2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Basketball Lineup Performance Prediction Using Network Analysis

Mahboubeh Ahmadalinezhad, Masoud Makrehchi, Neil Seward
Social Computing and Collective Intelligence Lab (SciLab)
University of Ontario Institute of Technology (UOIT), Oshawa, Canada
Email: {mahboubeh.ahmadalinezhad,masoud.makrehchi,neil.seward}@uoit.ca

***Abstract*—Winning a game in professional sports is the most significant matter for a team. All teams strive to bring their best performance to a game, and this requires considering all the possible lineups which coaches have available. Therefore, determining the lineup is more and more significant for a team in their winning endeavour. The ongoing result during a game defines the next decision coaches have to make to maintain or improve the outcome. Adaptive changes in a lineup of a team requires a complex decision making system. This system must consider the advantages, drawbacks, and previous experience about both teams' performance under similar situations. In order to analyze and predict lineups' performance, the authors create a directed, weighted, and signed network of all lineups that teams use against each other from 2007-2016 seasons in National Basketball Association (NBA) games. The proposed model uses machine learning and network analysis techniques to predict the performance of a lineup under a given situation by utilizing graph theory and Inverse Squared Metric.**

In order to evaluate the performance of the proposed method, several baseline models are established and results are compared. The final results over the span of ten years show that the proposed method in this paper improves the baseline results by 10% accuracy. The average of the best baseline results has an accuracy of 58% in lineup outcome prediction; however, the new method yields accuracy of 68%.

I. Introduction

In many sports, a coach has a vital responsibility to choose the best lineup of players for the game. In fact, choosing an efficient lineup has a direct impact on lineups performance. Therefore, a coach makes a decision based on some criteria such as: the performance history of a given lineup and the estimation of the given lineup's performance against the others. Examples of sports with consistent lineup changes are volleyball, basketball, hockey, and lacrosse. In these type of sports, there is a dynamic change of players during the game and, there are no limitation in terms of the number of changes. Therefore, the coach's strategy can quickly change from offensive to defensive.

ASONAM '19, August 27-30, 2019, Vancouver, Canada

ACM ISBN 978-1-4503-6868-1/19/08
http://dx.doi.org/10.1145/3341161.3342932

Since basketball is one of the most common sports, there is an extensive demand for analysis of the games. Moreover, basketball has a significant share of commercial marketing and investments in the sport industry. The National Basketball Association (NBA) is the most professional and popular league in the world. Many fans, participants and also some companies are interested in predicting the outcomes of the games [1].

Many studies have focused on data mining methods and machine learning algorithms such as neural networks and decision trees to predict the outcome of a game with acceptable accuracy [2]. One of the problem of neural network and decision tree is overfitting [3], [4]. Some models such as the Maximum Entropy and the Support Vector Machine are proposed to address overfitting problem [5]. Furthermore, Naive Bayes and multivariate Linear Regression methods are used to predict the outcome of games [6]. A major drawback of the proposed methods is dependency between the features used in the task of sport outcome prediction [7]. In addition, other studies employed the probability graph models [8].

In this paper, the lineup prediction is applied to NBA dataset for the 2007-2016 seasons. Network notations are used to represent all lineups and games. For instance, consider that two teams Home *A* and Away *B* have a game, and they use different lineups during the game. Each unique lineup is shown by a node and if two lineups played against each other, an edge from the node of home's team lineup to the node of the away's team is established. The direction always is from a home team to an away team. The result of the game is shown by the edge sign. For example if a lineup *A* outperforms lineup *B*, the sign of link from *A* to *B* is 1 ($sgn(A, B) = 1$), otherwise it will be -1. Figure 1 illustrates the network representation of lineup datasets. In this paper, for the first time, sign information is embedded in the network and the effect of the sign for the lineup prediction is investigated.

The rest of the paper is arranged as follows: In section 2, a review of related works are presented. Basic notations of graph theory and problem statement are defined in section 3. The proposed method is demonstrated in section 4, and datasets and results of experiments are discussed in sections 5 and 6, respectively. In section 7, network visualisation is demonstrated. Finally, section 8 summarizes the outcome of the study.

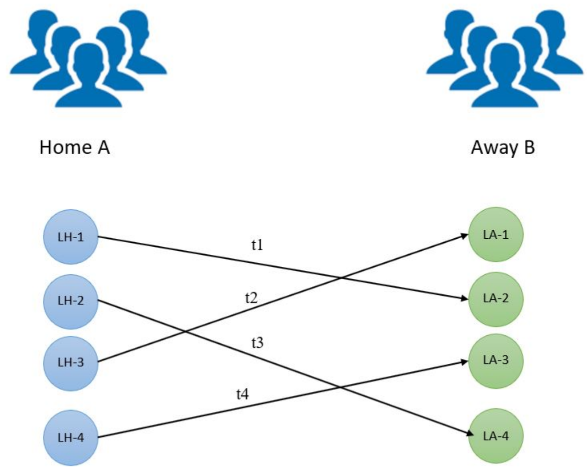


Fig. 1: Example of network representation for lineup prediction problem. There are two teams: Home A and Away B. Each team has four different lineups in four time intervals (LH: Lineup Home, LA: Lineup Away).

II. Related Works

Recently, the statistical modeling of sport data and sport analysis have become an attractive research area. Since dataset of all matches with details (lineup of each team, name of players, home team, away team, etc.) are available through the internet, many studies have been done in sport analysis. Moreover, sport analysis can help players, coaches and sport managers in evaluation of players performance, matches' results and arrangement of the best lineup for a team.

Majority of studies has focused on predicting outcome of the matches including final score of the game or the winner of the match and predicting players performances. Atlas et al. [9] proposed a model to predict players performance and players statics based on statistical data and fuzzy evaluation. The strength of each player is used to predict a match outcome [10]. Various classifiers such as logistic regression, Support Vector Machine, and Naive Bayes are applied to outcome prediction problem [11].

Moreover, team performance can be measured by the "Four Factors" which is defined by the following metrics: Effective field goal percentage (EFG), Free throw rate (FTR), Turnovers per possession (TPP), and Offensive rebounding percentage (ORP). There are some proposed models that predict outcomes of the matches using the four factors, and box scores [12], [13].

This paper has a distinct definition and application for lineup performance prediction from most research done so far in this field. Popular lineup performance research is mostly focused on fantasy sports, where fictional lineups are made from different teams. The performance is gauged over entire games, instead of being evaluated for each time period that players are on the court.

There has been one paper that has explored the use of network embedding to predict performances of non-fantasy lineups in basketball. The work presented by Pelechrinis [14] shows that a logistic regression model that uses input produced from a node2vec framework can outperform the baselines established in [14].

In this study, lineup prediction problem is studied by graph theory and network prospective. A link in social networks is translated as a relation among interacting units. An input network of the link prediction problem contains only one type of links, which is positive. On the other hand, the advent of signed social networks bring a notation called negative relations. In fact, signed social networks are a platform for users to express the sentiment of their interactions. A signed link in a social network can be interpreted as like or dislike, friend or foe. In this paper, a signed link is defined as a winner or looser lineup. Thus, the lineup prediction problem is investigated based on the structure of sign prediction problem. In this section we have an overview in the related works of sign prediction problem. Then, in the next section , the problem statement is explained.

The sign prediction problem is defined such that the sign of all edges except one are available [15]–[17]. The objective is to predict the sign of an edge based on the information acquired from the structure of the network [18], [19]. Propagation of trust and distrust in the signed network of Epinions is introduced by Guha et al. [20], in which the adjacent matrix is used as the features of the model. Kunegis et al. [21] have investigated the features of the nodes and links in the friend/foe network of Slashdot Zoo. In [22], the effect of users' behavior and their social interactions are analyzed. In [23] an edge sign prediction problem is defined as a matrix factorization problem.

The applicability of structural balance and status theories from social psychology to sign prediction problem is demonstrated in [24], [25], which provide valuable insight into the networks. Each link in the network is represented by a high-dimensional feature space, which is combination of node properties and the number of cliques and their types.

Zhang et al. [26] have extended the clique to rectangle pattern and used it along with a modified version of the PageRank measure as the features representing the edges in signed social networks.

In [27], the frequent subgraph patterns method is proposed. By modeling the social neighborhood of nodes u and v as a synthesis of frequent subgraphs, the existence of certain graph patterns is shown, which helps to estimate the link sign. Community-based algorithms, which group similar nodes, are used in link sign prediction [28] as well. Transfer learning approach [29] is proposed to predict the signs for a newly formed signed social network based on the information of the existing and a mature signed network. The link signs can represent trust and distrust in social networks too. Trust and distrust prediction based on a low-rank matrix factorization method is proposed in [30].

III. The Problem Statement

Let us consider G=(V, E, S, W) be a directed, weighted, and signed graph with n nodes and m edges (links), where *n* and *m* are the number of nodes and the number of edges, respectively. In the graph, $\mathrm{V} = \{\mathrm{v_1}, \mathrm{v_2}, \ldots, \mathrm{v_n}\}$ is the set of nodes, $E \subset V \times V$ is a subset of all possible links between the nodes, and *S* is the sign of the edges ($|E| = |S|$). In a directed network, the links between nodes *u* and *v* is not symmetric ($< u, v > \neq < v, u >$). In signed social networks, each edge has a positive or negative label. For example, $sgn(< u, v >) = 1$ indicates that the sign of the edge $< u, v >$ is positive, $sgn(< u, v >) = -1$ shows a negative edge from *u* to *v*, and $sgn(u, v) = 0$ means no link exists.

W shows the weight of a link between two nodes.

Due to the sparsity of social networks, we have $|E| \lll |V \times V|$. A social network can be represented by an adjacency matrix $A = (a_{ij})_{n\times n}$ such that:

$$a_{ij} = \begin{cases} -W & < v_i, v_j > \in E, sgn(v_i, v_j) = -1 \\ W & < v_i, v_j > \in E, sgn(v_i, v_j) = 1 \\ 0 & Otherwise \end{cases} \quad (1)$$

where $weight(v_i, v_j) = W$ and $a_{ij} \neq a_{ji}$ in an asymmetric or directed network.

In sport analytic, a lineup of a team is represented by a node and matches between lineups are represented by the links. Since all matches are laid out for entire seasons, the goal is to predict the winner of the matches. Then, the lineup prediction problem is similar to sign prediction problem in social networks. Based on network theories, we can analyses changes of each team's lineups in the matches and how to use an appropriate lineup in any circumstance for having the best performance. In other words, lineup prediction problem helps players, coaches, and sport managers to analysis rival team tactically and technically.

For the evaluation, lineup datasets during 2007-2016 seasons are used which are available from *basketball-reference.com*. The dataset includes complete information for all lineups used in entire seasons. We define a lineup as λ, consisting of five players from a team that are on the court at the same time from time m to n during a single game.

A matchup consists of the two lineups λ_i and λ_j that are both on the court from time m to n. Matchup is represented by an edge in a network and defined as:

$$edge(\lambda_i, \lambda_j) = Hometeam(\lambda_i) + Awayteam(\lambda_j) \quad (2)$$

Each matchup has an associated outcome to determine which lineup outperformed the other. The outcome of a matchup between a lineup of a home team λ_i and a lineup of an away team λ_j is defined as:

$$sgn(\lambda_i, \lambda_j) = \begin{Bmatrix} 1|\lambda_i > \lambda_j \\ -1|\lambda_i <= \lambda_j \end{Bmatrix} \quad (3)$$

where the outcome of the matchup is 1 when the home lineup, λ_i, outscores the away lineup, λ_j, and the outcome is -1 when the home lineup scores less than or equal to the away lineup. Table 1 shows statics of each seasons.

IV. The Proposed Method

A directed, weighted and signed network is created based on a lineup dataset. Each lineup is represented by a node and a matchup of two lineups against each other is represented by an edge. Direction is always from a home team to an away team. The weight is shown as a performance margin of two lineups in a matchup. If two lineups have played against each other several times, the summation of their scores is used as the weight of the edge. The sign of an edge identifies the winner of the game.

In this paper, we utilize graph theory to represent the NBA datasets to analyze lineup changes in a game. Inverse Squared Metric (ISM) which is based on the preferential attachment model on signed social networks [31] is used. Preferential attachment [32], the small-world model [33], and microscopic properties [34], [35] offer some models to understand and predict the growth pattern and behavior of complex networks. In [31], preferential attachment is used to represent an edge in a network. Barabasi and Albert [32] proposed a network growth model based on preferential attachment. In the proposed model, when a new node is added to the network, it prefers to get connected to highly connected nodes. It means the likelihood of the links between the new node and other existing nodes are not uniform.

ISM is proposed in [31] for edge representation using local and global information from a network. ISM has two main components: node degree and shortest path. In fact, the importance and intensity of both nodes are estimated by professional attachment which is calculated by multiplying their node degrees. On the other hand, the preferential attachment which measures the attraction of two nodes penalizes by the distance between the two nodes. The distance between the nodes is estimated by the shortest path. The ISM metric is defined as follows:

$$ISM(u, v) = \frac{Deg(u) \cdot Deg(v)}{|path(u, v)|^2} \quad (4)$$

Since the input is a signed and directed network, 16 different variants of ISM are calculated. For instance, for each node, there are indegree and outdegree. Since the network is signed, there are positive indegree (outdegree), negative indegree (outdegree). Thus, 4*4 = 16 different interactions may exist between two nodes. In one variant is to use positive indegree of two nodes, *u* and *v*:

$$ISM_{In_P In_P}(u, v) = \frac{Indeg_P(u) \cdot Indeg_P(v)}{|path(u, v)|^2} \quad (5)$$

Based on ISM metric, each edge is shown by a 16-dimension vector.

V. Dataset

To evaluate the proposed method, all lineups dataset of NBA during season 2007-2016 are used. The dataset contain all information for all lineups in each season. To implement the proposed method, we only use the following information:

TABLE I: Dataset Statics

	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
Number of Lineups	10146	9444	9308	9278	10841	8749	9965	10408	12504	11255
Number of matchups	27499	23408	24056	22633	30741	22407	26350	27260	28767	28759

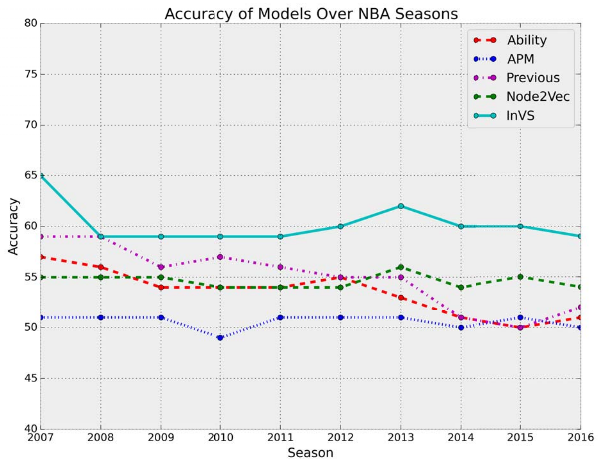


Fig. 2: Accuracy in All Datasets

1) Lineup of the home team,
2) Lineup of the away team,
3) Duration of the matchup between the lineups.

Table I shows the number of lineups which represents the number of nodes in a graph and the number of matchups which represents the number of edges in a graph.

VI. Results

In this paper, lineup performance prediction based on ISM metric is studied. We conducted an extensive number of experiments on all datasets with different feature vectors. The full dataset is divided into two parts, training and test data. For the training part, 90 percent of the data is included and test dataset has the remaining 10 percent. This dividing is known as 10-fold cross-validation. In order to apply the proposed method, three steps are defined:

1) A graph is created based on training dataset, in which all information including: edge, direction, and sign are available. Each edge is represented by a 16-dimension vector.
2) Support Vector Machine classifier is used to train a model.
3) The test dataset is added to the training dataset to create a full network. In fact, all information are available except sign and direction. The missing information are from the test dataset. This is the same definition of sign prediction problem. Thus, we follow the same set up for the experiments.

A. Baseline Methods

In this study, different methods are implemented as baseline methods and defined as follows:

1) Previous Performance: The previous performance prediction model consists of predicting the performance of the current matchup θ_k over the shift during the period of time from m to n using the performance vectors of the previous matchup θ_{k-1}. The objective of this model is to observe whether immediate performance history impacts performance prediction.

2) Lineup Adjusted Plus Minus: Adjusted Plus Minus (APM) is a metric used to gauge the individual impact a player can have while they are on the court. Before APM, just Plus Minus was used to evaluate player's performance. Plus Minus (PM) was determined by the points scored for vs. the points scored against.

With APM, the value of a player is determined by controlling against the different players on the court. This is done through regression [14]. For lineup λ_i playing against lineup λ_j, the lineups outscore each other by margin of n points over x possessions, represented by the dependant variable y. As input to the regression model, the independent variable p is a binary representation of the home and away players on the court. Players on the court from the home team are represented by 1, players on the away team on the court are represented by -1, and all players not on the court are represented by 0. The p vector is of length m, which is the total number of players in the league for that season. Using the binary representation, the regression model can be expressed as

$$y = a^T p \tag{6}$$

where a represents the regression coefficients. Once the regression model is trained, coefficients are obtained for all players. The APM for player n can be represented as a_{p_n}.

With the APM of all players given, the APM of lineup λ_i can be represented as the average APM across all players in the lineup:

$$p_{\lambda_i} = \frac{a_{p_n}}{5}, \forall p_n \in \lambda_i \tag{7}$$

For each matchup that our model tries to predict the performance outcome, we represent the final input to the classification model as the differential of the two lineup APM values for the home and away lineup:

$$\triangle p_{i,j} = p_{\lambda_i} - p_{\lambda_j} \tag{8}$$

3) Abilities: The abilities prediction model uses performance averages over an entire season as input features to the model. For each lineup in each matchup,

TABLE II: Definition of Binary Operators for edge representation

Operator	Definition
Hadamard	$[FV(u) \cdot FV(v)]_i = FV_i(u) * FV_i(v)$
Average	$[FV(u) + FV(v)]_i = \frac{FV_i(u)+FV_i(v)}{2}$
Weighted-L1	$\|FV(u) \cdot FV(v)\|_{\bar{l}_i} = \|FV_i(u) - FV_i(v)\|$

there is an associate performance average used as input to train the model. The performance average for each lineup is derived from the performance vectors for each matchup. When training the model, we try to observe the impact that performance averages have when predicting individual performances.

4) Node2Vec: In this approach, there are two main steps. First, constructing network, and second learning feature embedding. Grover et al. [36] proposed node2vec, which uses random walks to learn the latent features of the nodes. There are two parameters, p and q, which define the strategy of the search. In fact, two parameters determine a domain for breadth-first search and depth-first search.

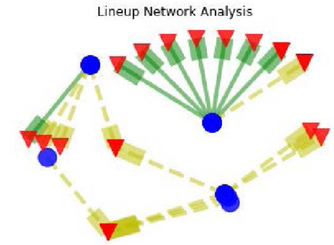
The output is a d-dimension vector representing a node. For edge representation, we need to apply binary operators on two nodes representations. For example, for edge *(u, v)*, we have two d-dimension vectors of u and v. Edge *(u, v)* is represented by one of the binary operators which are defined in Table II.

Figure 2 shows that graph theory methods such as Node2Vec and ISM can predict the lineup performance as efficient as baseline methods. ISM methods only based on the graph theory predicts the lineup performance and achieves average 65% accuracy in all datasets. Abilities method is trained the model based on the previous lineup performance. APM is used the effect of each player in a lineup and focused on players' abilities to predict the performance of a lineup. In fact, ISM method tries to recommend to a coach the best lineup of a team without considering the information of a lineup such as: home (away) lineup's defensive rebounds percentage or home (away) lineup's offensive rebounds percentage.

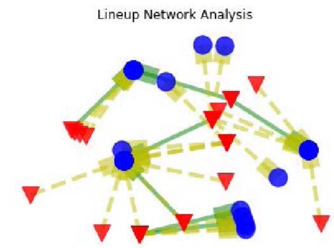
VII. Visualization

In this section, the lineup changes of two teams are visualized in two scenarios. First, team A is the home and B is the away team, which is shown in Figure 3a. Second, team B is the home team and team A is the away team, which is shown in Figure 3b. Blue circles show lineups of team A and red circles show the lineups of team B. Direction goes from the home team to the away team. Thus, direction in Figure 3a goes from the blue circles to the red circles. Color of an edge illustrates the sign and the winner of the matcheup. If a lineup of team A outperforms a lineup of team B, the edge color from A to B will be green. Otherwise the edge color will be yellow.

A team may have a different strategy when playing as a home or an away team. For example, in Figure 3a, team A chose five lineups against fifteen lineups of the away team. However, team A used more lineups when playing as the away



(a) Team A is the home team and B is the the away team. Circles show the lineup of the team A and triangles show the lineups of team B.



(b) Team A is the away team and B is the the home team. Circles show the lineup of the team A and triangles show the lineups of team B.

Fig. 3: Lineup's networks of two teams in season 2008.

team. Figure 3b shows team A chose more than ten lineups against fourteen lineups of team B. Linuep changes analysis of team A A of season 2009 and 2010 validate the strategy that team A uses when playing as home or away team over the span of three seasons of 2008-2010. Lineup changes of team A during three seasons illustrates that team A uses less lineups when playing as a home team compared to the time playing as an away team.

The network representation developed in this paper gives valuable information about the performance history of different lineups. For instance, in Figure 3a, one lineup is very successful and resulted in seven wins and only one loss. Network visualization presents precious information about both home and away teams.

VIII. Conclusion

The objective of lineup prediction problem is to determine the best lineup of a team against the rival's lineup. The point of lineup performance prediction is to investigate lineup of a team technically and tactically. Coaches can analyze team's lineup performances and choose the best lineup in the games.

In this paper, graph theory is applied to a lineup performance prediction problem. Lineups of a team and their games against other lineups are represented by nodes and edges in a directed, weighted, and signed network. In this paper, the NBA basketball league's games over then years of 2007-2016 are studied. The lineup performance prediction model is proposed based on Inverse Squared Metric (ISM).

In fact, each lineup is represented by a node and its matchup against other lineups is represented by an edge. Each edge is represented by a 16-dimension vector. A classifier, SVM, is used to train a model and predict the sign of the edge which identifies the wining lineup.

The proposed model is implemented on the NBA dataset from 2007-2016 seasons. Different methods are used as the baseline and results are compared. The proposed method, ISM, achieved 68% of accuracy and improve approximately 10% of the baselines' accuracy. Furthermore, the results show that graph theory can be a potential approach to predict lineup performance. Moreover, many characteristics of a game such as home team, lineup, differences in previous performance, and time can be embedded in the network.

The purpose of this study is to apply graph theory and analyze lineup performance using information about a team including lineup changes, home team, away team, and duration of using a specific lineup.

ACKNOWLEDGMENT

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number RGPIN-2014-06591].

REFERENCES

[1] D. J. Berri, "National basketball association," in *Handbook of Sports Economics Research*. Routledge, 2017, pp. 21–48.

[2] M. Haghighat, H. Rastegari, and N. Nourafza, "A review of data mining techniques for result prediction in sports," *Advances in Computer Science: an International Journal*, vol. 2, no. 5, pp. 7–12, 2013.

[3] M. Beckler, H. Wang, and M. Papamichael, "Nba oracle," *Zuletzt besucht am*, vol. 17, no. 20082009.9, 2013.

[4] B. Loeffelholz, E. Bednar, and K. W. Bauer, "Predicting nba games using neural networks," *Journal of Quantitative Analysis in Sports*, vol. 5, no. 1, 2009.

[5] D. Delen, D. Cogdell, and N. Kasap, "A comparative analysis of data mining methods in predicting ncaa bowl outcomes," *International Journal of Forecasting*, vol. 28, no. 2, pp. 543–552, 2012.

[6] D. Miljković, L. Gajić, A. Kovačević, and Z. Konjović, "The use of data mining for basketball matches outcomes prediction," in *Intelligent Systems and Informatics (SISY), 2010 8th International Symposium on*. IEEE, 2010, pp. 309–312.

[7] G. Cheng, Z. Zhang, M. N. Kyebambe, and N. Kimbugwe, "Predicting the outcome of nba playoffs based on the maximum entropy principle," *Entropy*, vol. 18, no. 12, p. 450, 2016.

[8] M.-h. Oh, S. Keshri, and G. Iyengar, "Graphical model for baskeball match simulation," in *Proceddings of the 2015 MIT Sloan Sports Analytics Conference, Boston, MA, USA*, vol. 2728, 2015.

[9] M. Atlas and Y.-Q. Zhang, "Fuzzy neural agents for online nba scouting," in *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society, 2004, pp. 58–63.

[10] H. O. Stekler, D. Sendor, and R. Verlander, "Issues in sports forecasting," *International Journal of Forecasting*, vol. 26, no. 3, pp. 606–621, 2010.

[11] C. Cao, "Sports data mining technology used in basketball outcome prediction," 2012.

[12] T. Baghal, "Are the" four factors" indicators of one factor? an application of structural equation modeling methodology to nba data in prediction of winning percentage," *Journal of Quantitative Analysis in Sports*, vol. 8, no. 1, 2012.

[13] M. Teramoto and C. L. Cross, "Relative importance of performance factors in winning nba games in regular season versus playoffs," *Journal of Quantitative Analysis in Sports*, vol. 6, no. 3, 2010.

[14] K. Pelechrinis, "Linnet: Probabilistic lineup evaluation through network embedding," *New England Symposium on Statistics in Sports*, 2017.

[15] Z. Liaghat, A. H. Rasekh, and A. Mahdavi, "Application of data mining methods for link prediction in social networks," *Social Network Analysis and Mining*, vol. 3, no. 2, pp. 143–150, 2013.

[16] S. Kumar, F. Spezzano, V. Subrahmanian, and C. Faloutsos, "Edge weight prediction in weighted signed networks," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 221–230.

[17] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu, "Signed network embedding in social media." SDM, 2017.

[18] K.-Y. Chiang, N. Natarajan, A. Tewari, and I. S. Dhillon, "Exploiting longer cycles for link prediction in signed networks," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 1157–1162.

[19] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 641–650.

[20] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proceedings of the 13th international conference on World Wide Web*. ACM, 2004, pp. 403–412.

[21] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The slashdot zoo: mining a social network with negative edges," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 741–750.

[22] S.-H. Yang, A. J. Smola, B. Long, H. Zha, and Y. Chang, "Friend or frenemy?: predicting signed ties in social networks," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 555–564.

[23] P. Agrawal, V. K. Garg, and R. Narayanam, "Link label prediction in signed social networks." in *IJCAI*, 2013.

[24] F. Heider, "Attitudes and cognitive organization," *The Journal of psychology*, vol. 21, no. 1, pp. 107–112, 1946.

[25] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1361–1370.

[26] T. Zhang, H. Jiang, Z. Bao, and Y. Zhang, "Characterization and edge sign prediction in signed networks," *Journal of Industrial and Intelligent Information Vol*, vol. 1, no. 1, 2013.

[27] A. Papaoikonomou, M. Kardara, K. Tserpes, and T. A. Varvarigou, "Predicting edge signs in social networks using frequent subgraph discovery," *IEEE Internet Computing*, vol. 18, no. 5, pp. 36–43, 2014.

[28] S. R. Shahriary, M. Shahriari, and R. Noor, "A community-based approach for link prediction in signed social networks," *Scientific Programming*, vol. 2015, p. 5, 2015.

[29] J. Ye, H. Cheng, Z. Zhu, and M. Chen, "Predicting positive and negative links in signed social networks by transfer learning," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 1477–1488.

[30] C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon, "Low rank modeling of signed networks," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 507–515.

[31] M. Ahmadalinezhad and M. Makrehchi, "Sign prediction in signed social networks using inverse squared metric," in *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 2018, pp. 220–227.

[32] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[33] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[34] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 462–470.

[35] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 177–187.

[36] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 855–864.