An Efficient Link Prediction Technique in Social Networks based on Node Neighborhoods

Gypsy Nandi
Assam Don Bosco University
Guwahati, Assam, India

Anjan Das
St. Anthony's College
Shillong, Meghalaya, India

***Abstract*—The unparalleled accomplishment of social networking sites, such as *Facebook*, *LinkedIn* and *Twitter* has modernized and transformed the way people communicate to each other. Nowadays, a huge amount of information is being shared by online users through these social networking sites. Various online friendship sites such as *Facebook* and *Orkut*, allow online friends to share their thoughts or opinions, comment on others' timeline or photos, and most importantly, meet new online friends who were known to them before. However, the question remains as to how to quickly propagate one's online network by including more and more new friends. For this, one of the easy methods used is list of *'Suggested Friends'* provided by these online social networking sites. For suggestion of friends, prediction of links for each online user is needed to be made based on studying the structural properties of the network. Link prediction is one of the key research directions in social network analysis which has attracted much attention in recent years. This paper discusses about a novel efficient link prediction technique *LinkGyp* and many other commonly used existing prediction techniques for suggestion of friends to online users of a social network and also carries out experimental evaluations to make a comparative analysis among each technique. Our results on three real social network datasets show that the novel *LinkGyp* link prediction technique yields more accurate results than several existing link prediction techniques.**

Keywords*—Link prediction; online social networks; common neighbors; Jaccard's coefficient; Adamic/Adar; preferential attachment; FriendLink

I. Introduction

Online Social Networks (OSNs) have become a means for millions of online users to express and share their opinions with other users. These OSNs provide an excellent opportunity for allowing interactions and exchange of thoughts, opinions and ideas among the online users in a group or community. Such networks can be represented as graphs, where a node or a vertex corresponds to a user present in the graph and an edge corresponds to any form of association between the nodes or users, such as friendship ties. Also, these OSNs are dynamic and this raises a question as to: how does the graph structure of these networks change over time? Based on this question, this paper studies about the likeliness of any two nodes of a graph to be associated in the near future, considering that presently there is no connection in the current snapshot of the OSN graph being studied. This problem, commonly called the link prediction problem, is a research area being studied by many researchers in this field to generate faster and more appropriate result with special consideration to scalability and dynamic nature of the graph. Fig. 1 gives a basic idea about how link prediction is done by studying the structural links of a network. In this figure, five nodes have been considered at time t and future predictions are being made at time t+1. By studying the existing links, two predictions are made which are marked as dashed lines in the figure.

Liben-Nowell and Kleinberg [1] were the first to study the link prediction problem and propose a prediction model for the same. Their model mainly studies the linkage structure of a social network and discusses several link prediction methods for inferring new links. In [2], Hasan et al. studied several classification models for possible link prediction in co-authorship domain that aimed to provide a comparison of several features using different feature analysis algorithms. Zheleva et al. [3] studied binary classification algorithm that mainly studies friend or family networks for link prediction. They have mainly worked on the predictive power of overlaying friendship and family ties on three real world social networks. In [4], Tylenda et al. studied time-aware and time-agnostic maximum entropy methods in which time-based weighting of edges were used. Chen et al. [5] made a detailed study and comparison of four algorithms related to link prediction, namely, Friend of a Friend (FOAF), SONAR, Content-plus-Link (CplusL) and content matching algorithms. Schifanella et al. [6] considered a sampling link prediction algorithm that can help users find friends with similar topical interests as well as facilitate the formation of topical communities. They also introduced a null model to show that a part of the similarity between online users is due to the correlations between user activity and user degree centrality in the OSN. In [7], Papadimi-triou et al. found the FriendLink Algorithm for fast and accurate link prediction in OSNs which outperforms many other related algorithms in terms of accuracy and time-complexity. Bayesian network has been also considered as a consistent model to understand the relations between future links to be predicted in networks [19], [20]. Recently, negative link prediction in social networks has attracted the attention of many researchers and considerable research work is being carried out to find efficient techniques for the same [15], [16], [18]. Such techniques aim to perform link prediction across multiple signed networks. Recent work has also focused on noise-filtering technique to predict links in complex networks [17].

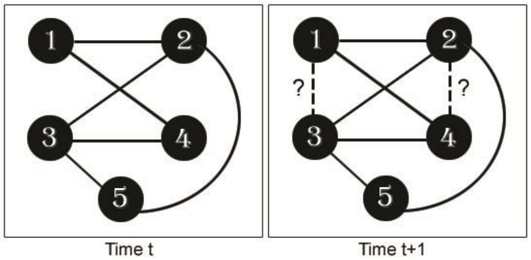

Fig. 1. Example of prediction of links in a given network.

The rest of the paper is organized as follows. In Section 2, a discussion on five standard link prediction techniques, namely, the *Common Neighbors*, *Jaccard's coefficient, Adamic/Adar, Preferential Attachment*, and *FriendLink* are given. Section 3 discusses about a novel link prediction technique *LinkGyp* which aims to provide better result than the above mentioned existing link prediction techniques. Section 4 illustrates the experimental results by comparing the various predictions of links made by the existing techniques with the novel technique. A conclusion of the paper and discussion on the scope for future work is given in Section 5.

II. Discussion on Standard Link Prediction Techniques

In this section, we at first discuss five standard link prediction techniques used in social networks and give a comparative analysis of the same. All these five techniques use the local-based features of a graph. There are, however, many global-based approaches also, which utilize the entire path structure in a network, but such approaches are computationally expensive for even a decent-sized social networks.

As seen in Fig. 1, vertices 1-5 indicates that there are five nodes or users in the network and the edges represent the existing links between each two nodes. In all the techniques explained below, an OSN is considered which is modeled as an directed or undirected graph *G=(V, E)*, where *V* denotes a set of vertices and *E* denotes a set of edges between two vertices in the network. Given below are some simple local similarity approaches based on node neighborhoods.

A. Common Neighbors

The technique of finding common neighbors for link prediction is considered as the most basic and significant method for prediction of links among nodes in the network. This approach was initially applied in the context of collaboration networks by Newman [8]. The basic idea of this technique is to find out the number of common nodes or neighbors or friends between two non-neighbors or non-friends. Now, the higher the number of common neighbors, the more likely is the chances of those two non-neighbors of being linked in the near future.

Using this concept, a link prediction score can be calculated between any two nodes p and q, where p and q are non-neighbors or non-friends at a given time t. The probability that these two nodes p and q will be linked in the near future is based on the score value given below:

$$score(p,q) = Neighbors(p) \cap Neighbors(q)$$

Considering Fig. 1, studying the network at time *t*, a prediction of future links (at time *t+1*) can be made between nodes 1 and 3 as well as 2 and 4. This is so because the number of common neighbors or the score value in both the cases is 2 (which are higher compared to the rest of the non-neighbors in the network). The technique of common neighbors is very simple and easy to analyze; yet this technique is very effective and it has been experimentally evaluated that it often outperforms several other complicated techniques used for link prediction. Algorithm 1 explains the link prediction technique based on the concept of common neighbors scores. In this algorithm, the social network *G* and the top *'n'* link predictions are taken as input and the top *'n'* nodes based on the score value is displayed as output. The score is calculated for each non-neighbors of a node and top-*n* predictions for the node are made based on the descending order of score values.

```
Algorithm 1: Common-Neighbors(G, n)

1: for each vertex v do
  1.1: $N_2 \notin$ Neighbors(v)
  1.2: for each vertex j $\in N_2$
  1.2.1: score = |Neighbors(v) $\cap$ Neighbors(j)|
  1.2.2: Store value of j and corresponding
         value of score
  1.3: end for
  1.4: Sort values of j in descending order of
       score
2: end for
3: Display top n values of j
```

B. Jaccard's Coefficient

Jaccard's coefficient [9] is another simple technique of link prediction which is similar to common neighbors' technique discussed above as this technique also relies on the number of common neighbors between two nodes. In case of *Jaccard's coefficient*, the probability that two nodes *p* and *q* will be linked in the near future is based on the score value given below:

$$score(p,q) = \frac{|\,Neighbors(p) \cap Neighbors(q)\,|}{|\,Neighbors(p) \cup Neighbors(q)\,|}$$

As can been seen from the score calculation mentioned above, in case of *Jaccard's coefficient*, the number of common neighbors is simply divided by the number of total neighbors. For instance, in Fig. 1, the score of nodes 1 and 3 at time t, using *Jaccard's coefficient* of link estimation is 0.67. Similarly, score of nodes 1 and 5 at time t is 0.33. Hence, there is a higher chance of nodes 1 and 3 being linked in the near future compared to nodes 1 and 5. Hence, the score value between two nodes will always remain between 0 and 1; 0 when there is no single common neighbor between two nodes and 1 when the two nodes being compared is the same node. Algorithm 2 explains the link prediction technique based on the concept of *Jaccard Coefficient* scores. In this algorithm, the social network G and the top *'n'* link predictions are taken as input and the top *'n'* nodes base on the score value is displayed as output.

```
Algorithm 2: Jaccard-Coefficient(G, n)

1: for each vertex v do
   1.1: $N_2 \notin Neighbors(v)$
   1.2: for each vertex $j \in N_2$
    1.2.1: $score = |Neighbors(v) \cap Neighbors(j)| / |Neighbors(v) \cup Neighbors(j)|$
    1.2.2: Store value of j and corresponding value of score
        1.3: end for
        1.4: Sort values of j in descending order of score
2: end for
3: Display top n values of j
```

C. Adamic/Adar

Adamic and Adar [10] have found another approach to predict links between two nodes in a network. In *Adamic/Adar* technique, all the common neighbors of two non-friends or non-neighbors are taken into consideration, and how many connections each of these common neighbors have are also considered. Thus, the probability that two nodes p and q will be linked in the near future is based on the score value given below:

$$score(p,q) = \sum_{x \in N(p) \cap N(q)} \frac{1}{\log |Neighbors(x)|}$$

Here x represents the set of common neighbors of nodes *p* and *q* and in the score calculation the number of neighbors of x is taken into consideration. In such a case, if a neighbor of *p* and *q* has two links or friends, a weight of 1/log(2) = 1.4 is considered. And again, if a neighbor of *p* and *q* has five links, a weight of 1/log(5) = 0.62 is considered. Hence, the more links a neighbor has, a better score value is obtained. Algorithm 3 explains the link prediction technique based on the concept of *Adamic/Adar* scores. In this algorithm, the social network G and the top *'n'* link predictions are taken as input and the top *'n'* nodes based on the score values is displayed as output.

```
Algorithm 3: Adamic-Adar(G, n)

 1: for each vertex v do
   1.1: $N_2 \notin Neighbors(v)$
   1.2: for each vertex $j \in N_2$
     1.2.1: Initialize score to 0
     1.2.2: for each $k \in (Neighbors(v) \cap Neighbors(j))$
        1.3.2.1: $score = score + 1 / (\log| Neighbors(k))$
     1.2.3: end for
     1.2.4: Store value of k and corresponding value of score
    1.3: end for
   1.4: Sort values of k in descending order of score
  2:   end for
  3:  Display top n values of k
```

D. Preferential Attachment

The technique of *Preferential Attachment* for predicting links in a network is based on the concept that two non-neighbors or non-friends have higher chances of being connected by a link in the future if the product of their number of individual neighbors is high. This results in the calculation of score value as given below:

$$score(p,q) = Neighbors(p) \, . \, Neighbors(q)$$

The term *preferential attachment* refers to the observation that in networks that grow over time, the likelihood that an edge is added to a node with n neighbors is proportional to n [11]. Experiments conducted by researchers have revealed that co-authorship is correlated with the product of the neighborhood sizes and the similar concept is applied for link prediction in social networks. Algorithm 4 explains the link prediction technique based on the concept of *Preferential Attachment* scores. In this algorithm, the social network *G* and the top *'n'* link predictions are taken as input and the top *'n'* nodes based on the score values is displayed as output.

```
Algorithm 4: Preferential-Attachment(G, n)

1:  for each vertex v do
   1.1: $N_2 \notin Neighbors(v)$
   1.2: for each vertex $j \in N_2$
      1.2.1: $score = |Neighbors(v) * Neighbors(j)|$
      1.2.2: Store value of j and corresponding value of score
    1.3: end for
    1.4: Sort values of j in descending order of score
 2: end for
 3: Display top n values of j
```

E. FriendLink

The *FriendLink* [7] approach of link prediction studies user's neighborhood by making use of paths of greater length. Here, two users connected with many unique pathways have a higher likelihood to know each other. Algorithm 5 explains the *FriendLink* approach for link prediction which takes as input the social graph *G*, the adjacency matrix *A* of graph *G*, number of nodes *'n'* present in the graph, and the maximum length of paths *'l'* explored in *G*. The algorithm provides as output the similarity matrix between two nodes in *G*. Based on the weights of similarity matrix friends can be recommended for a target node.

In the main program of Algorithm 5, the adjacency matrix of the graph is modified so that instead of holding the traditional values 0 or 1, the matrix is filled up with values 0 or vj, where vj is a node to which node vi is connected. In the function *ComputePaths*(), matrix multiplication of this modified adjacency matrix is performed with itself to produce all paths from node v_i to node v_j. Lastly, in the function *ComputeSimilarity*(), the similarity value between two nodes is measured to estimate the strength of connections between two non-linked nodes.

```
Algorithm 5: FriendLink(G, A, n, l) [7]

Main Program
1: for $v_i$ = 1 to n do
   1.1:  for $v_j$ = 1 to n do
     1.1.1: if A($v_i$,$v_j$) = 1 then
              A($v_i$,$v_j$) = $v_j$
           else A($v_i$,$v_j$) = 0
     1.1.2: end if
    1.2: end for
2: end for
3: for i = 2 to l
   3.1:  CombinePaths()
   3.2:  ComputeSimilarity(i)
4: end for
End Main Program

Function CombinePaths()
5: for $v_i$ = 1 to n do
   5.1:  for $v_j$ = 1 to n do
       5.1.1:  for k = 1 to n do
         5.1.1.1: if A($v_i$,k) <> 0 and A(k,$v_j$)<>
                  0 then
                  A($v_i$,$v_j$)=concatenate(A($v_i$,k),
                  A(k,$v_j$))
         5.1.1.2: end if
       5.1.2: end for
   5.2: end for
6: end for
7: return A($v_i$,$v_j$)
End Function

Function ComputeSimilarity()
8: for $v_i$ = 1 to n do
   8.1:  for $v_j$ = 1 to n do
     8.1.1:  denominator = 1
     8.1.2:  for k = 2 to i do
        8.1.2.1: denominator=denominator*(n-k)
     8.1.3:  end for
     8.1.4:  sim($v_i$,$v_j$) = sim($v_i$,$v_j$) + $\frac{1}{i-1}$ *
             $\frac{|paths^{i}_{vi..vj}|}{denominator}$
   8.2: end for
 9:  end for
 10: return sim($v_i$,$v_j$)
 End Function
```

There are also several other methods of link prediction which are based on the ensemble of all paths [22] such as *Katz* [13], *Hitting Time* and *SimRank* [14]. There are also other higher level approaches for link prediction such as clustering and low-rank approximation, which can be combined with the above mentioned link prediction techniques to give a more accurate output. The authors of [23] have used *Maximal Entropy Random Walk (MERW)* for link prediction, which emphasizes the centrality of nodes of the network. Other link prediction techniques consider temporal information to accurate predicts among non-edged nodes. Several other techniques focus different other issues such as giving weightage to more influential nodes, considering a subgraph based on the closed knit group in the graph, and so on. However, the primary focus on link prediction circles around which technique can give better accurate results along with better efficiency.

III. Novel *LinkGyp* Link Prediction Technique

In this section we first give a brief outline of our novel approach, named *LinkGyp* and then analyze the steps of the proposed algorithm.

A. Outline of the LinkGyp Technique

The *LinkGyp* prediction technique is a new approach proposed for prediction of links keeping in mind the scalability issue needed to be taken care of for huge-sized social networks. The basic idea of this technique is to initially take into consideration only those non-neighbors of a node whose product of their individual neighbors are among the top in descending order of list. A list is generated that includes the highly potential 'could be friends but currently non-friends' of a node and their corresponding scores. Using this list, a smaller sized graph for the node is now considered that is dependent on the number of top recommendations to be made. This results in a truncated graph where not all non-neighbors of a node are to be considered for a node. In fact, for a large-sized graph that involves huge number of non-friends for a node, the ultimate consideration of number of potential non-friends gets limitized to a great extent.

Once the smaller sized-graph is selected, the selection of top-n nodes results in a much faster execution by considering the *Adamic*/*Adar* approach where the simple counting of common features is refined by weighting rarer features more heavily [7]. As explained before, the *Adamic*/*Adar* method computes the similarity between two nodes p and q by means of a common feature of the two, say x. The similarity measure is then $\sum_x 1/|\log(\text{frequency}(x)|$ where, frequency(x) refers to frequency of occurrence of the common features between nodes *p* and *q*. The result obtained is the top-*n* prediction of links for each node of the graph.

B. The LinkGyp Algorithm

Algorithm 6 explains a novel link prediction technique *LinkGyp* that aims to provide better results than the above mentioned local similarity approaches of link prediction. In this algorithm, the social network graph *G=(V,E)* and the value of 'n' are taken as input. Here, *'n'* represents the number of link predictions to be made for each node. Steps 1.1 to 1.4 concentrate on calculating scores for two non-edged nodes based on the product of the size of their individual neighbors.

Based on the descending order of their scores, the top *2n* non-edged nodes are considered for a node *v*. The reason behind choosing *2n* as the threshold value for selection of the subgraph is that more than *2n* lead to a bigger sized subgraph and less than *2n* may lead to consideration of very less nodes. Hence the choice of *2n* is considered due to performance considerations and it represents a performance-quality-tradeoff. For a reasonably-sized *'n'*, experiments have been conducted with different multiples of *'n'* and it has been found that *2n* is the optimum consideration for choosing the top *2n* non-edged nodes. However, if 'n' is too small, then the subgraph will also contain limited information and may lead to lower quality results. Again, if *'n*' is too large, it may lead to very small or no improvement in the result. Hence giving a right input value of n (5 ≤ *n* ≤ 50) will lead to better and more accurate results. The rest of the other non-edged nodes are discarded and further steps are carried out only for the top *2n* resulting nodes. These few steps are carried out keeping in mind the ground truth in economics that "the rich get richer". Also, it results in a smaller choice of nodes with least computational complexity.

Next, a similar approach to *Adamic*/*Adar* explained above is followed to find the *'n'* best predicted non-edged nodes for vertex 'v' from the set of only *2n* number of nodes selected in steps 1.1 to 1.4. The reason behind choosing the *Adamic*/*Adar* approach is that this technique considers the case that an affair owned by less objects, compared to owned by more objects, has greater effect on link prediction. In this way, the scores are calculated for each of the two non-edged nodes and ultimately the output for top-*n* predicted nodes is displayed for each unique user v based on the descending order of score.

The idea behind using this algorithm is mainly the scalability issue while dealing with dense social networks. As mentioned before, from steps 1.6 to 1.10, the estimation of links is done for a very small subgraph consisting of only *2n* nodes, where *'n'* is the number of prediction of links to be made. Prior to step 1.5, the calculation of score1 is simple and does not involve studying in-depth the entire social network. It is only basically studying how many neighbors two non-edged nodes have. Hence, this algorithm proves to be an efficient method of prediction of links in social networks.

C. *Complexity Analysis of the LinkGyp Algorithm*

Online social networks are usually largely populated with information. Link prediction algorithms based on global based features, such as *Katz index* or Random Walk with Restart, are computationally too expensive for large graphs as it involves the inversion of matrix for link prediction. However, the standard existing link prediction algorithms discussed above are based on local based features, and comparatively have less time complexity than global based feature algorithms.

If we specifically consider the time complexity of our proposed *LinkGyp* technique, it is mainly O(2n), where *'n'* is the number of link predictions to be made per node. This is much effective in terms of complexity analysis as the value of *'n'* will be significantly much smaller compared to the total number of nodes 'g' for the entire graph. However, most of the other discussed link prediction techniques (such as, *Jaccard Coefficient*, *Adamic/Adar*, etc.) consider the entire nodes of the graph for prediction of links for a particular node that in real-time would be in terms of thousands, lakhs or even more. Hence, the complexity of *Jaccard coefficient* and *Adamic/Adar* techniques is O(g), where the value of *'g'* is significantly greater than *'n'*. For *Friendlink* algorithm, the time complexity is O(g x al), where *'a'* is the average nodes degree in a graph and *'l'* refers to the path lengths. Thus, the basic idea of *LinkGyp* algorithm is that the estimation of links is done for a very small subgraph consisting of only *2n* nodes, which gives better results as far as complexity of time is to be considered. The next section discusses the experimental results which prove that the above discussed novel link prediction algorithm gives a considerably better output compared to several basic existing link prediction techniques.

```
Algorithm 6: LinkGyp(G, n)

1: for each vertex v do
    1.1: cnt1 = |Neighbors(v)|
    1.2: for each j ∉ Neighbors(v) do
      1.2.1: cnt2 = |Neighbors(j)|
      1.2.2: score1 = cnt1*cnt2
      1.2.3: Store value of j and corresponding
             value of score1
    1.3: end for
    1.4: Sort values of j in descending order
         of score1 in arr1
    1.5: for i = 1 to 2n do
      1.5.1: Initialize score2 to 0
      1.5.2: Initialize cnt3 to 0
      1.5.3: for each k ∈ arr1 do
        1.5.3.1:  Initialize score2 to 0
        1.5.3.2: for each z ∈ (Neighbors(v) ∩
                 Neighbors(k))
          1.5.2.2.1:  cnt3=cnt3+|Neighbors(z)|
        1.5.3.3:  end for
      1.5.4: score2 = score2 + (1/log(cnt3))
      1.5.5: Store value of z and corresponding
             value of score2
      1.5.6: end for
    1.6: Sort values of z in descending order
         of score2
    1.7: end for
    1.8: Display top n values of z
2:  end for
```

IV. Experimental Evaluation

For conducting the experiments, three publicly available real-world datasets have been used that contains friendship network between users of social networking websites, namely the *facebook* dataset [24], the *hamsterster* dataset [12] and the *brightkite* location-based social networking website [21]. Table I gives few statistical information of all the three datasets.

TABLE I. Statistics of the Various Datasets

Dataset	*facebook*	*hamsterster*	*brightkite*
#Nodes	63731	1858	55228
#Edges	817035	12534	214078
Average Degree	25.640	13.492	7.353
Maximal Degree	1098	272	272
Average Path Length	2.832	3.453	2.76

a) The *facebook* dataset is an undirected network containing 63,731 nodes and 817035 edges that describes friendship data of *facebook* users. A node represents a user and an edge represents a friendship between two users. Fig. 2 illustrates the graphical view of the *facebook* dataset (nodes having degree less than six have not been considered) As can be seen from the figure, the yellow colored, bigger-sized nodes have highest degrees, the blue colored moderate-sized nodes have average degrees and red colored small-sized nodes have fewer degrees.

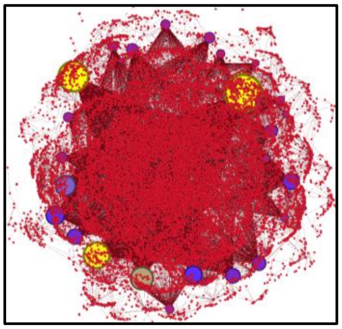

Fig. 2. The *facebook* dataset represented as a graph having different sized and colored nodes based on degree.

b) The *hamsterster* friendship dataset contains 1858 distinct nodes and 12534 edges which indicates the ties or friendship among all users in the network. The entire dataset has been represented in a graph as shown in Fig. 3 in which the yellow colored, bigger-sized nodes have highest degrees, the blue colored moderate-sized nodes have average degrees and red colored small-sized nodes have fewer degrees.

c) The *brightkite* friendship dataset contains undirected user-user friendship relations that have been gathered from a former widely used location-based social network. This dataset contains 55,228 distinct nodes and 214,078 edges that indicate friendship ties between two users. Fig. 4 illustrates the graphical form of a slice of the *brightkite* dataset (nodes having degree of one have not been considered) in which, the yellow colored, bigger-sized nodes have highest degrees, the blue colored moderate-sized nodes have average degrees and red colored small-sized nodes have fewer degrees.

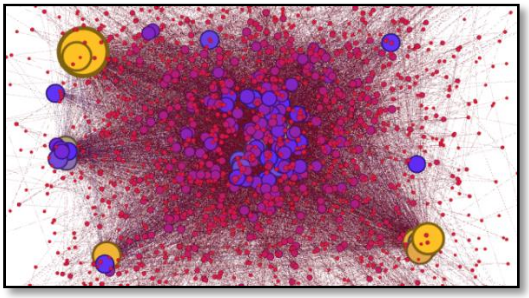

Fig. 3. The *hamsterster* dataset represented as a graph having different sized and colored nodes based on degree.

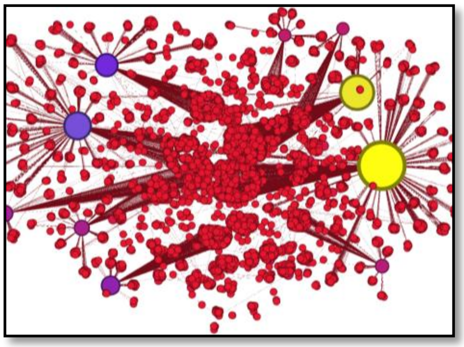

Fig. 4. The *brightkite* dataset represented as a graph having different sized and colored nodes based on degree.

Several experiments were conducted for the link prediction techniques mentioned above on all the three datasets. These experiments mainly aim at illustrating the performance comparison of the above mentioned link prediction techniques when compared to the random method generation of links for predicting future associations among nodes.

To conduct all the experiments, each of the entire dataset was divided into training and testing datasets consisting of 60% and 40% records respectively. Care was taken to include at least all core nodes in the training data set ('core' is the set containing nodes they have a direct link to minimum 10 other nodes). Tables II to VII illustrate the number of common predictions made between each two techniques for the two datasets which basically demonstrates which techniques are similar to each other in generation of link prediction results.

From the results obtained from Tables II to VII, it can be concluded that the random generation technique yields the least common predictions compared to the other six link prediction techniques. It can also be considered from all these tables that *Common Neighbors* and *Jaccard's Coefficient* predicts more similar friend suggestions for future links compared to the rest of the link prediction techniques.

TABLE II. THE NUMBER OF COMMON PREDICTIONS MADE ON THE FACEBOOK DATASET OUT OF 10000 (1000 USERS X 10) PREDICTIONS

	Common Neighbors	**Jaccard's Coefficient**	**Adamic/ Adar**	**Preferential Attachment**	**FriendLink**	**LinkGyp**	**Random Generation**
Common Neighbors	10000	8275	2504	2486	8175	3742	167
Jaccard's Coefficient		10000	1859	2144	6879	3301	173
Adamic/ Adar			10000	2570	4269	4957	204
Preferential Attachment				10000	3578	8146	196
FriendLink					10000	7686	185
LinkGyp						10000	168
Random Generation							10000

TABLE III. THE NUMBER OF COMMON PREDICTIONS MADE ON THE FACEBOOK DATASET OUT OF 20000 (1000 USERS X 20) PREDICTIONS

	Common Neighbors	**Jaccard's Coefficient**	**Adamic/ Adar**	**Preferential Attachment**	**FriendLink**	**LinkGyp**	**Random Generation**
Common Neighbors	20000	17751	6687	4591	15239	6157	427
Jaccard's Coefficient		20000	5826	4044	13486	6312	394
Adamic/ Adar			20000	3746	10017	7916	267
Preferential Attachment				20000	9810	16984	205
FriendLink					20000	14012	196
LinkGyp						20000	278
Random Generation							20000

TABLE IV. THE NUMBER OF COMMON PREDICTIONS MADE ON THE *HAMSTERSTER* DATASET OUT OF 18580 (1858 USERS X 10) PREDICTIONS

	Common Neighbors	Jaccard's Coefficient	Adamic/ Adar	Preferential Attachment	FriendLink	LinkGyp	Random Generation
Common Neighbors	18580	14833	5758	4218	16891	2983	208
Jaccard's Coefficient		18580	3384	2991	13567	1260	85
Adamic/ Adar			18580	3710	13001	3258	131
Preferential Attachment				18580	11763	13728	169
FriendLink					18580	12023	148
LinkGyp						18580	133
Random Generation							18580

TABLE V. THE NUMBER OF COMMON PREDICTIONS MADE ON THE *HAMSTERSTER* DATASET OUT OF 37160 (1858 USERS X 20) PREDICTIONS

	Common Neighbors	Jaccard's Coefficient	Adamic/ Adar	Preferential Attachment	FriendLink	LinkGyp	Random Generation
Common Neighbors	31760	27940	13572	5589	27438	6413	444
Jaccard's Coefficient		31760	9154	3093	24890	3018	360
Adamic/ Adar			31760	5122	22769	7200	285
Preferential Attachment				31760	20026	26728	294
FriendLink					31760	24374	251
LinkGyp						31760	263
Random Generation							31760

TABLE VI. THE NUMBER OF COMMON PREDICTIONS MADE ON THE *BRIGHTKITE* DATASET OUT OF 20000 (10000 X 10) PREDICTIONS

	Common Neighbors	Jaccard's Coefficient	Adamic/ Adar	Preferential Attachment	FriendLink	LinkGyp	Random Generation
Common Neighbors	20000	13076	9394	3186	10782	4280	176
Jaccard's Coefficient		20000	6702	1090	8201	2132	154
Adamic/ Adar			20000	2080	7658	2824	168
Preferential Attachment				20000	7105	162s62	186
FriendLink					20000	15396	154
LinkGyp						20000	202
Random Generation							20000

TABLE VII. THE NUMBER OF COMMON PREDICTIONS MADE ON THE BRIGHTKITE DATASET OUT OF 40000 (10000 X 20) PREDICTIONS

	Common Neighbors	Jaccard's Coefficient	Adamic/ Adar	Preferential Attachment	FriendLink	LinkGyp	Random Generation
Common Neighbors	40000	31750	19206	6236	28106	8872	184
Jaccard's Coefficient		40000	17544	3844	25871	7608	256
Adamic/ Adar			40000	5818	24712	12512	280
Preferential Attachment				40000	23108	26792	358
FriendLink					40000	27115	298
LinkGyp						40000	407
Random Generation							40000

Experiments were also conducted to find the number of correct predictions made on the testing datasets so as to find which techniques yield better results. A result of this is depicted in the Fig. 5-7 which again compare the above mentioned link prediction techniques against the novel *LinkGyp* link prediction technique for all the three datasets. Considerations were made for values of *'n'* as 10 and 20, where *'n'* is the number of predictions to be made for a particular node. Link predictions, in turn, were made for 1000 random distinct nodes present in the *facebook* dataset, and for each of the 1858 distinct nodes present in the *hamsterster* dataset, as well as for 2000 distinct nodes present in the *brightkite* dataset. Experiments reveal that the novel *LinkGyp* technique yields more accurate results followed by the *FiendLink* (considering lengths of path 2), Preferential Attachment and Adamic/Adar techniques. However, the random generation technique of link prediction, which randomly chooses the 'n' non-friends of a node, fails to come at par with all the other five prediction techniques.

The *hamsterster* dataset consists of densely-edged connections compared to the other dataset taken into consideration, namely the *facebook* dataset and the *brightkite* dataset which consists of comparatively sparsely-edged connections. Hence, it can be concluded that the novel *LinkGyp* link prediction technique can be considered as an efficient technique for link prediction keeping in mind the performance, scalability and execution time while dealing with social networks that comprise of thousands, lakhs or even more unique users and this technique is suitable for both densely-edged and sparsely-edged connections.

In summary, the results displayed in Fig. 5-7 indicate that results might slightly differ based on the scalability and sparseness of the dataset we are working upon. However, our novel *LinkGyp* technique outperforms other mentioned link prediction techniques in terms of accuracy.

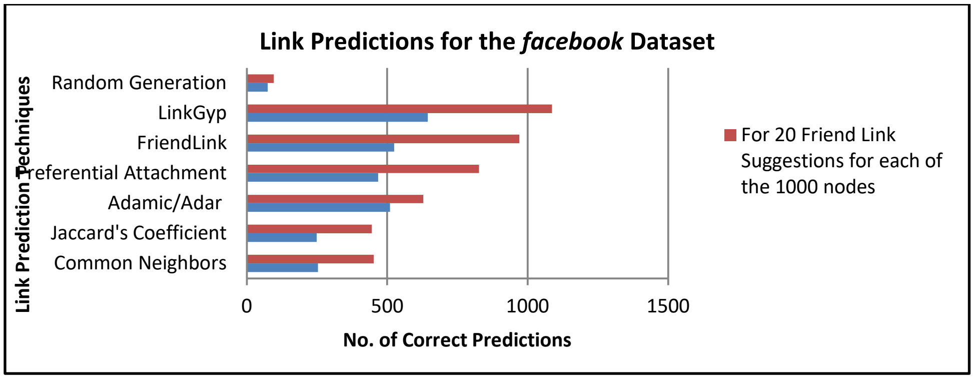

Fig. 5. Number of correct link predictions made on the *facebook* dataset.

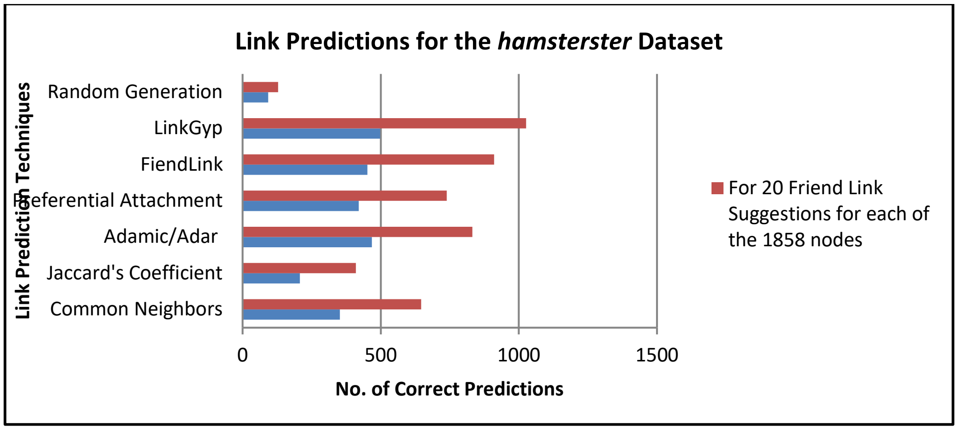

Fig. 6. Number of correct link predictions made on the *hamsterster* dataset.

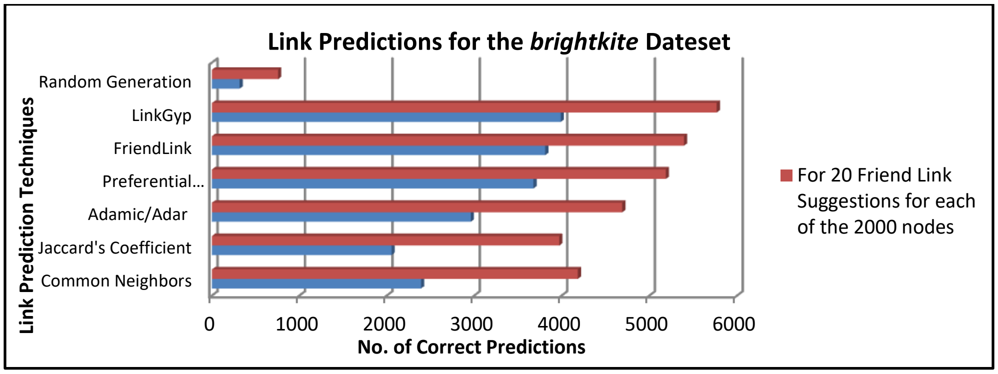

Fig. 7. Number of correct link predictions made on the brightkite dataset.

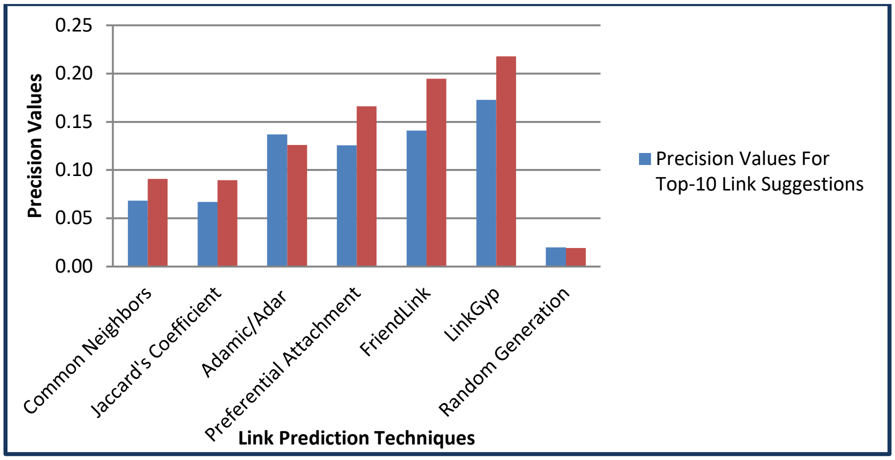

Fig. 8. Precision values of various link prediction techniques for the facebook dataset.

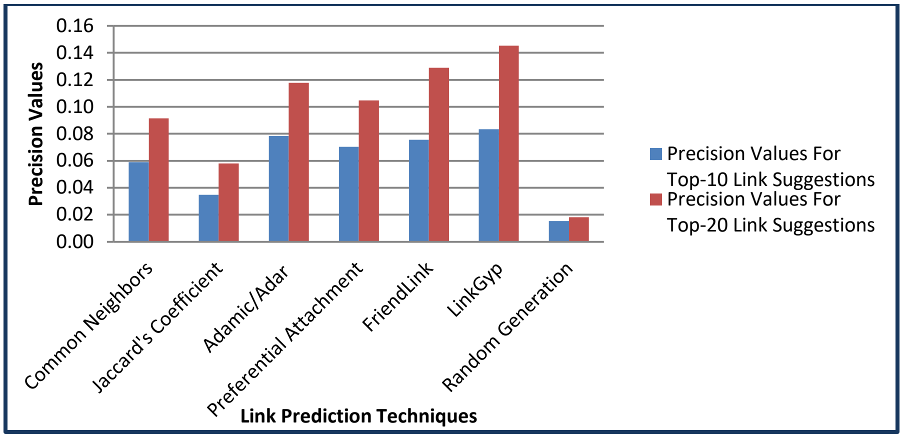

Fig. 9. Precision values of various link prediction techniques for the *hamsterster* dataset.

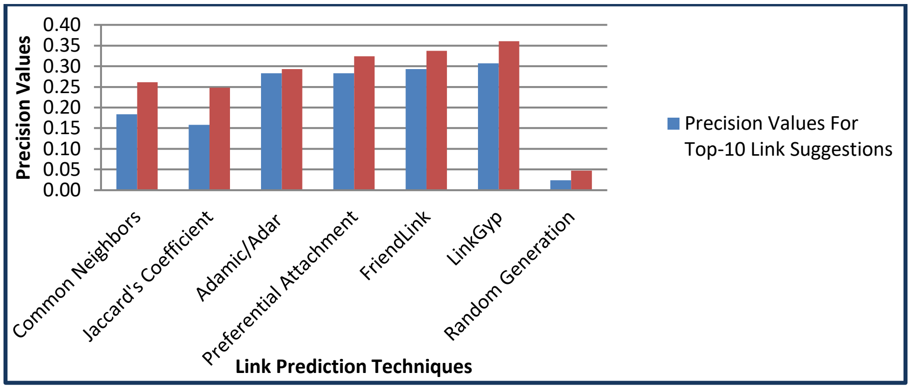

Fig. 10. Precision values of various link prediction techniques for the *brightkite* dataset.

Usually, for quantifying the accuracy of link prediction techniques, two standard metrics are commonly used: area under the receiver operating characteristic curve (AUC) and Precision [22]. Precision for a link prediction algorithm is calculated by considering the ratio of correct links selected to the total number of links selected. For example, if prediction of two new links has been made for a particular user, out of which one is correct and the other is incorrect, prediction value will be 0.5. This indicates that higher the precision value, higher will be the prediction accuracy. In this paper, we have used precision as the metric for evaluation of all the link prediction techniques and the results for the three different datasets are given in Fig. 8-10. The results of each of these figures below take into consideration the predictions made for 1000 random distinct nodes present in the *facebook* dataset, and for each of the 1858 distinct nodes present in the *hamsterster* dataset, as well as for 2000 distinct nodes present in the *brightkite* dataset.

V. Conclusions and Future Work

The tremendous growth in the use of online social networking sites has forced the researchers to carry out in-depth studies in social network mining. The link prediction technique in social networks is one such important research area that is in constant focus and is being studied and analyzed for better results. Our proposed work in this paper related to the proposed technique can be summarized as follows:

- This paper initially discusses the five basic standard techniques of link prediction and then gives a comparative analysis of these techniques using experimental results for the same. It can be concluded that these techniques will remain the simplest and basic techniques for studying and analyzing the concept of link prediction for OSNs and can assist a researcher in this field to get a preliminary idea about the same.

- The paper also discusses a new technique of link prediction namely '*LinkGyp'* that aims to provide a significantly better result in terms of more correct link predictions among non-linked nodes. We performed several extensive experiments on three different real-time datasets (Facebook, brightkite, and hamsterster) to arrive at a common result which proves that the *'LinkGyp'* technique can prove more efficient in prediction of links in social networks compared to several existing approaches.

- Considering link prediction to be one of the key research areas in social network mining, we have made an attempt to further improve the efficiency of link prediction with relate to number of correct predictions as well as run-time complexity.

- Finally, we can conclude that the proposed *'LinkGyp'* technique can be considered as the base model for link prediction technique to further carry out experiments on link predictions for complex networks.

As a future work, we plan to study other features of nodes along with their structural properties for generating better and more accurate results for link prediction in social networks. Also, further directions of study are needed to be carried out to improve the algorithm in order to deal with edges having negative weights (signed networks). The proposed algorithm can also be further enhanced to study the cold-start issue and link prediction for signed networks. If all these mentioned issues can also be considered while developing the link prediction techniques, it will provide new insight for modeling prediction of links in social networks.

References

[1] David Liben-Nowell , Jon Kleinberg, *The link prediction problem for social networks,* Proceedings of the twelfth international conference on Information and knowledge management, (2003), New Orleans, LA, USA [doi 10.1145/956863.956972]

[2] David Liben-Nowell , Jon Kleinberg, *The link prediction problem for social networks,* Proceedings of the twelfth international conference on Information and knowledge management, (2003), New Orleans, LA, USA [doi 10.1145/956863.956972]

[3] M. Hasan, V. Chaoji, S. Salem, and M. Zaki. *Link prediction using supervised learning*. In Proceedings of the Workshop on Link Discovery: Issues, Approaches and Applications, (2005)

[4] Elena Zheleva , Lise Getoor , Jennifer Golbeck , Ugur Kuter, *Using friendship ties and family circles for link prediction,* Proceedings of the Second international conference on Advances in social network mining and analysis, p.97-113, (2008), Las Vegas, NV, USA

[5] Jilin Chen , Werner Geyer , Casey Dugan , Michael Muller , Ido Guy, *Make new friends, but keep the old: recommending people on social networking sites,* Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (2009), Boston, MA, USA [doi>10.1145/1518701.1518735]

[6] Rossano Schifanella, Alain Barrat , Ciro Cattuto , Benjamin Markines , Filippo Menczer, *Folks in Folksonomies: social link prediction from shared metadata,* Proceedings of the third ACM international conference on Web search and data mining, (2010), New York, New York, USA [doi 10.1145/1718487.1718521]

[7] Papadimitriou, P. Symeonidis, and Y. Manolopoulos, *"Fast and accurate link prediction in social networking systems",* The Journal of Systems and Software 85, (2012), pp. 2119–2132

[8] M. E. J. Newman, *Clustering and preferential attachment in growing networks.* Physical Review Letters E, (2001)

[9] Gerard Salton and Michael J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill, (1983).

[10] Lada A. Adamic and Eytan Adar. *Friends and neighbors on the web, Social Networks,* 25(3):211, (2003).

[11] Jérôme Kunegis, Marcel Blattner, Christine Moser, *"Preferential attachment in online networks: measurement and explanations",* in Proceedings of the 5th Annual ACM Web Science Conference, pp. 205-214, (2013), Paris, France [doi>10.1145/2464464.2464514]

[12] Hamsterster friendships network dataset - KONECT, Available at http://konect.uni-koblenz.de/networks/petster-friendships-hamster, accessed June 2015.

[13] Leo Katz. *A new status index derived from sociometric analysis.* Psychometrika, 18(1), pp. 39-43, (1953).

[14] Glen Jeh and Jennifer Widom. *SimRank: A measure of structural-context similarity*. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July (2002).

[15] J. Tang, S. Chang, C. Aggarwal, and H. Liu, *"Negative link prediction in social media",* In WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining, pp. 87-96, ACM (2015).

[16] J. Tang, C. Yi, C. Aggarwal, and H. Liu, *"A Survey of Signed Network Mining in Social Media"*, arXiv preprint arXiv:1511.07569 (2015).

[17] B. Ouyang, L. Jiang and Z. Teng*, "A Noise-Filtering Method for Link Prediction in Complex Networks",* DOI: 10.1371/journal.pone.0146925 (2016).

[18] F. Liu, B. Liu, C. Sun, M. Liu and X. Wang, *"Deep Belief Network-Based Approaches for Link Prediction in Signed Social Networks",* Entropy 17.4, pp. 2140-2169; doi:10.3390/e17042140 (2015).

[19] S. H. Shalforoushan and M. Jalali, *"Link prediction in social networks using Bayesian networks",* in IEEE International Conference on Artificial Intelligence and Signal Processing, pp. 246-250 (2015).

[20] B. Zhang, S. Choudhury, M. A. Hasan, X. Ning, K. Agarwal, S. Purohit, and P. P. Cabrera *"Trust from the past: Bayesian Personalized Ranking based Link Prediction in Knowledge Graphs",* in SDM Workshop on Mining Networks and Graphs - MNG 2016, arXiv: 1601.03778, (2016).

[21] E. Cho, S. A. Myers, J. Leskovec. *Friendship and Mobility: Friendship and Mobility: User Movement in Location-Based Social Networks,* in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2011.

[22] L Lu and T Zhou, *Link prediction in complex networks: A survey*. Physica A: Statistical Mechanics and its Applications, 390: 1150–1170, 2011.

[23] R.-H. Li, J. X. Yu, and J. Liu, *"Link prediction: the power of maximal entropy random walk,"* in CIKM, 2011

[24] Facebook friendships network dataset - *http://konect.uni-koblenz.de/networks/facebook-wosn-links,* accessed June 2016