Analyzing and Comparing the Behavior of Real-life Web-like Network

Implemented by Nabila Shahnaz Khan

***Abstract*— In this era of Internet and World Wide Web, where everything is connected, study of large and random web-like network has become of great eminence. The authors Deo and Cami proposed a discrete-time dynamic random graph model where nodes are both added (along with edge) and deleted preferentially. They studied different properties such as growth in the number of nodes and edges with respect to steps, degree distribution of deleted nodes and overall degree distribution of the graph model and showed that this model follows the power-law degree distribution. Here, using the dynamic random graph model proposed by Deo et al, (1) I have analyzed real-life data collected from a survey done on the users of the well-known social networking site 'Facebook'. (2) Then I have compared the results with the results obtained from Deo et al. proposed model and showed how much the discovered properties of dynamic random graph model correspond with the real life example. (3) I have also improvised the proposed model by adding preferential addition and deletion of edges to the existing model and showed how the number of nodes and edges change with respect to the number of iterations. (4) I also proposed a different equation for the preferential deletion of nodes and showed how the results changes using that equation. (5) Finally, I tried to introduce a new property of the existing dynamic random graph model by separately analyzing the degree distribution of the old nodes and newly added nodes in the graph and showed how their cumulative probability distribution differ for higher and lower values of degrees.**

***Index Terms*—Web-like network, Dynamic random graph model, discrete-time, Preferential deletion, Scale-free, Real-life data**

I. Introduction

According to previous observational studies, Internet and Web-like networks are statistically similar to various large real-life networks such as transportation and communication, ecology, economic dependency, research connection, power distribution, brain cell signaling, food chain, etc. In fact any connected structure in real-life can be presented and analyzed as a web-like network. Understanding such huge and dynamic networks can be really complex. The increasing prominence and applications of such networks have engendered a research field dedicated to analyzing and understanding their properties along with the functionalities.

Web-like networks are every large, dynamic and they are considered to have no centralized control. They are seemed to follow scale-free power-law and have a greater degree of clustering [1]. A dynamic random graph model is a graph model which starts with a fixed number of nodes and at different points of discrete-times, nodes with edges are added and existing nodes are deleted randomly following some stochastic or preferential rules. Different research works [2]–[7] related to such graph models have already been done and still going on. Among them, Birth-only ones are the most common which use either preferential attachment [8] or copying [7]. In contrast, models including both birth and deletion have been studied much less. In 2000, Dorogovtsev and Mendes [5] proposed a model with an uniform deletion of edges while Chung et al. [9] and Cooper et al. [4] separately proposed a model that combines birth with uniform deletion of edges and nodes.

In the paper proposed by Deo and Cami [10], the authors proposed a dynamic random graph model which combines addition of nodes and edges with a preferential deletion of nodes where nodes having small-degree have higher probability of getting deleted. They analyzed properties of their proposed graph model and according to their yielded results, it was pretty evident that their proposed preferential deletion model follows the known asymptotically power-law degree distribution.

Here, I implemented their model on a real-life network (sub-network of Facebook users) and compared the outcome with the result of the actual model. From the results, it was seen that growth in number of nodes and edges follow a linear pattern with respect to number of iterations but it didn't exactly coincide with the predicted value as the ratio of edges doesn't match with the ratio proposed in the existing model. I also tried to extend the model by adding the concept of addition and deletion of edges independently. In the existing model, addition and deletion of edges are completely dependent on the addition and deletion of nodes. The newly proposed model showed some promising results as the growth of nodes and edges follow a certain linear pattern. Next, I tried to analyze the result of the existing model by using a different preferential equation for deletion and it worked slightly better than the one used here. Specially, when the number of iterations is not very high, the new equation gives better aligned result to the predicted line compared to the existing one and it also performs well in case of large number of iterations. Finally, I tried to introduce a new property for the existing model. I showed how the behavior of degree distribution varies among the previously added nodes and the newly added nodes in the graph.

The rest of the paper is organized as follows: Section II further defines the dynamic random graph model proposed in [10]. Section III analyzes a real-life network using the existing model and also compares the obtained results with the results of the existing model. In Section IV, the model has been extended by adding preferential addition and deletion of edges. After that, Section V and Section VI respectively discusses about modifying the equation for preferential deletion and the newly introduced property of the model. Finally, Section VII concludes the paper.

II. Random Graph Model with Preferential Deletion

In this model, initially the graph G contains one node and one-loop. At each time step t, either a new node is added along with edge (Birth Process) or a node is deleted (Death Process). Both of the processes are described below:

1) Birth Process:: If probability $\leq$ p (user-defined), a new node (u) is added with an incident edge. From the existing nodes, a node is selected which will be the other end-node (v) of that edge. This node v is selected based on Equation 1:

$$\frac{degree(EachNode)}{2 * TotalEdges} \tag{1}$$

Based on this formula, a weight is assigned to each existing node and then a node is selected randomly. As nodes with higher degree will be assigned higher weight, they are more likely to get selected.

2) Death Process:: When probability is q (q = 1 - p), a node u is chosen for deletion using the Equation 2 given below:

$$\frac{TotalNodes - degree((EachNode)}{TotalNodes - 2 * TotalEdges} \tag{2}$$

Here, the nodes with lower degree are more likely to get deleted as they are assigned lower weights. While deleting a node, all the edges incident to it are also deleted.

III. Analyzing a real life network using the existing model

A. Generating Graph for Real-life Data:

Here, First I tried to analyze a real life network by figuring out how the number of users and the relation between them diminish over number of iterations and see if there is any common pattern. To do so, I generated a graph using the data of a 'Facebook' sub-network [Code Used: **Graph_generate.py**]. The generated graphs have been stored in the **'Generated_graph'** folder.

B. Analyzing the Real-life Data:

Then I implied the existing random graph model on the generated graph in a reverse manner [Code Used: **Analyzing.Facebook.Data.py**]. Generally, in the existing model, a graph is generated from a initial state, containing two nodes and one edge, by giving a higher probability of node birth. But the real-life network presented here is a already built network consisting of large number of nodes (users) and edges (relations). So, I gave a lower probability for node addition (ex: 0.1, 0.25, 0.4), which eventually made the network smaller with increasing number of iterations. From the results shown in Figure 1 and 2, it was clear that the reduction in the number of nodes and edges with respect to the number of iterations followed a certain linear pattern. So, we can say that this model does maintain the scale-free power-law property of real-life network.

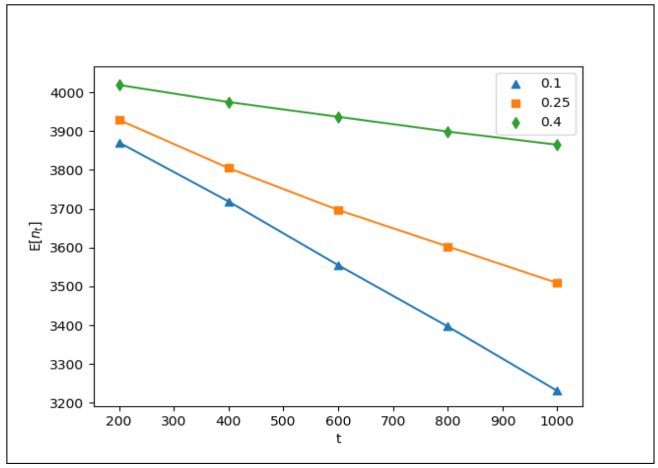

Fig. 1. Reduction in the number of nodes with respect to steps for three different probabilities (0.1, 0.25, 0.4) in the real-life network

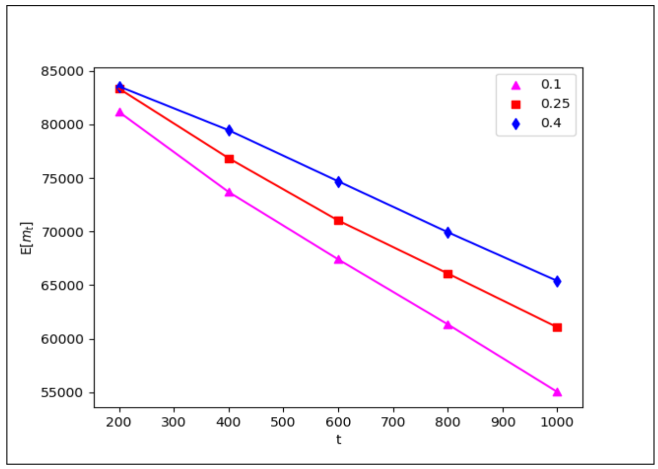

Fig. 2. Reduction in the number of edges with respect to steps for three different probabilities (0.1, 0.25, 0.4) in the real-life network

C. Dataset Used:

I collected the data of the real-life Facebook network from a well-acknowledged data source called 'Stanford Large Network Dataset Collection' [11] maintained by Stanford University. This particular data was collected from survey participants using Facebook app. The network contains 4039 nodes and 88234 edges. Here the nodes represent each individual user and edges represent interactions between the users.

D. Runtime:

When the model is implemented using a higher probability for addtion of node, the runtime is exponential and depends on the increasing number of nodes. But in the reverse mode, with higher probability for deletion, as the number of nodes don't increase and rather decreases, the runtime is linear and has a very flat slope, as we can see in Figure 3.

E. Comparison:

I implemented the model reversely on a previously generated graph [**graph_0.9_100000.pickle**, stored in **'Data'** folder]

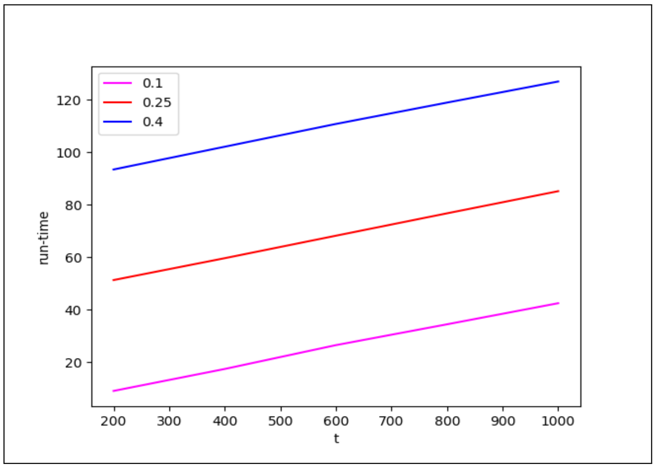

Fig. 3. Runtime comparison of the real-life network with respect to the number of iterations for probabilities (0.1, 0.25, 0.4)

which was generated using the same model for 100000 iterations [Code Used: **Comparison.With.Facebook.Data.py**]. This graph has 79994 nodes and 71418 edges which is comparable to the graph generated from real-life data. As we can see from Figure 4 and 5 (plotted for the graph previously generated by the model), the curves for showing the decrease in the number of nodes and edges with respect to runtime is almost linear. But, the decrease of the edges seems comparatively more linear for real-life network than the model-generated network. It might be because, the structure of edges in the real-life network is more practical where edges can be added and deleted independently. But overall, the result generated from real-life network and randomly generated dynamic network seems quite similar, both following an almost linear decrease line.

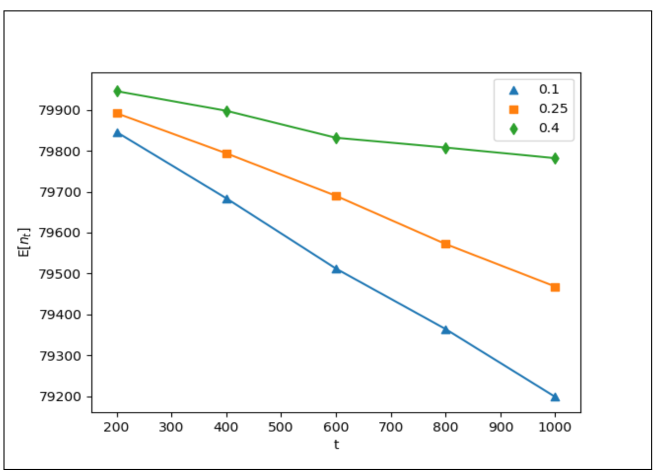

Fig. 4. Reduction in the number of nodes with respect to steps for three different probabilities (0.1, 0.25, 0.4) in the randomly generate dynamic network

IV. Introducing Preferential Addition and Deletion of Edges to the Existing Model

In the existing model, the authors did not consider the addition or deletion of edges independently. Rather, in their

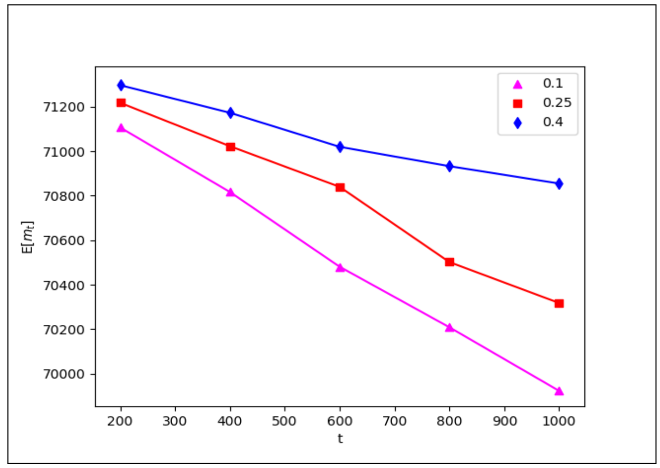

Fig. 5. Reduction in the number of edges with respect to steps for three different probabilities (0.1, 0.25, 0.4) in the randomly generate dynamic network

model, a single edge is added every time a single node is added (adjacent to the node) or whenever a node is deleted, all the edges adjacent to it are deleted along with it. So, in their model, addition and deletion of edges are completely dependent on the addition and deletion of nodes. But in any real-life network, edges can be added and deleted independently. For example, in a social-network like Facebook, edges can be added (creating new relation) between existing users or existing edges between two users can be deleted without deleting any of the nodes (removing someone from friend list). Here, I have extended the existing model by introducing the concept of preferential addition and deletion of edges to the model [Code Used: **Preferential_Edge_Model.py**]. Instead of one probability, in this model I used three types of probabilities. One probability will decide if the current iteration will handle a node or an edge, the second probability represents the probability of adding a node and the third one represents the probability of adding an edge.

A. Preferential Addition of Edge:

In case of adding a new edge, it is more likely that the edge will be added between two nodes which have more common nodes. For example, in case of Facebook, if two users have more mutual friends, they are more likely to get connected. In the proposed preferential model of edge addition, I used this strategy. Initially I selected a node N_1 randomly from the graph. Then I calculated which nodes have more nodes in common in the graph with node N_1 and randomly selected one node (node N_2) with a higher probability among them. To reduce search space, instead of considering all other nodes in the graph, I only considered the nodes which are secondarily adjacent to N_1 (means the nodes which are adjacent to the adjacent nodes of N_1). Then I added an edge between the nodes N_1 and N_2.

B. Preferential Deletion of Edge:

In case of edge deletion, the more likely scenario is that, an edge gets deleted when the end-nodes don't have much in common (less number of common nodes). For example, in Facebook, if two users don't have many mutual friends, that might mean they don't know each other very well and as a result, might remove each other from their friend lists. Following this strategy, I deleted an edge between two nodes which have less number of common nodes. Here also, I first selected one node N_1 randomly. Then for all the ancestors of N_1, I calculated number of common nodes N1 has with its ancestors and created a probability distribution. From that distribution, I randomly selected an ancestor node N_2 which has comparatively less number of common nodes with N_1. Then I removed the edge between N_1 and N_2.

C. Result Analysis

In Figure 6 and 7, we have shown the growth of nodes and edges respectively with respect to number of iterations for the proposed model with preferential edge addition and deletion. In both of the figures, the green straight line represents the predicted line generated by using the original equation written in the paper. But, this model doesn't only depend on the probability of adding or deleting nodes, rather it depends on two other probabilities as mentioned before. So, I have updated the equations for predicted lines of growth of nodes and edges (Equation 3 and 4 respectively) using the basic guideline of their equations. As we can see in the figures, The orange straight line represents the updated predicted line using the new equation and the scatter plot represents the actual simulated values. Suppose here p = probability of handling node, q = probability of handling edge, p1 = probability of birth of node, q1 = probability of death of node, p2 = probability of birth of edge, q2 = probability of death of edge and t represents t^{th} iteration. Now,
Updated Equation for Predicted Values of Nodes:

$$p * (q + 1) * \frac{p1}{2} * ((p1 - q1) + (p2 - q2)) * t + 2 * q1 \quad (3)$$

Updated Equation for Predicted Values of Edges:

$$0.5 * p * (q + 1) * p2 * ((p1 - q1) + (p2 - q2)) * t \quad (4)$$

From the figures, we can see that the simulated values are completely inline with the updated predicted line and slightly misplaced from the previously predicted line (according to the original model). But, both growth of nodes and edges follow a clear linear pattern in the newly proposed model which indicates towards a promising result.

V. Modifying the Equation for Preferential Deletion:

I have modified the preferential equation for deletion of nodes [Equation 2] and formulated a different equation instead [Equation 5]. Here, in this equation we consider a value called "Highest Possible Degree" which represents the maximum degree any node can have in the graph and it can be any high integer value. In our example, "Highest Possible Degree" = 150. Using the previous equation, the simulated values for small number of iterations were comparatively less linear

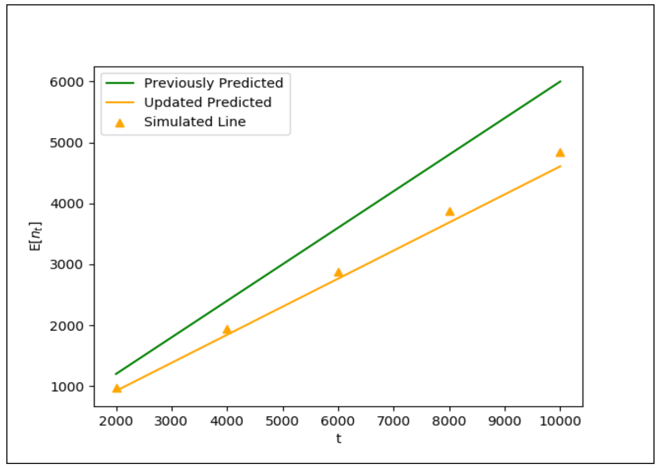

Fig. 6. Growth in the number of nodes with respect to steps in the proposed model

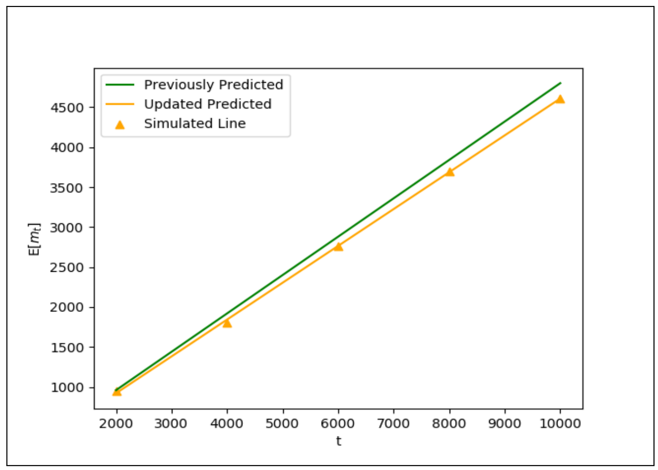

Fig. 7. Growth in the number of edges with respect to steps in the proposed model

and they aligned less with the predicted line [Code Used: **Parameter_Change.py**]. But using Equation 5, I got better results for small number of iterations and the simulated values aligned more closely with the predicted line. This is because, in my equation, nodes with higher degree have been given a higher weight and it increase the chances of selecting a node with higher degree even more. It also performs well for large number of iterations. The comparison between Equation 2 and Equation 5 has been shown in Figure 8 and 9.

$$\frac{150 - degree((EachNode)}{150 * TotalNodes - 2 * TotalEdges} \quad (5)$$

VI. Analyzing Degree Distribution of New and Old Nodes:

I tried to explore another property of the graph model. I analyzed the degree distribution of old nodes and newly created nodes separately in a graph [Code Used: **Old.New.Degree_distribution.py**]. Here, for the graph [**graph_0.9_100000.pickle**, the first 30000 nodes were considered as old nodes and the nodes added after that were

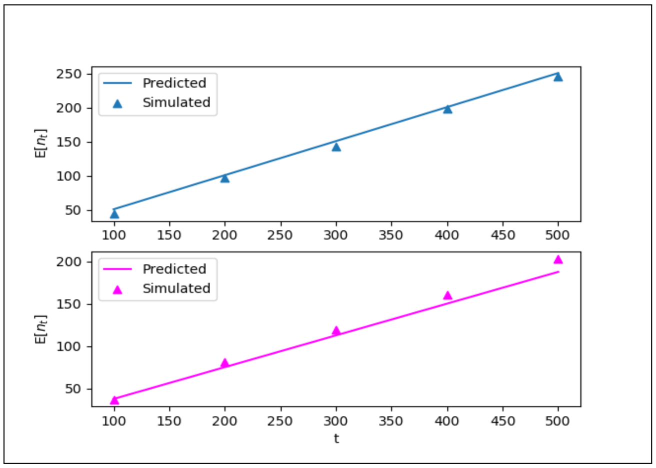

Fig. 8. Growth of nodes and edges with respect to number of iterations using Equation 5

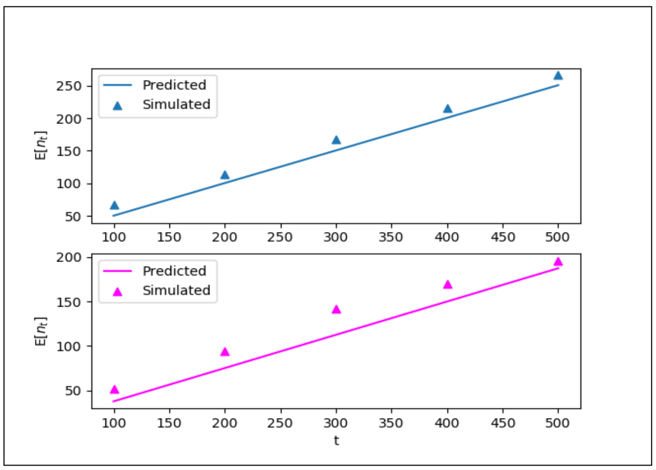

Fig. 9. Growth of nodes and edges with respect to number of iterations using Equation 2

considered as new nodes. From the result, it was clearly visible that, old nodes have the tendency to get higher distribution for large degrees while new nodes more likely to have a lower degree. In Figure 10 we can see that the overall degree distribution property remains same for both old nodes and new nodes and is comparable with the predicted value. But if we consider the cumulative distribution of degree as shown in Figure 11, then it becomes clear that older nodes have a higher distribution value for larger degree and new nodes have a higher distribution values for lower degree. It points towards the fact that, in this model, most of the edges are added to already existing nodes.

VII. Discussion and Conclusion

The dynamic random graph model presented in the paper [10] follows the power-law distribution and hence can be categorized as a scale-free graph. The result of this model corresponds with the real life example used in this paper which opens the possibility that this model is is a good representation of real-life web-like networks. Also, after extending the model

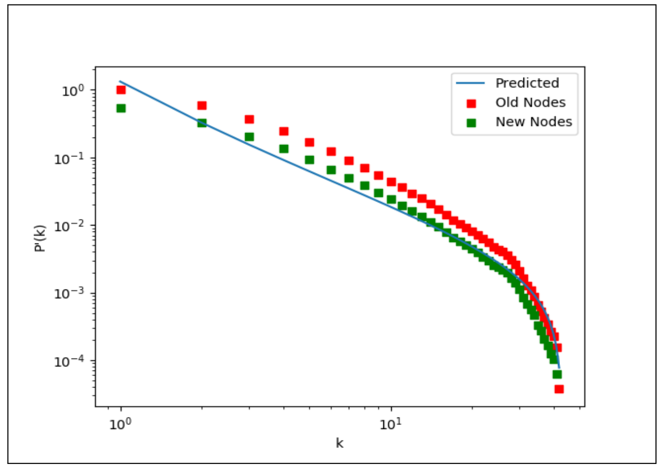

Fig. 10. Log-log plot of the cumulative degree distribution of the graph generated by the preferential model for both old nodes and new nodes

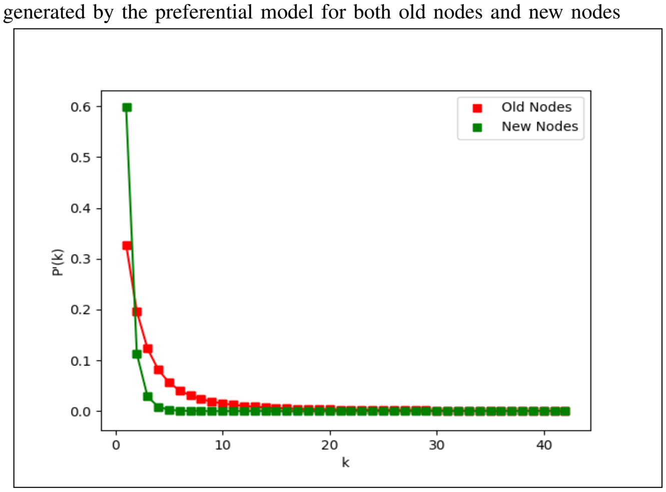

Fig. 11. Comparison of cumulative degree distribution of old nodes and new nodes

with preferential addition and deletion of edges, the output result still followed the properties of scale-free network, so this proposed model should be analyzed further to make the existing model more realistic. The updated equation for the preferential deletion of nodes also gave promising results, so this can be used as an alternative equation to select a node for deletion. Finally, apart from the four properties of the model already discussed in the paper, a new property has been analyzed here which points towards the fact that newly added edges are more likely to become adjacent to older nodes in this model. Overall, using all these works this paper can be extended further with some new implications and property analysis.

References

[1] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, p. 167–256, Jan 2003. [Online]. Available: http://dx.doi.org/10.1137/S003614450342480

[2] B. e. Bollobás, O. Riordan, J. Spencer, and G. Tusnády, "The degree sequence of a scale-free random graph process," *Random Structures & Algorithms*, vol. 18, no. 3, pp. 279–290, 2001.

[3] C. Cooper and A. Frieze, "A general model of web graphs," *Random Structures & Algorithms*, vol. 22, no. 3, pp. 311–335, 2003.

[4] C. Cooper, A. Frieze, and J. Vera, “Random deletion in a scale-free random graph process,” *Internet Mathematics*, vol. 1, no. 4, pp. 463–483, 2004.

[5] S. N. Dorogovtsev and J. F. F. Mendes, “Scaling behaviour of developing and decaying networks,” *EPL (Europhysics Letters)*, vol. 52, no. 1, p. 33, 2000.

[6] P. Krapivsky and S. Redner, “Rate equation approach for growing networks,” in *Statistical mechanics of complex networks*. Springer, 2003, pp. 3–22.

[7] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, “Stochastic models for the web graph,” in *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE, 2000, pp. 57–65.

[8] A.-L. Barabási, R. Albert, and H. Jeong, “Mean-field theory for scale-free random networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 272, no. 1-2, pp. 173–187, 1999.

[9] F. Chung and L. Lu, “Coupling online and offline analyses for random power law graphs,” *Internet Mathematics*, vol. 1, no. 4, pp. 409–461, 2004.

[10] N. Deo and A. Cami, “Preferential deletion in dynamic models of web-like networks,” *Information processing letters*, vol. 102, no. 4, pp. 156–162, 2007.

[11] Stanford large network dataset collection. Stanford University. [Online]. Available: https://snap.stanford.edu/data/ego-Facebook.html