*Avrupa Bilim ve Teknoloji Dergisi*
*Sayı 27, S. 311-317, Kasım 2021*

***Araştırma Makalesi***

www.ejosat.com ISSN:2148-2683

*European Journal of Science and Technology*
*No. 27, pp. 311-317, November 2021*

***Research Article***

# Network Embedding for Link Prediction in Bipartite Networks

Özge Kart[1*]

[1*] Dokuz Eylül University, Faculty of Engineering, Departmant of Computer Engineering, İzmir, Turkey, (ORCID: 0000-0001-6954-4928), ozge@cs.deu.edu.tr

(First received 22 May 2021 and in final form 25 August 2021)

(**DOI:** 10.31590/ejosat.937722)

**ATIF/REFERENCE:** Kart, Ö. (2021). Network Embedding for Link Prediction in Bipartite Networks. *European Journal of Science and Technology*, (27), 311-317.

**Abstract**

Many social networks have a bipartite nature. Link prediction in social networks has been the focus of interest for many researchers recently. Network embedding, which maps each node in the network to a low-dimensional feature vector is used to solve many problems. The aim of this study is to investigate how network embedding enhance the link prediction performance in bipartite networks. A network embedding and a supervised learning based link prediction model has been presented for bipartite networks. The input of the supervised learning model is learned embedding vectors of node pairs obtained from network embedding method. The target feature of prediction is a binary label indicating the existence or absence of a link between these node pairs. Ensemble learning algorithms have been applied for supervised link prediction. The experiments performed on two bipartite social networks built from public datasets led promising results with 0.939 and 0.974 AUC values. Random Forest models trained with embedding vectors obtained from BiNE method achieved the highest performances.

**Keywords:** Network embedding, Node2vec, BiNE, Link prediction, Bipartite network, Ensemble learning.

# İki Parçalı Ağlarda Bağlantı Tahmini İçin Çizge Gömme

**Öz**

Birçok sosyal ağ doğası gereği iki parçalı yapıdadır. Sosyal ağlarda bağlantı tahmini, son zamanlarda birçok araştırmacının ilgi odağı olmuştur. Ağdaki her düğümü düşük boyutlu bir özellik vektörüne eşleyen çizge (ağ) gömme işlemi birçok problemin çözümü için kullanılmaktadır. Bu çalışma, çizge gömme yöntemlerinin iki parçalı ağlarda bağlantı tahmin modelinin performansını nasıl iyileştirdiğini incelemeyi amaçlamaktadır. İki parçalı ağlar için çizge gömme ve makine öğrenmesi tabanlı bir bağlantı tahmini modeli sunulmuştur. Makine öğrenme modelinin girdisi, düğüm çiftlerinin çizge gömme yönteminden elde edilen öğrenilmiş gömme vektörleridir. Tahminleme işleminin hedef özniteliği, bu düğüm çiftleri arasında bir bağlantının varlığını veya yokluğunu gösteren ikili bir etikettir. Gözetimli bağlantı tahmini için topluluk öğrenme algoritmaları uygulanmıştır. Herkese açık veri kümelerinden oluşturulan iki parçalı iki sosyal ağ üzerinde gerçekleştirilen deneyler, 0.939 ve 0.974 AUC değerleriyle umut verici sonuçlara ulaşmıştır. BiNE yönteminden elde edilen gömme vektörleri ile eğitilen Random Forest modelleri en yüksek performansları elde etmiştir.

**Anahtar Kelimeler:** Çizge gömme, Node2vec, BiNE, Bağlantı tahmini, İki parçalı ağ, Topluluk öğrenmesi.

[*] Corresponding Author: ozge@cs.deu.edu.tr

# 1. Introduction

A social network is a type of network, in which nodes represent people or other entities in a particular social context, and connections represent the influence, interaction or cooperation between nodes. Networks representing researchers in a particular discipline having a co-authorship relationship or employees in a large company working on a joint project are natural examples of social networks. As a part of researches on large and complex networks, considerable attention is paid to computational analysis of social networks. Social networks grow and develop, with the addition of new connections (links) showing that new interactions in the social structure are emerging. Social networks are very dynamic because of these features (Li et al., 2017).

In many examples of natural social networks, there are relationships not only between entities of the same kind but also between different types of entities. Such relationships form a bipartite network (Gao et al., 2018). For example, a bipartite network represents products on an e-commerce site and customers with a purchasing relationship.

One of the common problems in social network analysis is the link prediction problem, which tries to predict the probability of a new connection between two nodes based on existing connections and the properties of the nodes (Hasan & Zaki, 2011). Link prediction algorithms can be used to predict future connections that may arise in the growing and developing dynamic networks. For example, in social networks, links that are not yet available but have high potential can be suggested as new links.

Peng et al. (2015) addressed the metrics used in link prediction in three main categories according to the basic network information used in prediction: social theory-based, node-based and topology-based. Topology-based techniques are addressed in three subgroups: neighborhood-based, path-based, and random walk-based (Peng et al., 2015). Preferential Attachment index, Jaccard and Adamic-Adar are examples of neighborhood-based metrics that use the common neighborhood information of node pairs to predict whether there will be a link between them. In addition to neighborhood information, path-based metrics such as Local Path use paths through links between nodes to calculate the similarities of node pairs (Lü & Zhou, 2011). Random walk-based methods such as ItemRank use the transition probabilities from a node to its neighbors in a random walk (Gori & Pucci, 2007). Kart et al. (2020) implemented ItemRank metric and their proposed weighted and bipartite extensions of topological metrics such as Jaccard, Adamic-Adar and Preferential Attachment for supervised link prediction. They compared performance of these metrics on machine learning models. ItemRank metric achieved the best performance in their expleriments on public MovieLens and Goodreads poetry datasets (Kart et al., 2020).

One of the successfully used methods in the link prediction problem is the network embedding method, which has recently attracted the attention of researchers. Network embedding maps each node in the network to a low-dimensional feature vector, trying to preserve the strength of the connection between nodes (Goyal & Ferrara, 2018). This representation of nodes allows them to be used in link prediction by capturing the natural dynamics of networks. It has been shown that the learned node representations are used successfully in link prediction on different types of networks such as Collaborative networks (Wang et al., 2016) , social networks (Ou et al., 2016) and biological networks (Grover & Leskovec, 2016). However, the use of network embedding methods for link prediction in bipartite networks has been relatively less explored than in other networks (Gao et al., 2018).

This study investigates the effect of network embedding on performance of supervised link prediction in bipartite networks. Network embedding methods node2vec and BiNe are applied on bipartite social networks to solve the link prediction problem. Embedding vectors of node pairs obtained by network embedding methods are fed into supervised machine learning algorithms. Binary classifier models trained in this way are used to predict whether a new link will occur or not in the future. The real-world datasets MovieLens and GoodReads poetry have been used in the experiments. They contain large volume data of ratings given to movies and books by users.

The content of this paper is organized as follows: In the second section, network embedding methods, machine learning algorithms and evaluation methods used in this study are presented. In the third section, datasets and construction of networks are detailed. Conducted experiments on network embedding methods are presented and obtained results are evaluated. Finally, in the fourth section the study is concluded.

# 2. Material and Method

Figure 3 illustrates the supervised link prediction process in this study. Firstly, embedding vectors are computed for each node in the network by applying network embedding methods. Embedding vector pairs of user and item nodes are concatenated and given as input to the machine learning model. The output of the model is the prediction of occurrence of a new link between the node pairs.

## 2.1. Network Embedding Methods

Network embedding is mapping each node in the network into a low dimensional space. It aims to encode the nodes in the network such that the similarities in the embedding space approximates the similarities in the original network. Thus the low dimensional node embeddings can be used as input to machine learning models instead of adjacency matrix which can be very high dimensional for the large networks.

Random Walk is an efficient method for defining similarity of nodes and creating node embeddings. Given a network and a starting point, the algorithm randomly selects one of its neighbors is and move to this neighbor. Then, it randomly selects one of the neighbors of this node, then move on to it, and so on. The sequence of nodes selected randomly in this way is a random walk on the network. Therefore, similarity between two nodes u and v is defined as the probability of their co-occurrence on a random walk through the network. In this study, two different extensions of Random Walk-based node embedding methods Node2vec and BiNE have been employed.

### *2.1.1. Node2vec*

Node2vec is a network embedding algorithm which generates vector representations of nodes on a graph. Each node in the graph is represented by a low-dimensional continuous feature vector. It aims to learn a mapping of nodes to a low-

dimensional space of features using random walks through a graph starting from a target node. The node embedding process of Node2vec is illustrated in Figure 1. Unlike Deepwalk algorithm (Perozzi et al., 2014), which uses uniform random walks, Node2vec designs a biased random walk procedure, for efficiently exploring diverse neighborhoods. The search strategy in node2vec, gives control to us over the explored neighborhoods through parameters p and q. p and q parameters allow us to control over the walk, whether it will further explores the neighborhood of a starting node or leave the neighborhood quickly.

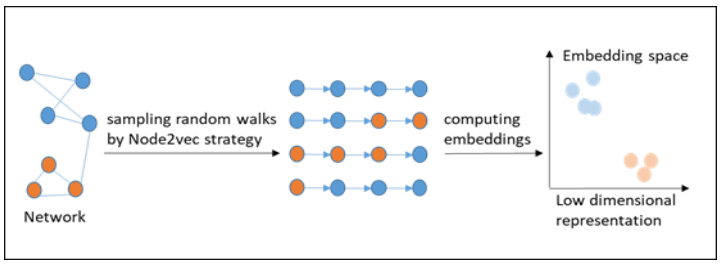


*Figure 1. Node2vec embedding process*

### *2.1.2. BiNE*

Generic network embedding algorithms such as Node2vec can be implemented to learn node embeddings for bipartite networks without considering the node type information. However, real-world applications of bipartite networks involve the relationship between two different types of nodes having different properties and patterns. For example, social network-based recommender systems need to capture the relations between users and items that the users rate. To address this problem, Gao et al. proposed BiNE (Bipartite Network Embedding) algorithm (Gao et al., 2018). It concerns long-tail distribution of node degrees and implicit connectivity relations between nodes of the same type. BiNE designs a biased random walk generator to generate node sequences that preserve the long-tail distribution of node degrees. They proposed an optimization framework that simultaneously models explicit relations (i.e. direct links) and implicit relations (i.e. indirect but transitive links). The node embedding process of BiNE is illustrated in Figure 2.

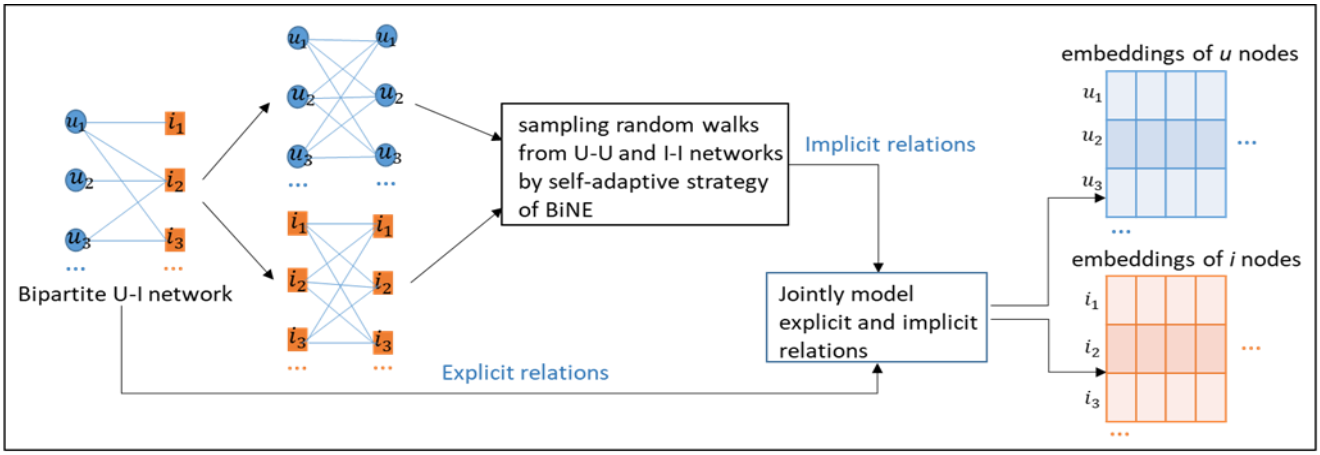


*Figure 2. BiNE embedding process*

## 2.2. Supervised Learning Methods

In this study, ensemble learning methods are applied as supervised machine learning methods for the binary classification problem. Ensemble methods use multiple learning models to attain better prediction performance than could be attained from any of the individual constituent model. (Kelleher et al., 2015)

The decision tree-based ensemble models trained using the embedding vectors obtained by network embedding methods detailed in section 2.1 are: Random Forest (RF), Adaptive Boosting (AdaBoost) and Extreme Gradient Boosting (XGB). Scikit-learn and XGBoost libraries in python were utilized for applying ensemble learning algorithms with their default hyper parameters. Number of estimators in ensemble models were set to 500.

### *2.2.1. Random Forest*

Random Forest classifier proposed by Breiman (2001) is a bagging based ensemble learning method. It combines "Bagging" and "Random Subspace" approaches (Breiman, 2001). The algorithm generates a forest of decision trees. Multiple random decision trees are trained and all the predictions of the trees are combined to make the most accurate classification instead of a single decision tree (Erdem & Bozkurt, 2021).

### *2.2.2. Adaptive Boosting*

Adaptive Boosting (AdaBoost) is a boosting based ensemble learning method. It continuously applies weak learners to resampled training data to create a set of hypothesis functions that are finally combined by a weighted linear vote to build the ensemble classifier. A misclassified sample takes a higher weight. Thus, in the next iterations, the weak learner is forced to concentrate on these hard-to-predict situations and a stronger learner is built (Freund & Schapire, 1996) .

### *2.2.3 Extreme Gradient Boosting*

The Extreme Gradient Boosting algorithm proposed by Chen and Guestrin (Chen & Guestrin, 2016). It is a novel implementation method of Gradient Boosting. The main improvement of XGB is the normalization of the loss function to reduce model variances. It also reduces the model's complexity and hence the likelihood of overfitting.

### 2.3. Evaluation Metrics

Different evaluation metrics can be applied to measure the performance of the supervised learning models RF and XGB. In this study, Area Under the ROC Curve (AUC), accuracy, precision and recall metrics were applied for the evaluation. These metrics are formulated by using TP, TN, FP and FN parameters. True-Positives (TP) denote the positive instances correctly labeled as positives. True- Negatives (TN) are negative instances correctly labeled as negative. False-Positives (FP) correspond to negative instances wrongly labeled as positive. Lastly, False-Negatives (FN) refers to positive instances wrongly labeled as negative. The ROC Curve is a plot of the true positive rate (TPR) against the false positive rate (FPR). AUC refers to the area under ROC curve (Fawcett, 2006). These metrics are formulated as follows.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$TPR = \frac{TP}{TP + FN} \quad FPR = \frac{FP}{FP + TN} \quad (4)$$

In the experiments, 5-fold cross-validation scheme was applied by randomly assigning equal number of positive and negative samples to each fold.

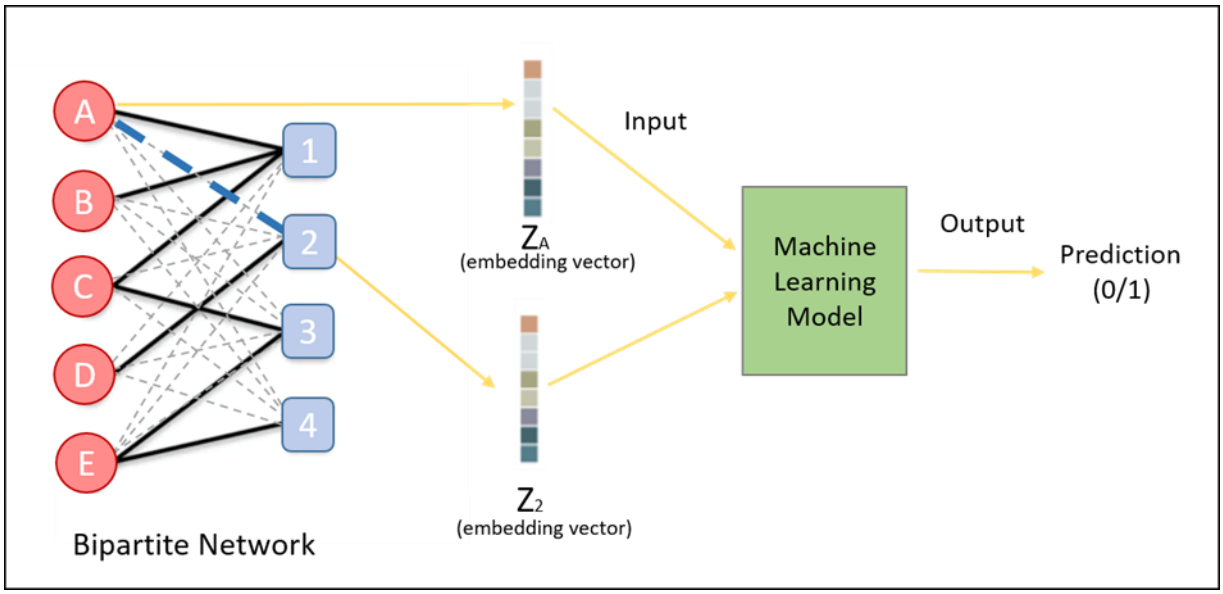


*Figure 3. The illustration of supervised link prediction process in this study*

## 3. Results and Discussion

### 3.1. Datasets

In this study, two real-world social networks from different fields has been considered: Goodreads poetry (*Goodreads*, 2021) and MovieLens (*MovieLens*, 2021). Description of the datasets is provided in Table 1.

- Goodreads: In Goodreads application, users rate the books they have read ranging from one to five. When the user read the book and doesn't rate, rating value become zero. Zero ratings were excluded from dataset.
- MovieLens: It is a web-based recommendation system and social network that recommends new movies to its users based on the movie preferences they have watched before. The dataset contains users' ratings (1-5) to movies they have watched. The rating time of the items (books/movies) is also available in both datasets.

*Table 1. Details of the datasets*

| Dataset | # Users | # Items | # Ratings (Weights) |
|---|---|---|---|
| GoodReads | 377,799 | 36,514 | 1,636,718 |
| MovieLens | 6,040 | 3,900 | 1,000,209 |

### 3.2. Network Generation

Weighted bipartite networks are constructed from the datasets whose details are given in Table 1. Users and items are two different types of nodes of the network. The rating information links the users and items (books/movies). The rating values given to items by users are the weights of the links.

We need to create a labeled dataset for our supervised link prediction problem. The similar process to (Bütün et al., 2018) has been applied in this study, since they used network data including timestamps similar to this study. Firstly, each network is divided into two equal parts. The time intervals for the equal partitioning is given in Table 2. The first network is constructed by using "first dataset" and the second network is constructed by

using "labeling dataset". In GoodReads network, 382 subcomponents not connected to the main component in the network were omitted from the dataset. If a non-existing link in first network, appears in second network, then this link is labeled as positive, else it is labeled as negative. The number of positive-labeled samples obtained from GoodReads and MovieLens datasets are given in Table 2. To obtain a balanced dataset, the same number of random negative samples were selected among all negative samples which have a large volume.

The network embedding algorithms (Node2vec and BiNE) described in method section have been implemented on the labeled datasets. Each node in the network is represented by a low- dimensional feature vector. These feature vectors are used to train the ensemble learning models.

*Table 2. Details of the network generation process*

| Dataset | Years of first dataset | Years of labeling dataset | Positive instances obtained |
|---|---|---|---|
| GoodReads | April 2000 – October 2000 | November 2000 –December 2003 | 146,756 |
| MovieLens | 2007 - 2013 | 2014 - 2017 | 47,636 |

## 3.3. Experimental Results

Performances of the ensemble learning models built using learned embedding vectors have been evaluated. The concatenated embedding vectors of node pairs obtained by applying two different network embedding methods Node2Vec and BiNe are given as input to supervised machine learning algorithms. The embedding size is 100 in Node2Vec implementation and it is 128 in BiNe implementation. Binary classifier models trained in this way have been used to predict potential links. The performance of the models have been evaluated by cross validation in terms of evaluation metrics stated in methods section.

In all ensemble learning models; Random Forest, AdaBoost and XGBoost trained with the Goodreads dataset, BiNE representation achieved slightly more successful results in terms of average accuracy, precision and AUC values compared to Node2vec representation, with an exception of recall values.

All performance values are given in Table 3. BiNE representation increased accuracy and AUC values of Random Forest model by approximately 6% and 4% respectively, compared to Node2vec.

Table 4 shows that, BiNE representation achieved more successful results in terms of accuracy, precision, recall and AUC values compared to Node2vec representation in all models trained with the Movielens dataset. In the experiments using this data set, a higher performance increase was observed in BiNe representation compared to those conducted with the Goodreads dataset. BiNE representations increased accuracy and AUC values of Random Forest model by approximately 12% and %8 compared to Node2vec.

In experiments using both Goodreads and Movielens data sets, the best performance is observed in Random Forest models trained with BiNe representations with 0.939 and 0.974 AUC values respectively.

*Table 3. Performance values of ensemble learning models trained with Goodreads dataset*

| | Node2vec | | | BiNE | | |
|---|---|---|---|---|---|---|
| **Model:** | **RF** | **AdaBoost** | **XGBoost** | **RF** | **AdaBoost** | **XGBoost** |
| Average Accuracy: | 0.811 | 0.780 | 0.810 | 0.877 | 0.863 | 0.863 |
| Average Precision: | 0.799 | 0.770 | 0.795 | 0.895 | 0.889 | 0.884 |
| Average Recall: | 0.954 | 0.964 | 0.965 | 0.923 | 0.907 | 0.912 |
| Average Auc: | 0.899 | 0.906 | 0.922 | **0.939** | 0.927 | 0.927 |

*Table 4. Performance values of ensemble learning models trained with MovieLens dataset*

| | Node2vec | | | BiNE | | |
|---|---|---|---|---|---|---|
| **Model:** | **RF** | **AdaBoost** | **XGBoost** | **RF** | **AdaBoost** | **XGBoost** |
| Average Accuracy: | 0.814 | 0.765 | 0.808 | 0.930 | 0.880 | 0.838 |
| Average Precision: | 0.770 | 0.754 | 0.784 | 0.933 | 0.872 | 0.816 |
| Average Recall: | 0.897 | 0.792 | 0.852 | 0.928 | 0.894 | 0.873 |
| Average Auc: | 0.895 | 0.847 | 0.890 | **0.974** | 0.950 | 0.917 |

## 3.4. Comparison with Topology Based Metrics

In this section, the performances of ensemble learning models RF and XGB trained by using network embedding methods Node2vec and BiNE are compared with conventional topology-based metrics Jaccard and Local Path adapted to weighted bipartite networks and Item Rank metric presented in a previous study (Kart et al., 2020). Table 5 and Table 6 compare the AUC results of classification models trained with Node2vec and BiNE representations and topology-based metric values, computed from Goodreads and Movielens datasets, respectively.

A substantial increase is observed in the AUC values of ensemble machine learning models trained with the network embedding-based representations compared to other topology-based metrics.

Between the models trained using topology-based metrics and Node2Vec representation, XGBoost achieved a better AUC value than Random Forest (except that node2vec representation of MovieLens network has approximately equal AUC in both models). Random Forest achieved the best AUC results in experiments performed for both data sets when BiNE method was used to represent the networks.

The results of the experiments has shown that supervised link prediction using network embedding methods are more successful than conventional topology-based link prediction methods described above.

*Table 5. AUC results of ensemble learning models trained with Goodreads dataset by using topology based metrics and network embedding methods*

| | **Model:** | **RF** | **XGBoost** |
|---|---|---|---|
| Results of Topology-based metrics from (Kart et al., 2020) | Jaccard: | 0.863 | 0.890 |
| | Local Path: | 0.831 | 0.848 |
| | Item Rank: | 0.880 | 0.911 |
| Results of Network embedding methods | Node2vec: | 0.899 | 0.922 |
| | BiNE: | **0.939** | 0.927 |

*Table 6. AUC results of ensemble learning models trained with MovieLens dataset by using topology based metrics and network embedding methods*

| | **Model:** | **RF** | **XGBoost** |
|---|---|---|---|
| Results of Topology-based metrics from (Kart et al., 2020) | Jaccard: | 0.684 | 0.773 |
| | Local Path: | 0.719 | 0.803 |
| | Item Rank: | 0.774 | 0.846 |
| Results of Network embedding methods | Node2vec: | 0.895 | 0.890 |
| | BiNE: | **0.974** | 0.917 |

## 4. Conclusion and Future Work

In this study, a supervised link prediction model based on network embedding methods for bipartite networks has been presented. Network embedding methods have been applied on bipartite networks. They represented each node in a network as a learned embedding vector encoding the structure information of the network. These embedding vectors of node pairs have been fed into ensemble learning models to predict potential links.

The experiments conducted on Goodreads and MovieLens datasets have demonstrated that the models obtained 0.939 and 0.974 AUC values respectively, which are promising results. Network embedding methods node2vec and BiNe outperformed weighted and bipartite extensions of topology based network similarity metrics Jaccard, Local Path and original Item Rank metric. The highest AUC values have been provided by Random Forest models trained with BiNe representations for both datasets. Since different node types of bipartite networks are taken into consideration in BiNe method, it has yielded higher performance than node2vec. The results of this study present that network embedding methods enhance supervised link prediction in bipartite networks.

In the experiments of this study, the network embedding algorithms has run with one parameter setup. By performing parameter tuning on these algorithms, the optimal parameter setup and its effect on the performances of models can be investigated. Deep learning algorithms can be implemented in addition to ensemble machine learning methods for supervised link prediction, in order to extend the analysis in this study.

## References


Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Bütün, E., Kaya, M., & Alhajj, R. (2018). Extension of neighbor-based link prediction methods for directed, weighted and temporal social networks. *Information Sciences*, *463–464*, 152–165. https://doi.org/10.1016/j.ins.2018.06.051

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Erdem, E., & Bozkurt, F. (2021). A comparison of various supervised machine learning techniques for prostate cancer prediction. *Avrupa Bilim ve Teknoloji Dergisi*, *21*, 610–620.

Fawcett, T. (2006). ScienceDirect.com - Pattern Recognition Letters - An introduction to ROC analysis. *Pattern Recognition Letters*.

Freund, Y., & Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. *Proceedings of the 13th International Conference on Machine Learning*. https://doi.org/10.1.1.133.1040

Gao, M., He, X., Chen, L., Liu, T., Zhang, J., & Zhou, A. (2018). *Learning Vertex Representations for Bipartite Networks*. 1–14.

*Goodreads*. (2021). https://www.goodreads.com/

Gori, M., & Pucci, A. (2007). ItemRank: A random-walk based scoring algorithm for recommender engines. *IJCAI International Joint Conference on Artificial Intelligence*, 2766–2771.

Goyal, P., & Ferrara, E. (2018). Knowledge-Based Systems Graph emb edding techniques, applications, and performance : A survey. *Knowledge-Based Systems*, *151*, 78–94. https://doi.org/10.1016/j.knosys.2018.03.022

Grover, A., & Leskovec, J. (2016). *node2vec*. https://doi.org/10.1145/2939672.2939754

Hasan, M. Al, & Zaki, M. J. (2011). A Survey of Link Prediction in Social Networks. In *Social Network Data Analytics*. https://doi.org/10.1007/978-1-4419-8462-3_9

Kart, O., Ulucay, O., Bingol, B., & Isik, Z. (2020). A machine learning-based recommendation model for bipartite networks. *Physica A: Statistical Mechanics and Its Applications*. https://doi.org/10.1016/j.physa.2020.124287

Kelleher, J., Mac Namee, B., & D'arcy, A. (2015). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.* MIT Press.

Li, Z., Fang, X., & Sheng, O. R. L. (2017). A survey of link recommendation for social networks: Methods, theoretical foundations, and future research directions. *ACM Transactions on Management Information Systems*.

https://doi.org/10.1145/3131782
Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and Its Applications*, *390*(6), 1150–1170. https://doi.org/10.1016/j.physa.2010.11.027
*MovieLens*. (2021). https://grouplens.org/datasets/movielens/
Ou, M., Cui, P., Pei, J., Zhang, Z., & Zhu, W. (2016). Asymmetric transitivity preserving graph embedding. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939751
Peng, W., Baowen, X. U., Yurong, W. U., & Xiaoyu, Z. (2015). *Link Prediction in Social Networks : the State-of-the-Art arXiv : 1411 . 5118v2 [ cs . SI ] 8 Dec 2014*. *58*(January), 1–38. https://doi.org/0.1007/s11432-014-5237-y
Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2623330.2623732
Wang, D., Cui, P., & Zhu, W. (2016). Structural deep network embedding. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939753