# Football Match Statistics Prediction using Artificial Neural Networks

[1]K. Sujatha, [1]T. Godhavari and [2] Nallamilli P G Bhavani
[1]Dept of Electrical and Electronics Engineering/CSE Dept, Center for Electronics Automation and Industrial Research (CEAIR), Educational and Research Institute University, Chennai, India,
drksujatha23@gmail.com
[2]Electrical and Electronic Engineering, Educational and Research Institute University, Chennai, India,

*Abstract*-The predictions of the outcomes of football (American soccer) matches are widely done using the if-else case based Football Result Expert System (FRES). The proposed prediction technique uses a neural network approach to predict the results of football matches. The neural network detects patterns from a number of factors affecting the outcome of a match making use of historical cases of training. This paper describes the inputs, outputs and compares the results of this kind of a system.

*Keywords:* - Artificial Neural Networks, Back propagation, sport result prediction, pattern prediction, FRES, Activation function

## 1 Introduction

To predict the outcome of a match between two teams, a person will generally take into account certain factors like the recent performances of the teams, whether the match is going to be played at home or away, recent player transfers, recent coach and staffing changes, etc. This is what human 'intuition' is made up of, a decision based on a combination of solid tangible facts available to the person making the decision. The problem when a human predicts the outcome of a match is that the persons decision will be influenced by factors like the humans' preference in teams, the humans' perception of certain players in a team, some studies show decisions may even be influenced by the color of the kit of the team. [1]

One of the methods widely used for predicting match outcome is the Football Result Expert System used widely by media and book makers. The problem with FRES is that it uses a rule based approach or an if-else statement based approach that considers and rejects various factors one by one.

The approach described in this paper takes advantage of the fact that neural networks are good at recognizing patterns and mapping these patterns to outputs. The approach proposed in this paper uses a number of factors, some which may even not be available to a standard human decision maker as inputs to the neural network. The neural network is given a number of previous performances of the teams in question as a training set after which it is used to predict a match in the future.

## 2 Input factors considered

The input factors are considered for a Bundes Liga (German League) match between FC Bayern Munich and FC Borussia Dortmund. Training data available is for all meetings between the two sides in the Bundes Liga during the period 2005 to 2011, making a total of 14 matches. The various inputs considered and their effects on the outcome of the match are as follows:

### 2.1 Transfer money spent

Generally, twice during a season (during the transfer window) the teams are allowed to buy or sell players to change the makeup of a team. The indication is that if a team spends more money on transfers then the team is looking to improve its bench strength or team makeup. Historically the performance of a team generally tends to improve after spending during the mid-season transfer window. Money spent during the transfer season is directly normalized as shown and used as an input to the neural network.

Generally, the first step for normalization involves applying a median filter to normalize data as in equation 1. [2] This step was not performed for the simple reason that there were not enough training sets available to allow for this process of elimination of outlying data. This lead to certain mid-range values getting squashed, but was acceptable since the two teams in consideration, Borussia Dortmund and FC Bayern Munich generally tended to be among the top spenders.

The formula used for normalization is

$$v' = (v - \text{average}) / SD \tag{1}$$

where v' – normalized value, v – previous value, average – average of the training set, SD – standard deviation of the set.

## 2.2 UEFA co-effecient

UEFA is the European football governing body that organizes the annual Champions League in which all the top European teams appear. Even though the Champions league is not related to the German Bundes Liga, this rank has been taken into consideration since both Bayern Munich and Borussia Dortmund have been regular fixtures in the league since 2005 [3]. UEFA rank is given by the "Champions league point system". A certain number of points are awarded to a team at each stage of the Champions league as given in Table 1.

**Table 1: Champions league point system**

| S.no | Stage | Points earned |
|---|---|---|
| 1 | 1st qualifying round elimination | 1 |
| 2 | 2nd qualifying round elimination | 2 |
| 3 | Group stage participation | 4 |
| 4 | Group stage win | 2 |
| 5 | Group stage draw | 1 |
| 6 | Round of 16 participation | 4 |

Since 2009 onwards, an additional point is awarded to a team taking part in the round of 16. Eliminated teams are not awarded points since they move to the second tier Europa league and earn points there.

The above system is used for awarding points to a team taking part in the Champions league. The total team co-efficient is given by the sum of points earned by the team in the past five years plus 33% of the association co-efficient during the same period. This results in the UEFA coefficient of a team increasing with time.

## 2.3 League rank

The position of the team in the league standings is an indication of their performance so far, and as such is used as one of the inputs to the neural network. The number of teams in the Bundes Liga has remained constant at 18. Normalization is done by simply dividing position of team in the league by 18.

## 2.4 Goals scored and conceded

Goals scored are used as a number of inputs to the neural network. The categories these come under include:

- Away goals scored – Goals scored by a team playing away fixtures
- Away goals conceded – Goals conceded by a team playing in away fixtures.
- Home goals scored – Goals scored by a team in home fixtures
- Home goals conceded – Goals conceded by a team playing home matches.

Goals scored either home or away have to be considered for all matches prior to that meeting in the league. It was found that most sites kept a record of goals scored in all formats including the Champions league and the German cup. Care should be taken to ensure that only records pertaining to that league are considered as the number of matches played may otherwise tend to be different for the two teams.

## 2.5 Team cost

Each player on the team is given a figure that is indicative of how much the club would have earned from the sale of that player to another club. [4]
The sum of all the individual player costs gives the team cost. These values were found to vary between 2005 and 2012, reflecting changes in the value of the Euro.

## 2.6 Year of match

The inclusion of year of match is found to improve the performance and the prediction capabilities of the neural network. This is analogous to a human being coming to the conclusion that the recent performance of the team is more important than past performances. In simple terms Borussia Dortmund is on a rising trend with more and more victories than Bayern Munich while in the past Bayern Munich was the superior team. [5]

The neural network was amazingly able to pick up on this trend and predict more accurately the recent matches played out between the two teams. Normalization of the year of match is easy since the upper and lower bounds are fixed with a difference between the maximum and minimum value less than ten. For future predictions, the value used for normalization will have to be changed appropriately

## 2.7 Wins and losses

Home and away wins, home and away losses and home and away draws are very vital inputs to the neural network. These statistics are to be considered for the league upto the point of the match. Home and away performance are considered separately since there was found to be a difference in the home and away performance of a team, with a team generally performing better in home matches than in away matches [6].

## 2.8 League points

League points awarded in the Bundes Liga are as shown in Table 2.

**Table 2: League point award system**

| S.no | Number of points | Result |
|---|---|---|
| 1 | 3 points | Win |
| 2 | 1 point each | Draw |
| 3 | 0 points | Loss |

The points are used to rank the team within the league. The final standings in the league are given in decreasing order of points earned by a team, meaning the team that wins the maximum points takes the top position in the league while the team with the minimum number of points takes the bottom position in the table. Points earned till the match day in consideration is used as one of the neural network inputs.

### 2.9 Home advantage

Home advantage is given by the last two nodes. These nodes are used to indicate which team has the home ground advantage. For example, if the match is being played at home for team A, the input for 'home ground A' node will be 1 and correspondingly will be 0 for 'home ground B' input node.

### 2.10 Factors rejetced

A number of factors that, if were taken into consideration would improve the performance of the predictor, had to be rejected as being too impractical included the following:

- Possession – ball possession by a team is generally an indication of the performance of the team as a whole.
- Passing – passing gives an indication of how well the team is performing as a unit
- Clean sheets – Clean sheets are credited a team that hasn't conceded any goals in a match. The number of clean sheets in a league prior to a match gives a fair indication of that teams' defense performance.
- Bookings – Bookings are of two types, yellow or red. A yellow card is shown for a minor infraction, while two yellow cards in a match amount to a player being red carded whereupon he will no longer be allowed to participate in the match and his team will be reduced to ten players. A team which has a large number of bookings may indicate an inferior team.

The inclusion of the above factors would have benefited the prediction capability of the neural network but had to be rejected since records of these values could not be found even upon considerable searching.

## 3 Output of the network

More than predicting the winning team, the network also attempts to predict a number of other factors given below. This was met with a certain degree of success as illustrated in the results section re-enforcing the idea that a neural network is as if not more capable than a human at providing unbiased predictions and at detecting complex patterns.

### 3.1 Match Winner

Match winner is provided by the neural network as a percentage. One output provides the win percentage of team A and another separate output the win percentage of team B. For training, if team A has won the match, the input team A will be 1 while the input team B will be 0 and vice-versa.

The resulting predictions therefore are not clear cut true and false cases (or 1s and 0s), but instead give a value between 0 and 1 which when multiplied by 100 gives the prediction of a team A or a team B win. For all purposes, the team showing the greater win percentage can be thought of as the ultimate winner.

### 3.2 Goals scored

Goals scored are another two outputs of the neural network. The neural network is expected to predict the goals scored based on past patterns resulting in certain number of goals scored. The performance of the neural network for prediction of goals scored was found to be as good as a human guess. That is, close at best and way of mark at worst.

In cases where the win percentage of the two teams was close to one another (indicative of a drawn match), the goals scored would often be the same for both teams indicating that the neural network is able to make associations between goals scored and other factors such as previous goals scored and conceded and previous wins and losses.

### 3.3 Bookings

The ability of the neural network to predict the number of red and yellow cards shown to each team was found to be close at best. The reason for the neural network performing so badly at predicting the number of cards is because the number of cards depends on a lot of intangible factors that cannot be predicted like, player temperament, referee temperament and match day ground conditions.

The neural network used was the standard back propagation neural network as explained below. A few features were used to enhance the performance of the network such as the addition of logarithmic and exponential input and output neurons. In practice, any kind of neural network algorithm can be used to model a predictor, but back propagation was found to provide the best controllability in terms of rate of learning and generalization capabilities of the neural network.

### 3.4 Backpropogation Algorithm

A back propagation neural network was written in MATLAB for a single hidden layer. The inputs are assumed to be stored in a column wise matrix, and the outputs of the training set likewise in a column wise matrix. The number of nodes in the hidden layer is the same as the number of input layer nodes. The BPN performs updating operations taking the entire input matrix into consideration. The algorithm for back propagation is as follows:

Step 1: Inputs and outputs are normalized as explained above. Total number of inputs to the network is given by l = $i_O$×3 and number of outputs is given by n = $j_O$×3, where $i_O$ and $j_O$ are the number of inputs and outputs. The number of hidden layer neurons m equals the number of input neurons i.e. m = l.

Step 2: Let 'nTest' be the number of training sets. This means the size of the input matrix will be lxm and the size of the output matrix will be nxm.

Step 3: The input and output pattern are stored in rows of the input and output matrices leaving two places for the augmented neurons. Every second row of the input and output matrix is given by computing the ln ($\log_e$) of the preceding row to give the logarithmic neuron. Every third row of both matrices is computed by the exponential of the original pattern.

Step 4: Assign learning rate and momentum factor to some initial value.

Step 5: Initialize the input layer – hidden layer weight matrix v (lxm) and the hidden layer – output layer weight matrix w (mxn) to some random values.

Step 6: Let the thresholds given by delv and delw, both be zero matrices initially.

Step 7: Variable 'iterate' is used to store the number of iterations that training is going to take place for.

Step 8: Since the input neurons use a linear activation function, output of input layer '$O_i$' is made equal to input to input layer '$I_i$' for each pattern (stored as a column).

Step 9: Input to the hidden layer is calculated by multiplying the output of the input layer with corresponding weight values. That is, $I_h = v' \times O_i$, where '$I_h$' represents the input to the hidden layer and is a column matrix of length m.

Step 10: Hidden layer outputs '$O_h$' are calculated using the sigmoidal function as shown-

$$O_O = \frac{1}{1+e^{-I_o}} \qquad (2)$$

Step 11: Target output '$T_o$' (nx1) is calculated from the output matrix by taking the appropriate column.

Step 12: Error is calculated in two steps. First, the part error ePart is calculated as-

$$ePart = \sum (T_O - O_O)^2 \qquad (3)$$

Final error is given as Root Mean Square (RMS) value of ePart or $E_{RMS} = \frac{\sqrt{ePa\imath}}{n}$ (4)

Step 13: Calculated output 'Y' (mxn) is given by $Y' = O_h \times d'$ . Where d is given by $d = (T_O - O_O) \times O_O \times (1 - O_O)$.

Step 14: 'delW' is updated using the formula given as $delW = (momentum \times delw) + (learningRate \times Y)$

Step 15: The complete training set 'nSet' error is calculated as $e = w \times d$ .

Step 16: 'X' is calculated as $X = O_i \times dStart'$ . Where $dStar = e \times O_h \times (1 - O_h)$ . (5)

Step 17: Change in input layer weights is given by $delv = (momentum \times delv) + (learningRate \times X)$ and weights are adjusted as $v = v + delv$ and $w = w + delw$ .

Step 18: Repeat steps 9 to 18 until the error rate is lesser than tolerance value. Save the weights and exit.

The given algorithm explains the learning algorithm used and how the weights have been adjusted.

### 3.5 Number of hidden nodes

The number of nodes in the hidden layer is generally kept to the minimum required to classify all input pattern areas accurately. This is done to keep the memory requirement of the nodes to a minimum. The number of separable regions in space is one less than the number of hidden nodes. It is also argued that the number of hidden nodes should be one less than the number of training sets.

Based on this, the number of training sets is found to be13 for matches played between 2005 and 2011, meaning that the number of hidden nodes should be 12 (given as 13-1). But, since the number of inputs being passed to the network is 29, or 87 taking into account logarithmic and exponential neurons, the number of hidden nodes is kept at 87 to ensure separability of each region within the training set.

General rule of thumb dictates that the number of hidden layer nodes for a single hidden layer feed forward network equals the mean (rounded up) of the number of input layer and output layer nodes.

### 3.6 Logarithmic and exponential neurons

Logarithmic and exponential neurons are generally used to augment the training capabilities of the network. For every one input neuron, its logarithmic and exponential neurons are computed as shown and passed as additional inputs to the network. On the output side also, the logarithmic and exponential neurons are calculated and used for every output neuron.

These neurons are collectively termed as augmented neurons. The addition of these neurons provides accurate boundary mapping and increases the speed of training. These increase the accuracy of the neural network at the cost of the generalization capabilities.

Though these neurons play no role in the actual inputs and outputs of the network, they are used to modify the interconnecting weights of the neural network and therefore improve training.

## 4 Performance Parameters

### 4.1 Input nodes

The list of inputs given to the neural network (given separately for team A and team B):

- Transfer money spent A, B
- UEFA ranking A, B
- League position A, B
- Away goals scored A, B
- Home goals scored A, B
- Away goals conceded A, B
- Home goals conceded A, B
- Player cost A, B
- Year of match
- Away wins A, B
- Home wins A, B
- Away losses A, B
- Home losses A, B
- Total draws A, B
- League points A, B
- Home ground A, B

The above given 29 inputs are used with the network. Addition of augmented neurons gives a total of 87 input neurons.

### 4.2 Output nodes

The following are the outputs of the neural network:

- Match winner A, B
- Goals scored A, B
- Yellow cards A, B
- Red cards A, B

The total number of bookings or cards shown for each team is given by the sum of the corresponding teams red and yellow cards.

### 4.3 Selection of learning rate

Experimentally, the optimal learning rate is found to be 0.7. As shown below, a learning rate of 0.7 provides the best convergence of error rate in 500 iterations.

### 4.4 Selection of momentum factor

When momentum factor is zero, the neural network takes 854 iterations to achieve a convergence of 0.02. The most suitable momentum factor which does not make the neural network overshoot its minima is found to be 0.9. The figure given below compares the error rate for different values of momentum factor.

### 4.5 Generalisation

Often the problem with training for a number of fixed iterations or training with the goal of reducing error is that the neural network will lose its generalization capability. That is, since the error will now be small, the borders will be clearly defined and fresh data that falls just outside a border will be disregarded even though it may belong to that data set.

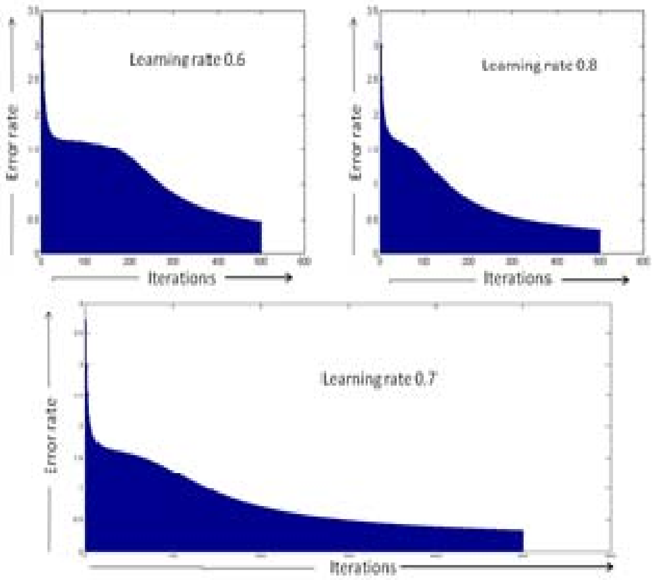


**Figure 1: Comparison of learning rates**

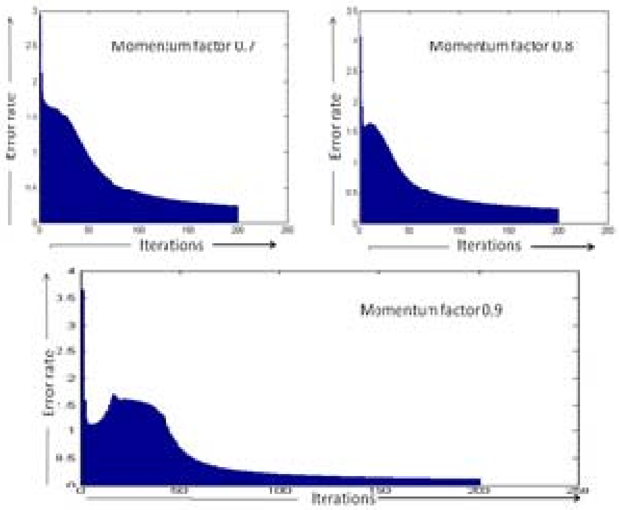


**Figure 2: Comparison of various momentum factors**

The ideal solution involves classifying all training data with minimum error, while at the same time maintaining a minimum error for fresh data. In order to decide when to stop training, 3 out of 13 training patterns are used as the control group. Training is done on the remaining 10 patterns while error is calculated at the end of each set of weight iterations for the 3 control patterns. It is found the error value for the control set initially decreases along with decrease in overall error of the training set, but starts to increase at a certain point. This point is the minimum error for the control pattern after which it has started to lose its generalization capabilities.

The 3 control patterns are selected randomly from the given input test patterns, choice of different control patterns is found not to noticeably affect the neural network.

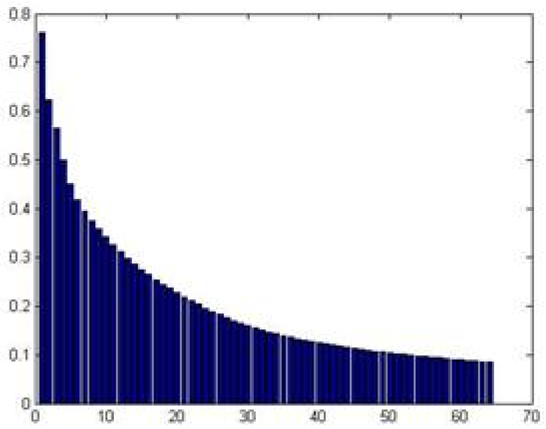

**Figure 3: Error rate of training set for 65 iterations**

# 5 Prediction Capabilities

## 5.1 Teams considered

The two teams used for testing the prediction capabilities of the neural network are FC Bayern Munich and FC Borussia Dortmund. Both teams perform in the top flight German league the Bundes Liga.

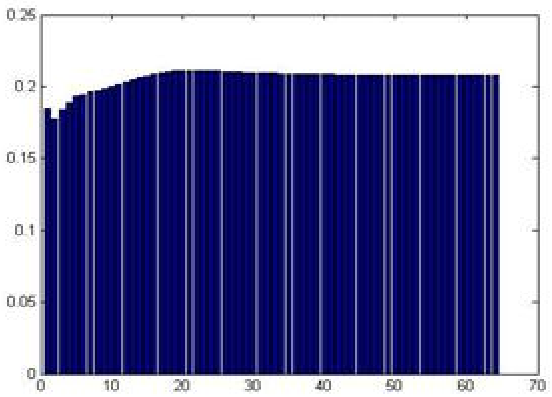

**Figure 4: Error rate of control set in 65 iterations**

Both teams have also participated at every Champions league since 2003 and feature in their ranking system. One reason that two top teams have been selected is that in matches involving a weaker team and a stronger team, the result is generally unpredictable, both Bayern Munich and Borussia Dortmund are top teams and matches between them are keenly contested. The other reason is that it is much more practical to select two well known and popular teams since past statistics of these teams will be easy to locate.

In practice, this method of sports prediction can be applied to any two teams, including for instances where a good team is playing against a weak team. During such matches, the better team tends to win by a smaller margin [7] than expected due to factors such as the 'underdog factor' where the weaker team performs much better than expected against the stronger team. In such a case, the neural network has to be trained separately for such matches using suitable historical test cases, so that the neural network makes the association for a reduced goal margin.

This method of prediction is also appropriate for matches in other ball sports. For example, such a model is readily applicable to predict the outcome of a hockey match with more or less identical inputs and outputs.

## 5.2 Matches predicted with actual result

The predictions of goals scored by each team by the neural network have been compared against actual results in the table given below.

**Table 3: Predicted results Vs actual results**

| Date | Predicted result | | Actual result | |
|---|---|---|---|---|
| | Munich | Dortmund | Munich | Dortmund |
| 13/08/2012 | 2.3 | 0.6 | 2 | 1 |
| 13/05/2012 | 1.5 | 3.3 | 2 | 5 |
| 12/04/2012 | 0.7 | 1.3 | 0 | 1 |
| 20/11/2011 | 0.2 | 0.8 | 0 | 1 |

## 5.3 Matches predicted against human guesses

To test how the neural network predictions compare to human predictions, the results of the neural network were compared against book maker odds. Book maker odds are figures released by bookmakers or betting managers that represent the chance of a team winning the match as a fraction. The fraction represents how much a better can expect to earn for backing a particular team. Conversion of the fraction into each team's odds involves multiplying by the denominator for both teams.
For example, if Bayern Munich is given odds of 1/7, it means that for every seven rupees put on Bayern Munich, there is a chance of earning back one rupee. This means that the booker thinks Dortmund have a one in seven odd of winning or a 14.3% chance of winning. The below table is a comparison of booker odds with predicted win percentages to test how the neural network compares to human intuition.

**Table 4: Predicted results Vs Booker odds**

| Date | Predicted result | | Booker odds | |
|---|---|---|---|---|
| | Munich | Dortmund | Munich | Dortmund |
| 13/08/2012 | 70% | 40% | 20% | 80% |
| 13/05/2012 | 10% | 95% | 20% | 80% |

| | | | | |
|---|---|---|---|---|
| 12/04/2012 | 44% | 72% | 50% | 50% |
| 20/11/2011 | 43% | 80% | 66.7% | 33.3% |

Note that the sum of percentages of prediction of either team winning may not be equal to 100%. This is because the neural network has not created the connection that sum of returned percentages should be equal to 100%, as the training pattern makes use of a 100 - 0 relationship to indicate a winning team.

### 5.4 Comparison with FRES

FRES makes use of a rule-based network akin to an if-else system to predict the outcome of the match. FRES is found to be very successful in predicting the final outcome of the match closer to match completion. This is because FRES divides the match into a number of time segments and inputs are used for each given time segment.

The FRES approach involves factors such as emotional state (depending on current score line), the teams' offensive and defensive capabilities etc. That is, it is more of a here and now kind of system used to predict matches based on numerous current factors. What it lacks is the capability to make deductions based on past factors or performances of the team.
When FRES was applied to predict the result of a match it was found that given the first half data input, FRES would predict the full time winner and score line very accurately. The performance of FRES is better in this regard when compared to the system of prediction using neural networks. It is also found that the use of neural networks to predict matches before the start of the match produces results more accurate than those of FRES. Another advantage is that this approach can predict matches weeks even months in advance, whereas FRES only starts predicting matches once they have begun.

**Table 5: Predicted RMS error Vs FRES RMS error**

| S. No. | FRES RMS error | | Prediction RMS error | |
|---|---|---|---|---|
| | Goals | Winner | Goals | Winner |
| 1 | 6.593 | 6.108 | 1.500 | 6.403 |
| 2 | 5.920 | 5.306 | 1.772 | 9.014 |
| 3 | 6.809 | 6.358 | 0.761 | 2.280 |
| 4 | 5.702 | 6.212 | 0.283 | 5.237 |

## 6 Conclusion

From the table it is clear that this method of prediction is much better at predicting goals than FRES. This is mainly because of the vast number of factors taken into consideration that give an inkling of the defensive and offensive capabilities of the two teams in question. In the case of predicting the winner, the system is found have an RMS error value slightly more than the FRES system of prediction. Another feature of the system of prediction is that the later the match predicted, better the accuracy. This clearly highlights the fact that this method of prediction is a learning based method of prediction and that the system predictions improve with more test cases.

**CITATIONS**

1. "The jersey colors losers wear", www.news.menshealth.com/the-jersey-colors-winners-wear/2012/05/02/
2. "Normalizing NN data", www.elitereader.com/vb/showthread.php?threadid=118179
3. "UEFA club coefficients", www.uefa.com/memberassociations/uefarankings/club/index.html
4. "Bundesliga stats", www.bundesliga.com/en/about/questions/marketing/
5. "Bayern Munich Wikipedia page", en.wikipedia.org/wiki/FC_Bayern_Munich
6. "Home advantage", en.wikipedia.org/wiki/Home_advantage
7. "How teams rise to the occasion", http://www.soccerhelp.com/Soccer_Tactics_Weak_Teams.shtml

**ACKNOWLEDGEMENTS**

The author would like to thank Mr. Arun Prakash, manager and coach of Andrew football club, Hyderabad for his valuable guidance on factors affecting a teams' performance in a match. The author also owes a debt of gratitude to a number of Bundes Liga fans on the Bundes Liga forums who answered questions about the league and the two teams and provided certain hard to find statistics.

*REFERENCES*

[1] Byungho Min, Jinhyuck Kim, Chongyoun Choe and Robert Ian, "A compound framework for sports prediction: The case study of football", Knowledge-Based Systems, Vol. 21, No. 7, pp. 551-562, 2008.

[2] J. Sindik and N. Vidal, "Uncertainty coeffecient as a method for optimization of the competition systems in various sports", Sport Science, Vol 2, No. 1, pp. 95-100, 2009.

[3] Y. Y. Petrunin, "Analysis of the football performance: from classical methods to neural network", Conference Paper, unpublished, 2011.

[4] A. C. Constantinaou, N. E. Fenton and M. Neil, "A Bayesian network model for forecasting Association Football match outcomes", Working Papers, Queen Mary University, 2012.

[5] Andreas Heuer and Oliver Rubner, "Towards he perfect prediction of soccer matches", Westfalische Wilhelms University, Germany, In press, 2012.

[6] Andrew James Moore, "Predicting football results", Unpublished, Module code: COM 3021, May 6, 2004.