# Literature Review

## Introduction

With the introduction of new technology, and the surge of interest in using machine learning to help solve problems, the realm of sports result prediction has developed considerably. The need for accuracy when making a prediction is in large parts due to use the of large amounts of monetary interest that exist in the sporting world, particularly in sports betting. This need for artificial intelligence to help solve this problem is due to the inability of humans when it comes to processing large amounts of information (Gabriel Fialho et al., 2019). Although human decision-making ability is powerful, there are underlying issues that lead to errors and biases in forecasting a prediction (Arnu Pretorius & Douglas A. Parry 2016).

Sports prediction is also of great interest to those who have a stake in the sporting team, such as investors (Pakawan Pugsee & Pattarachai Pattawong, 2019). Football clubs would be able to use a machine learning model to find relevant factors that contribute to winning a football match and use this knowledge to improve their teams.

## Machine Learning Models

### Artificial Neural Networks

Artificial Neural Network (ANN) algorithms are heavily inspired by the function of biological neural networks, such as the human brain. ANNs have been used to solve problems such a natural language understanding in our phones to anomaly detection in data. ANNs are perhaps one of the most common approaches to sports result prediction. (Grossberg, 1988). One of the main advantages of using a neural network is the lack of restriction placed onto the network, allowing the network to produce an output that is not limited to the input provided to them, meaning that the data provided can be incomplete or insufficient. (Purucker, 1996).

Purucker (1996) conducted one of the first-ever attempts into predicting the results of a sporting event, in this instance predicting the outcome of NFL (National Football League) matches. A total of five features were used in this model; time of possession, rushing yards gained, turnover margin, yards gained, and betting line odds. First, unsupervised learning methods based on clustering were used to determine which teams were good and which were bad. One of the methods used was a backward propagation (BP) algorithm to estimate the weights of the neural network. (QingCaoa & Mark E.Parry 2009). BP was able to successfully predict 11 of the 14 games (78.6%) in Week16 correctly compared to the Kohonen SOM (Self-Organising Maps) which only predicted 8 of the 14 games correctly. The target vectors used were able to provide more information to the BP model than to the Kohonen SOM model (Purucker, 1996). However, a limitation of this study is that a relatively small number of features were used, and arguably five features alone cannot determine to a higher accuracy the result of a match (Rory Bunker & Fadi Thabtah, 2017).

BP was also used by Davoodi & Khanteymoori (2010) to predict the result of horse races. They used one ANN per horse during a race, using 100 races from the event as input data. The output of the models was the finishing time of that horse. They found that the optimal network architecture was one input layer consisting of 8 inputs/features, two hidden layers, and an output layer predicting that horses' finishing time. They used five different training algorithms on the data they had collected. They observed that with 400 epochs, BP, and gradient descent with a momentum parameter (BPM) shared the highest accuracy of the 5 algorithms, both boasting a 77% accuracy. The downside of using BP however is the lengthy training time, which means that it takes a long time to train the model to make a prediction (Rory Bunker & Fadi Thabtah, 2017). “**applicable to a wide range of areas”.**

Following on from the work of Purucker, Kahn (2003) was able to extend the former's work with greater accuracy. This was achieved by adding new features, such as total yardage differential, turnover differential, rushing yardage differential, away team indicator, and home team indicator. The approach was treated as a binary classification problem. These two classes were the away team outcome and the home team outcome (-1 for a loss, +1 for a win). The results of this study were compared to the predictions of eight ESPN sportscasters. These domain experts predicted an average of 63% of the games correctly (Gabriel Fialho et al., 2019). An issue with this study is that it does not address the likely outcome of a tied game, which would mean that a binary classification approach would not be fully suitable for a game like football where the chances of the outcome being a tie are very high. **What was Kahn’s accuracy?**

Similarly, Igiri et al. (2014) also treated this as a problem which could be resolved using binary classification (win or lose). 110 matches played from the 2014/2015 Premier League season were used as inputs for the neural network. **Present this section better.** A larger range of features was used: Home and Away goals (GHA), Home and Away shots (HAS), Home and Away corners (HAC), Home and Away Odds (HAOD), Home and Away attack strength (HAAT), Home and Away Players' performance index (HAPPI), Home and Away Managers' performance index (HAMPI), Home and Away streak (HASTK), Home and Away managers' win (HAMW). 20 matches from the 10th and 11th week of the same season were used to test the model. An 85% accuracy was observed, with the features of the model being optimised by weighting. This model was arguably more successful than the Purucker model not only due to the greater number of features used but also because this model took the individual players into account by using the HAPPI as a feature.

### Random Forest Classification

A Random forest (RF) is a construction made from a multitude of decision trees, trained, and then predicts an output into a class. RFs outperform standard decision trees as decision trees have the habit of overfitting to their training data (Hastie et. al., 2001), leading to a poorer performance as a result.

## Evaluation Methods