Performance based Comparison between Several Link Prediction Methods on Various Social Networking Datasets (Including Two New Methods)

Ahmad Rawashdeh
Department of Computer Science and Math
University of Central Missouri (UCMO)
Lee's Summit, MO, USA

***Abstract*—This work extends my previous work on link prediction in Social Networks. In this research, I used two additional datasets, Twitter dataset and Facebook Social Circles Dataset and I ran link prediction methods on these datasets. In my previous work, I performed experiment on the Facebook dataset and proposed two new link prediction methods: Neighbors Connectivity and Common Neighbors of Neighbors (CNN). As in my previous work, in this work, I ran the link prediction methods for several training and testing sizes. Results showed that For Facebook dataset, random had the highest precision, followed by Neighbors Connectivity, then Preferential Attachment, followed by Jaccard/CC, Adamic-Adar, finally CNN. For Twitter dataset, random achieved the highest precision. Preferential Attachment achieved the next highest precision, and Adamic-Adar achieved the least precision. For Facebook Social Circles dataset, Preferential-Attachment achieved the highest precision of 1.08891 followed by random for a training and testing sizes of (1535, 2504) respectively. That is said with slight variation on the orders depending on the training and testing size. The low precision values achieved with Facebook and Twitter datasets are due to the graph types which are sparse as indicated in the datasets websites which confirms Kleinberg finding.**

Keywords*—Social networks; link prediction; comparison; experiment

I. INTRODUCTION

Social Networks have become an essential part of our lives nowadays. Social Networks' providers are now competing to offer users the best platform, services, and safe environment for all social activities including social collaboration, networking, sharing of textual, image, or video posts and even reaction to such posts in the form of likes or replies. One of the most popular Social networking sites are Facebook[1] (and the likes including Friendster[2] before changing to entertainment website, Zorpia[3] which is now two, and Myspace[4]), Twitter[5], Instagram[6], YouTube[7], Reddit[8], LinkedIn[9], and some of the new ones such as TikTok[10], Snapchat[11] to name a few. Knowing that every social networking website has a unique as well as common audience with others, some are known to be different than the others in providing certain services. For example, Facebook is used for friendship and content (videos, images) sharing and replies. Twitter is for posting short posts, called tweets, and for using Hashtags which helps in getting the newest posts about a certain important event. Instagram is for images and videos sharing. YouTube is for video sharing (originally was part of what was known as Web 2.0). Reddit is for news. LinkedIn is for professional networking. This work focuses on the two-social networking: Facebook and Twitter. The reason for that follows in the next paragraphs.

Facebook and Twitter remain two of the most popular social networking websites. With 2.70 billion users (see https://www.omnicoreagency.com/facebook-statistics/), Facebook is the largest Social Network to date (see https://makeawebsitehub.com/social-media-sites/). Twitter is the 8th largest (see https://makeawebsitehub.com/social-media-sites/) with users count of 340 million (see https://www.omni coreagency.com/twitter-statistics/). That is enough said about the importance of social networking and their applications. In Facebook, most users are familiar with the "People you May Know" feature which is an example of a link prediction application/service (see People You May Know at https://www.facebook.com/help/www/336320879782850).
This work is concerned with this kind of application of social network, namely the application or problem of link prediction in social network. So, what is link prediction?

Link Prediction is still one of the active areas in research due to the importance of its various applications which range from predicting links of friendship/followship in social network such as Facebook and Twitter respectively to areas such as biology, co-authorship, networking, and medicine. Which are only few of the examples of the areas or domains where link prediction could be used. More about the application of link prediction can be found at [1]. The importance of this work can be realized by understanding the

[1] www.facebook.com
[2] www.Friendster.com
[3] www.zorpia.com
[4] www.mysapce.com
[5] www.twitter.com
[6] www.instagram.com
[7] www.youtube.com
[8] www.reddit.com
[9] www.linkedin.com
[10] www.tiktok.com
[11] www.snapchat.com

importance of applications of link prediction. The reader may refer to [2], [3], [4], [5], [6], [7], [8], [9] for detailed applications.

Limitation of this work lies in the relatively small dataset sizes, even though it is still huge, compared to the actual data sizes of social networks. However, this current attempt to use the largest possible dataset sizes, on the current system (16384 MB Memory), was a success.

II. Related Work

So much work has been conducted on social network, particularly on link prediction on social networks. John Kleinberg [10] was the first to term this kind of research as link prediction. He compared between several link prediction methods including, to name a few: Common Neighbors, Jaccard, Adamic-Adar, Preferential Attachment, and random. Also, he considered global link prediction methods such as Kats which uses the ensembles of all paths between the nodes/vertices of interest. Kleinberg ran link prediction algorithms on co-authorship network. Since the prediction algorithms had very low performance values, he measured the relative performance of various predictors versus the common neighbors as well as random predictor (factor improvement). He found that Adamic-Adar and Kats had the highest factor improvement over random.

A survey of link prediction method as well as an experiment was carried out in [11], in which the authors classified link prediction methods into: node based, link based, and path based. The experiment was conducted on the Epinion (a website of reviews) dataset. 10% of the links from the graph were removed for testing purposes. The results showed superiority of Local Random Walk (LRW) algorithm, even though nodes' neighbors, Jaccard, and Supervised Random Walk (SRW) were close. It was concluded that LRW was the best in terms of precision among the compared methods (about 12 methods). However, only one dataset was used, and no new methods were proposed. Also, no detailed information about the used dataset was provided.

In [12] another survey of link prediction methods was performed, but it is theoretical based survey which clearly implies that no experiment was carried out as in this work.

In [13] an experimental comparison of five link prediction methods were performed. The paper also introduced a new link prediction method called LinkGyp. The experiment was carried on three datasets (not the same that are used in this work).

Other works in which more link prediction methods were proposed can be found at [14] and [6] which investigated a machine learning classifier to predict links. The earlier, constructed a feature vector from topological information and node attributes and was evaluated on a co-authorship dataset. The later evaluated the algorithm on 10 different datasets. Also, the research in [15] evaluated two new proposed methods.

In [16] a new link prediction method was proposed, called Time-aware Multi-relational Link Prediction (TMRLP) which combine the dynamic or of the graph topology and interaction history. Results showed that it outperformed the existing methods when ran on DBLP dataset.

An intensive comparison between link prediction methods was conducted in [2]. However, even though the comparison included so many link prediction methods, the datasets used are different from the one used in this work. This work's focus is in using link prediction in Social Network. The research work just been cited found that new links can be better predicted using only local or quasi-local information in most networks. Considering indirect connections only adds noise and computational complexity to the link prediction problem.

All the cited work before differs from this current work in either the used datasets, the application of link prediction, the existing and proposed link prediction methods which are studied, the results, or the experiment setting (running the experiment for several training and testing sizes which has been done in this work).

In [17], I investigated the semantics in link prediction. Also, our work in [18] was about using semantic in finding similarity in Social Networks which can be used in link prediction.

This research extends the work in [1], which compares between link prediction methods and proposes two new methods, by applying the link prediction methods (including the two proposed) on two additional datasets.

III. Problem Definition

In this section, we define formally the problem of link prediction. Given a graph G represented as G = (V, E). Where V is the set of all vertices/nodes, and E is the set of all edges/connection. Which edges might be formed in the future or which missing edges can be predicted? [19] These are two are two different version of the problem. So, given the graph at time t_0, at time t_1, which new edges can be predicted or simply given an instance of the graph, if some edges had been deleted, which missing edges could be predicted. [17].

IV. The Algorithms

The algorithms considered in this paper are: Common Neighbors, Jaccard, Adamic-Adar, Preferential-Attachment, Random, Common-Neighbors-of-Neighbors and Node-Connectivity. The last two methods were proposed by my work at [1].

A. The Formulas for the Algorithms

The formula for links predictions considered in this paper are as follows (more details in [1]):

1. Common neighbors [10]
   Score (x and y) = $|\Gamma(x) \cap \Gamma(y)|$ (1)
2. Jaccard's coefficient [10]
   Score(x and y) = $\frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$ (2)
3. Adamic-Adar [10]
   Score (x and y) = $\sum_{z \in |\Gamma(x)| \cap |\Gamma(y)|} \frac{1|}{\log |\Gamma(z)|}$ (3)
4. Preferential attachment [10]
   Score (x and y) = $|\Gamma(x)| . |\Gamma(y)|$ (4)
5. Random
   Score (user x and user y) =
   true or false depending on the binary random value computed using Math.Random (5)
6. Neighbors Connectivity, Hybrid (proposed in my work in [1])
   a) If Common Neighbors in (1) gives a Score(x and y) >= 1, use that score
   b) Else use the following formula:
   Score (x, and y) =
   Average degree of neighbors of neighbors of user x +
   Average degree of neighbors of neighbors of user y
   $= \frac{\sum_{z \in \Gamma(x)} |\Gamma(z)|}{|\Gamma(x)|} + \frac{\sum_{z \in \Gamma(y)} |\Gamma(z)|}{|\Gamma(y)|}$ (6)
7. Common Neighbors of Neighbors is calculated as follows (proposed in my work in [1])
   Score (x and y) =
   $|(\text{neighbors}(\text{neighbors}(x)) \cap (\text{neighbors}(\text{neighbors}(y))| = |\Gamma(\Gamma(x)) \cap \Gamma(\Gamma(y))|$ (7)

V. Datasets

This work extends my work at [1] by comparing between link prediction on additional datasets. I used Facebook (friendship: undirected) in that work. In this work, I am also using Twitter (fellowship: directed), and Facebook SocialCircles (contains friendship, profiles' features, and circles), and possibly MoviesGalaxies Social datasets (edges connecting similar movies) in the near future.

I used the Facebook (more about it in my work in [1]), Twitter, and Facebook Social Circles datasets. Information about them can be found in Table I.

More about Twitter dataset can be found and downloaded at the link http://networkrepository.com/ego-twitter.php. The Twitter dataset was created by the work at [20]. The dataset, as indicated on the website, contains followship: user to user following information. A node represents a user. An edge indicates that the user represented by the left node follows the user represented by the right node. It is worth noting that the graph is sparse, with not so many edges (that explains the low precisions listed in the result section which is fine since also Kleinberg got similar low performance values, so he measured the factor improvement over selected algorithm [10]).

TABLE I. Information about used Datasets (Size: Number of Nodes, and Number of Edges)

	nodes	edges	Average clustering coefficient
Facebook	2699 nodes used all used in this and in previous work	2981 edges all used in this and in previous work	0.0272474
Twitter	Total: 23400 nodes used: only nodes whose indices are from 1 to 20000 are included in the processed dataset. For example, the node 20011 was not included.	Total:33101 edges used: < 33101 edges	0.1014
Facebook Social Circles	4039 nodes	edges 88234	0.6055

Facebook Social Circles can be downloaded at the link https://snap.stanford.edu/data/egonets-Facebook.html. The dataset was created from the work in [21]. The dataset contains circles (or friends list from Facebook). Also, the dataset contains node features (profiles), circles, and ego networks.

VI. Running the Algorithms

To know more about the program which I wrote for the experiment, the reader may refer to my previous paper at [1]. I have extended the program and ran it on Facebook dataset (again), Twitter dataset, and Facebook Social Circles Dataset, and then generated output for the three datasets. The precision of every studied link prediction method was calculated for Twitter, Facebook, and Facebook Social Datasets for various training and testing sizes. The link prediction methods used are Common Neighbors, Jaccard, Adamic-Adar, Common Neighbors of Neighbors (CNN), Node Connectivity and Random. CNN and Node Connectivity were proposed by me and experimented with in my previous paper. Several training and testing sizes of the datasets were used to reach the most ever possible generic conclusion based on the current experiment on the datasets. The training sizes attempted are 20%, 25%, 40%, 50%, and 62% of the total dataset size (see Table I). The following were the training and testing sizes for the Facebook dataset in the format of (training, testing): (2159 ,540), (2000,699), (1620 ,1079), (1350 ,1349), and (1000 ,1699) [1]. Initially, for Twitter dataset, I used the following training sizes: 2196, 2745, 4392, 5491, 6808. Then used later the following training sizes and testing sizes (since I reduced total nodes size to 20000) in the format (training size, testing size): (16000, 4000), (15000, 5000), (12000, 8000), (10000, 10000), (7600, 12400). The later sizes of Twitter dataset were used because I encountered an "out of memory" exception (see the sub section below) but still the same percentage of training data sizes were used. The following were the training and testing sizes for the Facebook Social Circles dataset in the format of (training, testing): (3232, 807), (3030, 1009), (2424, 1615), (2020, 2019), and (1535, 2504).

A. Problem: Memory

The twitter data is very large (could not create a two-dimensional array of size more than 20000 x 20000). So, I had to reduce the graph data to include only edge information for

source nodes starting from node 1 to node 20000. That explain the exception which I encountered "OutOfMemory" exception during the running of CC algorithm (Common neighbors), so I had to focus on resolving this issue before running all algorithms on the dataset.

Initially, I was allocating memory for more than 20000 entries, however, running into the exception, "OutOfMemory" made me change my coding. So, technically speaking, I had to modify the class Graph.cs (the class file used to store the graph data) to use linked adjacency matrix (GraphLinkedData of type Dictionary<int, List<int>>) to reduce the used memory. I only stored the links without having to allocate wasted memory as in the case of when using the matrix GraphData of type int [,]. That was enough to make it work, and by the time I had a running program that stores the edges' information in a dictionary (not matrix), the program was ready to run all algorithms. So, lesson learned, only the links need to be stored, and no memory needs to be allocated for any non-existing link, missing link (between pair of nodes). Sine that correction, I had not encountered any similar error. The keys of the GraphLinkedData dictionary were 20000 of count so I succeeded in using most of the dataset.

As mentioned previously, I had to modify the algorithms (CC, jaccard, Adamic-Adar, preferential-attachment, random, CCNeighbors_of_neighbors, NodeConnectivity) to use the linked list adjacency matrix (GraphLinkedData) when using twitter dataset to avoid the "OutOfMemory" exception.

Another related issue is that random took so much time (to initialize the candidate links and made the random prediction). In [1], For Facebook dataset, I was able to generate a random number for every possible edge between a pair of nodes, however for Twitter, since the number of nodes is very large, 20000, I generated the random number for only MaxTestingSize=12400 (which matches the last testing size for running the experiment on Twitter dataset), because that what was needed.

In summary, I found out having enough memory is important sometimes, however, it turns out that it is the limit of the C# programming language which I'm using. However, the program ran with no additional problems after that.

For Facebook Social Circles Dataset, the memory consumption was as follows while running the program (shown is the status of the program and memory):

- Reading the Facebook Social Circles Dataset (memory consumption less than 500MB).
- End reading the Facebook Social Circles Dataset (memory consumption less than 500MB).
- Common neighbors (memory consumption less than 500MB).
- Jaccard coefficient (memory consumption less than 500MB).
- while running preferential (memory 633 MB).
- while running Adamic (memory 860 MB then 915 MB and it started slowing).
- while running random (memory 1000MB speed got back well again).
- while running neighbor's connectivity (memory 1900 MB then 1700 MB), at this stage the program become very slow in processing the data.

so, I had to stop the program, modify the code then run it once again but only using Common neighbors, Jaccard, Adamic-Adar, Preferential-Attachment, and random. Then start the application again and run it for remaining algorithms (neighbors' connectivity, common neighbors of neighbors) after reading the dataset. So, the results for Neighbors Connectivity and Common Neighbors of Neighbors were obtained from a different run than from the remaining algorithms. However, it seemed as if the algorithms Neighbors Connectivity and Common Neighbors of Neighbors were slow because of the processing (higher complexity than the other algorithms) and the program ran them slowly for this dataset not because all algorithms were running altogether (required more than 6 hours to finish). Next section discusses the results of the work.

VII. Results and Discussion

After generating the predictions and calculating the precision, one can refer to Table II, Table III, and Table IV which show the precision of link prediction methods on Facebook Dataset, Twitter dataset, and Facebook Social Circles Dataset, respectively. Each table shows the precision of the algorithms for several training and testing sizes. See Section VI for the training and testing sizes used for each dataset.

The precisions are very low. Which confirms the finding by Kleinberg [10], that is due to the nature of the graph, which is sparse, where there are very few edges compared to all possible edges. Possible edges are equal to:

Possible edges

= all possible source nodes × all possible destination nodes

= (total number of nodes × (total number of nodes -1))

I am not counting self-loops, so,

Possible edges (twitter dataset) =

$$= 20000 \times 19999$$

$$= 399980000 \text{ possible edges.}$$

Table II shows the precision of all studied link prediction methods which were run on the Facebook dataset using several training and testing data sizes. As it can be observed, overall, random has the highest precision, followed by Neighbors Connectivity, then Preferential Attachment, followed by Jaccard/CC, Adamic-Adar, finally CNN. For CNN, the number of decimal digits used to format the display of the precision value was not enough, so even though it shows as 0, it is not actually 0, because there were positive predictions, but very few compared to the overall possible edges to be predicted. The used String Format method wasn't using enough decimal digits.

As illustrated in Table III, overall, the algorithms ordered from the one with highest precision to the one with the lowest are as follows: Random, Preferential-Attachment, Neighbors-Connectivity, Common-Neighbors-of-Neighbors (CNN), Jaccard/Common Neighbors, and finally Adamic-Adar.

All algorithms were run at once (together in a single execution). The dataset was read only once; the algorithms were run (one by one) and made their predictions. Then, testing data nodes were selected from the original dataset, and finally evaluation was performed, and results were plotted. All automated by on command to the program console. The algorithms were executed on the Twitter dataset for several values for the sizes of training and testing data (see Section VI). Jaccard and Common Neighbors have the same precision values that is may be due to logical error or simply they produced same output.

Just to confirm the ordering of the algorithms in terms of the precision mentioned earlier, and by checking Table III, one can observe the following: For a training data size of 7600, the random was the highest, followed by preferential attachment, followed by Neighbors_Connectivity, then CNN, then Jaccard/CC, finally Adamic-Adar. For a training data size of 10000, the random had the highest precision, followed by preferential attachment, then Neighbors-Connectivity, then CNN, then Jaccard/CC, finally Adamic-Adar. The same is true about other training data sizes.

Random algorithm predicted a link if the number 1 is generated (see [1]), 0 otherwise. Generating a series of binary random number shouldn't exhibit any pattern (that is what completely random mean) unless it is pseudo random. So, there shouldn't be any pattern among all sequence of generated binary values of 0 or 1 (which are used to decide whether a link is predicted or not in this research) and that how it was implemented, see [1]. On a related note, in an ideal situation the chance of predicting a link using random should 50% and the chance of not predicting a link should also be theoretically 50%. And since the graphs in the Twitter dataset [20] and Facebook dataset[12] are sparse graphs as indicated on the datasets websites, random beat other link prediction methods since there is not enough structural information in the graph (few links and the number of neighbors is few and that is what sparse graph means) and these methods are structural. That is not the case with Facebook Social Circles Dataset (as the graph is more dense and less sparse see Table I).

The third dataset, Facebook Social Circles, is not as sparse as the others, so we see higher precision values for all prediction algorithms. The precisions for training and testing data sizes of (1535, 2504), respectively for one algorithm is the highest for all algorithms compared to other training and testing sizes (for the same algorithm). However, among all algorithms, preferential attachment achieved the highest precision of 1.08891, and that occurred for a training and testing sizes of (1535, 2504). The next highest precision is for random for a value of 0.56517 for the same training and testing sizes. The next highest are the remaining algorithms.

Fig. 1, Fig. 2, and Fig. 3 show the precision values for link prediction methods on Facebook, Twitter, and Facebook Social Circles Datasets respectively for the maximum training data size 1000 for Facebook (training: 1000, testing: 1699), training data size of 7600 for Twitter (training: 7600, testing: 12400), and training data size of 1535 for Facebook Social Circles Dataset (training: 1535, testing: 2504). For Facebook Dataset, Neighbors Connectivity produced the highest precision after random, while for Twitter dataset, Random produced the highest precision and all remaining produce low values.

Fig. 4 shows the precision for link prediction methods for the maximum training data size for all datasets. The reader can see that the Facebook dataset produced the least precision values compared with other datasets, followed by Twitter, and finally Facebook Social Circles dataset which produced the highest precision values for Random and Preferential-Attachment link prediction methods.

[12] http://networkrepository.com/ego-facebook.php

TABLE II. PRECISION OF LINK PREDICTION METHODS ON FACEBOOK DATASETS FOR DIFFERENT TRAINING AND TESTING SIZES (TRAINING SIZE, TESTING SIZE)

	Facebook (http://networkrepository.com/ego-facebook.php)				
	← increase →				
Common Neighbors	(2159, 540)	(2000, 699)	(1620, 1079)	(1350, 1349)	(1000, 1699)
	0.00005	0.00003	0.00001	0.00001	0.00000
Jaccard	(2159, 540)	(2000, 699)	(1620, 1079)	(1350, 1349)	(1000, 1699)
	0.00005	0.00003	0.00001	0.00001	0.00000
Adamic Adar	(2159, 540)	(2000, 699)	(1620, 1079)	(1350, 1349)	(1000, 1699)
	0.00005	0.00002	0.00001	0.00001	0.00000
Preferential Attachment	(2159, 540)	(2000, 699)	(1620, 1079)	(1350, 1349)	(1000, 1699)
	0.00035	0.00021	0.00008	0.00005	0.00002
CNN (Common Neighbors of Neighbors)	(2159, 540)	(2000, 699)	(1620, 1079)	(1350, 1349)	(1000, 1699)
	0.0001	0.0000	0.0000	0.0000	0.0000
Neighbors Connectivity	(2159, 540)	(2000, 699)	(1620, 1079)	(1350, 1349)	(1000, 1699)
	0.01852	0.01128	0.00486	0.00310	0.00194
Random	(2159, 540)	(2000, 699)	(1620, 1079)	(1350, 1349)	(1000, 1699)
	0.02650	0.01615	0.00673	0.00428	0.00276

TABLE III. PRECISION OF LINK PREDICTION METHODS ON TWITTER DATASET FOR DIFFERENT TRAINING AND TESTING SIZES (TRAINING SIZE, TESTING SIZE)

	Twitter (http://networkrepository.com/ego-twitter.php)				
	← decrease →				
Common Neighbors	(16000, 4000)	(15000, 5000)	(12000, 8000)	(10000, 10000)	(7600, 12400)
	0.00034	0.00008	0.00038	0.00051	0.00096
Jaccard	(16000, 4000)	(15000, 5000)	(12000, 8000)	(10000, 10000)	(7600, 12400)
	0.00034	0.00008	0.00038	0.00051	0.00096
Adamic Adar	(16000, 4000)	(15000, 5000)	(12000, 8000)	(10000, 10000)	(7600, 12400)
	0.00013	0.00003	0.00015	0.00020	0.00038
Preferential Attachment	(16000, 4000)	(15000, 5000)	(12000, 8000)	(10000, 10000)	(7600, 12400)
	0.01132	0.00255	0.01271	0.01727	0.03338
CNN (Common Neighbors of Neighbors)	(16000, 4000)	(15000, 5000)	(12000, 8000)	(10000, 10000)	(7600, 12400)
	0.0006	0.0001	0.0007	0.0009	0.0017
Neighbors Connectivity	(16000, 4000)	(15000, 5000)	(12000, 8000)	(10000, 10000)	(7600, 12400)
	0.00913	0.00204	0.01022	0.01390	0.02688
Random	(16000, 4000)	(15000, 5000)	(12000, 8000)	(10000, 10000)	(7600, 12400)
	0.10981	0.02478	0.12418	0.16908	0.32700

TABLE IV. PRECISION OF LINK PREDICTION METHODS ON FACEBOOK SOCIAL CIRCLE DATASET FOR DIFFERENT TRAINING AND TESTING SIZES (TRAINING SIZE, TESTING SIZE)

	Facebook Social Circles				
	https://snap.stanford.edu/data/egonets-Facebook.html				
Common Neighbors	(3232, 807)	(3030, 1009)	(2424, 1615)	(2020, 2019)	(1535, 2504)
	0.00765	0.01287	0.02808	0.02274	0.06879
Jaccard	(3232, 807)	(3030, 1009)	(2424, 1615)	(2020, 2019)	(1535, 2504)
	0.00765	0.01287	0.02808	0.02274	0.06879
Adamic Adar	(3232, 807)	(3030, 1009)	(2424, 1615)	(2020, 2019)	(1535, 2504)
	0.00765	0.01287	0.02808	0.02274	0.06879
Preferential Attachment	(3232, 807)	(3030, 1009)	(2424, 1615)	(2020, 2019)	(1535, 2504)
	0.12079	0.20386	0.44354	0.35981	1.08891
CNN (Common Neighbors of Neighbors)	(3232, 807)	(3030, 1009)	(2424, 1615)	(2020, 2019)	(1535, 2504)
	not available took so much time				
Node Connectivity	(3232, 807)	(3030, 1009)	(2424, 1615)	(2020, 2019)	(1535, 2504)
	not available took so much time				
Random	(3232, 807)	(3030, 1009)	(2424, 1615)	(2020, 2019)	(1535, 2504)
	0.06255	0.10574	0.23033	0.18669	0.56517

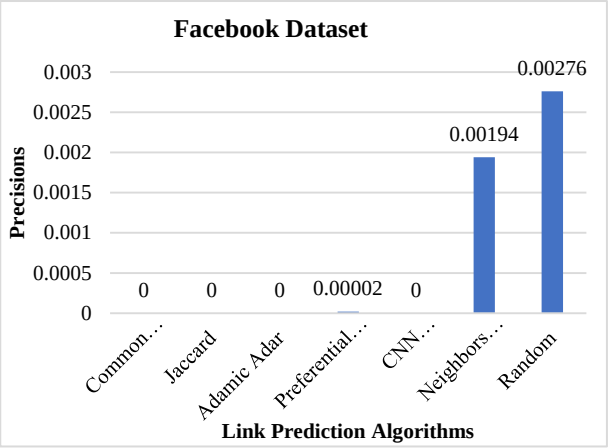

Fig. 1. Precision for the Algorithms on Facebook Dataset for the Max Training Size.

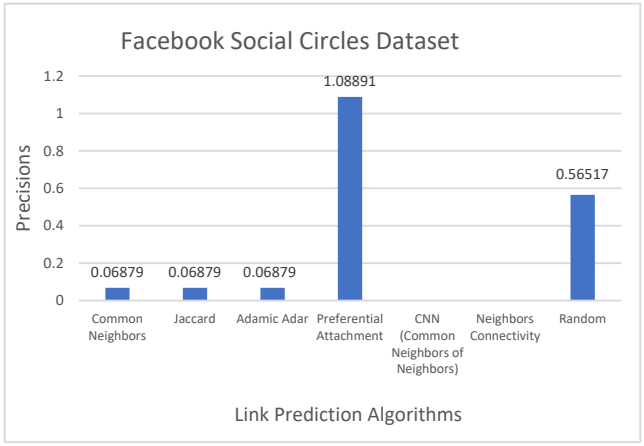

Fig. 3. Precision for the Algorithms on Facebook Social Circles Dataset for the Max Training Size.

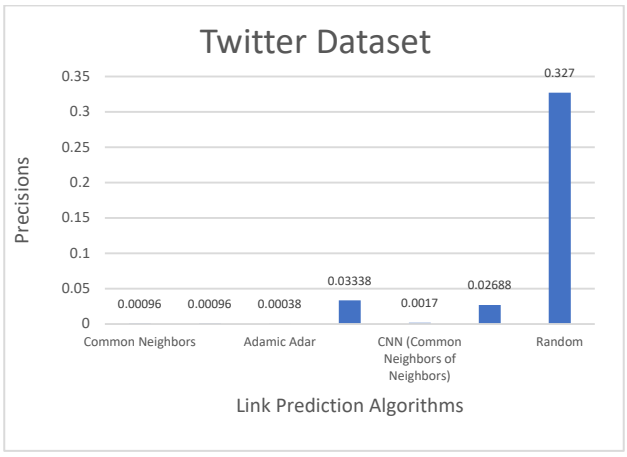

Fig. 2. Precision for the Algorithms on Twitter Dataset for the Max Training Size.

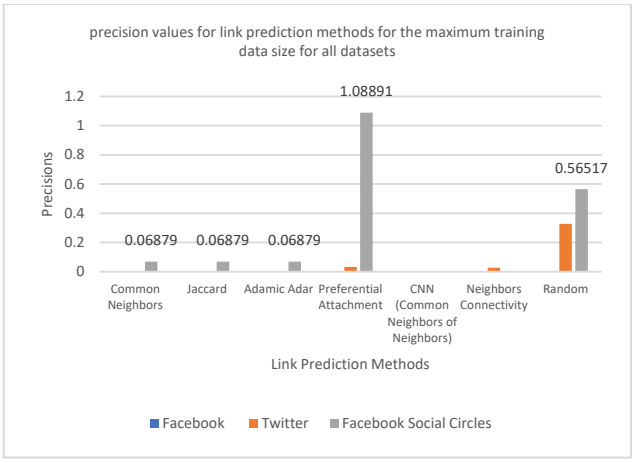

Fig. 4. Precision Values for Link Prediction Methods for the Maximum Training Data Size for All Datasets.

VIII. Conclusion

For Facebook dataset, random had the highest precision, followed by Neighbors Connectivity, then Preferential Attachment, followed by Jaccard/CC, Adamic-Adar, finally CNN. For Twitter dataset, random achieved the highest precision. Preferential Attachment achieved the next highest precision, and Adamic-Adar achieved the least precision. The running time of the algorithms was about (an estimate) 2-4 hours for Twitter and less than an hour for Facebook dataset. For Facebook Social Circles dataset, Preferential-Attachment achieved the highest precision of 1.08891 followed by random for a training and testing sizes of (1535, 2504), respectively.

IX. Future Work

Future work lies in considering new link prediction methods which could achieve better results and focus on factor improvement over certain predictor algorithm, similar to what Kleinberg did who also got low performance values. Also, finding the precision for the remaining methods for Facebook Social Circles Dataset. Another open area for research is using content as well as structural information of the graph in link prediction, and this could be done using the Facebook Social Circles Dataset. Previously, I considered the semantic in link prediction and that is also another very interesting area of link prediction.

References

[1] Rawashdeh A. An Experiment with Link Prediction in Social Network: Two New Link Prediction Methods. In: Arai K, Bhatia , Kapoor , editors. Future Technologies Conference; 10 October 2019. p. 563-581. Available from: https://link.springer.com/book/10.1007/978-3-030-32523-7.

[2] Martínez V, Berzal F, Cubero JC. A survey of link prediction in complex networks. ACM computing surveys (CSUR). 2016;49(4):1-33.

[3] Jalili M, Orouskhani Y, Asgari M, Alipourfard N, Perc M. Link prediction in multiplex online social networks. Royal Society open science. 2017;4(2):160863.

[4] Yang JX, Zhang XD. Revealing how network structure affects accuracy of link predictio. The European Physical Journal B. 2017;90(8):157.

[5] Wang T, He XS, Zhou MY, Fu ZQ. Link prediction in evolving networks based on popularity of nodes. Scientific reports. 2017;7(1):1-10.

[6] Fire F, Tenenboim-Chekina L, Puzis R, Lesser O, Rokach L, Elovici Y. Computationally efficient link prediction in a variety of social networks. ACM Transactions on Intelligent Systems and Technology (TIST). 2014;5(1):1-25.

[7] Al Hasan M, Chaoji V, Salem S, Zaki M. Link prediction using supervised learning. In: SDM06: workshop on link analysis, counter-terrorism and security; 2006. p. 798-805.

[8] Ibrahim NMA, Chen L. Link prediction in dynamic social networks by integrating different types of information. Applied Intelligence. 2015;42(4):738-750.

[9] Dai C, Chen L, Li B. Link prediction based on sampling in complex networks. Applied Intelligence. 2017;47(1):1-12.

[10] Liben-Nowell D, Kleinberg J. The link prediction problem for social network. Journal of the American Society for Information Science and Technology. 2007 March;58(7):1019–1031.

[11] Sharma D, Sharma U, Khatri SK. An Experimental Comparison of the Link Prediction Techniques in Social Networks. International Journal of Modeling and Optimization. 2014;4(1):21-24.

[12] Kushwah AKS, Manjhvar AK. A review on link prediction in social network. International Journal of Grid and Distributed Computing. 2016;9(2):43-50.

[13] Nandi G, Das A. An Efficient Link Prediction Technique in Social Networks based on Node Neighborhoods. International Journal of Advanced Computer Science And Applications. 2018;9(6):257-266.

[14] Liang Y, Huang L, Wang Z. Link prediction in social network based on local information and attributes of nodes. Journal of Physics: Conference Series. 2017;887(012043):10-1088.

[15] Dong L, Li Y, Yin H, Le H, Rui M. The Algorithm of Link Prediction on Social Network. Mathematical Problems in Engineering, vol. 2013. 2013;2013.

[16] Sett N, Basu S, Nandi S, Singh SR. Temporal link prediction in multi-relational network. World Wide Web. 2018;21:395--419.

[17] Rawashdeh A. Semantic Similarity of Node Profiles in Social Networks. Electronic Thesis and Dissertations Center. 2015 119.

[18] Rawashdeh A, Rawashdeh M, D´ıaz I, Ralescu. Measures of semantic similarity of nodes in a social network. In: Laurent , Olivier S, Bernadette BM, Ronald RY, editors. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems; 2014. p. 76-85. Available from: http://www.lirmm.fr/~lafourcade/pub/IPMU2014/papers/0443/04430076.pdf.

[19] Yang Y, Lichtenwalter RN, Chawla NV. Evaluating link prediction methods. Knowledge and Information Systems. 2015;45(3):751-782.

[20] Rossi RA, Ahmed NK. The Network Data Repository with Interactive Graph Analytics and Visualization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence; 2015. Available from: http://networkrepository.com.

[21] Leskovec J, Mcauley J. Learning to Discover Social Circles in Ego Networks (Advances in Neural Information Processing Systems 25). 2012.