Complex Network

Home work4

Dr.Rahmani

Sara Sadat Younesi – 98533053

Question :

- **Consider the three network datasets of assignment#2**

– If you want, you may also select other datasets.

Answer : I used the first three grids that I made myself

- **Fit ER, WS, BA, and X models to these networks**

– Select (find) X as the best fitting model

– Generate artificial graphs similar to that networks

Model Fitting Steps

1. Model Selection – Question: Which model is better for a specific application? – Find the best model for the target network which generates networks similar to the target network 2. Parameter Tuning – Question: What are the best parameters for the selected model? – Estimate the model parameters which generate the most similar graphs to the target network

Model Fitting Process

• Goal: – to fit a model to a target network instance, – or to fit it to a set of target feature values • Fitted model is supposed to synthesize networks similar to the target network. • Model fitting involves: – select an appropriate network model – then tune model parameters to generate networks with desired properties

BA Model

The Barabási–Albert (BA) model is an algorithm for generating random scale-free networks using a preferential attachment mechanism. Several natural and human-made systems, including the Internet, the World Wide Web, citation networks, and some social networks are thought to be approximately scale-free and certainly contain few nodes (called hubs) with unusually high degree as compared to the other nodes of the network. The BA model tries to explain the existence of such nodes in real networks. The algorithm is named for its inventors Albert-László Barabási and Réka Albert.

Concepts[edit]

Many observed networks (at least approximately) fall into the class of scale-free networks, meaning that they have power-law (or scale-free) degree distributions, while random graph models such as the Erdős–Rényi (ER) model and the Watts–Strogatz (WS) model do not exhibit power laws. The Barabási–Albert model is one of several proposed models that generate scale-free networks. It incorporates two important general concepts: growth and preferential attachment. Both growth and preferential attachment exist widely in real networks.

Growth means that the number of nodes in the network increases over time.

Preferential attachment means that the more connected a node is, the more likely it is to receive new links. Nodes with a higher degree have a stronger ability to grab links added to the network. Intuitively, the preferential attachment can be understood if we think in terms of social networks connecting people. Here a link from A to B means that person A "knows" or "is acquainted with" person B. Heavily linked nodes represent well-known people with lots of relations. When a newcomer enters the community, they are more likely to become acquainted with one of those more visible people rather than with a relative unknown. The BA model was proposed by assuming that in the World Wide Web, new pages link preferentially to hubs, i.e. very well known sites such as Google, rather than to pages that hardly anyone knows. If someone selects a new page to link to by randomly choosing an existing link, the probability of selecting a particular page would be proportional to its degree. The BA model claims that this explains the preferential attachment probability rule.

Later, the Bianconi–Barabási model works to address this issue by introducing a "fitness" parameter. Preferential attachment is an example of a positive feedback cycle where initially random variations (one node initially having more links or having started accumulating links earlier than another) are automatically reinforced, thus greatly magnifying differences. This is also sometimes called the Matthew effect, "the rich get richer". See also autocatalysis.

Networkx BA Model

barabasi_albert_graph

`barabasi_albert_graph(n, m, seed=None, initial_graph=None)` [source]

Returns a random graph using Barabási–Albert preferential attachment

A graph of n nodes is grown by attaching new nodes each with m edges that are preferentially attached to existing nodes with high degree.

Parameters:

n : *int*
Number of nodes

m : *int*
Number of edges to attach from a new node to existing nodes

seed : *integer, random_state, or None (default)*
Indicator of random number generation state. See Randomness.

initial_graph : *Graph or None (default)*
Initial network for Barabási–Albert algorithm. It should be a connected graph for most use cases. A copy of `initial_graph` is used. If None, starts from a star graph on (m+1) nodes.

Returns:

G : *Graph*

Raises:

NetworkXError

Ws Model

The Watts–Strogatz model is a random graph generation model that produces graphs with small-world properties, including short average path lengths and high clustering. It was proposed by Duncan J. Watts and Steven Strogatz in their article published in 1998 in the Nature scientific journal.[1] The

model also became known as the (Watts) beta model after Watts used � to formulate it in his popular science book Six Degrees.

Rationale for the model[edit]

The formal study of random graphs dates back to the work of Paul Erdős and Alfréd Rényi.[2] The graphs they considered, now known as the classical or Erdős–Rényi (ER) graphs, offer a simple and powerful model with many applications.

However the ER graphs do not have two important properties observed in many real-world networks:

They do not generate local clustering and triadic closures. Instead, because they have a constant, random, and independent probability of two nodes being connected, ER graphs have a low clustering coefficient.

They do not account for the formation of hubs. Formally, the degree distribution of ER graphs converges to a Poisson distribution, rather than a power law observed in many real-world, scale-free networks.[3]

The Watts and Strogatz model was designed as the simplest possible model that addresses the first of the two limitations. It accounts for clustering while retaining the short average path lengths of the ER model. It does so by interpolating between a randomized structure close to ER graphs and a regular ring lattice. Consequently, the model is able to at least partially explain the "small-world" phenomena in a variety of networks, such as the power grid, neural network of C. elegans, networks of movie actors, or fat-metabolism communication in budding yeast.[4]

Ws Networkx

Notes

First create a ring over n nodes [1]. Then each node in the ring is joined to its k nearest neighbors (or $k-1$ neighbors if k is odd). Then shortcuts are created by replacing some edges as follows: for each edge (u, v) in the underlying "n-ring with k nearest neighbors" with probability p replace it with a new edge (u, w) with uniformly random choice of existing node w.

In contrast with `newman_watts_strogatz_graph()`, the random rewiring does not increase the number of edges. The rewired graph is not guaranteed to be connected as in `connected_watts_strogatz_graph()`.

watts_strogatz_graph

`watts_strogatz_graph(n, k, p, seed=None)` [source]

Returns a Watts–Strogatz small-world graph.

Parameters:

n : *int*

The number of nodes

k : *int*

Each node is joined with its `k` nearest neighbors in a ring topology.

p : *float*

The probability of rewiring each edge

seed : *integer, random_state, or None (default)*

Indicator of random number generation state. See Randomness.

See also

`newman_watts_strogatz_graph`
`connected_watts_strogatz_graph`

ER Model

In the mathematical field of graph theory, the Erdős–Rényi model refers to one of two closely related models for generating random graphs or the evolution of a random network. These models are named after Hungarian mathematicians Paul Erdős and Alfréd Rényi, who introduced one of the models in 1959.[1][2] Edgar Gilbert introduced the other model contemporaneously with and independently of Erdős and Rényi.[3] In the model of Erdős and Rényi, all graphs on a fixed vertex set with a fixed number of edges are equally likely. In the model introduced by Gilbert, also called the Erdős–Rényi–Gilbert model,[4] each edge has a fixed probability of being present or absent, independently of the other edges. These models can be used in the probabilistic method to prove the existence of graphs satisfying various properties, or to provide a rigorous definition of what it means for a property to hold for almost all graphs.

A network model: an algorithm which generates artificial networks

• It generates artificial graphs which are similar to real-world networks

ER Networkx

erdos_renyi_graph

`erdos_renyi_graph(n, p, seed=None, directed=False)`

Returns a $G_{n,p}$ random graph, also known as an Erdős-Rényi graph or a binomial graph.

The $G_{n,p}$ model chooses each of the possible edges with probability p.

Parameters:

n : *int*

The number of nodes.

p : *float*

Probability for edge creation.

seed : *integer, random_state, or None (default)*

Indicator of random number generation state. See Randomness.

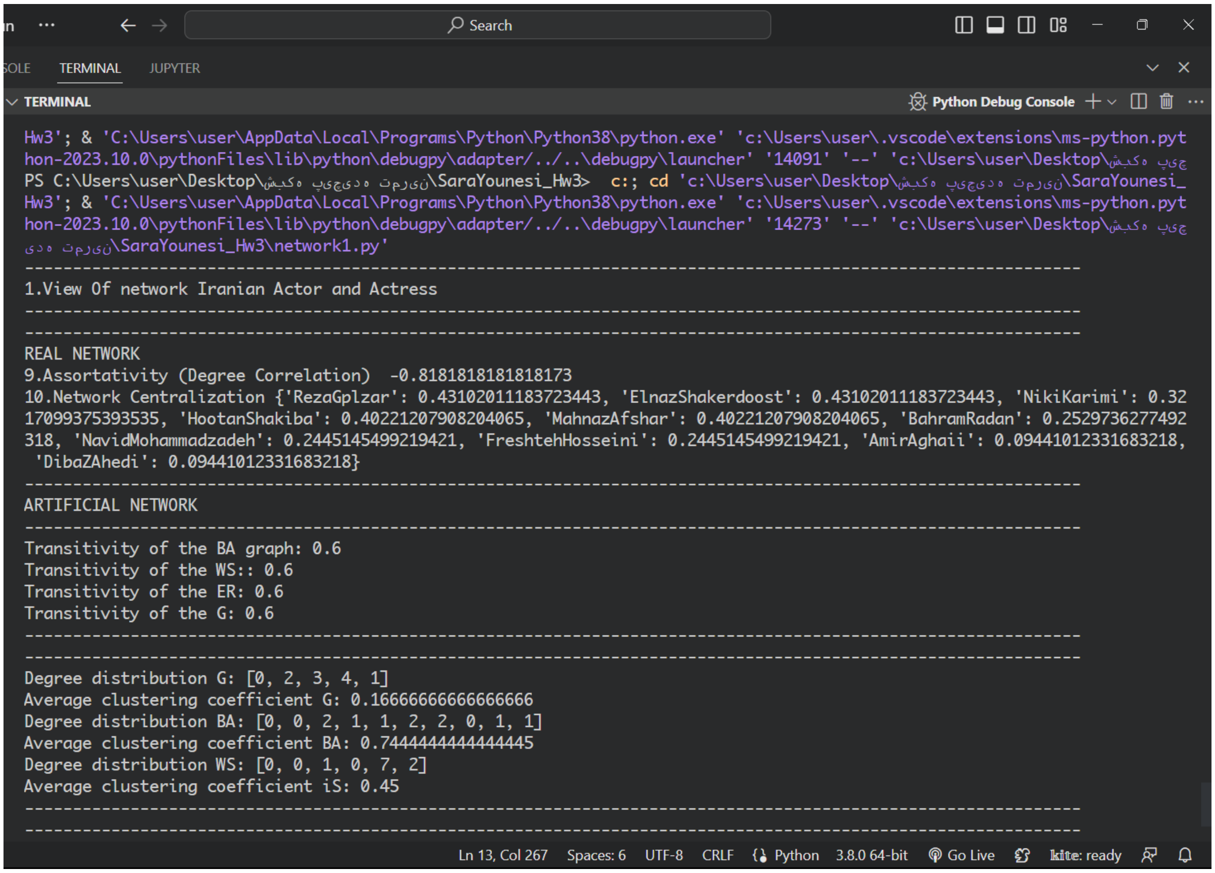
directed : *bool, optional (default=False)*

If True, this function returns a directed graph.

- **Specify the suitable generative parameters**

– Compute some macro-level metrics

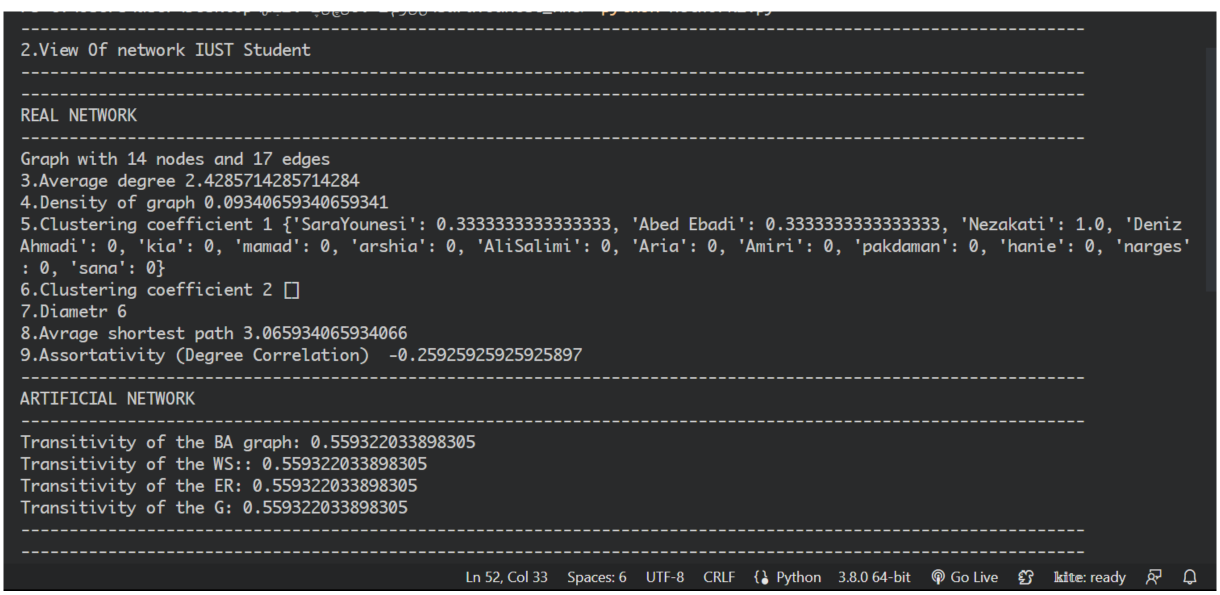
1.METRIC Of network Iranian Actor and Actress



```
Hw3'; & 'C:\Users\user\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.pyt
hon-2023.10.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '14091' '--' 'c:\Users\user\Desktop\شبکه پیچ
PS C:\Users\user\Desktop\شبکه پیچیده تمرین\SaraYounesi_Hw3>  c:; cd 'c:\Users\user\Desktop\شبکه پیچیده تمرین\SaraYounesi_
Hw3'; & 'C:\Users\user\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.pyt
hon-2023.10.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '14273' '--' 'c:\Users\user\Desktop\شبکه پیچ
یده تمرین\SaraYounesi_Hw3\network1.py'
-------------------------------------------------------------------------------------------------------------
1.View Of network Iranian Actor and Actress
-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
REAL NETWORK
9.Assortativity (Degree Correlation)  -0.8181818181818173
10.Network Centralization {'RezaGplzar': 0.43102011183723443, 'ElnazShakerdoost': 0.43102011183723443, 'NikiKarimi': 0.32
17099375393535, 'HootanShakiba': 0.40221207908204065, 'MahnazAfshar': 0.40221207908204065, 'BahramRadan': 0.2529736277492
318, 'NavidMohammadzadeh': 0.2445145499219421, 'FreshtehHosseini': 0.2445145499219421, 'AmirAghaii': 0.09441012331683218,
 'DibaZAhedi': 0.09441012331683218}
-------------------------------------------------------------------------------------------------------------
ARTIFICIAL NETWORK
-------------------------------------------------------------------------------------------------------------
Transitivity of the BA graph: 0.6
Transitivity of the WS:: 0.6
Transitivity of the ER: 0.6
Transitivity of the G: 0.6
-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
Degree distribution G: [0, 2, 3, 4, 1]
Average clustering coefficient G: 0.16666666666666666
Degree distribution BA: [0, 0, 2, 1, 1, 2, 2, 0, 1, 1]
Average clustering coefficient BA: 0.7444444444444445
Degree distribution WS: [0, 0, 1, 0, 7, 2]
Average clustering coefficient iS: 0.45
-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
Ln 13, Col 267   Spaces: 6   UTF-8   CRLF   Python   3.8.0 64-bit   Go Live   kite: ready
```

Best model : BA most similar

2. METRIC Of network IUST Student



```
-------------------------------------------------------------------------------------------------------------
2.View Of network IUST Student
-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
REAL NETWORK
-------------------------------------------------------------------------------------------------------------
Graph with 14 nodes and 17 edges
3.Average degree 2.4285714285714284
4.Density of graph 0.09340659340659341
5.Clustering coefficient 1 {'SaraYounesi': 0.3333333333333333, 'Abed Ebadi': 0.3333333333333333, 'Nezakati': 1.0, 'Deniz
Ahmadi': 0, 'kia': 0, 'mamad': 0, 'arshia': 0, 'AliSalimi': 0, 'Aria': 0, 'Amiri': 0, 'pakdaman': 0, 'hanie': 0, 'narges'
: 0, 'sana': 0}
6.Clustering coefficient 2 □
7.Diametr 6
8.Avrage shortest path 3.065934065934066
9.Assortativity (Degree Correlation)  -0.25925925925925897
-------------------------------------------------------------------------------------------------------------
ARTIFICIAL NETWORK
-------------------------------------------------------------------------------------------------------------
Transitivity of the BA graph: 0.559322033898305
Transitivity of the WS:: 0.559322033898305
Transitivity of the ER: 0.559322033898305
Transitivity of the G: 0.559322033898305
-------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------
Ln 52, Col 33   Spaces: 6   UTF-8   CRLF   Python   3.8.0 64-bit   Go Live   kite: ready
```

Best model : ER most similar

3. METRIC Of network IUST Professors

```
---------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------
REAL NETWORK
---------------------------------------------------------------------------------------------------------------
Graph with 16 nodes and 18 edges
3.Average degree 2.25
4.Density of graph 0.075
5.Clustering coefficient 1 {'Dr.Rahmani': 1.0, 'Dr.Khanjari': 0.13333333333333333, 'Dr.Ashtiani': 1.0, 'Dr.etemadi': 0.33
Transitivity of the ER: 0.47592067988668557
Transitivity of the G: 0.47592067988668557
---------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------
Degree distribution G: [0, 5, 5, 5, 0, 0, 1]
Average clustering coefficient G: 0.21666666666666667
Degree distribution BA: [0, 0, 0, 0, 2, 2, 4, 2, 2, 3, 0, 1]
Average clustering coefficient BA: 0.5034181096681096
Degree distribution WS: [0, 0, 3, 5, 2, 2, 3, 1]
Average clustering coefficient iS: 0.2669642857142857
---------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------
PS C:\Users\user\Desktop\شبکه پیچیده تمرین\SaraYounesi_Hw3>

Ln 52, Col 33   Spaces: 6   UTF-8   CRLF   Python   3.8.0 64-bit   Go Live   kite: ready
```

Best model : WS most similar with respect to parametrs

- **Degree distribution, avg clustering coefficient, …**

– Compare the features of the real and artificial graph counterparts

– Analyze the results (comparison)

Answer :

Fig. 8. The normalized confusion matrix of the Random Forest on the Food network domain (left) and the t-SNE embedding of the corresponding dataset (right).

According to the Random Forest classifier, when the goal is to predict whether the network is real or model-generated, the most distinguishing graph metrics are the normalized average path length, the average clustering coefficient, the maximum eigenvector centrality, and the assortativity.

Figure 9 illustrates the structural properties of the real networks that the network models cannot capture. For example, in the cheminformatics domain, the real graphs typically have relatively large average path length and a large clustering coefficient at the same time. However, most of the models – except for the forest fire – can either generate a graph with a high clustering coefficient but a small average path length or the other way around. Furthermore, most of the infrastructural networks have so large average path lengths that none of the models can recreate (Figure 9). The network models usually generate small-world or ultra-small-world networks, while the diameter of a grid-like infrastructure network, for example, a road network, does not scale logarithmically.

In the case of social networks, average clustering coefficient and assortativity were the most important distinguishing metrics. As the bottom left scatterplot of Figure 9 shows, the stochastic block model and the 2K model can capture the assortativity of the real networks but cannot mimic the high average clustering coefficient. On the other hand, the clustering Barab´asi–Albert model and the forest-fire model can generate highly clustered networks, but the assortativity of these model-generated graphs is usually very close to zero. Hence, a subgroup of the social networks with a rather assortative and highly clustered structure cannot be synthesized with these network models.

By studying the correlation profile of the graph metrics of the real networks, we identified a small, uncorrelated subset of metrics that efficiently describe the real networks. Using the selected nonredundant graph metrics, we fitted each network model to each real network, i.e., we calibrated the models' parameters such that they are as close to the real networks as possible. Note that, we also showed that the distribution of the graph metrics of the random network models is concentrated enough to be able to use them as a base to perform parameter calibration. To be able to measure the similarity of the graphs we used the Canberra distance of the vectors of the selected metrics.

With the help of machine learning techniques, we found that the network models are unable to capture the highly clustered structure and relatively large average path length of the real-world networks. On the other hand, the models are able to capture the degree-distribution-related metrics such as the degree centrality, and interval degree probabilities. We also found that the networks from different domains are not equally modelable, for example, the brain and cheminformatics networks are the most difficult and the food and web networks are the easiest to synthesize.

Node triangle participation: Edges in real-world networks and especially in social networks tend to cluster (Watts and Strogatz, 1998) and form triads of connected nodes. Node triangle participation is a measure of transitivity in networks. It counts the number of triangles a node participates in, that is, the number of connections between the neighbors of a node. The plot of the number of triangles Δ versus the number of nodes that participate in Δ triangles has also been found to be skewed (Tsourakakis, 2008)

Small diameter: Most real-world graphs exhibit relatively small diameter (the “small- world” phenomenon, or “six degrees of separation” Milgram, 1967): A graph has diameter D if every pair of nodes can be connected by a path of length at most D edges. The diameter D is susceptible to outliers.

Results Of 3 Networks

We assign a random number from 0 to 10 to each node and start working.

1.View Of network Iranian Actor and Actress

WS , BA , ER Before Fit

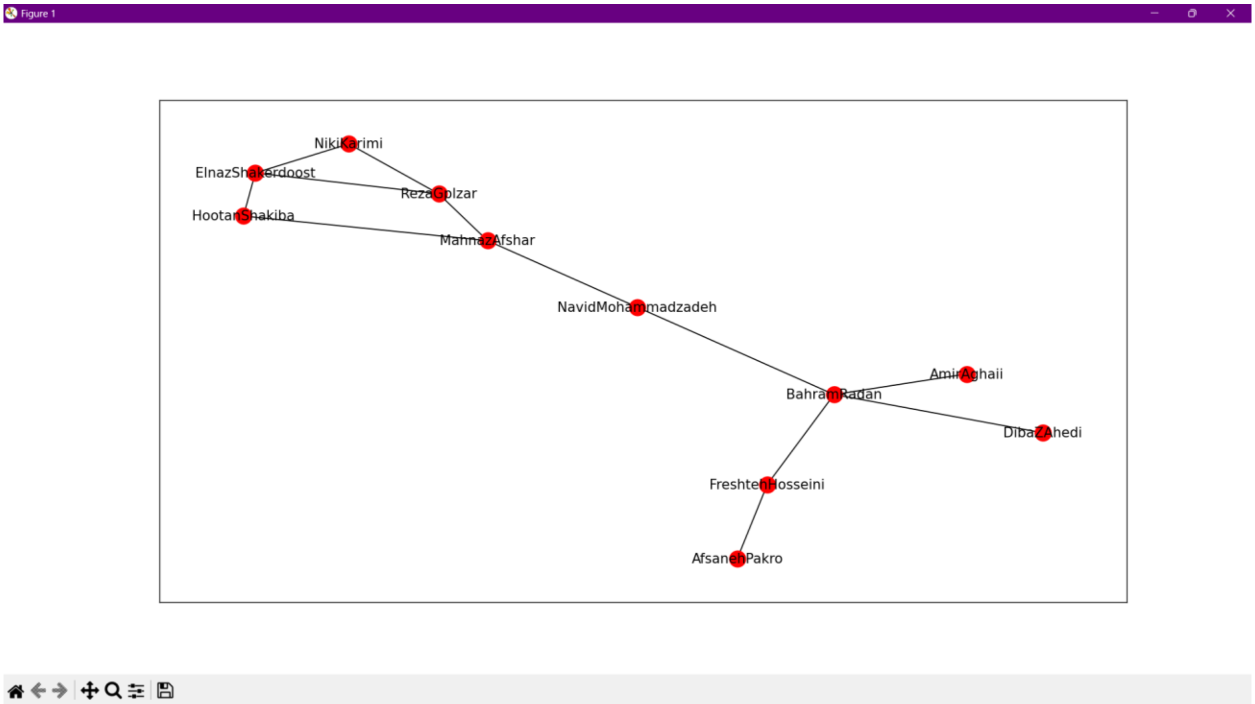


Figure 1
NikiKarimi
ElnazShakerdoost
RezaGolzar
HootanShakiba
MahnazAfshar
NavidMohammadzadeh
AmirAghaii
BahramRadan
DibaZAhedi
FreshtehHosseini
AfsanehPakro

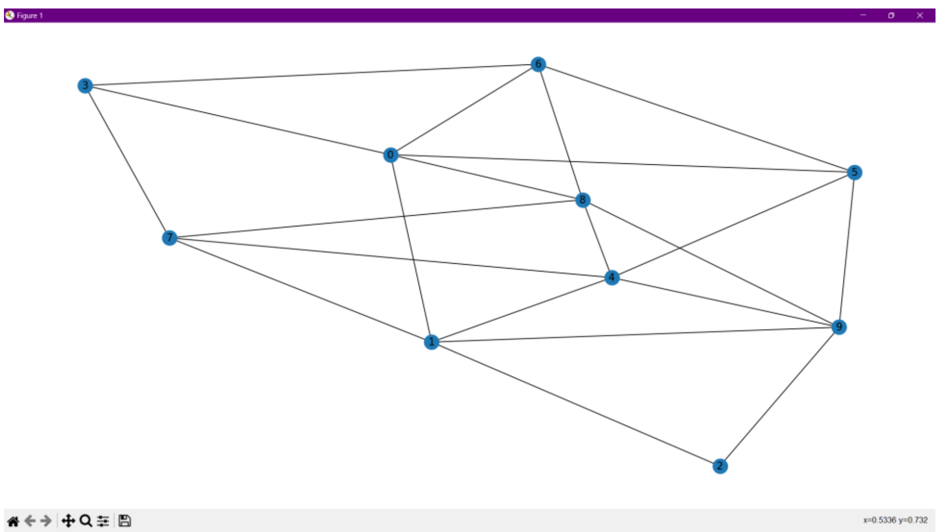


Figure 1
6
3
0
5
8
7
4
9
1
2
x=0.5336 y=0.732

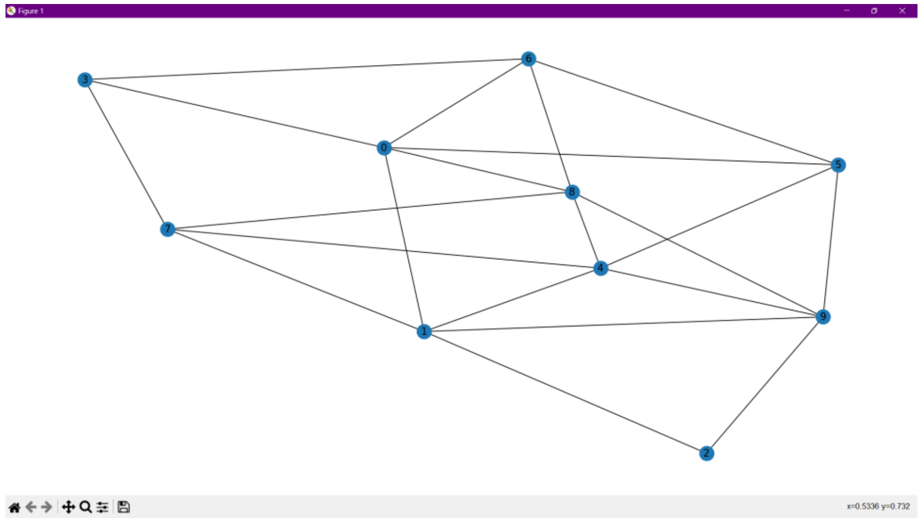


Figure 1
x=0.5336 y=0.732

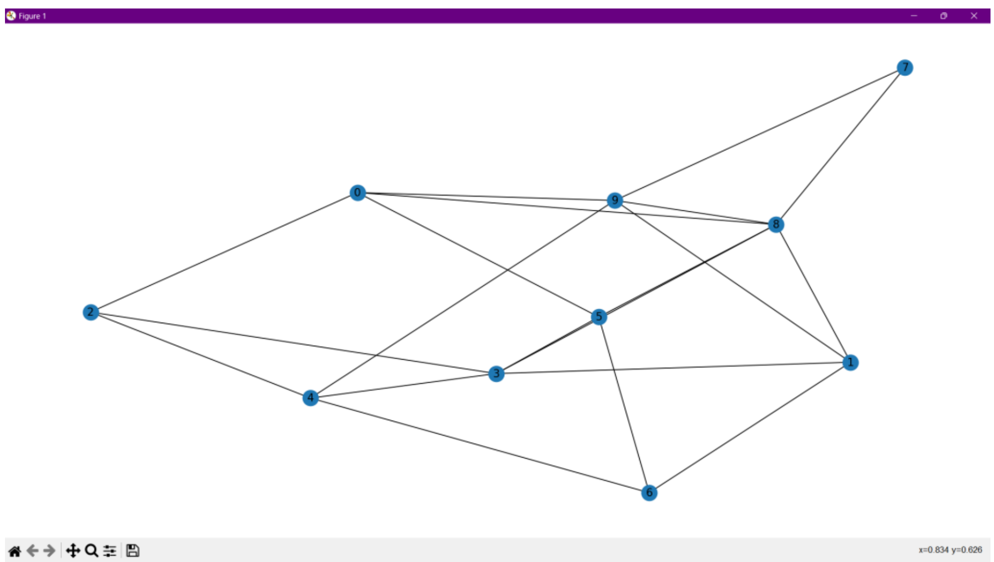


Figure 1
x=0.834 y=0.626

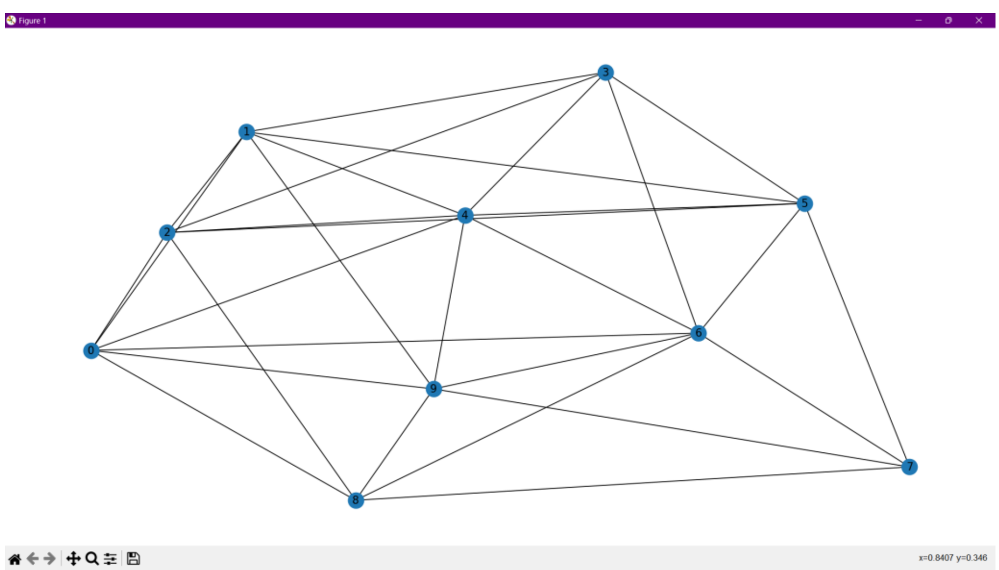
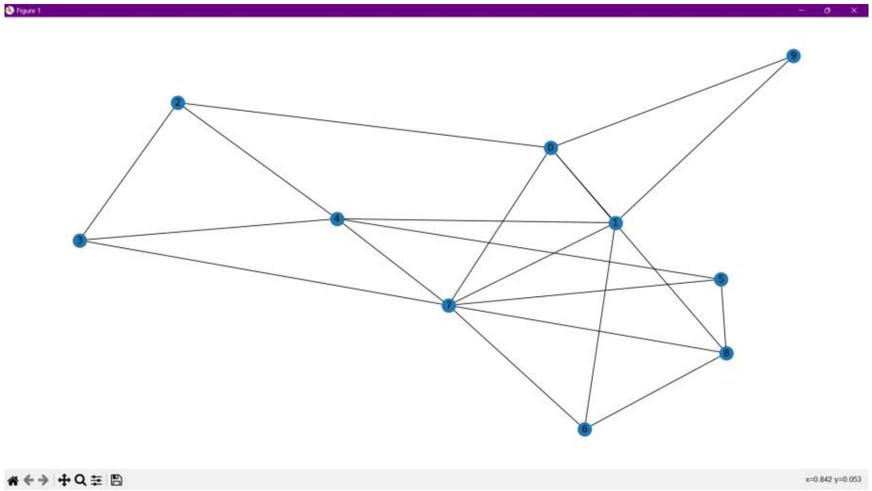
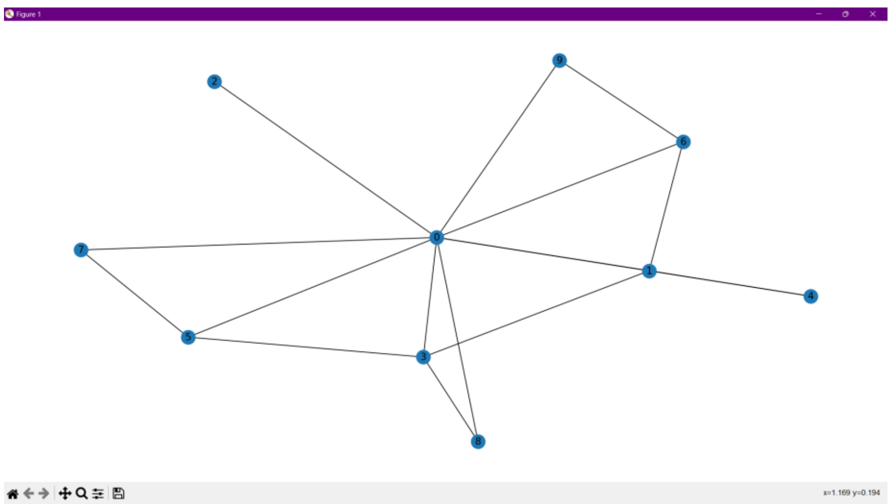
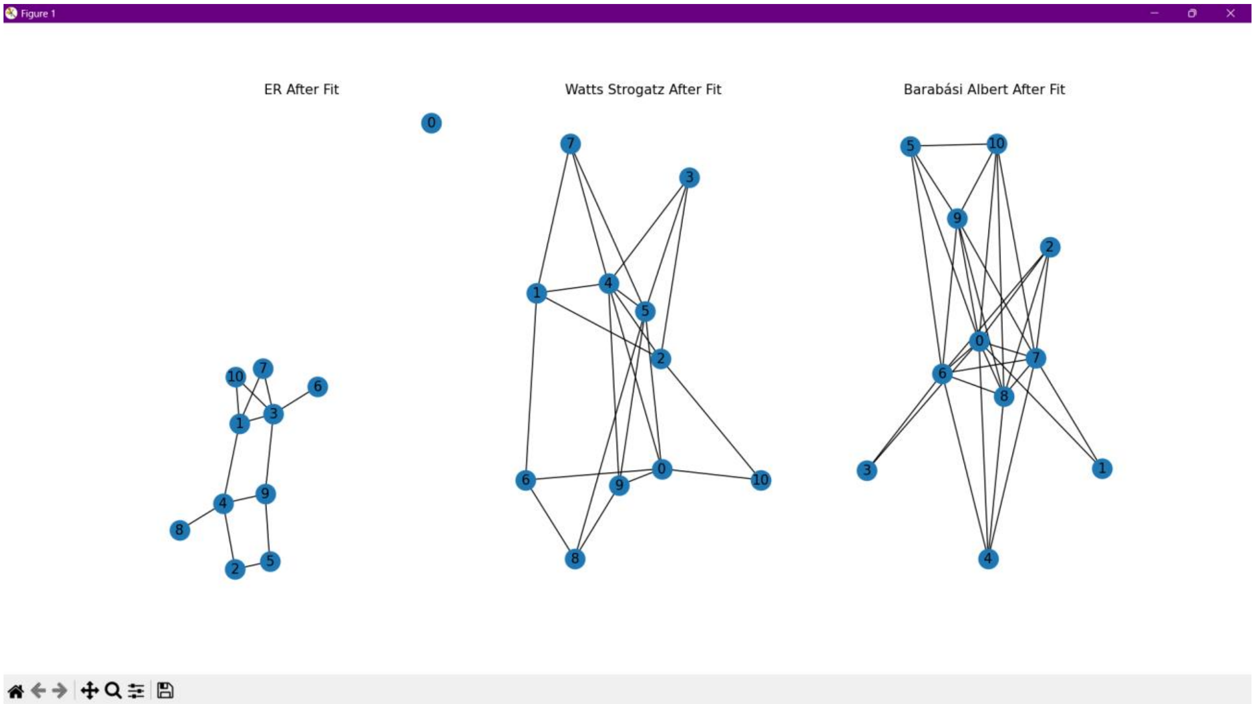


Figure 1
x=0.8407 y=0.346





WS , BA , ER After Fit



2.View Of network IUST Student

WS , BA , ER Before Fit

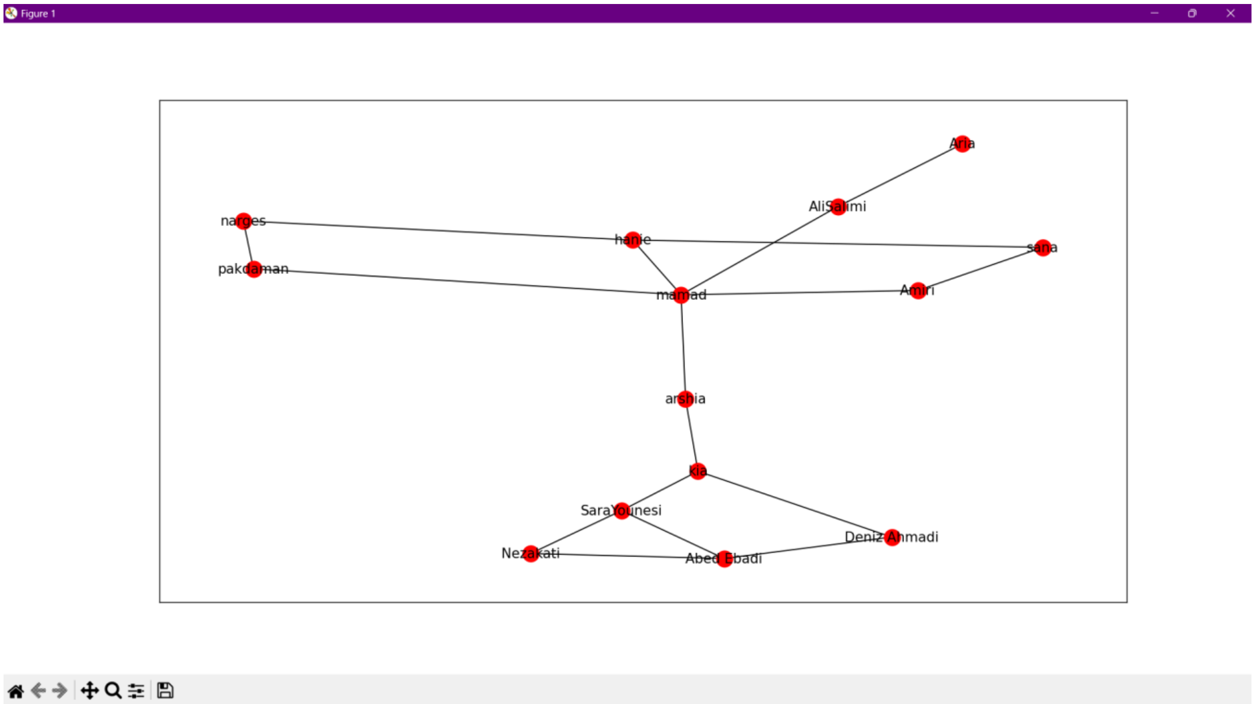


Figure 1
Aria
AliSalimi
narges
hanie
sana
pakdaman
mamad
Amiri
arshia
kia
SaraYounesi
Deniz Ahmadi
Nezakati
Abed Ebadi

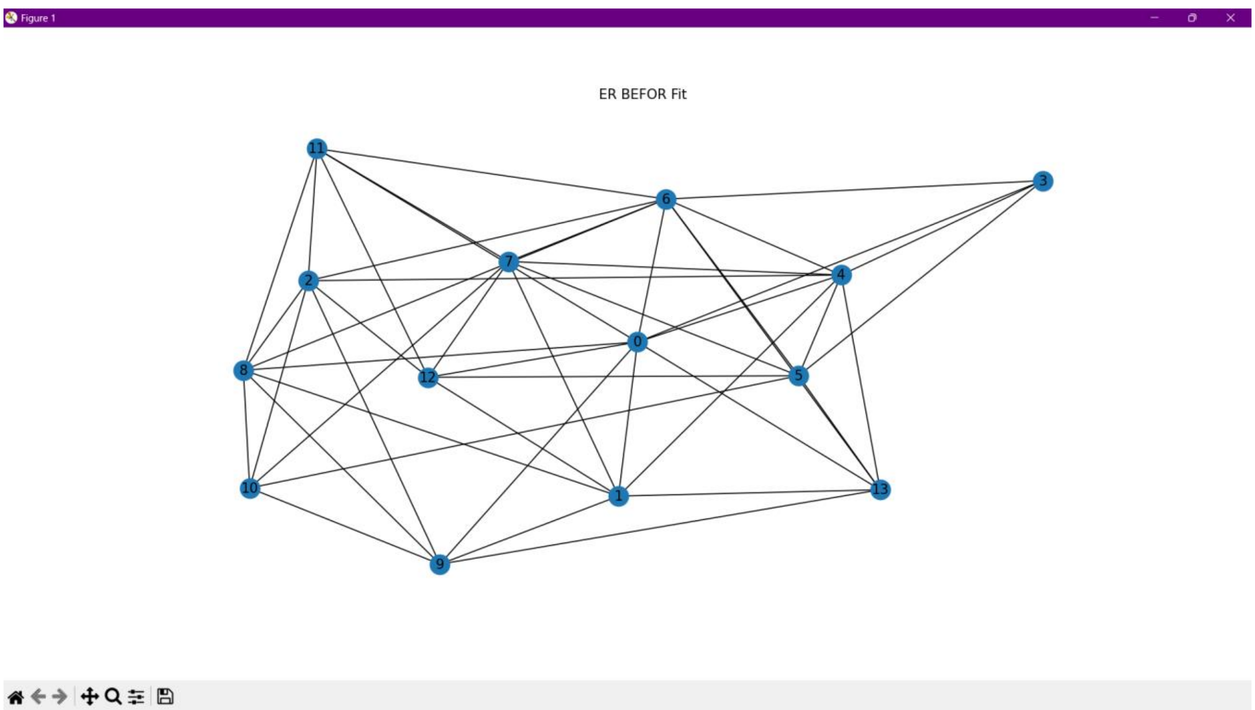


Figure 1
ER BEFOR Fit
11
6
3
7
4
2
0
8
12
5
10
1
13
9

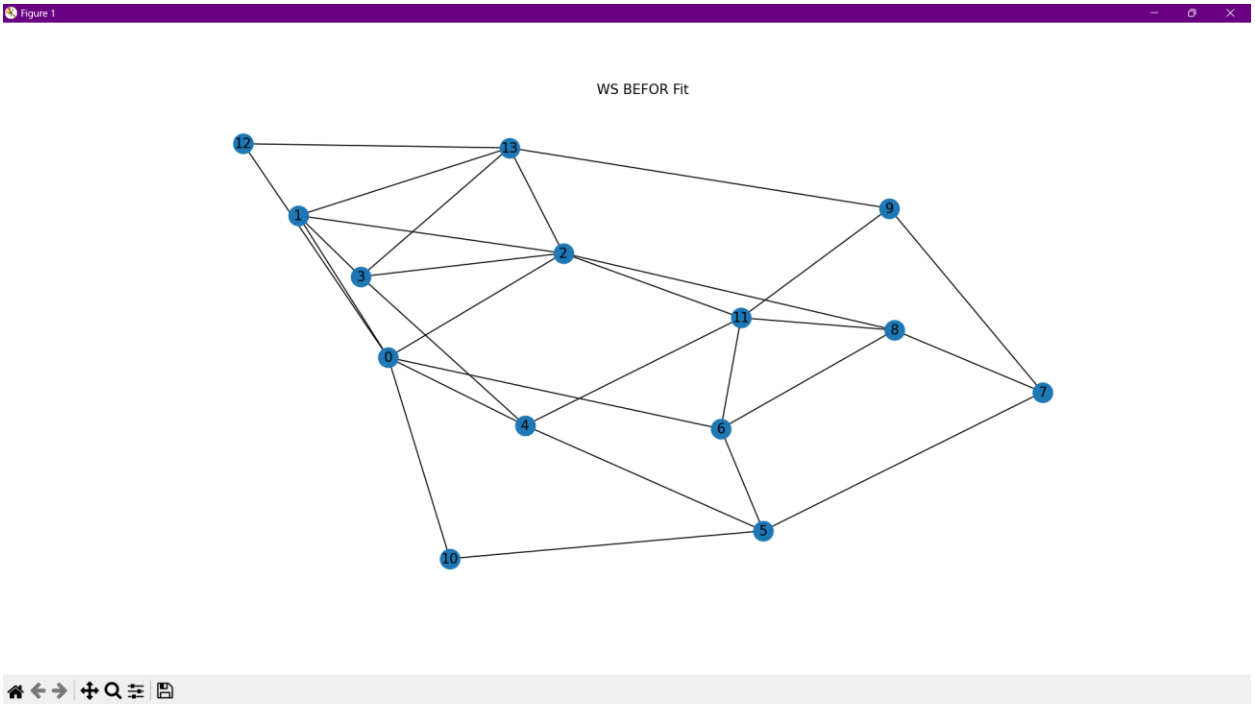


Figure 1
WS BEFOR Fit
12
13
9
1
2
3
11
8
0
7
4
6
5
10

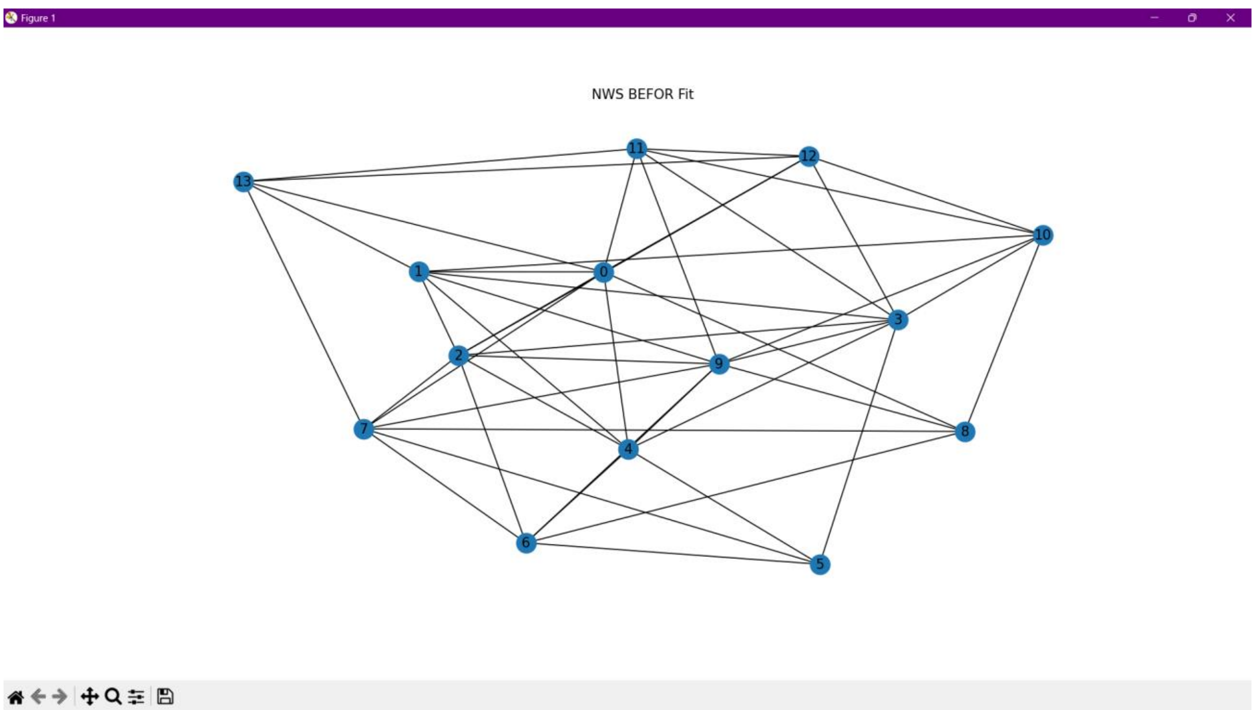
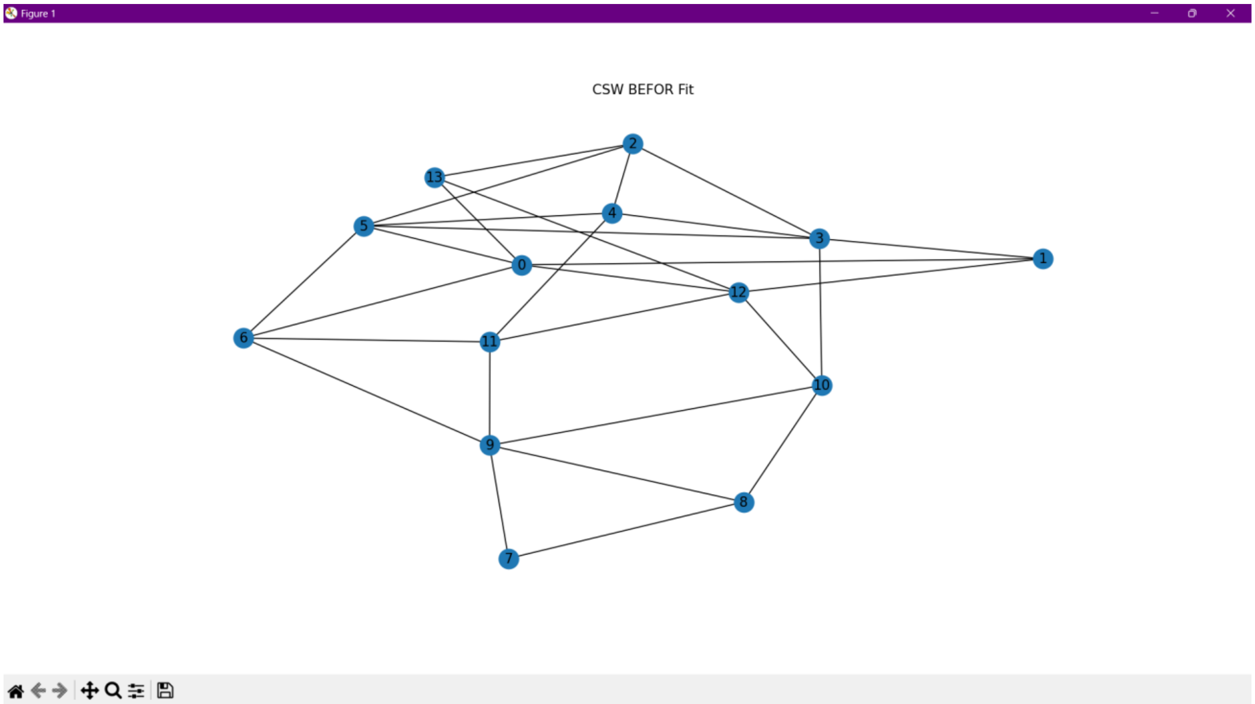
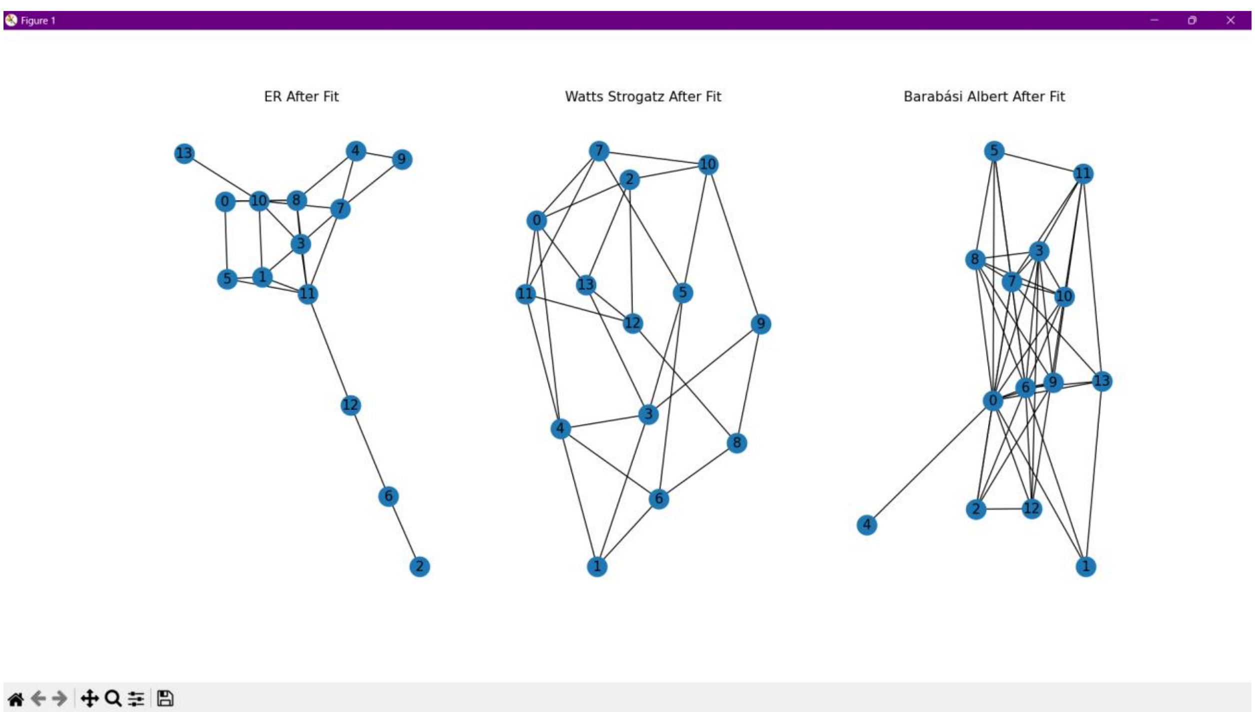


Figure 1
NWS BEFOR Fit
11
12
13
10
1
0
3
2
9
7
8
4
6
5

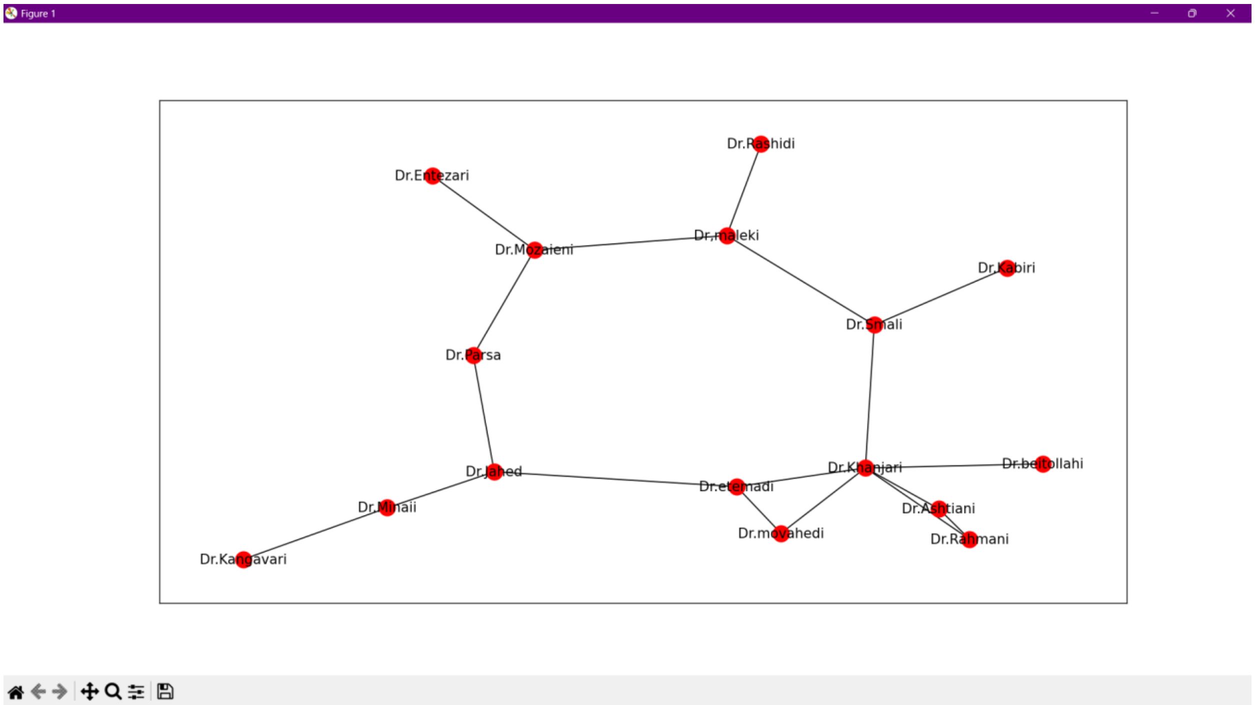


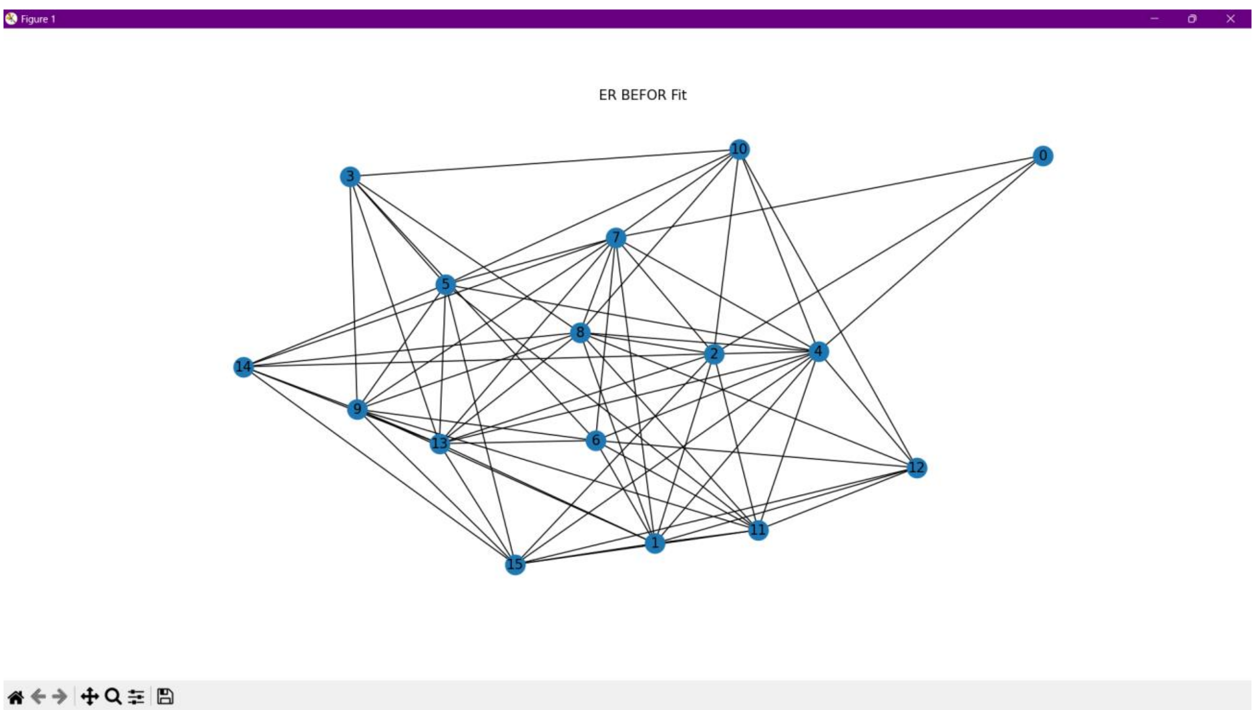
WS , BA , ER After Fit



3.View Of network IUST Professors

WS , BA , ER Before Fit





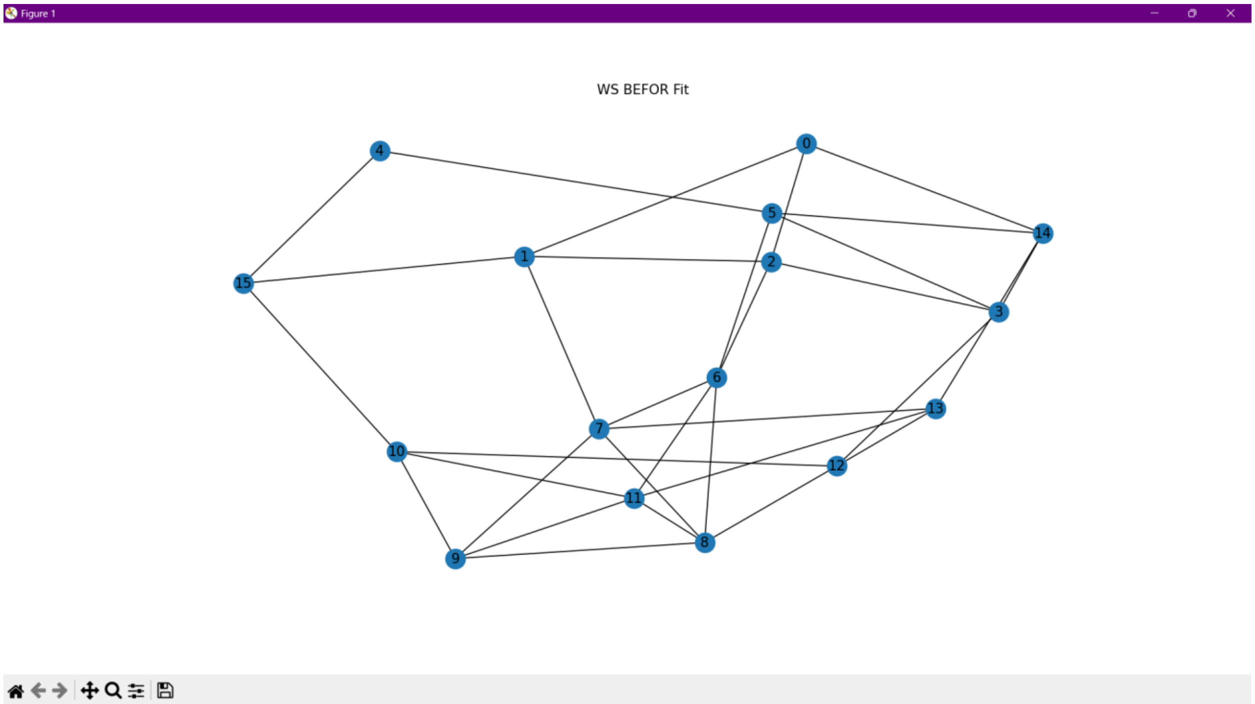


Figure 1
WS BEFOR Fit

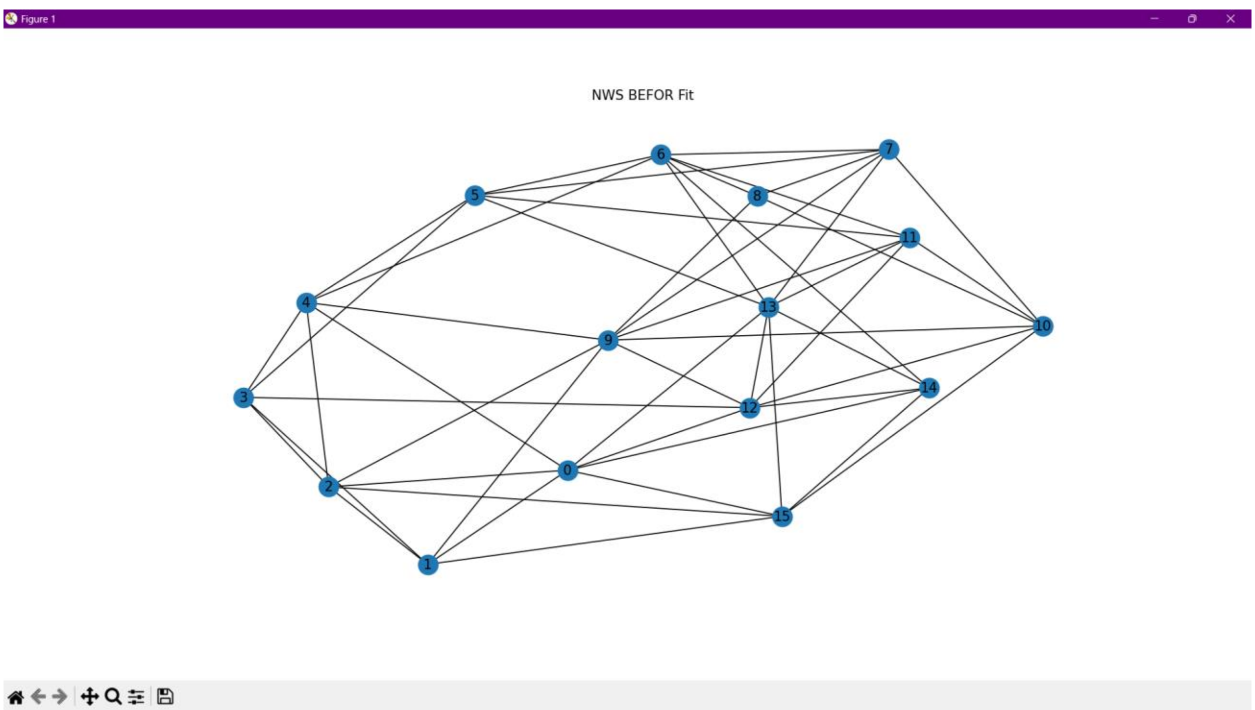
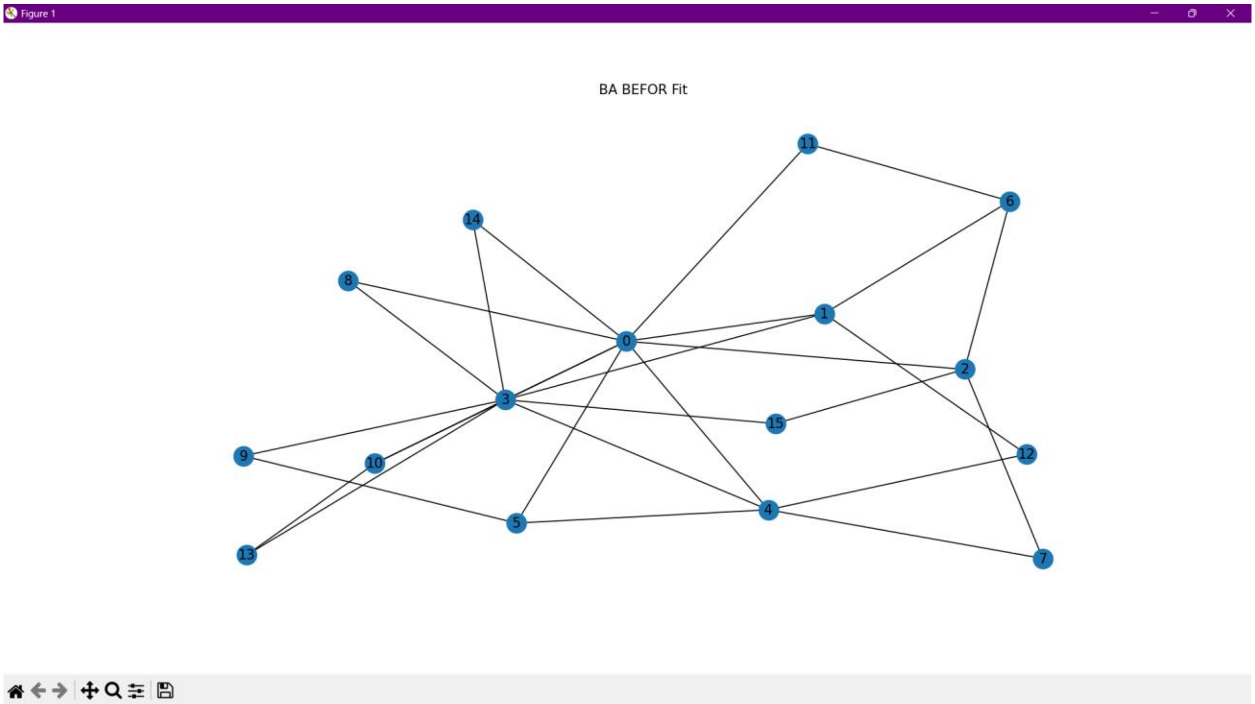
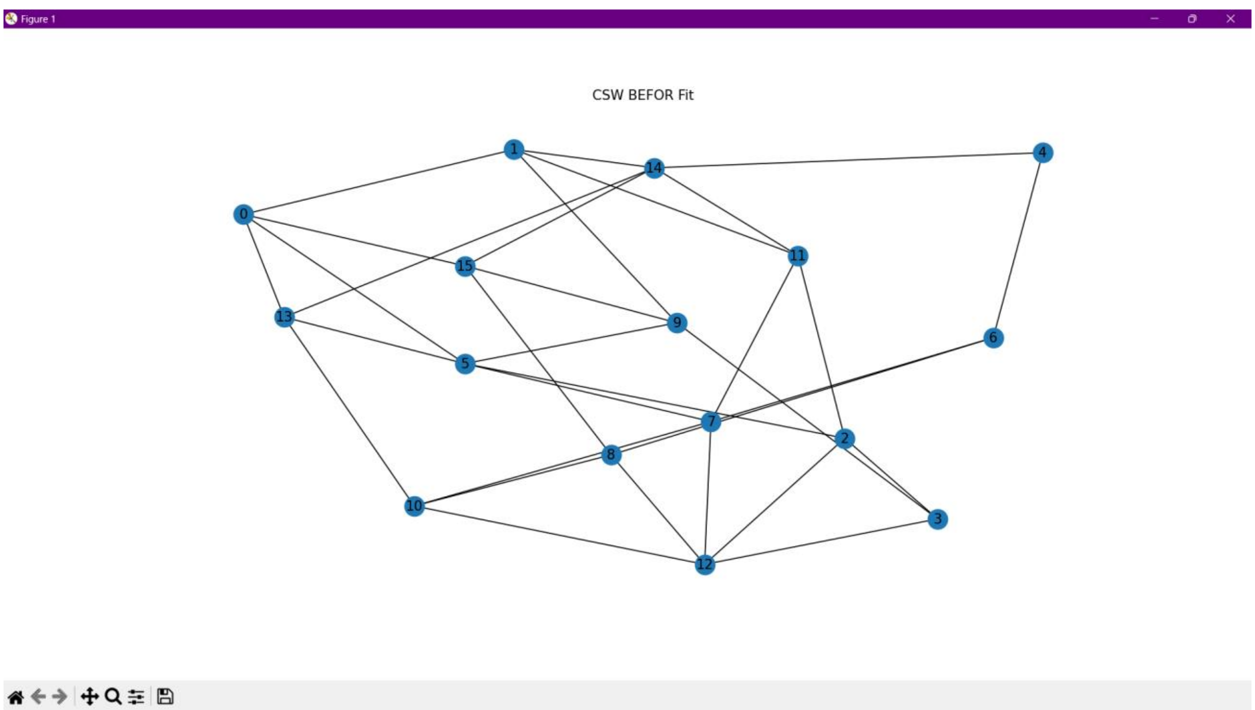
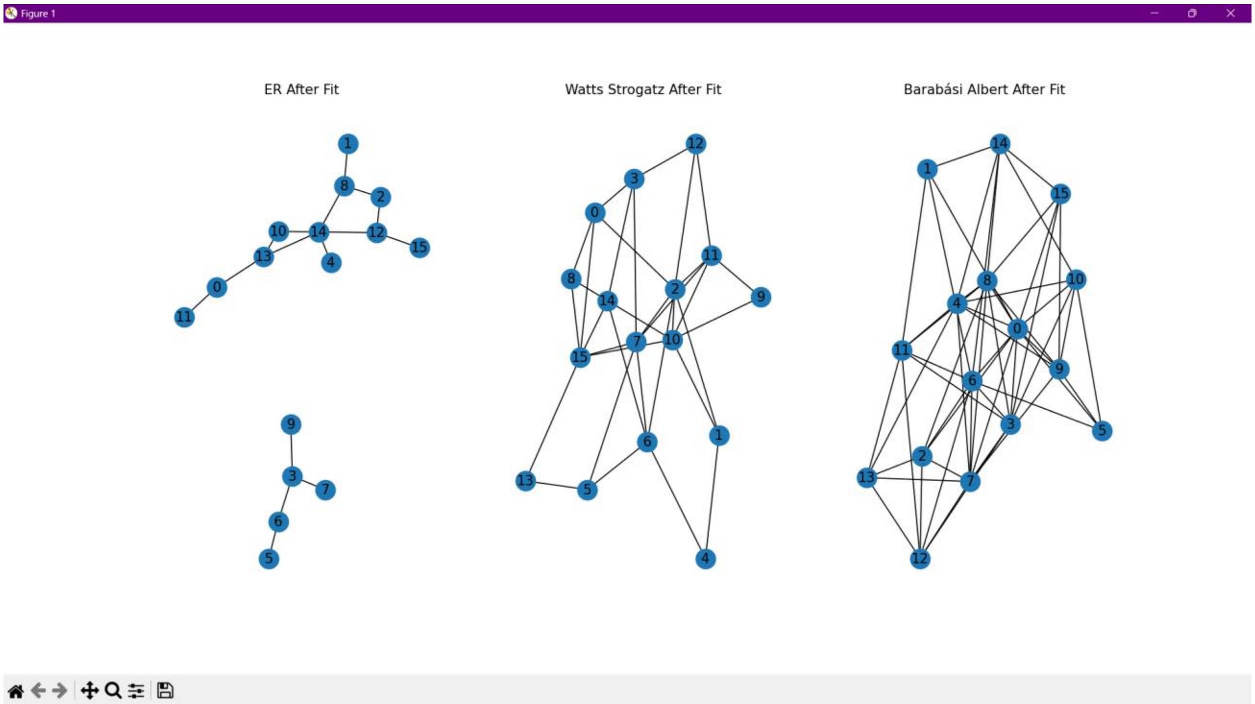


Figure 1
NWS BEFOR Fit





WS , BA , ER After Fit



Source

[— NetworkX 3.1 documentation]

https://arxiv.org/pdf/1810.08498.pdf

https://cs.stanford.edu/people/jure/pubs/kronecker-jmlr10.pdf