Link Prediction in Complex Networks using Embedding Techniques and Similarity Measures

Sanjay Kumar, Vipul Gupta, Sudhanshu Shekhar Singh

Abstract**:** ***Networks have proved to be very helpful in modelling complex systems with interacting components. There are various problems across various domains where the systems can be modelled in the form of a network with links between interacting components. The Problem of Link Prediction deals with predicting missing links in a given network. The application of link prediction ranges across various disciplines including biological networks, transportation networks, social networks, telecommunication networks, etc. In this paper, we use node embedding methods to encode the nodes into low dimensional embeddings and predict links based on the edge embeddings computed by taking the hadamard product of the participating nodes. We further compare the accuracy of the models trained on different dimensions of embeddings. We also study how the introduction of additional features changes the accuracy when introduced to various dimensions of node embeddings. The additional features include overlapping measures such as Jaccard similarity, Adamic-Adar score and dot product between node embeddings as well as heuristic features i.e. Common Neighbors, Resource Allocation, preferential attachment and friend tns score.***

Keywords: Complex Networks, Network Embedding, Link Prediction, Online Social Networks, Similarity Measures.

I. INTRODUCTION

Complex systems which have components that interact with each other can be represented as a network, and the interactions between two components can be shown by a link. Most of the networks are dynamic in nature. We can predict how the network will evolve and what new links will be formed by the help of link prediction [1,2,3]. Link prediction has a vast domain of application spread across various disciplines. It aims to exploit the behavior and structure of a network to accurately predict new links that don't already exist in the network. Since, Link prediction allows us to predict how the network will evolve, it has become an interesting area of study. A lot of research work in the field of link prediction is based on assumptions made on a particular type of network, such methods perform well on networks where these assumptions hold true, however they perform poorly when applied to the networks where these assumptions don't hold true.These applications include friend suggestions in social networks, recommendation of relevant entities across platforms like e-commerce websites and digital streaming platforms, detection of spam emails, recommending alternative trip routes based on the current traffic patterns etc. [4]. Link Prediction can also be used in silico drug-target interaction [5] for the discovery of new uses for existing drugs. Link prediction has been used to identify many protein-protein interactions which are biologically significant but not identified [6]. Link Prediction also provides efficient solutions to various problems across the fields such as Social Networks, Bioinformatics, Transportation networks, Advertisement, Telecommunication networks and many other important fields [7]. There has been a lot of research in Link Prediction. Most of the work is focused on exploiting structural information using hand engineered features in which specific aspects and assumptions are taken into consideration [8]. Most of the work in link prediction revolves around exploiting the structure of the network in order to compute features based on various assumptions about the given network and predicting links based on those features. The results in such methods reflect the effectiveness of the assumptions about the network. However, networks are vast and represent complex systems that are significantly different from each other. The specific features designed for a particular type of network do not perform well with a different network. To sum up, methods exploiting the structure of the network to calculate similarity between nodes only take a certain aspect of the network into consideration while other insights can also be inferred which are not acknowledged by these methods.

Revised Manuscript Received on March 30, 2020.
* Correspondence Author

Sanjay Kumar*, Department of Computer Science and Engineering Delhi Technological University, New Delhi, India.
Vipul Gupta, Department of Computer Science and Engineering, Delhi Technological University, New Delhi, India.
Sudhanshu Shekhar Singh, Department of Computer Science and Engineering, Delhi Technological University.

Previously, a lot of work has been done in the field of link prediction which exploited node heuristics such as Jaccard coefficient, metrics based on the in-degree and out-degree of the nodes and other complex mathematical operators for finding the overlapping neighbors. The stated features estimate the node similarities and thus, predict the behavior of the newly observed nodes with respect to the general trends observed on the similar nodes. Most of the research work in this field revolves around solving the specific problem at hand by taking into account the assumptions associated with that problem. We use network embeddings to represent nodes. Each node is represented using the low dimensional vector. Similar nodes have vector representations close to each other. The vector representation of the node acts as the feature set for the node. The higher the dimension of representation of a node into a vector, the more features are represented and thus, the model is expected to learn better in the case of high dimensional embeddings.

We also use other features along with embeddings to improve the accuracy of our model [9]. The other similarity measures used are :

First Set of Features

Jaccard Index: It is the ratio of common neighbors between two nodes to the total number of neighbors of the node [10].

Adamic Adar: It is defined as the addition of the inverse logarithmic degree centrality of the common neighbors of the two nodes [10].

Dot Product of Node Embeddings: The use of vectors and a dot product instead of simply points and a distance measure allows us to understand the propensity to connect along two different axes.

Second Set of Features

Common Neighbors: Common neighbors algorithm is based on the idea that two stranger individuals who have a mutual friend are more likely to get to know each other than those who don't have any friends in common.

Resource Allocation: Resource allocation is the process of allocating and handling the resources in accordance with an organization's principles and strategies.

Preferential Attachment: A preferential attachment process is one of those processes in which some asset is distributed among the network users in a biased way so that the rich user nodes receive more than others. [14]

Friend TNS score: The basic function of the friend TNS algorithm is to evaluate the similarities from a specific user node to all other user nodes present in a network, using our fundamental as well as derived node similarity measures [11].

Our link prediction algorithm, mainly, leverages the network embedding techniques, Deep Walk based on biased random walks and node2vec based on unbiased random walks. Initially the edge embeddings are obtained by the hadamard product of the node embeddings of the nodes present in the edges. We intend to obtain edge embeddings of various dimensions.

Further, we calculate the combined prediction result of edge embeddings along with the two set of features. The first set of features include the overlapping measures evaluated on the edge embeddings such as Jaccard similarity, Adamic-Adar score and dot product between node embeddings. These measures signify the overlapping similarity among the edges considered. The second set of features includes important heuristic features such as Common Neighbors, Resource Allocation, preferential attachment & friend tns score.

The models are built by exploiting the edge embeddings and the two sets of features separately in order to provide the independent result from the embeddings and the other feature space. Later, we perform the ensembling procedure on the two separate model results applied on the same set of edges through the averaging method in which we take an average of predictions from all the models and use it to make the final prediction. Finally, we assign a threshold value and classify the data with respect to the average value generated.

The rest of the paper is organized as follows: Section 2 represents the related work performed in the concerned field; Section 3 discusses the dataset used; Section 4 includes the methods used in our approach; Section 5 explains the implementation followed by Section 6 expounding the results. Finally, Section 7 concludes the research paper.

II. RELATED WORK

The link prediction research is ever-evolving. The traditional link prediction approaches lead to the extraction of the implicit information available in the network [4]. Link Prediction mechanisms have been applied efficiently to biological networks to predict previously unknown interactions between proteins [7]. We also come across the link prediction applications in our daily life like when we see the recommendations for the people we may know based on the existing relationships on various social media platforms [12]. Networks have been observed carefully since the proposition of the basic models in order to interpret the structural features of the network and learn about the network formation protocols [13].

We learnt about a lot of significant work in the field of link prediction in social networks based on psychological theories like social status and balance theory [4, 15]. Most of the research work has been focused on the extraction of information from the network by making network specific assumptions. These traditional mechanisms perform well on the specific networks while they fail miserably on the networks in which the assumptions do not hold true [8]. There are techniques like node embedding that encode the nodes present in the network into low dimensional vectors. These vectors represent the features of the nodes. Node embedding techniques can be applied to various types of networks to obtain high accuracy results [9].

III. DATASET DESCRIPTION

A. Email-Enron Dataset

The Enron email dataset comprises approximately half a million emails sent by the employees working at the Enron Corporation. It includes data concerning 150 users, mostly the senior management executives of Enron, stored into well-organized folders. It has 36692 nodes and 183831 edges. Here nodes represent email addresses and edges represent that an email was exchanged between these two email addresses [16] [17].

B. Email-Eu-Core Dataset

It includes data regarding the network of email exchange systems across the European research organizations. It consists of 1005 nodes and 25571 edges. The edge is formed in the network if atleast one email is transferred from node u to node v [18][19].

IV. METHODOLOGY

A. Node Embedding Methods

1. Node2vec

Node2vec is an improvement over deepwalk. In Node2Vec the random walks are governed by two principles P and Q. Parameter P signifies the probability of the random walk returning to the previous node. Parameter Q defines the probability that the random walk will discover the undiscovered part of the graph.

The parameter P controls the discovery of microscopic view around the node. Parameter Q controls the discovery of undiscovered parts of larger undiscovered neighborhoods in order to infer about communities and complex dependencies [10].

2. Deep Walk

Deep Walk uses random walks to generate node embeddings. The random walk starts from a source node. From the current node we move to other nodes randomly for a defined number of steps [9].

This method has three steps:

Sampling: Sampling of graphs is done using random walks. About 32 to 64 random walks are performed from each node.

Training Skip-Gram: Skip-gram network takes a node from the random walk as an input as a one-hot vector. It typically predicts 20 neighbors, 10 on right and 10 on left.

Computing Embeddings: The hidden layer of network outputs embeddings for the nodes. Node embeddings for each node is computed.

Deep walk is an unbiased method which does not preserve local information of nodes to choose the next node.

B. Edge Embedding Method

An edge embedding e^x (u,v) , can be computed by the Hadamard Product of the corresponding node embeddings. The Hadamard product operates on two matrices of equal dimensions and outputs the resulting matrix with a number of dimensions same as the operand matrices. Each element i, j of the output matrix is produced by the multiplication of elements i, j of the input matrices [20].

V. IMPLEMENTATION

We use node embedding techniques to encode the nodes of the graph and evaluate prediction accuracy of two models on the datasets provided to us. We perform the following steps :

Input: List of edges

Output: model and accuracy of model

1. Divide the graph datasets into two parts i.e. network edges (80%) and test edges (20%) after random shuffling of the entire dataset.
2. Build the graph G using edges in the network edge list. The test edges are reserved for training the model later.
3. Select the embedding method to encode the nodes of the graph G.
4. Select the number of dimensions to determine the size of vector representation of the nodes.
5. Implement the selected embedding method on the graph G generated to compute node embeddings for the nodes of the graph.
6. Identify the edges that are non-existent i.e. the pairs of nodes present in the graph G with no edge formed between them and randomize the entire sample.
7. Calculate the edge embeddings for the edges present in our dataset by calculating the hadamard product of the respective node embeddings of the nodes which constitute the edge.
8. Split the data into training and testing data.
9. Build the neural network model with the input size same as the number of dimensions.
10. Train the model created in step 9 on the training data generated in step 8.
11. Calculate the accuracy of prediction of the model on testing data.
12. Extract the additional features including overlap function outputs and heuristic measures and prepare the dataset by assigning the respective labels for the edge.
13. Split the new dataset created in step 12 in training and testing datasets.
14. Create a SVM classification model for performing prediction on the new dataset prepared in step 11.
15. Train the model created in step 13 on the training dataset generated in step 12.
16. Ensemble the models created in step 9 and step 11 with different feature spaces to obtain the improved accuracy.
17. Evaluate the accuracy of prediction of the new model on test data.

We concatenate these embedding features with other set of features which are listed below:

First Set of Features: This set includes overlap functions. The features are dot products between node embeddings, Jaccard similarity and Adamic-Adar score.

Second Set of features: This set of features includes important heuristic features. The features include Common Neighbors, Resource Allocation, preferential attachment and friend tns score.

With the help of the final sets of features, we select the best values for the hyper parameters for node embeddings. The values of the hyper parameters such as learning rate, optimizer and assigning weighted sum are tuned in order to extract the embeddings that best suit the network being analyzed and that extracts and represents the features of underlying network data in the best possible way. We select machine learning techniques to train the model on the basis of the features extracted from the above features. We use SVM and Neural networks to train our model.

VI. RESULTS

We trained our model on the features extracted from the network data according to the implementation approach explained in section 5. We extracted node embeddings and the two sets of features (i.e. first and second sets of features based on overlap functions and heuristic measures respectively) from the data and we obtained the results from the combined model.

A. Enron- Email Dataset

The following Fig. 1 1and Fig. 2 depict the graphical representation of the accuracy of the model based on Node2Vec embeddings and Deep Walk embeddings respectively.

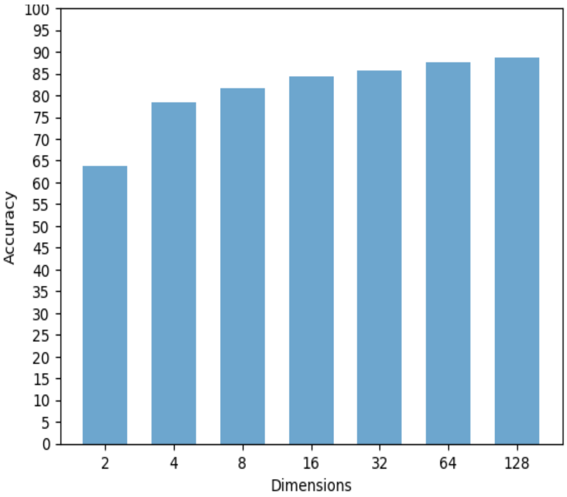

Fig. 1 Accuracy of the model based on the Node2vec model on Embeddings only.

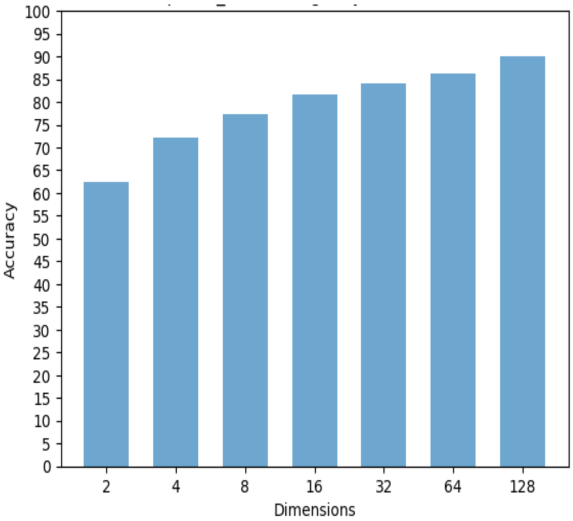

Fig. 2 Accuracy of the model based on the Deep Walk model on Embeddings only.

The following Fig. 3 and Fig. 4 shows the accuracy comparison of the model based on Node2Vec and Deep Walk embeddings respectively before and after the introduction of the first and second set of features.

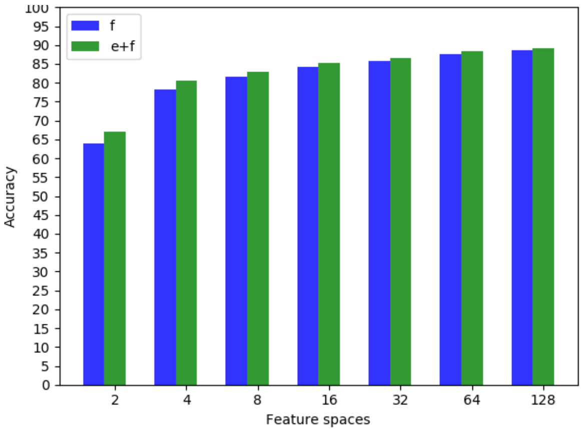

Fig. 3 Accuracy Comparison of the model based on the Node2vec model on Embeddings and additional features.

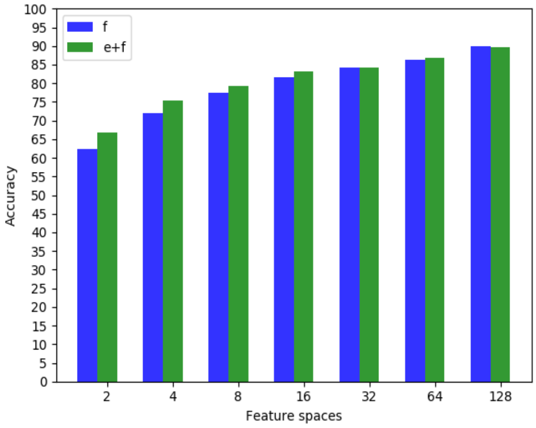

Fig. 4 Accuracy Comparison of the model based on the Deep Walk model on Embeddings and additional features.

The Receiver Operating Characteristic (ROC) curve is a plot of False Positive Rate(FPR) and True Positive Rate(TPR) at various threshold probability values. ROC analysis is used to select optimum models for binary classification problems. Precision is calculated as the ratio of total number of true positives and the number of all the instances predicted as positives (i.e. sum of true positives and false positives). Precision is the measure of how good the model predicts a positive class. Recall is the ratio of the number of true positives and the sum of true positives and the false negatives. The True Positive Rate is also known as Recall and Sensitivity.

Precision-Recall graphs are most useful in cases of imbalance in data. Precision-Recall graph is used as a measure when one type of event occurs more frequently and the frequency outnumbers the other event by a significant amount.

The following section consists of the figures Fig. 5, Fig. 6, Fig. 7 and Fig. 8 representing the Precision-Recall and the ROC curves for the 128 dimensions of the Node2Vec and Deep Walk embeddings.

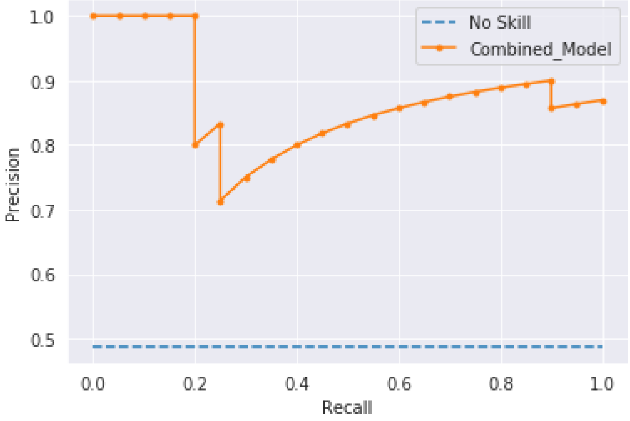

Fig. 5 Precision vs Recall graph for the model trained on 128 dimensions of Node2Vec Embeddings on Enron-Email Dataset

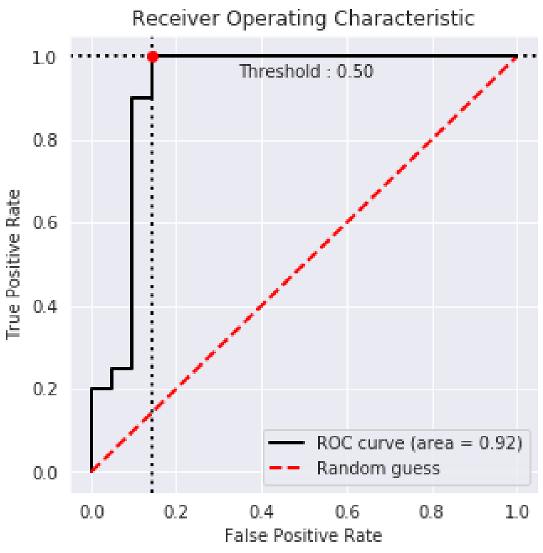

Fig. 6 ROC curve for the model trained on 128 dimensions of Node2Vec Embeddings on Enron-Email Dataset

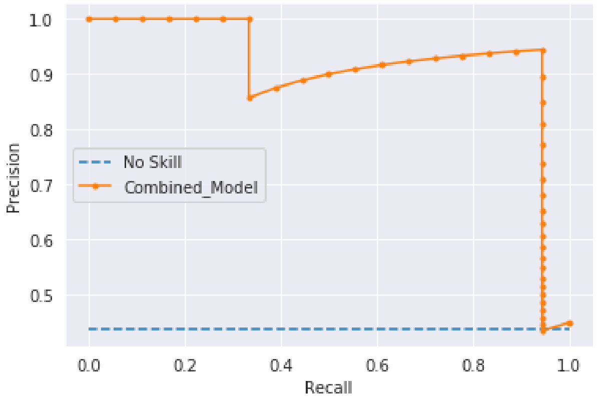

Fig. 7 Precision vs Recall graph for the model trained on 128 dimensions of DeepWalk Embeddings on Enron-Email Dataset

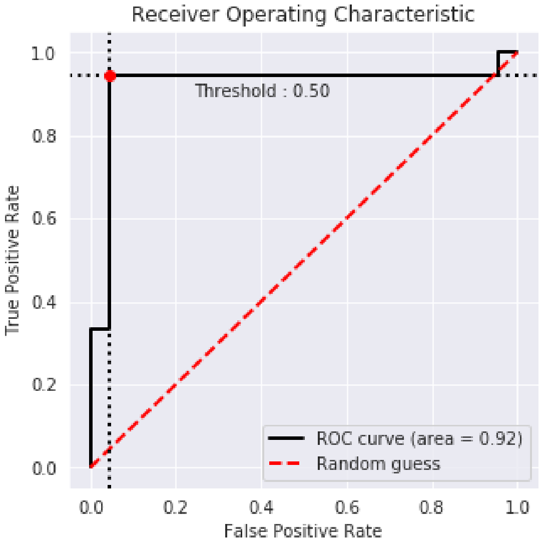

Fig. 8 ROC curve for the model trained on 128 dimensions of Deep Walk Embeddings on Enron-Email Dataset

B. Email-Eu-core Dataset

The following Fig. 9 and Fig. 10 depict the graphical representation of the accuracy of the model based on Node2Vec embeddings and Deep Walk embeddings respectively.

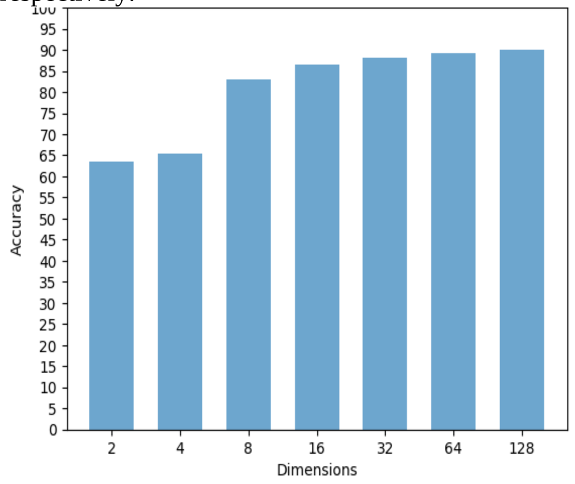

Fig. 9 Accuracy of the model based on the Node2vec model on Embeddings only

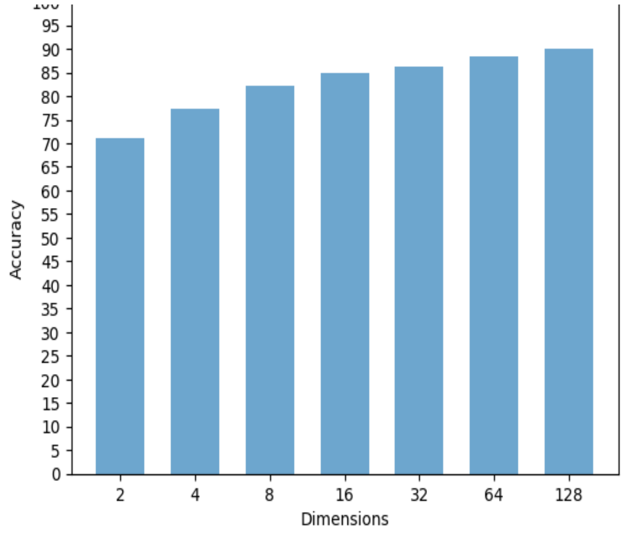

Fig. 10 Accuracy of the model based on the Deep Walk model on Embeddings only.

The following Fig. 11 and Fig. 12 show the accuracy comparison of the model based on Node2Vec and Deep Walk embeddings respectively before and after the introduction of the two sets of features including overlap functions and heuristic measures.

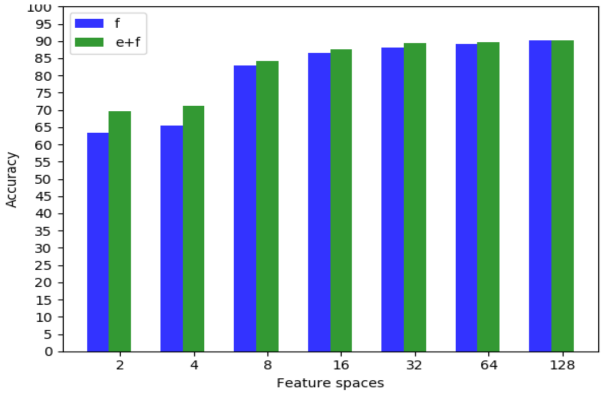

Fig. 11 Accuracy Comparison of the model based on the Node2vec model on Embeddings and additional features.

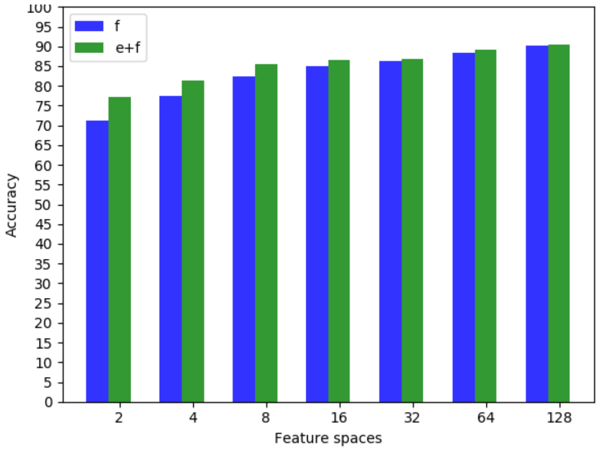

Fig. 12 Accuracy Comparison of the model based on the Deep Walk model on Embeddings and additional features.

The following section consists of the figures Fig. 13, Fig. 14, Fig. 15 and Fig. 16 representing the Precision-Recall and the ROC curves for the 128 dimensions of the Node2Vec and Deep Walk embeddings.

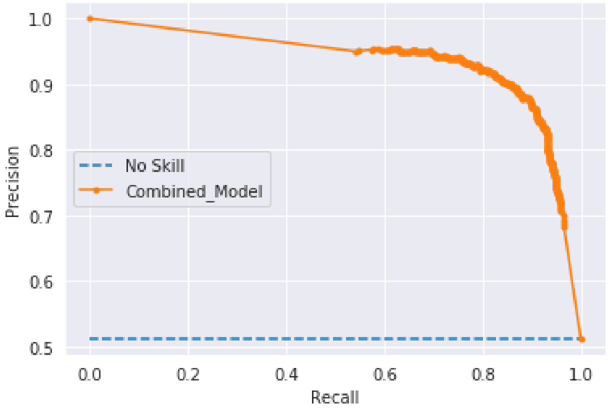

Fig. 13 Precision vs Recall Graph for the model trained on 128 dimensions of Node2Vec Embeddings on Email Eu-Core Dataset

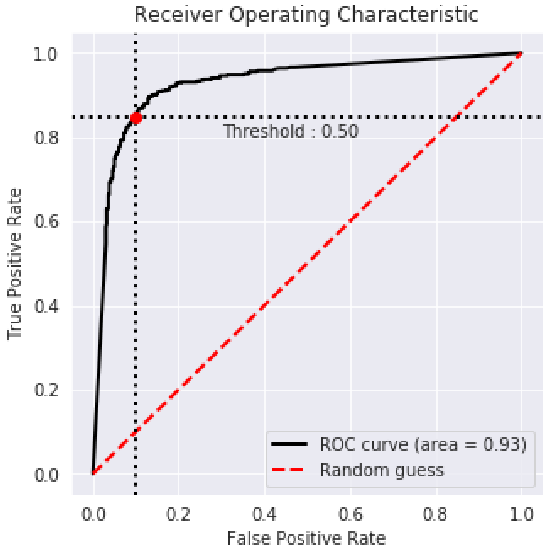

Fig. 14 ROC curve for the model trained on 128 dimensions of Node2Vec Embeddings on Email Eu-Core Dataset

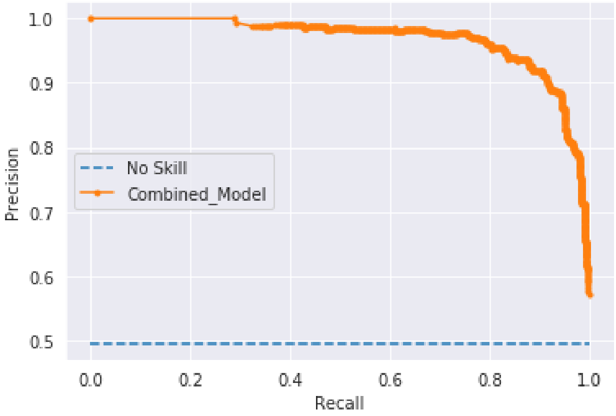

Fig. 15 Precision vs Recall Graph for the model trained on 128 dimensions of DeepWalk Embeddings on Email Eu-Core Dataset

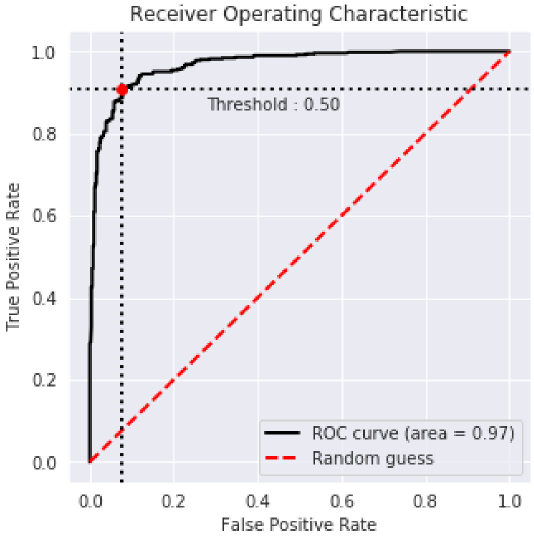

Fig. 16 ROC curve for the model trained on 128 dimensions of DeepWalk Embeddings on Email Eu-Core Dataset

The additional sets of features are integrated with the edge embeddings, and the accuracy of the combined model is evaluated. The "Improved Accuracy" part of Fig. 2, Fig. 4, Fig.6 and Fig. 8 shows the predictive accuracy of the combined model. There is a very significant increase in the predictive accuracy of the model.

VII. CONCLUSION

In this paper, we performed the link prediction task using the network embedding and similarity measures. Network embedding techniques, which signify the translation of high-dimensional vectors into the low-dimensional vector representation, help us to achieve high prediction accuracy when the edge embeddings are computed in a high-dimensional space. Moreover, the concatenation of the additional features including the overlap functions and the heuristic measures helps in the reduction of the prediction loss occurred mostly due to unavailability of the optimum number of features in the case of low-dimensional edge embeddings. Also, these features help us to improve the results obtained corresponding to the higher dimensional embeddings to some extent.

REFERENCES

1. Srinivas, V., Mitra, P. : Applications of link prediction. In Link Prediction in Social Networks (pp. 57-61). Springer, Cham. (2016).
2. Al Hasan, M., Zaki, M. J.: A survey of link prediction in social networks. In Social network data analytics (pp. 243-275). Springer, Boston, MA. (2011)
3. Wang, P., Xu, B., Wu, Y., Zhou, X. : Link prediction in social networks: the state-of-the-art. Science China Information Sciences, 58(1), 1-38.(2015)
4. O'Madadhain, J., Fisher, D., Smyth, P., White, S., & Boey, Y. B.:Analysis and visualization of network data using JUNG. Journal of Statistical Software 10.2 (2005): 1-35
5. Lu, Y., Guo, Y. Korhonen, A. : Link prediction in drug-target interactions network using similarity indices. BMC Bioinformatics 18, 39 (2017).
6. Kovács, I.A., Luck, K., Spirohn, K. et al. Network-based prediction of protein interactions. Nat Commun 10, 1240 (2019).
7. Liben-Nowell, David, and Jon Kleinberg. "The link-prediction problem for social networks." Journal of the American Society for information science and technology 58.7 (2007): 1019-1031.
8. Verma, Janu, et al. "Heterogeneous Edge Embedding for Friend Recommendation." European Conference on Information Retrieval. Springer, Cham, 2019.
9. Goyal, Palash, and Emilio Ferrara. "Graph embedding techniques, applications, and performance: A survey." Knowledge-Based Systems 151 :78-94 (2018):
10. Gao, Fei, et al. :Link prediction methods and their accuracy for different social networks and network metrics." Scientific programming (2015).
11. Rai, Abhay Kumar, et al. "A Survey on Link Prediction Problem in Social Networks."
12. Martínez, Víctor, Fernando Berzal, and Juan-Carlos Cubero. "A survey of link prediction in complex networks." ACM Computing Surveys (CSUR) 49.4 (2016): 1-33.
13. Newman, Mark EJ. "The structure and function of complex networks." SIAM review 45.2 (2003): 167-256.
14. Kunegis, Jérôme, Marcel Blattner, and Christine Moser. "Preferential attachment in online networks: Measurement and explanations." Proceedings of the 5th Annual ACM Web Science Conference. 2013.
15. Situngkir, Hokky, and Deni Khanafiah. Social balance theory: Revisiting heider's balance theory for many agents. Bandung Fe Institute, 2004.
16. J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. Internet Mathematics 6(1) 29--123, 2009.
17. B. Klimmt, Y. Yang. Introducing the Enron corpus. CEAS conference, 2004
18. Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. "Local Higher-order Graph Clustering." In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017.
19. J. Leskovec, J. Kleinberg and C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters. ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1), 2007.
20. Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016.