# Soccer Result Prediction Using Deep Learning and Neural Networks

**Sarika Jain, Ekansh Tiwari, and Prasanjit Sardar**

**Abstract** In the present world, the prediction of the results of football matches is being done by both machines and football experts. Football as a game produces a huge amount of statistical data about the players of the team; the matches played between the teams and the environment in which the match is being played. This statistical data can be exploited using various machine learning techniques to predict various information related to a particular football match, namely the result of a particular game, injury of a player, performance of a player in a particular match, and spotting new talents in the game, etc. In this work, previous works are reviewed on the prediction of the outcome of a football match, evaluate the merits and demerits of different approaches and then attempt to design a prediction system powered by Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTMs).

**Keywords** Neural network · Machine learning · Deep learning · Prediction · RNN · LSTM

## 1 Introduction

Football being one of the most popular sports around the globe has a huge amount of fans following the day to day events in the game. The investment involved in football matches is increasing in billions by the end of every season and every team wants to get the best result out of the investments they have made. Multiple big teams and sports organizations depend on data available from previous meetings of the teams as well as the present condition of the team to make proper adjustments to their squads to win the competitive matches with the best results. The people associated with the football teams as well as the followers of the teams often come across instances, where it is hard to guess how their team will perform in a particular game; this is

S. Jain (✉) · E. Tiwari · P. Sardar
National Institute of Technology, Kurukshetra, Haryana, India
e-mail: jasarika@nitkkr.ac.in

J. Hemanth et al. (eds.), *Intelligent Data Communication Technologies and Internet of Things*, Lecture Notes on Data Engineering and Communications Technologies 57,
https://doi.org/10.1007/978-981-15-9509-7_57

where game result prediction systems come into play. Soccer score prediction can be a helpful means to assess the readiness of a team before going into a game so that necessary changes can be made to the team to avoid a defeat. The prediction can also help the management of different football associations in getting their teams ready for the upcoming matches to get the best result possible out of the fixture. Accurate football match outcome prediction is valuable for small and big sports telecommunication companies as it can help them increase their revenue and make the game a lot more interesting for the viewers. They can also be used for finding new talents in the sport and making better investments in the teams' future and better returns for the investments.

There have been several attempts at trying to predict the outcome of a football match but none of them has been able to match the accuracy of the human prediction. Humans are still by far superior in predicting the outcome of the match as they take into account the technicality as well as the emotional factors that affect the outcome of the game; this helps them in predicting well but it has its drawbacks as well. Humans let their emotions overpower them in the prediction which more or less leads to a wrong prediction of the outcome. Various factors affect the outcome of a football match such as the number of scored goals in previous matches by a team, the winning streak that the team is coming to play within the present game, the environment in which the team is playing the current fixture, and many more other factors associated with the current form of the players in the team as well as the current form of the team itself. Not only the outcome of a game but also various other parameters can also be predicted like the performance of a player, injury of a player, and evaluating a game strategy.

This work aims to review previous efforts on the prediction of results of a football match, to evaluate the merits and demerits of different approaches and then attempt to design a prediction system powered by Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTMs) to predict the result of a football match. Increasing the accuracy of the prediction is aimed at by taking into consideration all the events that take place during a football match and their effects on the outcome of the match.

## 2 Background and Related Work

This section familiarizes the reader with the theory that works behind the proper functioning of the result prediction done by the RNN using LSTMs. The section starts with machine learning, deep learning, and neural networks and concludes with RNN and LSTMs. Most of the techniques aims at reducing the randomness with the game.

### 2.1 Machine Learning

Machine learning is a subfield of artificial intelligence whose popularity is growing in the field of computers at a very high rate. It is the field of study that helps a system learns on its own without any explicit intervention. Majorly machine learning is of three types supervised, unsupervised, and reinforcement learning. In *supervised learning,* the system searches for patterns and analyses the data to train itself for different scenarios according to the data provided. Once the system is well trained and starts giving the desired level of accuracy in the output in the training procedure; it is deployed on real-life data. In *unsupervised learning,* the algorithm is not provided with any structure of the data, and it is expected to find a pattern in the given data by itself. *Reinforcement learning* is the ability of the system to determine what is the best outcome taking into consideration the environment.

### 2.2 Neural Networks

The neural network works exactly like the working of the human brain. The human brain consists of nerve cells that are connected by means of axons. An artificial neural network can be considered as a pool of processing units that are connected with weighted links that process the data among themselves. It consists of nodes which are connected by links that send the data processed by one node to the other node. Each node executes a simple task to help the system learn something from the input fed to the system. The core of the neural systems, i.e. the neurons (nodes) is just simple activation functions that take multiple input and process it to give a particular output.

### 2.3 Deep Learning

Deep learning is a branch of learning that makes use of deep neural networks. They can be used to implement great real-world problems which involve complex hierarchy in themselves. While deep learning takes longer time to train the system, it gives a more accurate result after the processing and training of the data is done. One benefit of deep learning is that it can work on a very large dataset with different layers and hence give a better result as compared to other learning mechanisms like SVM and random forest.

### *2.4 Recurrent Neural Networks (RNNs)*

Neural networks have a disadvantage associated with them; that is, the input and output of the network is of fixed length. Recurrent neural networks try to overcome this drawback of ANNs. They allow loops between different nodes to allow the system to remember specific outputs related to some particular features. These loops act like memorizers which remember the data for some time in the system.

### *2.5 Long Short Term Memory (LSTMs)*

LSTMs are units of recurrent neural networks that are designed to remember the values for a long or a short period of time. If the LSTM feels that a feature is important in deciding the output, then it carries the feature in the memory for a longer period of time. The recurrent units in LSTM do not use activation function at every node so the data does not get squashed by the system by getting processed repeatedly.

### *2.6 Comparative Study*

Zdravevski and Kulakov employed classification techniques that can be found in WEKA to predict the winner of a game. Parameters that were used were, namely the no. of players injured in the team, winning streak of the team before playing the game, the fatigue of the team before playing the game, win percentage of the teams, and offensive and defensive ratings of the team [1]. Lam in their research has used a Bayesian linear regression model, Gaussian process regression model, and Sparse spectrum Gaussian process regression model to calculate the winning probabilities of a match. They use 11 attributes in total to infer the team's strength and the player's ability [2]. Igiri in their research used Gaussian combinational kernel type and generated a total of 79 support vectors at a count of 100,000 iterations [3]. Rotshtein et al. in their research propose a model which predicts the result of an upcoming match using the method of identifying non-linear dependencies in the knowledge base. Five levels were defined to construct a prediction model based on fuzzy logic, namely loss with a high score, loss with a low score, game with a draw, win with a low score, and win with a high score. To apply the fuzzy knowledge base a fuzzy knowledge approximator was used [4]. Pettersson and Nyquist used LSTMs in recurrent neural networks to predict the outcome of an ongoing football match. They considered all the players playing in the team to be at the same advantage and of the same capabilities [5]. In their system, Cui et al. [6] used 25 different variables related to a football game as terminal sets, and the fitness measure is set as the total number of the wrong prediction of the games.

**Datasets**: While a great deal of data is available related to football games across the globe, the data is not always useful or in accordance with the requirements of the researchers. Many times the researchers have to gather the data manually or through various sports companies or site to work upon. Zdravevski and Kulakov [1] used the data of two back to back seasons of National Basketball Association League from the basketball reference website. Max Lam [2] used the data of the 2014/15 season of NBA from the database available on the NBA site [7]. Igiri [3] in their research used the data available for all the seasons of the English Premier League [8] as their dataset to work upon. Rotshtein et al. [4] used the data of the championship of Finland. The dataset in Pettersson and Nyquist [5] included matches from leagues and tournaments across the globe from 54 different countries that come under the Union of European Football Associations (UEFA) [9] as well as the countries of American and Asian origin. Cui et al. [6] used the official site of the English Premier League [8]. Table 1 summarizes the just discussed works, i.e., references [1–6].

**Table 1** Comparing approaches

| Author | Accuracy | Remarks |
|---|---|---|
| Zdravevski and Kulakov [1] | The accuracy of reference classifier was comparatively low when compared to any other classifier (around 5-10% low) | The data was collected from the official NBA source, and hence, the data was more accurate. Low prediction accuracy |
| Lam [2] | TLGProb had an accuracy of 85.28% in NBA 2014/2015 season | TLGProb performs well when compared to the existing NBA predictive models. A small dataset was used |
| Igiri [3] | Prediction accuracy was 53.3% | Can handle nominal data. Low prediction accuracy. Does not support large datasets |
| Rotshtein et al. [4] | 85% accuracy in genetic tuning and 84% accuracy in neural tuning | High accuracy. Due to insufficient learning samples, the tuning of this kind of system is a very difficult task |
| Pettersson and Nyquist [5] | The accuracy was 96.63% for many to one approach and 88.68% for the many-to-many approach | High accuracy. The dataset comprised of only 2–3 seasons of the leagues and tournaments |
| Cui et al. [6] | Testing accuracy was 56%, and the overall accuracy of the system was near to 70% | Low accuracy. The accuracy rate of each function is not the same |

### 2.7 Discussion

Taking into account the popularity of the game in the present world, many organizations will be willing to invest a huge amount of money in the prediction systems for the better performance of their teams. As interesting as it may seem, prediction of the results of a football game is a very hard task and involves a large amount of uncertainty. However, it can be said that the result of football is not a completely random event, and hence, a few hidden patterns in the game can be utilized to predict the outcome. Based on the studies of numerous researchers that is being reviewed in our study as well as of those done in the previous years, one can say that with a sufficient amount of data an accurate prediction system can be built using various machine learning algorithms. While each algorithm has its advantages and disadvantages, a hybrid system that consists of more than one algorithm can be made that can increase the efficiency of the system as a whole. There also is a need for a comprehensive dataset through which better results can be obtained. Experts can work more toward gathering data related to different leagues and championships across the globe which may help in better understanding of the prediction system. Moreover, the different features of a footballer, as well as that of the team, can also be taken into consideration while predicting as this may produce a better result as compared to when all the players in a game are treated to be having an equal effect on the game.

RNNs gave a better result when compared to the other techniques for predicting the outcome of a football match. The more information the system was fed the more accurate the result of the RNN system showed. Hence, RNN as the technique to be used in our prediction system is chosen.

## 3 Operational Analysis

Previous works on predicting the results of football matches with machine learning techniques have mainly focused on the data available about the team. The focus has only been limited to a small number of leagues as well. The approach of this work is to deduce better features from the results of the previous matches that the team has played and taken into consideration the current form of the team. A long short term memory system is exploited for this work.

### 3.1 Dataset

The dataset for the project has been taken from "http://football-data.co.uk/data.php." The dataset contains the data of the seasons from 2010–11 to 2017–18 of the English Premier League. The advantage associated with this dataset is that the number of matches played by each team, and the total number of matches in the tournament is fixed. This was beneficial in removing unnecessary information from the system. In each dataset, there are around 380 records each which are in chronological order with around 60 attributes. From the given dataset, manual feature generation was done to get new attributes. For example, the values from the "Full Time Away Goals" (FTAG) and "Full Time Home Goals" (FTHG), two new attributes "Home Team Goals Scored" (HTGS), and "Away Team Goals Scored" (ATGS) were calculated by summing over the attributes through all the rows. All the contenders in every match had their form calculated by finding the winning streaks of the teams and recording if any team had won five games in a row or three games in a row. Finally, one hot encoding was done for classification purpose. The various rows that were considered in the final dataset are given below:

| *HTGS—Goal scored by the home team* | *HTGC—Goal conceded by the home team* |
|---|---|
| *ATGS—Goal scored by the away team* | *ATGC—Goal conceded by the away team* |
| *HTGD—Goal difference of the home team (HTGS—HTGC)* | *HM1(—4) —Results of the last four games won by the home team.* |
| *ATGD—Goal difference of the away team(ATGS—ATGC)* | *AM1(—4)— Results of the last four games won by the away team* |
| *HtLossStreak3—loss streak of the home team with three games (Hat trick)* | *HtWinStreak3 —win streak of the home team with three games (Hat trick)* |
| *AtLossStreak3—loss streak of away team with three games (Hat trick)* | *AtWinStreak3—win streak of the away team with three games (Hat trick)* |
| *HtWinStreak5—win streak of home team with five games* | *HtLossStreak5—loss streak of the home team with five games* |
| *AtWinStreak5—win streak of away team with five games* | *AtLossStreak5—loss streak of away team with five games* |
| *HTP—Points gained by the home team* | *HTGDGoal difference of the home team* |
| *ATP—Points gained by the away team* | *ATGD—goal difference of the away team* |
| *DiffPts-(HTP—ATP)* | *DiffFormPts— HTFormPts—ATFormPts* |

## 3.2 Methodology

There are a few parameters in every neural network that won’t change throughout the working of the model even when the calculation of the accuracy of the model is done. Such parameters are known as hyper-parameters. Once the model has made use of a certain set of hyper-parameters these parameters can be played around with to get a better understanding and accuracy of the model. RNNs can not only remember the sequence in one instance, but also the sequence of the various instances coming in because the nodes storing the weights, and the activation functions are the same. With respect to football match outcome prediction, the dataset is a relational database.

Also, the order of the various columns is unimportant as it doesn't affect the outcome of the match.

LSTM cells are a modification of RNNs. LSTM cells implement RNNs, and with that, they perform an additional calculation with previous output from the RNN cell and the new input to generate the new output. LSTM cells are mainly composed of three gates:

- Forget gate: This gate is known as the sigmoid function. It is responsible for forgetting information that is not required anymore by the system when moving to the next sequence.
- Input gate: This gate works as an input provider to the network. It uses tanh function as well as the sigmoid function.
- Output gate: This gate is responsible for giving the output that is calculated by the processing that is done at the previous two gates.

The given model is implemented with the help of TensorFlow library in python. All the data in this library is processed in the form of an array. A code written in Tensorflow works in two steps. Initially, a computational graph is formed, where all the connections among various layers of the neural network and the calculations of the weights are done but none of these variables has values. They are initialized with Tensorflow objects without any numerical values. The second step is the execution, where a session function is run, and all the variables that have to be calculated are passed as parameters. This generates numerical values that are assigned to the variables.

## *3.3 Implementation*

The code for the implementation of LSTM and RNN for the prediction can be seen below. The hyper-parameters include:

(1) n classes—total number of output classes.
(2) batch size—number of rows fed in one iteration to the model.
(3) hm epochs—number of epochs the model runs for.
(4) chunk size—number of attributes in each chunk.
(5) n chunks—number of chunks in one instance of data.
(6) rnn size—number of LSTM cells in the hidden layer of the RNN.

*Pseudocode*

```
Step 1: import the necessary libraries
```

*import Tensorflow as tf*

*from tensorflow.contrib import rnn*

*import pandas as pd*

***Step 2:*** *import the one hot dataset using pandas*

***Step 3:*** *define the hyper-parameters*

*hm_epochs = 10*

*n_classes = 2*

*batch_size = 1*

*chunk_size = 27*

*n_chunks = 1*

*rnn_size = 512*

***Step 4:*** *Create the RNN model with LSTM cell.*

***Step 5:*** *Feed the system with data and LSTM cell mechanism.*

***Step 6:*** *Train the model using the function created in step 4.*

### 3.4 Results and Discussion

The experiment has been performed on the above dataset using a basic LSTM cell from the TensorFlow library. The model was run on a various set of hyper-parameters to find out the best model. Table 2 shows the various train and test accuracies with chunk size = 3 and n chunks = 9, for batch sizes from 1 to 124. The model with chunk size as 27 was considered for further analysis as depicted in Table 3.

**Table 2** RNN with chunk size 3 and n chunks as 9 and varying batch size

| Batch size | 1 | 30 | 60 | 124 |
|---|---|---|---|---|
| Accuracy train | 0.922043 | 0.6973118 | 0.6655914 | 0.6704301 |
| Accuracy test | 0.79875 | 0.6525 | 0.64625 | 0.6525 |

**Table 3** RNN with chunk size 27 and varying batch size

| Batch size | 1 | 30 | 60 | 124 |
|---|---|---|---|---|
| Accuracy train | 0.9811828 | 0.7688172 | 0.74139786 | 0.7354839 |
| Accuracy test | 0.8075 | 0.69875 | 0.69375 | 0.69 |

**Table 4** RNN with different sizes of the hidden layer

| RNN Size | 32 | 64 | 256 | 512 |
|---|---|---|---|---|
| Accuracy train | 0.9731183 | 0.97043014 | 0.9919355 | 0.9892473 |
| Accuracy test | 0.8125 | 0.805 | 0.805 | 0.8025 |

Table 4 shows the accuracies of the model with varying size of the hidden layer. Here, for increasing RNN size, though the training accuracy is increasing, there is no sufficient increase in the test accuracy. This suggests that the model is overfitting the data. Hence, for the present dataset, the best model values are as follows:

| *n classes—2* | *batch size—1* | *hm epochs—10* |
|---|---|---|
| *chunk size—27* | *n chunks—1* | *rnn size—64* |

## 4 Conclusion

The popularity and international effect that football has made it an interesting problem to solve. Moreover, the number of factors that affect the outcome of a match is enormous. From the results can be said that RNNs with LSTM show a visible and obvious advantage over the original ANN and the traditional machine learning. Hence, other than the mainstream uses of LSTMs, this particular path of using it for prediction of the outcome for sporting events has also shown promising results. This model can still be improved. One way to do that is, by using a better set of attributes. They can include statistics of each player. This can also help in predicting the form of a particular player from season to season.

## References

1. Zdravevski E, Kulakov A (2009) System for prediction of the winner in a sports game. In: International conference on ICT innovations, pp 55–63. Springer, Berlin, Heidelberg
2. Lam MW (2018) One-match-ahead forecasting in two-team sports with stacked bayesian regressions. J Artif Intell Soft Comput Res 8(3):159–171
3. Igiri (2015) Support Vector machine—based prediction system for a football match result. IOSR J Comput Eng (IOSR-JCE) 17(3):21–26
4. Rotshtein AP, Posner M, Rakityanskaya AB (2005) Football predictions based on a fuzzy model with genetic and neural tuning. Cybern Syst Anal 41(4):619–630
5. Pettersson D, Nyquist R (2017). Football match prediction using deep learning. Doctoral dissertation, Master's thesis, Chalmers University of Technology
6. Cui T, Li J, Woodward JR, Parkes AJ (2013) An ensemble based genetic programming system to predict English football premier league games. In: 2013 IEEE conference on evolving and adaptive intelligent systems (EAIS), pp 138–143. IEEE

7. https://in.nba.com/?gr=www
8. https://www.premierleague.com/
9. https://www.uefa.com/