league results. The first is a direct comparison to the random forests on the same English Premier League data. We also give an example of a neural network predicting in three different rugby leagues as well as the English Premier League again using a different data set.

Figure 7: Foresight prediction with emphasis on betting profitability in English Premier League Matches.

Season	Decision Trees	Random Forest	Neural Network
2008/2009	1422 units	1157 units	1368 units
2009/2010	-918 units	54 units	816 units
2010/2011	366 units	1899 units	1387 units
2011/2012	-55 units	99 units	541 units
2012/2013	1815 units	1635 units	2010 units

Figure 8: Top figure shows results on McCabe's neural network on four different leagues tested. The bottom graph shows the prediction accuracy on the Super Rugby season as the season progressed [4].

League	Best	Worst	Average
AFL	68.1%	58.9%	65.1%
NRL	67.2%	52.2%	63.2%
Super Rugby	75.4%	58.0%	67.5%
EPL	58.9%	51.8%	54.6%



4.1 English Premier League

Given that neural networks were used on the same English Premier League match data, it is interesting to see how the two methods compared when using identical data. Back propagation neural networks are used to predict in this example. These types of neural networks are described in Section 2.4.

In the hindsight tests neural networks did not perform nearly as well as random forests or the decision trees. Their highest accuracy occurred in the 2009/2010 season when it predicted just 56.32% of the matches correctly, compared to 94.74% for random forests, which was the worst of its seasons at predicting. The authors attribute the poor performance of the neural network to its confusion between wins and draws [3].

Foresight prediction yielded similar results to hindsight prediction for their neural network method, which means it ranked among the most accurate of the methods. Its best season was 2008/2009 when it predicted with 56.84% accuracy [3]. An interesting result that was found was the neural network method almost never predicted a draw, which didn't really affect its overall accuracy, but distinguished itself from the other methods. When the authors used foresight prediction with betting in mind, neural networks produced the most profitable results, with three of the five years it earned profit, resulting in a solid net profit gain over five years [3].

4.2 Multiple Leagues

The next set of data given to a neural network contained three different rugby leagues. The features used for this data included points-for, points-against, win-loss record, home-away record, previous game result, previous n game performance, team ranking, points-for and against in previous n games, location, and player availability [4]. The main com-

parison made in this example was between human prediction and neural networks. The neural network method used here is a MLP, or *multi-layer perceptron*. MLPs give a weight to the initial set of input features. This weight can be set to an initial value to facilitate an input bias, or they can be random [4]. These weights are changed until there is a minimum error between the target output and actual output from the algorithm. Backpropagation is used to optimize the weights.

Human experts typically predict between 60% and 65% of matches correctly. Neural networks are found to be quite similar in accuracy to the human experts. In the Australian Rules Football League(AFL), their neural network method predicted an average of 65.1% of games correctly. In the Australian National Rugby League (NRL), the accuracy was an average of 63.2%. And in the Super Rugby season it predicted an impressive 67.5% of matches correctly with a peak season of 75.4% [4]. Figure 8 shows the learning process of the neural network in the graph. As the Super Rugby season progressed, the network showed a steady increase in accuracy. One interesting result was that in 2006, two new teams were added to the Super Rugby league. To compensate, all features of the other teams were averaged and added them to the two new teams. Performance for the new teams was normalized in the first two to three weeks (two to three matches) with little effect on the prediction accuracy [4].

5. DISCUSSION

Random Forests performed very well on hindsight prediction. The most obvious application of this comes in scouting, where teams can predict the effectiveness of certain lineup combinations given past performances against the same team. Something not mentioned by the authors is that hindsight prediction could also be useful when thinking about giving players new contracts. If a player is showing signs of decline in a similar pattern to another player or players from the training data, that team might avoid signing the player to a long-term contract given the possibility of a greater decline in performance. Random Forests also performed quite well in the Rugby World Cup, especially in terms of betting. Figure 4 shows the superior performance of betting over the course of the World Cup. The human prediction methods (OddsPortal and SuperBru, which take aggregations of user predictions to output a final prediction) lost all money rather early in the tournamemt, while random forests finished near the initial investment(both methods finished with less than the original investment).

While neural networks struggled with hindsight prediction, they do seem to be more successful in foresight prediction. The most obvious party of interest in foresight prediction would be the gambling market. All of the authors of the different tests said that with more data, the methods will continue to improve. Given the amount of data available to gambling organizations, it is reasonable to assume neural networks can become a trusted method in predicting for bettors. Neural networks also performed very well over multiple different leagues as shown in Section 4.2. As more data became available over the course of the season the algorithm showed marked improvement, highlighting the potential of neural networks accuracy when implemented over many seasons of data.

One interesting idea that was not brought up by any of the authors was the ability for machine learning algorithms to create betting lines. It is in the best interest of bettors and those creating the lines to have the most accurate line possible. If random forest and neural networks could be used to create more accurate lines, it would benefit all parties involved.

There are obviously a lot of interested parties when it comes to predicting sporting events, which brings a lot of questions on the availability of the most accurate methods. Because of the enormous industry that is sports betting, many of the most profitable groups keep their predictions and methods very close to the vest.

There appears to be a promising future for machine learning in sports prediction. With an increase in the availability of data, as well as the embracing of advanced statistics in the athletic world, it seems logical to see computing algorithms gain a stronger foothold in the industry, both for prediction and scouting for individual teams. In each experiment the two methods were shown to be at least as good as current methods, if not significantly more effective in some cases.

Acknowledgments

I would like to thank my advisor, Elena Machkasova for her help and feedback on my paper and presentation, and Andrew Latterner for his feedback on the paper.

6. REFERENCES

[1] M. K. Bhuyan, D. P. Mohapatra, and S. Sethi. A survey of computational intelligence approaches for software reliability prediction. *SIGSOFT Softw. Eng. Notes*, 39(2):1–10, Mar. 2014.

[2] G. Biau. Analysis of a random forests model. *J. Mach. Learn. Res.*, 13(1):1063–1095, Apr. 2012.

[3] G. Kyriakides, K. Talattinis, and S. George. Rating systems vs machine learning on the context of sports. In *Proceedings of the 18th Panhellenic Conference on Informatics*, PCI '14, pages 4:1–4:6, New York, NY, USA, 2014. ACM.

[4] A. McCabe and J. Trevathan. Artificial intelligence in sports prediction. In *Fifth International Conference on Information Technology: New Generations (itng 2008)*, pages 1194–1197, April 2008.

[5] A. Pretorius and D. A. Parry. Human decision making and artificial intelligence: A comparison in the domain of sports prediction. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, SAICSIT '16, pages 32:1–32:10, New York, NY, USA, 2016. ACM.

[6] Wikipedia. Artificial neural network, 2017. [Online; accessed 20-March-2017].

[7] Wikipedia. Backpropagation, 2017. [Online; accessed 6-April-2017].