PageRank++: European Soccer Team Ranking Prediction

Sheng Wang
shewang@umich.edu

Jiazhao Li
jiazhaol@umich.edu

Wenche Hsu
wencheh@umich.edu

Sisi Luo
luosisi@umich.edu

ABSTRACT

Soccer is the most popular sport all over the world, especially in Europe. The prediction of soccer competition results is always an attractive topic. However, traditional ranking method, Network of Networks (NoN), can only capture the similarity between players or teams rather than the win-lose relationship[7]. Broadly used method Deep Neural Network (DNN) cannot converge within limited dataset[10]. In this report, we proposed a graph-based method PageRank++ to predict the rank of teams in a league. In order to evaluate our results, we choose True-Skill ranking algorithm and Standard ranking algorithm as ground truth. Our algorithm converges faster without using large dataset compared with DNN. Moreover, it obtains higher accuracy compared with baselines, Naive Rank and Naive PageRank.

KEYWORDS

Graph, PageRank, True-Skill, Prediction

1 INTRODUCTION

European countries always have a great passion on soccer. People get involved in this national sport in lots of ways. One of the most interesting activity for those soccer fans during seasons is the prediction of match results. The lottery corporation also offers platforms for those who expect to earn money through soccer betting. Moreover, player evaluation and team evaluation can help team coaches with tactical analysis before match begins. The instructor can plan for strategies against them by observing opponents' weakness, or trade players to improve the overall team strength. This shows how important the evaluation is. Also, competition data is an enormous source waiting for us to analyze. In these cases, we provide an efficient method to predict the rank of the teams.

NoN can be used in cross-network ranking. Main graph consists of different teams, and each team is represented as a node. Subgraphs consist of different players, and each player is represented as a node. At first glance, NoN seems to be an ideal solution to this problem. However, the definition of edges in NoN is the similarity between nodes, which cannot be used to capture the win-lose relationship between teams. Deep learning is also broadly used to predict the result of competitions, especially to capture time sequence. However, deep learning algorithm requires enormous dataset to train in order to learn the model exactly. If the training data is insufficient, the parameters of the deep learning model could not be correctly "learned" by gradient descent method, which will lead to terrible prediction.

The definition of our problem is how can we capture time sequence in order to make predictions with limited dataset. We should identify the time sequence pattern included in our dataset without using deep learning algorithm, as well as develop an algorithm to capture the win-loss relationship among the network-of-networks framework.

There are two main observations that have been explored in our work. First, the performance of each team strongly depends on its player composition, which is highly correlated with the evaluation of certain player during certain period. We develop a process to effectively evaluate players using five main factors, cluster players to five groups by K-Means algorithm, and use radar chart visualization to interpret each group's properties. Secondly, there exists some interesting hidden relationship between teams. Some teams have the power to defeat top teams while keep losing to weak teams. If we simply evaluate them by standard ranking, the "three points for a win" evaluation, we cannot capture this interesting truth, which may lead to wrong evaluation between teams. We then implement our True-Skill algorithm to obtain the ability to discover these hidden factors based on previous match results. It allows us to capture time sequence among our datasets, which can be constructed as 8 snapshots for each league.

In this work, we obtain the edge information between two teams based on our clustering results and build graph network using the edge information.

When the graph is built, we apply three prediction algorithm on this graph, including PageRank++, Naive PageRank and NaiveRank, and obtain the final rank of all the teams in each league. We evaluate our results based on the ground truth, True-Skill algorithm and Standard ranking algorithm. According to our experiment, PageRank++ has higher prediction accuracy than that of NaiveRank and Naive PageRank. To summarize, there are three main contributions of our project:

1. We define a new criteria in evaluating player and team performance, which proves to work extremely well.
2. We take advantage of our time sequence dataset using True-Skill algorithm, which explores the hidden factor among team matches, delivering more convincing evaluation than standard ranking algorithm.
3. We design PageRank++ algorithm, which can converge faster without large dataset on our constructed graph with higher prediction accuracy, comparing with Naive ranking and Naive PageRank.

The rest of the report is organized as follows. In section 2, we introduce our dataset. Section 3 describes our methods in details, including the model architecture, feature selection, clustering algorithm and how we determine the weights of our graph and apply PageRank++ on it. Section 4 contains the performance criteria, results of our ground truth algorithm and preliminary experiments. In Section 5, we review related works. is the future improvements

we plan to do to our models and datasets. Finally, we draw the conclusion, work division , future work in section 6 to section 8.

2 DATASET

In order to test our models, we use the dataset from http://kaggle.com with some additional crawled data from European League matches from different websites, including http://www.whoscored.com and http://www.dongqiudi.com.

The original dataset from Kaggle includes more than 25,000 matches' and 11,000 football players' information from 2008 to 2015. Then we add about 3,000 matches' information from 2016 to 2017 into this dataset. The teams IDs and lineup of each match are included. A full list of match features can be seen in Table 1. The data can be separated into 11 leagues, coming from Spain, England, Belgium, France, Germany, Italy, Netherland, Poland, Portugal, Scotland and Switzerland.

Besides match information, this dataset also contains the player attributes in different match. Each player has 38 attributes, which can be further divided into 6 kinds of skills, including technical skill, mental skill, offense skill, defense skill, physical skill and goal-keeping skill.

home-player-1	away-player-1
home-player-2	away-player-2
home-player-3	away-player-3
home-player-4	away-player-4
home-player-5	away-player-5
home-player-6	away-player-6
home-player-7	away-player-7
home-player-8	away-player-8
home-player-9	away-player-9
home-player-10	away-player-10
home-player-11	away-player-11
home-goal	away-goal
match-index	match-date

Table 1: Attributes vector in European Football Match Dataset

3 PROPOSED METHODS

Our proposed method can be divided into seven parts, including feature selection, player performance evaluation, team attributes formation, PageRank implementation, league graph creation, Standard and True-Skill Ranking and Deep Neural Network prediction.

3.1 Feature Selection

Our dataset contains 33 attributes for each player. In order to filter out irrelevant or redundant features from the original dataset, we perform feature selection, which keeps a subset of our original features.

3.1.1 Variance threshold. We remove features whose value do not change much from observation to observation, which means this attribute is not a signal to represent useful patterns. Variance depends on scale, hence we normalize our features at the beginning, removing five features with the lowest threshold.

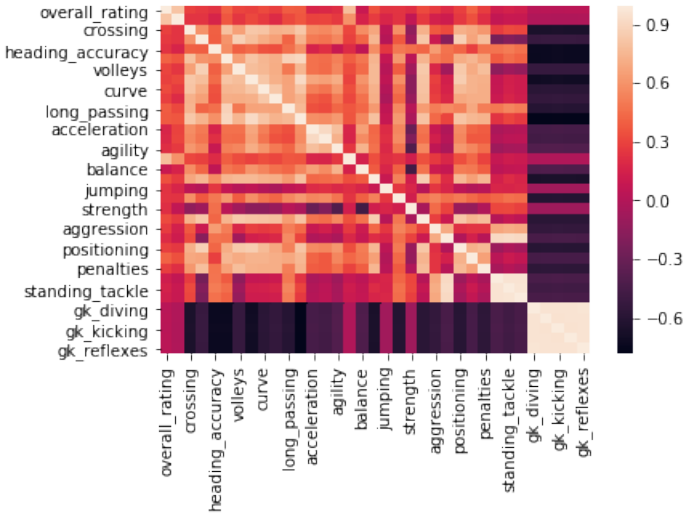


Figure 1: Correlation between player attributes

3.1.2 Correlation threshold. In Figure 1, we visualize the correlation between remaining features using heat map, as . Instead of removing features that are highly correlated with others (i.e. its value change very similarly to another), we combine them into five main factors, which are 'Technical', 'Offense', 'Defense', 'Physical' and 'Mental', shown in Table 2. We can see that the combination is reasonable. Crossing and dribbling are related to technical skills; jumping, balance and stamina undoubtedly should be regarded as player's physical ability.

3.2 Player Performance Evaluation

By combining the total 33 player attributes into 5 main factors, we successfully reduce the dimensionality of our dataset. Then we evaluate players' ability in each team using the five main factors. These main factors do not contain goalkeeper's attributes, because the evaluating criteria is different, we have to treat goalkeeper individually. Since our dataset does not contain player position information, we decide who the goalkeeper is based on given data. We loop over the entire player list and select the player who has the highest "gk-reflexes" value in each team as a goalkeeper. We choose this attribute as a criteria by data visualization, through which we find it is the only attribute goalkeeper generally have higher value than other players. Both goalkeeper and normal player could have similar values of some attributes such as "gk-kicking" and "gk-positioning"(player can act as a goalkeeper without touching the soccer by their hands), which will lead to wrong selection.

We use K-Means algorithm to divide total players (except for goalkeeper) into four clusters as we can see in Figure 2. Grouping player's ability score in one cluster and take the mean average, we can easily observe the difference between them. The mean ability score of different clusters are showed in Table 3 and Figure 3. We can see that cluster 0 generally has the highest ability score over other

Technical	Offense	Defense	Physical	Mental
crossing	shot power	interception	sprint speed	positioning
short passing	free kick accuracy	marking	agility	aggression
volleys	acceleration		reactions	vision
dribbling	finishing		balance	
heading accuracy	long shots		jumping	
long passing	penalties		stamina	
ball control			strength	
standing tackle				
sliding tackle				
curve				

Table 2: Attributes vector in European Football Match Dataset

clusters. This player cluster is labeled "All star", such as Christiano Ronaldo and Lionel Messi. Cluster 1 performs outstandingly in offense, and cluster 2 has high defense score, hence we name them as "Offense" and "Defense" respectively. Cluster 3 does not have evident performance in either main factors, hence we classified it to "Normal".

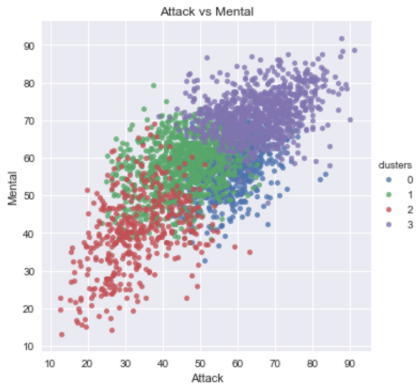


Figure 2: Bivariate plot between offense and mental score

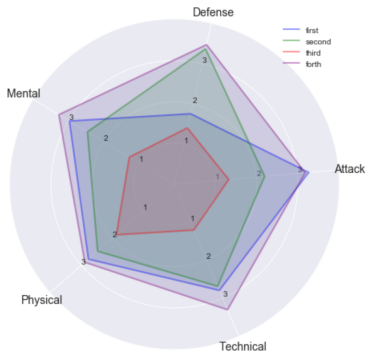


Figure 3: Radar chart for player evaluation

In our work, we treat goalkeeper separately. At the beginning, we divide total goalkeepers into 2 clusters. After visualizing their ability score through radar chart, as in Figure 4, we found that 2 clusters both perform similar pentagonal shape, which means that the clustering is redundant. Hence, instead of dividing them into multiple cluster, we simply use the average of their total ability score as their relative weights.

Figure 4: Radar chart for goalkeeper evaluation

We can conclude that the clustering is reasonable from the bivariate plot below (Offense vs Mental), as it divides players into four parts without too much overlap.

	Offense	Defense	Mental	Physical	Technical
cluster 0	67.9	62.4	72.1	73.5	70.0
cluster 1	63.2	29.4	61.1	69.6	55.9
cluster 2	52.0	67.2	60.4	69.2	61.7
cluster 3	38.7	43.1	47.6	62.4	52.0

Table 3: Mean ability score of different clusters

3.3 Team Attributes Formation

Since we have obtained the reduced player attributes stored in a 5*1 vector for each player, each team can be represented as a 5*11 matrix, including 10 players and 1 goal-keeper. An example of player attributes matrix of Barcelona in 15-16 season can be seen in Table 4. How to convert this player attributes matrix into a 5*1 team attributes vector will be introduced in this section.

	Technical	Offense	Defense	Physical	Mental
player-1	78.7	73.6	79.25	76.5	71.67
player-2	68.7	46.8	72.125	83.0	54.67
player-3	72.2	57.8	66.875	87.0	62.67
player-4	75.4	65.2	85.0	80.5	73.33
player-5	71.6	60.0	76.125	87.0	69.33
player-6	72.9	60.6	66.375	85.0	82.67
player-7	75.5	82.4	68.375	57.5	78.0
player-8	53.9	65.0	75.25	24.5	64.33
player-9	73.1	85.2	79.375	35.5	83.33
player-10	68.0	78.2	78.625	28.5	71.66

	Goal-keeper
gk-diving	83.0
gk-handling	82.0
gk-kicking	87.0
gk-position	78.0
gk-reflexes	86

Table 4: Player attributes matrix in a team

3.3.1 Calculating players scores. According to the radar chart, players in different clusters should be measured by different technical skills. For example, the defense skill of players in cluster one should be less important than their offense skill, shown in Figure 3. We calculate the score of each player through multiplying their attributes and weight matrix which is different for each cluster, releasing by radar chart. The weight matrix is shown in Table 5.

	Technical	Offense	Defense	Physical	Mental
cluster-1	0.251	0.103	0.221	0.222	0.200
cluster-2	0.178	0.237	0.192	0.194	0.199
cluster-3	0.155	0.269	0.174	0.204	0.197
cluster-4	0.211	0.192	0.208	0.185	0.203

Table 5: Weight matrix for player attributes

3.3.2 Obtaining team attributes. We assume the players in one cluster should contribute the same to the team. In other words, the players in the same cluster should be dependent while players in different clusters are considered as independent. Under this assumption, we sum up the scores of the players in one cluster as the score of this cluster. The team attributes, which is a 5*1 vector, is the score of five clusters in the team, respectively. An example of team attributes of Barcelona in 15-16 season is showed in Table 6.

Goal-keeper	Offense	Defense	All-star	Normal
83.2	55.34	132.892	508.040	0

Table 6: Team Attributes

3.4 PageRank Algorithm

In this section, we introduce PageRank Algorithm as the preliminary knowledge of our method.

PageRank is a method for rating webpages objectively and mechanically, effectively measuring the human interest and attention devoted to them. [8]. The importance of webpages should be measure by the number and quality of inlinks.

To be specific, given webpages numbered 1, ..., n, The webpage i should be ranked based on linking webpages with different weights. The weight is higher if the linking webpage has higher score. Let $L_{ij} = 1$ if webpage j links to webpage i (written $j \Rightarrow i$), and $L_{ij} = 0$ otherwise.Then let $m_j = \sum_{k=1}^{n} L_{kj}$, meaning the total number of webpages that j links to. Then BrokenRank rank vector p_i of webpage i is defined as

$$p_i = \sum_{j \Rightarrow i} \frac{p_j}{m_j} = \sum_{j=1}^{n} \frac{L_{ij}}{m_j} p_j \tag{1}$$

This is an almost PageRank because it is broken. The matrix notation should be

$$p = LM^{-1}p \tag{2}$$

where

$$p = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}, L = \begin{pmatrix} L_{11} & L_{12} & \cdots & L_{1n} \\ L_{21} & L_{22} & \cdots & L_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{pmatrix}, M = \begin{pmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_n \end{pmatrix}$$

Let $A = LM^{-1}$, and then $p = Ap$. This implies that p is the eigenvector of matrix A with eigenvalue 1. If we initialize p as $p^{(0)}$, we can obtain the final rank of the websites $p^{(t)}$ by left-multiply matrix A by t times until it converges and A can be thought as a transition matrix.

PageRank revises BrokenRank a little:

$$p_i = \frac{1-d}{n} + d \sum_{j=1}^{n} \frac{L_{ij}}{m_j} p_j \tag{3}$$

where d is damping factor designed to prevent rank sinks and dead ends. Intuitively, we can think of the iteration process as random walks. A random walk on a graph is a stochastic process where at any given time step we are at a particular node of the graph and choose an outedge uniformly at random to determine the node to visit at the next time step. Each mulipication of matrix A represents a random walk. The importance ranking of a webpage is essentially the limiting probability that the random walk will be at that one after a large time.

3.5 Prediction Algorithm

After obtaining the team attributes, our next step is to predict the rank of these teams with PageRank algorithm with appropriate graph. We propose three different prediction algorithms, which are Naive Rank, Naive PageRank and PageRank++. The original definition of edge in PageRank is similarity between two teams. However, in soccer team ranking problem, similarity is no longer useful since it fails to capture the strength of the team. Hence in the last two methods, we redefined the edge information to make it work. We worked out two definition of edge in Naive PageRank and PageRank++, respectively.

3.5.1 NaiveRank. NaiveRank algorithm is a naive prediction way which is purely based on the sum of the team attributes. Adding up the scores of all clusters in a team as the final score of the team, the rank would simply be the descending sorted final score. This algorithm is naive but reasonable, because the team with better player is more likely to win the match. However, players in different clusters didn't contribute to the team in a linear way. In other words, the team with lower score should still have chance to win the team with higher score. This information is neglected in NaiveRank.

3.5.2 Naive PageRank. Naive Pagerank algorithm is an application of Pagerank algorithm by redefining the edge information as the comparison of two team scores. The team with higher score should have the inlink, while the team with lower score should have the backlink. This definition is reasonable since the basic idea of PageRank is that a page is ranking high if its inlinks' ranks are high[8].

When calculating the team score using team attributes, instead of simply summing up all the attributes together, we weight each cluster differently based on the function of this cluster released by radar chart. For example, the attributes representing All-star cluster should have higher weight than others, while the attributes representing Normal cluster should have lower weight than others. The damping factor is set to be 0.85 since there may be a team whose score is lowest and can never be reached in PageRank.

Though this definition is reasonable, the main weakness of this definition is that it can not capture how much a team is stronger than others. In other words, it omit a lot of useful information.

3.5.3 PageRank++. In order to solve this problem, we work out another method to define the edge based on weighted PageRank algorithm[13]. The edge between u and v in original weighted PageRank algorithm is the number of inlinks of u divided by the number of inlinks of v. However, under this definition, there would be only 1 edge between two teams. In other words the team with weaker attributes could never have the change to win the team with higher attributes, which is not practical in real world.

Our assumption here is that even through the score of the stronger team is higher than that of the weaker team, it should still be possible for weaker team to have inlink in some aspect. For example, even though the score of Barcelona is higher than the score of Sevilla, the defense of Barcelona is lower than Sevilla because most of the player in Barcelona is clustered in All-Star cluster.

Under the this assumption, we calculate the edge by comparing every attributes based on their weight. Specifically, summing up all the winning scores as the weight of inlink and losing scores as the weight of backlink, the fully-connected weighted graph was created. The process of comparing is showed in Table 7. An example of final graph of Spain League is showed in Figure 5. The damping factor is set to be 1 because the graph is fully connected.

	Team-1	Comparison	Team-2
cluster-1	55.34	-65.15	120.49
cluster-2	132.89	-116.99	249.88
cluster-3	508.04	+244.34	265.45
cluster-4	0	0	0
goal-keeper	83.2	+9.41	73.79
edge-in	253.75		182.14

Table 7: Comparison between Barcelona and Real Sociedad

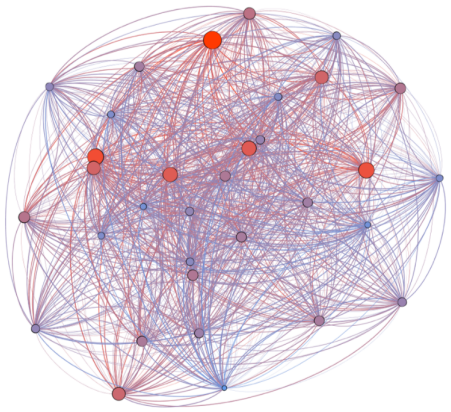


Figure 5: Graph of La Liga

The prediction is made through implementing the PageRank++ algorithm on Figure 5.

3.6 Standard and True-Skill Ranking Algorithm

In this section, we would like to introduce two ranking algorithm True-Skill Ranking algorithm and Standard Ranking algorithm as two ground truth.

3.6.1 Standard ranking algorithm. Standard ranking algorithm is also called "three points for a win". It is a standard adopted in many sports leagues and group tournaments, especially in soccer association. In this algorithm, three points are awarded to the winning team, no points for the losing team, and one point for each team if the game is drawn.[12]

It encourages more attacking play than "two points for a win", because teams will compete relatively fiercely to get the two more points rather than a draw gives. In this way, it will prevent the situation that a team needs only a draw to advance in a tournament

or avoid relegation. Hence, a commentator has stated that this point system will lead to more positive and attacking play.

The steps are following: we initialize each team with zero points of one league and update scores based on match results. The Standard ranking algorithm would be descending sorted total scores. Generally, the Standard ranking algorithm could reflect the average performance of team, though there might be some wave during the competition season.

However, assume A is a weak team, B is a medium team, and C is a strong team. If B wins A and C, it is clearly that the first win is predictable while the second one is more difficult and deserves more points on this win. However, this algorithm can't capture this latent information of the second win. It is obvious if we would like to get the rank of ability of teams, we need other alternative rank algorithms.

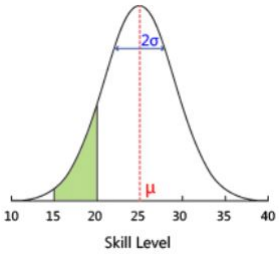


Figure 6: Gaussian distribution in True-Skill

3.6.2 True-Skill ranking algorithm. True-Skill ranking algorithm could handle previous problem in Standard ranking algorithm. True-Skill ranking algorithm is a skill-based ranking system that it not only identifies but also tracks the variance of skills of players[11]. In other words, True-Skill ranking algorithm can also capture time serious information since the update step is related with expectation based on present scores.

The algorithm characterizes its belief using a Gaussian distribution with the mean as μ and standard deviation σ[4]. These two parameters are namely the average ability of teams (μ), and the degree of uncertainty in the team's ability (σ). To be more specific, the team with high score and also high variance is considered to have strong ability but fluctuating performance, while the team with high score but low variance is believed to always keep a good state when it is competing. With the team competing in more matches, the uncertainty of the ability is likely to decrease, shown in Figure 6. Maintaining this variance allows the system to make big changes to the skill estimates early but small changes after a series of consistent matches. Consequently, the True-Skill ranking system can identify the skill of individual player from a very small number of matches. This will alleviate the problem of our limited data[2].

The True-Skill ranking system updates the parameters for players based on the outcome of a match[5]. The algorithm follows following equations.

$$\mu_{winner} \leftarrow \mu_{winner} + \frac{\sigma^2_{winner}}{c} \cdot v\left(\frac{(\mu_{winner} - \mu_{loser})}{c}, \frac{\epsilon}{c}\right) \quad (4)$$

$$\mu_{loser} \leftarrow \mu_{loser} + \frac{\sigma^2_{winner}}{c} \cdot v\left(\frac{(\mu_{winner} - \mu_{loser})}{c}, \frac{\epsilon}{c}\right) \quad (5)$$

$$\sigma^2_{winner} \leftarrow \sigma^2_{winner} \cdot \left[1 - \frac{\sigma^2_{winner}}{c^2} \cdot w\right]\left(\frac{(\mu_{winner} - \mu_{loser})}{c}, \frac{\epsilon}{c}\right) \quad (6)$$

$$\sigma^2_{loser} \leftarrow \sigma^2_{loser} \cdot \left[1 - \frac{\sigma^2_{loser}}{c^2} \cdot w\right]\left(\frac{(\mu_{winner} - \mu_{loser})}{c}, \frac{\epsilon}{c}\right) \quad (7)$$

$$c^2 = 2\beta^2 + \sigma^2_{winner} + \sigma^2_{loser} \quad (8)$$

where β is the variance of the performance around the skill of each player, $v(\cdot)$, $w(\cdot)$ are Gaussian functions, and μ_{winner}/μ_{loser}, $\sigma^2_{winner}/\mu^2_{loser}$ are the mean skill and variance of a winner/loser. Moreover, ϵ is the draw margin decided by match mode and c reflects the overall uncertainty. When a draw happens, the equations can still be applied.

In practice, we first initialize each team based on STD rank of last competitions, assuming the true ability in proportion to last year STD rank. Then we update the distribution of each team based on matched results following the match time serious. The final TS rank will be descending sorted mean of Gaussian distribution of teams. We can conclude the TS ranking algorithm can reflect the true ability of team, which can be proved in Exp.1.

3.7 DNN model

The deep neural network (DNN) is built with Python using Tensorflow package. The network has four layers, including three hidden layers and an output layer. The nodes in hidden layers will all use the sigmoid function for activations. The output layer has three probabilities of three labels: home, away and draw. The framework of our DNN model is showed in Figure 7.

We worked through each layer of our network calculating the out-puts for each neuron. All of the outputs from one layer become inputs to the neurons on the next layer, which is the forward propagation process. The number of nodes contained in three hidden layers are 256,128 and 64, respectively. The final output would be determined by the summation of each hidden nodes output:

$$\hat{y} = \sigma(w_1 x_1 + w_2 x_2 + b) \quad (9)$$

We use the weights to propagate signals forward from the input to the output layers in a neural network. We use the weights to also propagate error backwards from the output back into the network to update our weights, which is the back-propagation process.

The strategy is to find such hyper-parameters that is able to obtain the lowest error on the training set, while it would not lead to data overfitting. If we train a network that is too long or has too many hidden nodes, it can become overly specific to the training set and will fail to generalize to the validation set. That is, the loss

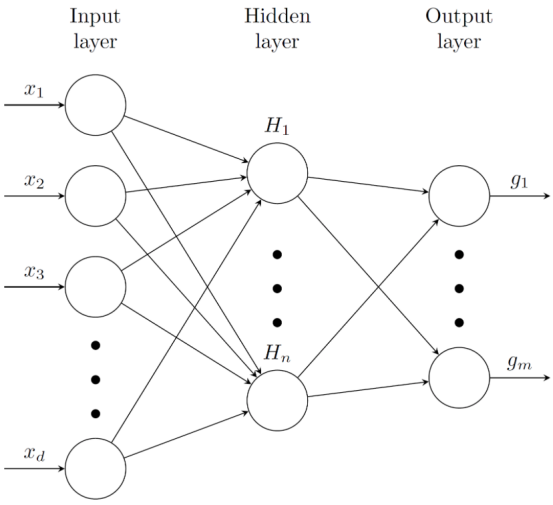


Figure 7: Framework of DNN

on the validation set will start increasing while the training set loss decreases.

A method known as Stochastic Gradient Descent (SGD) is used to train the network. The intuition of this method is that for each training pass, we grab a random sample of the data instead of using the whole data set. The error is calculated as follows:

$$Error(y, \hat{y}) = -ylog(\hat{y}) - (1 - y)log(1 - \hat{y}) \tag{10}$$

Here more training passes are used than with normal gradient descent with each passes much faster. This ends up with training the network more efficiently. The hyper-parameters include the number of iterations (the number of batches of samples from the training data wefill use to train the network), hidden nodes in each layers and the learning rate.

4 EXPERIMENTS

In this section, we design three experiments. In the first experiment, we develop two ranks based on True-Skill ranking algorithm(TS) and Standard ranking algorithm(STD) separately, and prove that TS rank can reflect the true ability of one team better than STD rank. In the second experiment, we compare the result of three prediction algorithms: PageRank++, Naïve and Naïve PageRank, by calculating the difference between their predictions and ground truth ranks obtained from experiment 1. From this experiment, we find out these three predictions are able to capture the true ability of team. Among those predictions, PageRank++ stands out with the lowest error. In the third experiment, we apply deep neural network(DNN) on our dataset as baseline in order to compare with our graph method. From this experiment, we find out that DNN is hard to converge with limited dataset.

Rank	TS Ranking	Standard Ranking
1	FC Barcelona	FC Barcelona
2	Real Madrid CF	Real Madrid CF
3	Atlético Madrid	Atlético Madrid
4	Valencia CF	Athletic Club de Bilbao
5	Sevilla FC	RC Celta de Vigo
6	RC Celta de Vigo	Sevilla FC
7	Málaga CF	Real Sociedad
8	Athletic Club de Bilbao	Málaga CF
9	RC Deportivo de La Coruña	Real Betis Balompié
10	RCD Espanyol	Valencia CF
11	Real Sociedad	UD Las Palmas
12	Rayo Vallecano	RCD Espanyol
13	Real Betis Balompié	SD Eibar
14	Real Sporting de Gijón	RC Deportivo de La Coruña
15	Getafe CF	Real Sporting de Gijón
16	UD Las Palmas	Granada CF
17	Granada CF	Rayo Vallecano
18	SD Eibar	Getafe CF
19	Levante UD	Levante UD

Figure 8: Team Ranking of Season 15-16 La Liga

4.1 True-Skill Rank and Standard Rank

In this part, we prove that True-Skill is more valid than STD when evaluating team ability. We choose match results data of Spanish league: La Liga from season 14-15 and 15-16.

For Standard rank part, we initialize each team with zero point and update the points of each team based on the match results of the 15-16 season. Updating rules is that one team gets 3 points for a win, 1 point for a drawn and 0 point for a loss. At the end of the season, the final standard rank will be sorted by total scores. For True-Skill rank part, each team is initialized as one random variable x with Gaussian distribution $N(m, var)$. The initial parameters m and var are determined by the result of last year's teams' ranking (14-15 season here). Then, we update the parameters of the Gaussian distribution of each team independently according to the result of each match by applying True-Skill algorithm. The final True-Skill rank will be sorted by the mean of Gaussian distribution of each team. We get the final ranking of these two algorithm as Figure 8 gives. In Figure 9, we also visualize the initial and final distribution of one team Valencia CF. The var has decrease significantly, which means that it reaches to convergence.

Though we can notice that some parts between two ranks are similar, for example the top 3 teams are the same, the other parts of two ranks have significant difference. Here we observe that there are interesting condition happening to Valencia CF. In order to clarify this point, we pays attention to Valencia CF and Real Betis

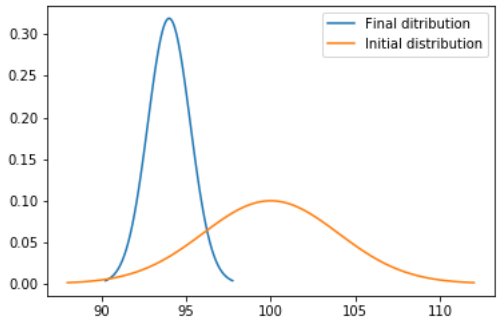


Figure 9: Gaussian distribution of Valencia CF in True-Skill

Balompié. In Figure 10, the overall performance of Real B.B. and Valencia CF are evaluated by two True-Skill rank and Standard rank. Firstly, if we compare their match results (Win/Drawn/Loss) against top 4 teams in their league, we can see that Valencia CF performs better. Secondly, Valencia CF double kills Real Betis Balompié in direct competition.These two facts can indicate Valencia CF has higher ability than Real Betis Balompié, which this phenomenon can be captured by True-Skill rank; however, Standard rank cannot capture this information. Hence, this verifies our first statement that Standard rank merely gives the average performance of one team, while True-Skill rank provides the true ability of certain team, which is more helpful for team evaluation and match prediction.

Team	TS	STD	Win/Draw/Loss
Real B.B	13	9	0 / 2 / 6
Valencia	4	10	2 / 2 / 4
Valencia : Real B.B		2 - 0	

Figure 10: Match Results Against Top 4 teams

4.2 Evaluation of prediction algorithms:

In this part, we apply three prediction algorithms separately on 11 leagues during 15-16 season and evaluate their results through loss function.

In Figure 11, we only show the first 10 of 20 teams in Spain league La Liga. The blue columns are two ground truth ranks based on real match results in 15-16 season, while orange columns are three prediction ranks based on team attributes information of 14-15 season.

In order to evaluate these predictions, we define the loss function as the one norm of ranking difference. We choose two ranks, True-Skill rank and Standard rank from Exp.1, as ground truth. We compute the average of 1-norm of difference between prediction rank and ground truth rank.

	TrueSkill Rank	Standard Rank	Naive Preidiction	Naive Prediction	PR Prediction
1	FC Barcelona	FC Barcelona	FC Barcelona	Real Madrid CF	FC Barcelona
2	Real Madrid CF	Real Madrid CF	Real Madrid CF	Atlético Madrid	Real Madrid CF
3	Atlético Madrid	Atlético Madrid	Atlético Madrid	Sevilla FC	Atlético Madrid
4	Valencia CF	Athletic CB	Valencia CF	FC Barcelona	Valencia CF
5	Sevilla FC	RC Celta de Vigo	Sevilla FC	Athletic CB	Sevilla FC
6	RC Celta de Vigo	Sevilla FC	Athletic CB	Real Sociedad	RC Celta de Vigo
7	Málaga CF	Real Sociedad	RC Celta de Vigo	Valencia CF	Athletic CB
8	Athletic CB	Málaga CF	Real Sociedad	Rayo Vallecano	Málaga CF
9	RC D.L.C	Real B.B	Rayo Vallecano	RC Celta de Vigo	RC D.L.C
10	RCD Espanyol	Valencia CF	Real S.G	Real S.G	Rayo Vallecano

Figure 11: Ground Truth Rank and Prediction Rank of La Liga in 15-16 season

$$Loss = |Prediction\ Rank - Ground\ Truth\ Rank|/Num\ of\ teams \tag{11}$$

As in Figure 12 and Figure 13, we plot the predictions and ground truth ranks together. The red lines are two ground truth: STD rank and TS rank, while the other three color lines are our prediction lines. It is obvious from plots that the error based on STD is larger than the error based on TS rank. Moreover, in error matrix (Figure 14), all three prediction algorithms are closer to TS rank rather than STD rank. Hence, we conclude these three prediction algorithms can capture more information about true ability than the average performance of a team.

On the other hand, we compare the accuracy of three predictions separately within one ground truth. In Figure 12 and Figure 13, the closer to the red line, the better the performance of prediction is. In error matrix, the smaller the error, the better the performance of the prediction. Hence, we could draw the conclusion below:

$$PageRank++ > NaiveRank > NaivePageRank$$

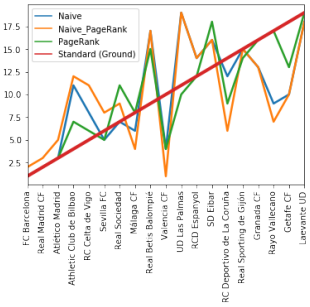


Figure 12: Standard Rank

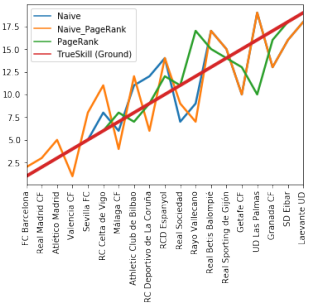


Figure 13: True-Skill Rank

We also test three prediction algorithms on larger dataset: 11 leagues of Europe from competition 08-09 season to 15-16 season. The average error results are in Figure 15. Compared with the error obtained merely from Spain, the error of the total dataset are much larger. After comparing the ground truth and the prediction rank, we find some outliers which significantly decrease our accuracy.

Prediction/Rank	TrueSkill Rank	Standard Rank
Naive PageRank	2.9	3.1
Naive	2.3	2.5
PageRank++	**0.4**	1.3

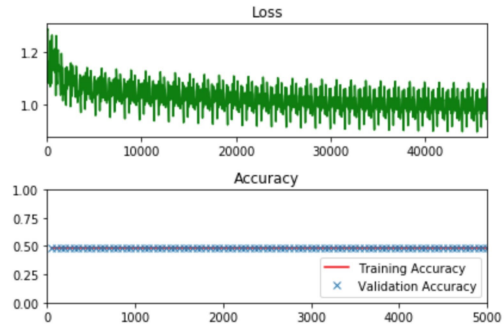
Figure 14: Error Matrix of La Liga 15-16 season

Prediction/Rank	TrueSkill Rank	Standard Rank
Naive PageRank	3.088	3.338
Naive	2.925	3.125
PageRank++	**2.215**	2.4

Figure 15: Average Error of 11 leagues from 08-09 to 15-16 season

Outliers are regarded as teams which consist of players with relatively low scores, resulting in low total score of their team. However, player's ability isn't the criteria to evaluate team's ability. Some teams are famous for their teamwork and tactics, or sometimes a talented coach can turns the table on the match. These factors can't be captured by our prediction algorithm.

4.3 Prediction with Deep Neural Network




```
Test Accuracy is 0.532
Validation accuracy at 0.523
Test Accuracy is 0.543
```

Figure 16: Loss and Accuracy of DNN

We test DNN model on European League dataset. First, we extract the match information of 15-16 season. The dataset contains 2742 matches and 4408 players in total. Then, we split the data into two parts, 2440 for training and 302 for testing. The prediction accuracy is around 50 percent, which is illustrated in Figure 16.

The prediction accuracy of DNN is poor due to three reasons. First, the performance of a team is not equal to the sum of players in the team. Secondly, the rating of a player is not convincing because the performance of a player varies through time. In other words, the rating of a player should be dynamically updated in different seasons, which is a time sequence. Additionally, the test accuracy could not increase, which means Stochastic Gradient Descent might just come to a local minimum instead of global minimum. Hence, if we expect to get a better performance using DNN, we need larger dataset and improve the feature construction.

5 RELATED WORK

The soccer and other sports prediction has attracted lots of interests with sports' popularity and the benefit of betting. The accuracy varies based on the kinds of sports they predict, dataset they use and methods they chose. Daniel Pettersson and Robert Nyquist researched this problem on their masters' thesis[9] with Recurrent Neural Network. Anito Joseph, Norman E Fenton, and Martin Neil predicts soccer match applying Bayesian nets and other machine learning method[6]. Burak Galip Aslan and Mustafa Murat Inceoglu used Neural Network to do the prediction[1]. Jordan Gumm, Andrew Barrett, and Gongzhu Hu predicted the march madness winners using machine learning strategy[3].

Most of them chose machine learning methods, one of which is "Football Match Prediction Using Deep Learning"[9]. One of the most important parts in prediction is dataset, which often decide the accuracy of the result. The dataset used in this paper is collected by author himself. It contains multiple attributes, including players, teams, period, position, card and goal. They used one hot vector to simplify the feature representation and word2vec representation to further represent match attributes. Finally, LSTM model is applied to predict the final result. However, the proposed method and all the other machine learning methods is applicable only when they have enough dataset and computing resource. In other words, those methods converge slowly with large dataset while failing to converge with small dataset.

6 CONCLUSIONS

In this paper, we compared two ranking algorithms, True-Skill Rank and Standard Rank, and evaluated four prediction methods, NaiveRank, Naive PageRank, PageRank++ and Deep Neural Network. In detail, we first find that True-Skill rank is able to reflect real time ability with time sequence matches updating process. Second, we analyzed that compared with Standard Rank, True-Skill Rank is more likely to capture the true ability of a team. We then evaluated four prediction algorithms based on these ground truth, True-Skill rank and Standard rank. After that, we find that except for deep neural network, the other three algorithms can converge faster without requiring large dataset. Additionally, among three algorithms based on graph, PageRank++ has the highest accuracy.

7 ACKNOWLEDGEMENTS

Our special thanks to Professor Danai Koutra for her insightful suggestions, and Graduate Student Instructor Yujun Yan for her helpful discussions with us. The dataset was supported by Kaggle, http://www.whoscored.com and http://www.dongqiudi.com.

REFERENCES

[1] Burak Galip Aslan and Mustafa Murat Inceoglu. 2007. A comparative study on neural network based soccer result prediction. In *Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on.* IEEE, 545–550.

[2] Thore Graepel. 2012. Score-based Bayesian Skill Learning. (2012). https://www.microsoft.com/en-us/research/publication/score-based-bayesian-skill-learning-2/

[3] Jordan Gumm, Andrew Barrett, and Gongzhu Hu. 2015. A machine learning strategy for predicting march madness winners. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on*. IEEE, 1–6.

[4] Ralf Herbrich and Thore Graepel. 2006. TrueSkill(TM): A Bayesian Skill Rating Sys- tem. Technical Report. (2006). https://www.microsoft.com/en-us/research/publication/trueskilltm-a-bayesian-skill-rating-system-2/

[5] Ralf Herbrich and Thore Graepel. 2007. TrueSkill(TM): A Bayesian Skill Rating System. MIT Press, 569fi?!576. (2007). https://www.microsoft.com/en-us/research/publication/trueskilltm-a-bayesian-skill-rating-system/

[6] Anito Joseph, Norman E Fenton, and Martin Neil. 2006. Predicting football results using Bayesian nets and other machine learning techniques. *Knowledge-Based Systems* 19, 7 (2006), 544–553.

[7] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. 2014. Inside the atoms: ranking on a network of networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1356–1365.

[8] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[9] Daniel Pettersson and Robert Nyquist. 2017. *Football Match Prediction using Deep Learning*. Master's thesis. Chalmers University of Technology, Gothenburg, Sweden.

[10] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[11] Daniel Tarlow, Thore Graepel, and Tom Minka. 2014. Knowing what we donfit know in NCAA Football ratings: Understanding and using structured uncertainty. In *Proceedings of the 2014 MIT Sloan Sports Analytics Conference (SSAC 2014)*. Citeseer, 1–8.

[12] Wikipedia. 2009. Three points for a win. (2009).

[13] Wenpu Xing and Ali Ghorbani. 2004. Weighted pagerank algorithm. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*. IEEE, 305–314.