Hashtag Co-occurrence Network during World Cup 2018

Social Network Analysis

Academic Year 2017/2018

Prof. Pedreschi
Dr. Rossetti

Narges Osmani
Aleksandra Krzywańska

1. The case study

The FIFA World Cup, often simply called the WorldCup, is an international football competition among different nations which is held once every four years. This event was one of the main highlighted sportive events during June and July 2018 and had so many fans and followers around the world. Twitter users are no exception and followed this event eagerly by tweeting about different topics including matches and events, using different hashtags. The main hashtag during this event was #WorldCup. Users appended this hashtag to their tweets along with some other hashtags and expressed their ideas. Also this event was a great opportunity for some of the users to express their off-topic ideas, including political, economical, cultural and environmental issues.

In this project, we study the various kinds of concerns raised by Twitter users, in terms of hashtags during the period of World Cup. We *envisioned* a network of hashtags, in which hashtags are related if they co-occur in a tweet and related hashtags are in clearly-cut communities.

Since World Cup 2018 started from June 14, 2018, we collected all the tweets containing #WorldCup in the first week of the games (14th - 21st). Among different spelling variants, the hashtag #WorldCup had %81 of frequency and #worldcup had %16 of frequency[1]. Thus, we also collected tweets containing #worldcup. For the first week, we gathered 1,809,545 tweets, containing #WorldCup or #worldcup. As the number of tweets was very large and this conflicted with our PC's computational power, we decided to work on tweets of just 1 day.

We selected the tweets of the day, 17th June 2018, because on that day some important matches played.

Matches on 17th June

Match	Result
Costa Rica vs. Serbia	**0-1**
Germany vs. Mexico	**0-1**
Brazil vs. Switzerland	**1-1**

For 17 July, we had a total of 290K tweets. After analyzing hashtags, we found out that there are several other related hashtags which are co-occurred with "#WorldCup" and we decided to download tweets containing these related hashtags for 17th. The list of all hashtags considered in this report is:

[WorldCup, worldcup, WorldCup2018, Russia2018, Rusia2018, FIFA, FifaWorldCup, FathersDay, GER, MEX, GERMEX, Germany, Mexico, BRASUI, BRA, SUI, Switzerland, Brazil, Costarica, Serbia, CRC, SRB, CRCSRB, copa2018, football]

At the end, we had a total of 1,251,627 tweets. Then, we built our hashtag co-occurrence network, in which the nodes are hashtags and the edges show the co-occurrence of endpoint hashtags in at least one tweet. In other words, two hashtags are co-occurring if mentioned by the same tweet. We set the weight of each edge to the total number of co-occurrences of two endpoints. Also, to deal with spelling variants, we converted all the hashtags to lowercase (Figure 1). The associated hashtag map shows the main topics of concern of Twitter users.

[1] Based on https://hashtagify.me/hashtag/worldcup

There were 4,648,241 co-occurrences among which 835,478 were unique. To improve the quality of the network, we decided to ignore the edges with weights lower than 4. Finally, we built our network with 12,624 nodes and 99,028 edges.

Figure 1. An example tweet and corresponding sub-graph

2. Network Analysis

In this section, we discuss the properties of our network and two randomly generated networks. We used networkX python library and Gephi to perform our analyses. The notebooks containing our code and dataset used for analyzing both networkX graph and Gephi csv output is available online[2].

2.1. Summary of network properties:

In this section, we summarized the properties of the crawled network (from now on, we call it GHT) and two synthetic networks. Table 1, shows general statistics of our network along with synthetic Erdos-Reyni (GER) and Barabasi-Albert (GBA) networks. Our network, GHT is undirected, weighted and has 12,624 nodes and 99,028 edges. To generate synthetic networks with the same scale (number of nodes and edges), we calculated the required parameters. For Erdos-Reyni graph, we computed p as:

$$p = 2\,|E| / (|N|\,(|N| - 1)) = 0.00124287$$

and the resulting network GER has roughly the same number of edges with GHT. For Barabasi-Albert network, the parameter m is calculated as:

$$m = \text{ceil}(\,|E| / |N|\,) = 8$$

We set m = 8 to get almost the same number of edges with G_{HT}.

Table 1. Networks Summary

	G_{HT}	G_{ER}	G_{BA}
# Nodes	12624	12624	12624
# Edges	99028	98799	100928
Density	0.00124287	0.00124000	0.00126672
Average Degree	15.6888	15.6525	15.9898

[2] https://github.com/Narges1365/NS-Project

Average Path Length	2.5079	3.7272	3.3112
Diameter	7	6	5
Average Clustering Coefficient	0.6817	0.0012	0.0077
# Connected Components	102	1	1
Size of Giant Component	12366	12624	12624

For average path length, we computed the shortest path distance among all pairs of nodes (almost 79M combinations) and then the average of all these distances. Our network has 102 connected components and the giant component has %97 of nodes and %99 of edges. The ER and BA networks are only composed of 1 connected component.

2.2. Degree Distribution

In this section we discuss the degree distribution of G_{HT} and compare it with G_{ER} and G_{BA}. The degree distribution plotted in a logarithmic scale is shown in figure 2. Our network has almost a power law degree distribution with a particularly long tail. The long tail is due to the presence of some hashtags with very high degrees (hubs) which are much greater than the average degree (15.68). There are 2778 nodes with degree 1 and 208 nodes with degree higher than 100.

We also plotted the degree distribution of G_{ER} and G_{BA}. For Erdos-Reyni network, the average degree is 15.65 and the distribution is Poisson. The degree of the nodes ranges from 2 to 37. There are many nodes with degree close to the average degree. In fact, the average degree of all three networks are almost the same.

For G_{BA} the degree distribution is a power law distribution with a relatively large number of high degree nodes which is caused by the preferential attachment phenomenon. The degree of the nodes ranges from 8 to 574 with an average of 16. As we set m=8 when generating this network, the minimum degree is 8 which is different from our real network which has a large number of nodes with lower degrees.

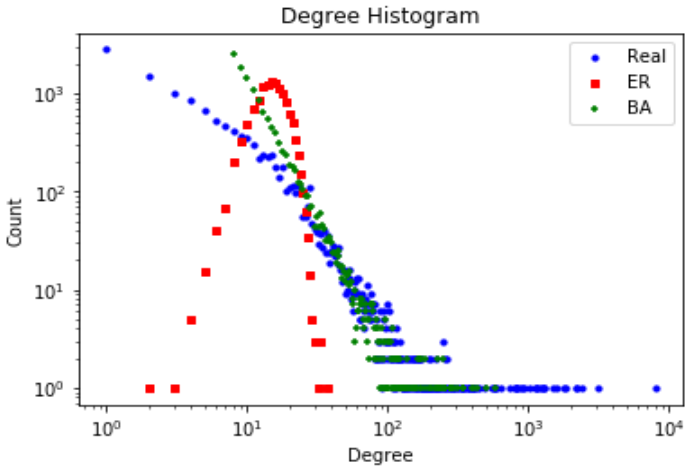


Figure 2. Degree Distribution

2.3. Hubs Analysis

Figure 3, illustrates highest degree hubs in our network. As can be seen in the figure, the high degree hashtags are mostly related to general world cup terms (in different languages) and also the matches in day 17th. There is a hashtag in the figure which is not related to football, i.e. FathersDay. We figured out that the father's day in Mexico and Costa Rica (2 of the countries who played on 17th) is 17th of June 2018 and there were a lot of tweets mentioning the hashtag #fathersday and relating it to the matches.
Another interesting point is the presence of both #rusia2018 and #russia2018 in our hubs. In fact, the former is mostly mentioned in Spanish tweets and is our second largest hub. The latter is the English spelling (the word "Russia" is spelled as "Rusia" in Spanish) and is mostly mentioned in English tweets. We believe that the reason why #rusia2018 has a higher degree is that the spoken language of 2 countries out of 6 competing teams is Spanish.

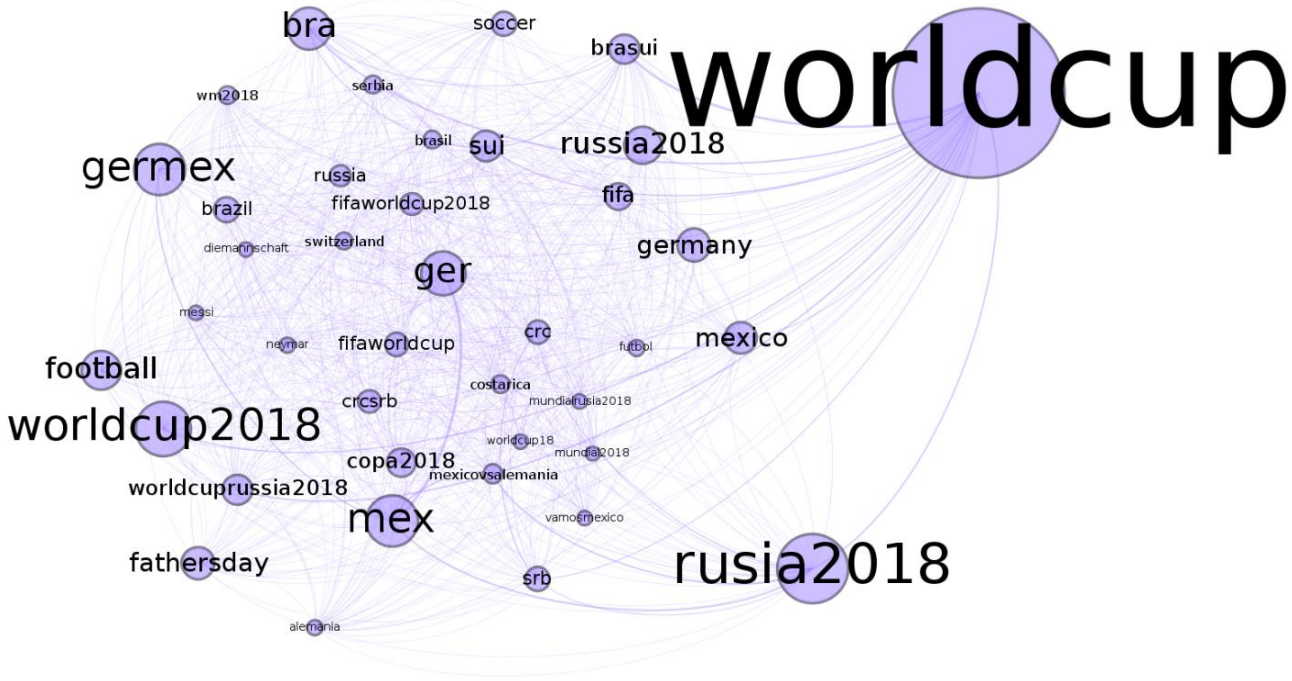


Figure 3. Largest hubs in our crawled network

2.4. Connected Components Analysis

In table 2, we summarized the information about 5 largest connected components of our network. The giant component consists of 12366 nodes and 98763 edges, which is clearly observable in figure 3. The second largest connected component is composed of 17 nodes and 72 edges. Figure 4 shows the second largest connected component, which seems to be popular Indonesian dating hashtags. In fact, the users who created these hashtags tried to co-locate them with hashtags related to world cup to make them more pervasive. As we described above, when we filtered the edges having a minimum weight of 4, this sub-graph has been separated.

Table 2. 5 largest connected components in the network

	# Nodes	# Edges
Giant Component	12366	98763
2nd Largest Component	17	72
3rd Largest Component	7	11
4th Largest Component	5	7
5th Largest Component	5	10

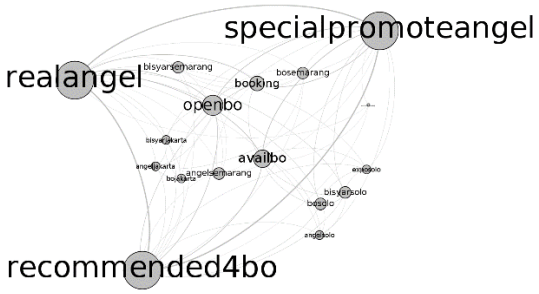


Figure 4. Second Largest Connected Component

2.5. Path Analysis

We computed the distribution of shortest path lengths for G_{HT}, G_{ER} and G_{BA}. To do this, we computed the shortest path length among all of the node pairs (~79M) in each graph and then the average of these distances. Then, we plotted the histogram for the shortest path lengths (Figure 5). As can be seen in the figure, the longest shortest path in G_{ER} and G_{BA} which are the diameter of the networks are 6 and 5, respectively, but for our network the diameter is 7. The average path length (<d>) for our network is 2.5, for the Erdos-Reyni graph is 3.7 and for Barabasi-Albert graph is 3.

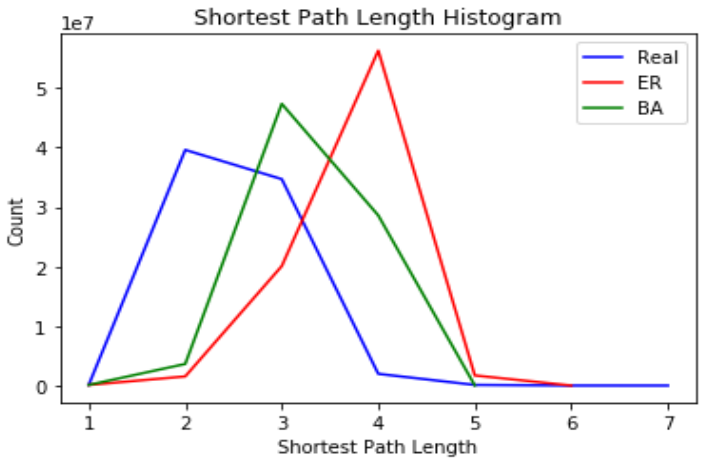


Figure 5. Shortest Path Length Distribution

2.6. Clustering Coefficient Analysis

In this section we studied the dependence of the local clustering coefficient C(k) on the node's degree for G_{HT}, G_{ER} and G_{BA}. For each network, we computed the local clustering coefficient for each of the nodes in the network and then calculated the average clustering coefficient of all nodes with the same degree k. As shown in Figure 6, for our real network, the clustering coefficient clearly depends on the node's degree. For small degree nodes, the clustering is high (around 1) and for high degree nodes, the clustering tends to zero.

The distribution of C(k) of BA and ER graphs are shown in Figure 7. As expected, there is no significant dependence among degree and local clustering and the values fluctuate around the <C> for both ER and BA.

For G_{HT} the average clustering coefficient is high (~0.7) compared to ER and BA graphs which shows the high local link density and connectivity of our network.

We also plotted the average number of triangles incident to each node for different degrees for three graphs. As shown in figures 8 and 9, there is linear dependence between the degree of a node and the number of local triangles in G_{HT} and G_{BA}. But this dependence is sublinear for G_{ER} and the number of triangles is very low for each degree which shows the sparseness of the network.

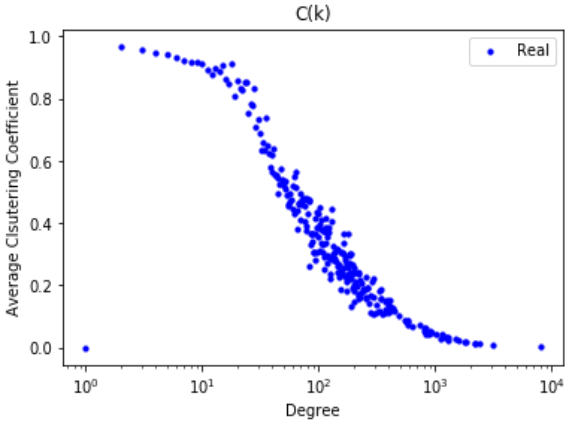


Figure 6. C(k) for hashtag network

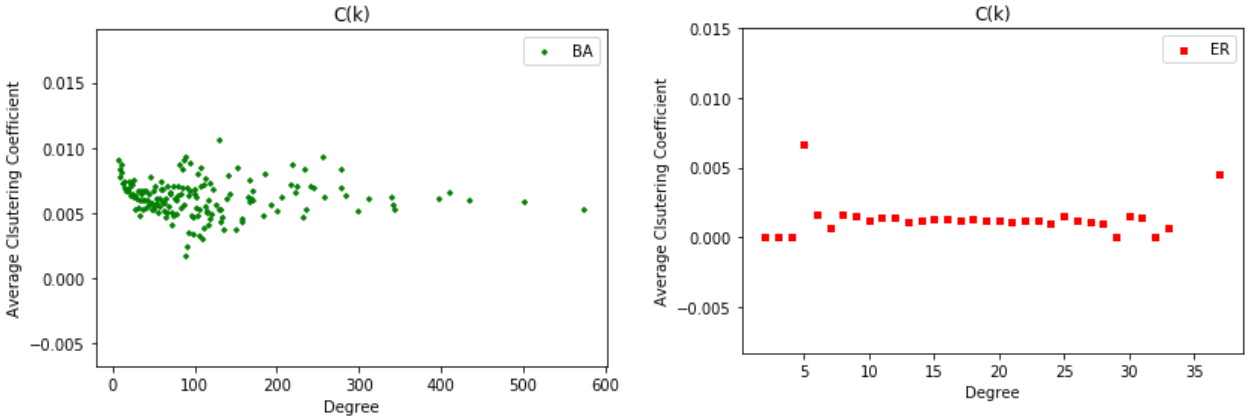


Figure 7. C(k) for BA and ER networks

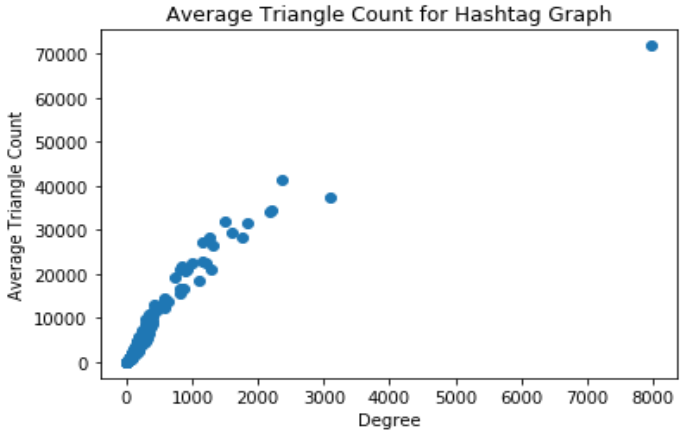


Figure 8. Triangle count for hashtag network

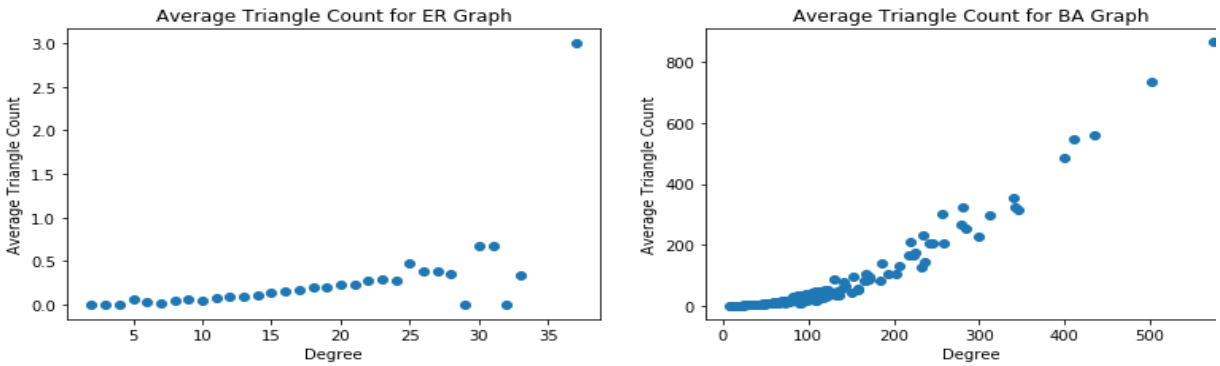


Figure 9. Triangle count for ER and BA networks

2.7. Centrality Analysis

In this section, we studied different centrality measures for three networks. For centrality measures, we used the Gephi software as it is faster in computing centralities. Then, we exported the results to parse in the python script. Again, we computed the average centrality measures for each degree and plotted the distributions in figures 10 and 11.

By analysing the plot of figure 10, we can see that in G_{HT} the nodes with high degrees also have high betweenness centralities and hence are more important. The 2 largest of our hubs, i.e., #worldcup and #rusia2018 have betweenness centralities of 41M and 12M, which shows the number of shortest paths crossing these two nodes. Almost 8700 nodes in G_{HT} has betweenness centrality of 0 which are mostly 1-degree nodes. Our network is mostly similar to the BA network since in BA the hubs are also prevalent and differs significantly from the ER network.

The average closeness centrality for different k is plotted in figures 12 and 13. As evident from the figures, the nodes with high degrees are closer to other nodes than the nodes with lower degrees, in average. We believe that there are a large number of low-degree nodes which are directly connected to one of the largest hubs and as a consequence they have a shortcut through these hubs to other nodes in the network. Again, our network is more similar to BA than ER in terms of closeness centrality.

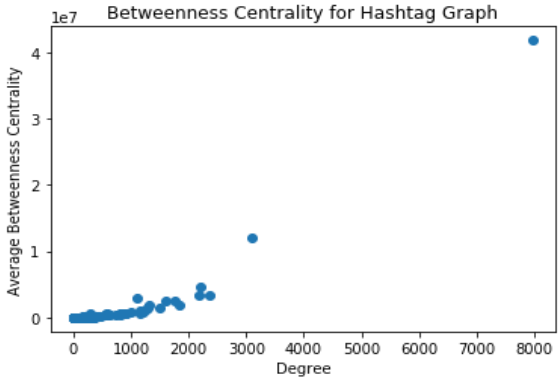


Figure 10. Average betweenness centrality for hashtag network

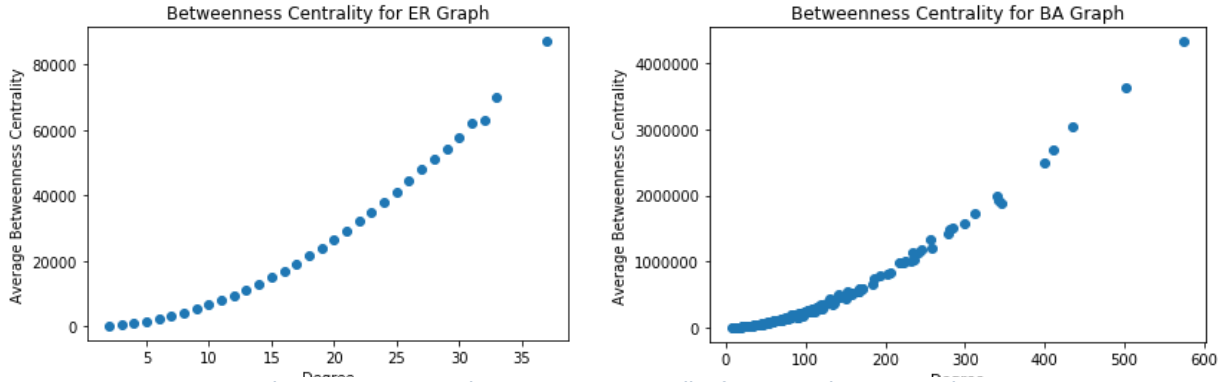


Figure 11. Average betweenness centrality for ER and BA networks

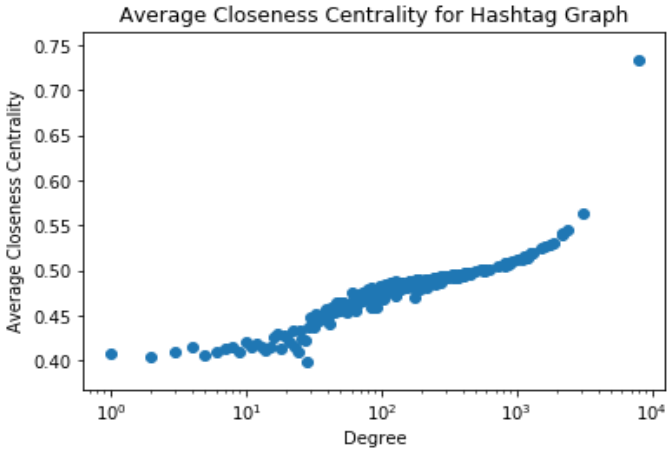


Figure 12.Average closeness centrality for hashtag network

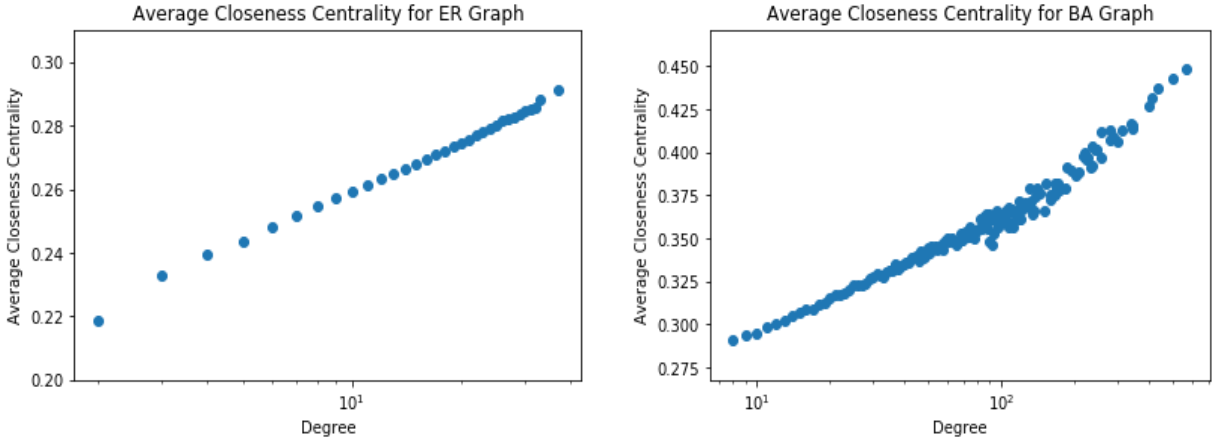


Figure 13. Average closeness centrality for ER and BA networks

3. Community Discovery

In this section we used different community discovery algorithms to identify communities. At first, we used our crawled network in community discovery, but, it took a very long time without any result (28 hours) for K-Clique and Girvan-Newman algorithms and we stopped the operation. Then, we exported a sample of our graph based on the weights. We removed all the edges having weights lower than 60. We used this network as the sample of G_{HT} in this section and we call it G_{HTS}. Figure 14, shows the degree distribution of G_{HTS} which has 955 nodes and 5425 edges.

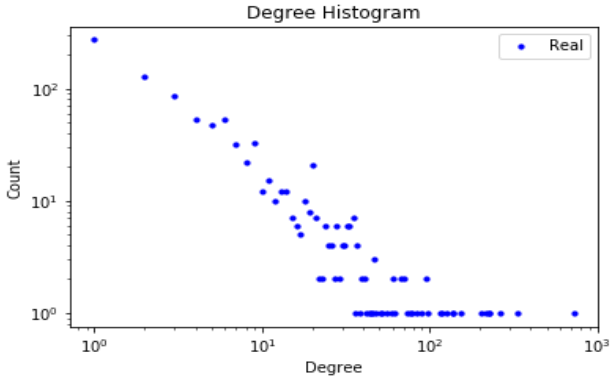


Figure 14. Degree distribution for hashtag network

In the following, we report the results of different community discovery algorithms along with the evaluation results.

3.1. K-Clique (KC)

As a first approach to identify communities, we used the k-Clique algorithm. We performed this experiment with 3 different values for parameter k, i.e., 3, 4 and 5. The results are shown in the table below. The best community structure with respect to modularity score is for k=3, which leads to 10 communities. However, the modularity score is very low compared with other methods. This partition provides a high internal edge density which equals to 0.22. We used the results of k=3 as the best results obtained by K-Clique method for computing the NF1 measure.

k	# Communities	time	IED	AND	MOD	CON
3	**10**	**1.408**	**0.225596**	**4.154671**	**-0.141082**	**0.775873**
4	13	1.114	0.229570	5.572970	-0.169055	0.871306
5	13	1.040	0.228584	6.562284	-0.201537	0.844977

3.2. Label Propagation (LP)

The second algorithm for community discovery is Label propagation which gives 13 communities. The modularity score is very low (almost 0) and shows that the quality of communities are not good.

# Communities	time	IED	AND	MOD	CON
13	0.050	0.231013	3.044641	0.007324	0.347779

3.3. Louvain (LO)

Louvain has the best modularity score among all other methods with a modularity of 0.38 which resulted in finding 15 communities.

# Communities	time	IED	AND	MOD	CON
15	0.140	0.156469	6.270793	0.381612	0.348484

3.4. Girvan-Newman (GN)

The Girvan-Newman algorithm took a considerable time to finish (6531 seconds). The result is a dendrogram which we should decide the cut level. To do this, we cut the dendrogram from all of the levels and computed the modularity score for the resulting partition. Figure 15 shows the value of modularity for each level. Starting from level 0, which results a partition composed of 2 communities, the modularity increases by increasing the level of cut. Then at level 714th, the partitioning has the highest modularity of 0.13, which gives the highest number of communities among all other methods. After that level, by increasing the level, modularity decreases. We report the results of Girvan-Newman algorithm for cut in level 714th in the following table.

# Communities	time	IED	AND	MOD	CON
716	**6531**	0.239103	3.506931	0.13234	0.583337

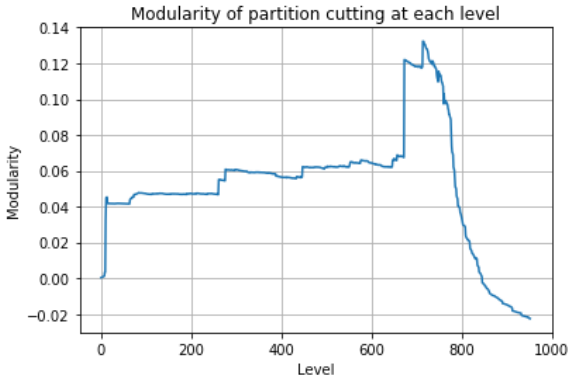


Figure 15. Value of modularity for each level

3.5. DEMON (DM)

As the last method, we used DEMON algorithm with different values for epsilon, i.e., 0.2, 0.4, 0.6, 0.8 and 0.95. In all of the iterations, we used 3 as minimum number of communities. The best modularity is obtained with eps = 0.95 which is not very high and leads to finding of 100 communities. We used the results of eps = 0.95 as the best results obtained by DEMON method for computing the NF1 measure.

We should note that, we executed the algorithm for each epsilon multiple times. We witnessed great variations in the modularity score for the same epsilon. Finally, we reported a stable (mostly seen) modularity score for each epsilon.

eps	# Communities	time	IED	AND	MOD	CON
0.2	8	2.758	0.041268	14.928566	-0.077668	0.431208
0.4	17	2.797	0.104181	13.420623	-0.259895	0.580462
0.6	37	2.775	0.141992	14.657617	-0.03669	0.721912
0.8	71	2.804	0.153736	13.459804	-0.009818	0.758772
0.95	**100**	**2.871**	**0.178599**	**12.491645**	**0.005437**	**0.816354**

In this section we compared the community structure obtained from different methods using the NF1 measure. The results are shown in the following table.

Method1	Method2	Ground Truth Communities	Identified Communities	Community Ratio	Ground Truth Matched	Node Coverage	NF1	F1 Mean
KC3	LP	13	10	0.76	0.46	0.71	0.14	0.51
KC3	LO	15	10	0.66	0.33	0.71	0.05	0.33
KC3	GN	716	10	0.01	0.01	0.71	0.01	0.77
KC3	DM	100	10	0.10	0.03	0.02	0.007	0.84
LP	LO	15	13	0.86	0.46	1	0.09	0.39
LP	GN	716	13	0.01	0.01	1	0.01	0.86
LP	DM	100	13	0.13	0.07	1.43	0.02	0.61
LO	GN	716	15	0.02	0.02	1	0.01	0.70
LO	DM	100	15	0.15	0.10	1.43	0.02	0.40
GN	DM	100	716	7.16	0.96	1.43	0.004	0.035

Finally, in Figure 16, we visualized the network graph used for community discovery and colored the nodes based on partitions obtained by the Louvain method. Although there are some uncertainties, but we can observe several obvious communities. The biggest community (purple) belongs to English tweets and the hashtags are mostly in English, such as “football”, “worldcup” and “brazil”.

The second biggest community is the green one, which is mostly composed of Spanish hashtags, such as “vamos mexico” and “nada nos detiene” which mean “go mexico” and “nothing stops us”, respectively. Since the speaking language of Mexico is Spanish, it is not surprising that most of the hashtags are related to the Germany-Mexico match.

The other community (gray) contains Portuguese hashtags, such as “copa2018”, “copa do mundo”, “vai Brasil” which mean “cup2018”, “world cup” and “go Brazil”, respectively. This community is mostly related to the match of Brazil-Switzerland.

Similarly, the blue community contains the hashtags related to Germany and also contains some French hashtags. The orange community, mostly contains the hashtags related to the Costa Rica-Serbia match, but also has some unrelated hashtags.

The other smaller communities are related to some unsuccessful attempts of trending unrelated hashtags by attaching them to the World Cup-specific hashtags. For example, the dark pink community is mostly about the Bitcoin and Blockchain technologies.

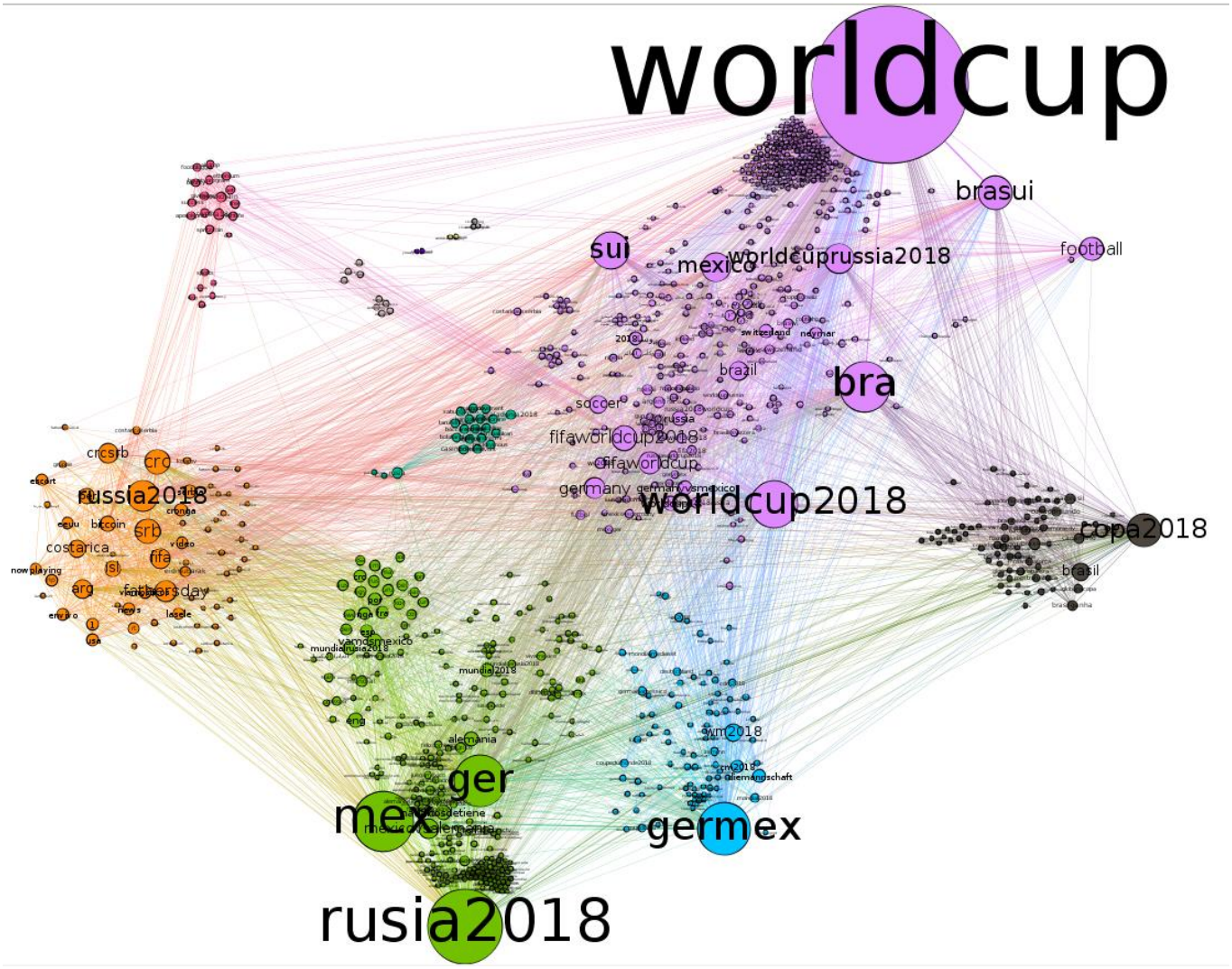


Figure 16. Communities identified by Louvain

4. Tie Strength

In this section we tested several scenarios to observe the resilience of our network to intentional attacks. Our attack scenarios is divided into two categories of node removal and edge removal attacks. In both cases we measured the number of connected components and the size of the giant component in the graph after each of the 20 removals (iterations). We report the node removal scenarios in section 4.1 and edge removal scenarios in section 4.2.

4.1. Node Removal Scenarios

The first scenario was removing nodes by degree centralities. In each iteration, we remove the node with the highest degree (hub) in the remaining graph.
The second scenario, is the removal of nodes based on node strengths which is defined as the sum of edge weights incident to the node. Again, we removed the node with highest strength in each iteration.
In our crawled graph, the edge with the highest weight is ('ger', 'mex'). Thus, there is a slight difference between the removals based on the first 2 scenarios. The list of nodes removed consecutively for both scenarios is as follows:
With degree centrality:
worldcup , rusia2018 , worldcup2018 , germex , mex , ger , bra , football , russia2018 , germany , fathersday , mexico , sui , brasui , worldcuprussia2018 , copa2018 , fifa , brazil , soccer , srb

With Strength:
worldcup , mex , rusia2018 , bra , germex , ger , worldcup2018 , brasui , copa2018 , mexico , sui , russia2018 , crc , worldcuprussia2018 , crcsrb , bitcoin , football , fathersday , fifa , srb

As the third scenario, we tested the effect of removing based on decreasing betweenness centrality of nodes. Again the result is not much different than the previous scenarios. Figure 17 illustrates the results of testing network robustness under different node removal scenarios. Also, we plotted the behavior of random node removal in figure 18. We can observe that, comparing to random removal, the three scenarios can break down the network in a faster way. In fact, in the first three scenarios, we have detailed information about the network and can perform intentional attacks which are more effective. In case of no information about the network, just the random removal can be done which is not effective and in fact does not fragmentize the network.

4.2. Edge Removal Scenarios

In this section, we designed three different scenarios to attack a network by removing edges. In the first scenario, in each iteration we remove the edge with highest weight. In second scenario, we remove by the highest edge betweenness centrality and in the third scenario, we remove by the overlap measure. None of the first two scenarios can affect the network in terms of number of connected components and the size of the giant component. In fact, the removed edges were

between the high degree hubs of the network, so removing them could not break apart the system, or even decrease the number of nodes.

The third scenario was a bit different, and we've selected the edge with the lowest overlap value (which acts as a bridge) in each iteration. We can see a linear relation between the number of nodes removed and the number of connected components in the network. This means, removing the edge in each iteration divides the giant component into two separate components. Actually, this division takes place in the most outer edges of the network, where the edge connects a degree-1 node to the rest of the network. Thus, after each removal the size of the giant component decreases by 1. Figure 19 shows the behavior of three edge removal scenarios.

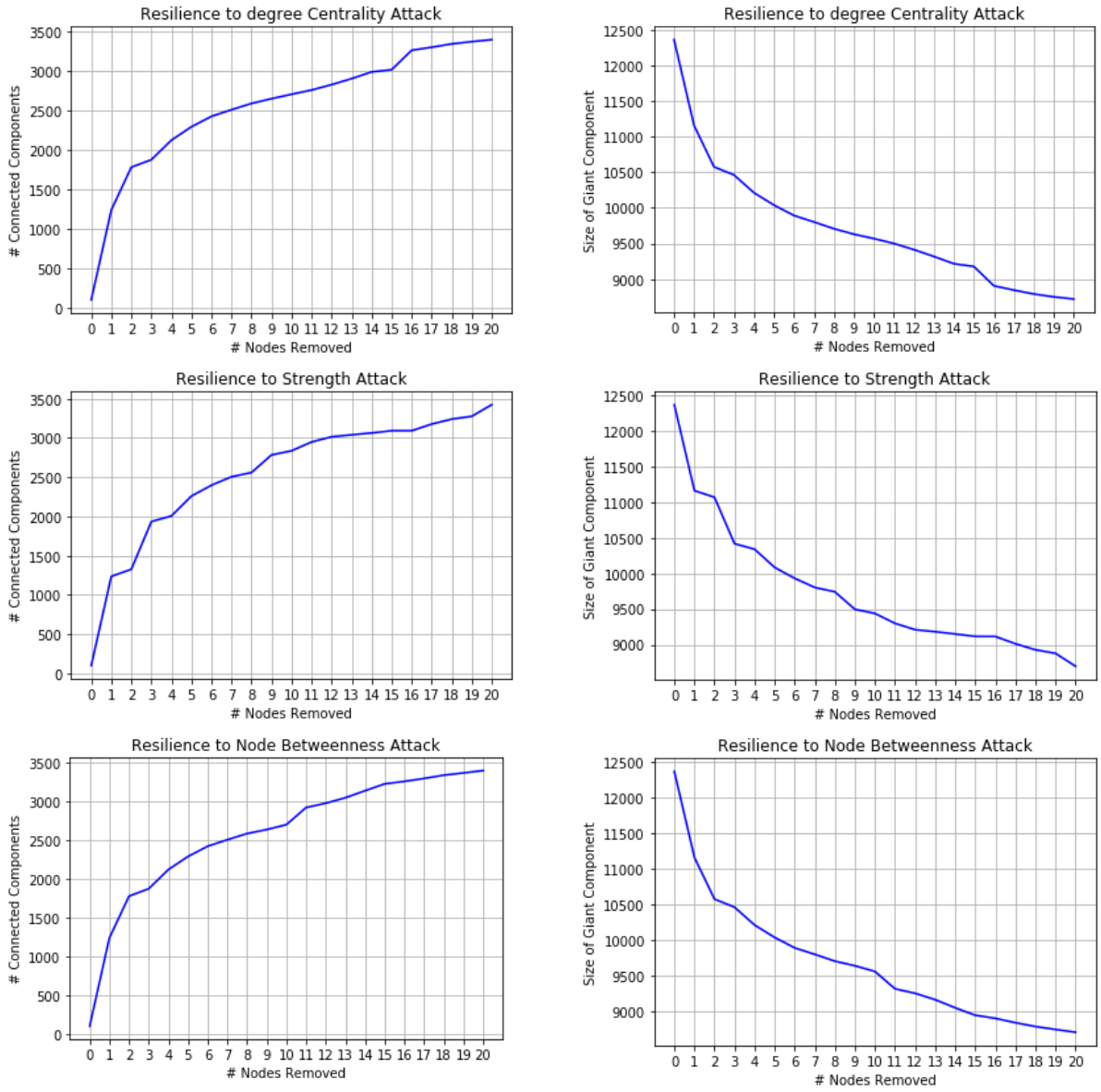


Figure 17. Intentional Node Removal Scenarios

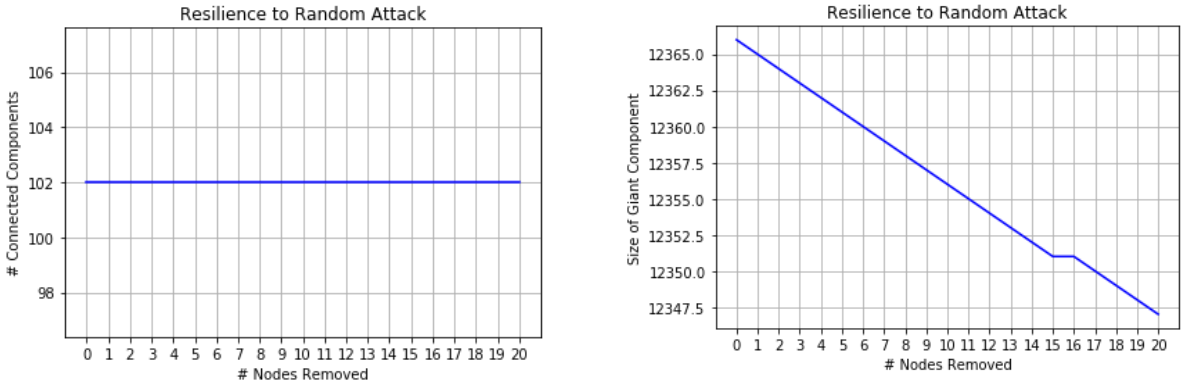


Figure 18. Random Removal Scenarios

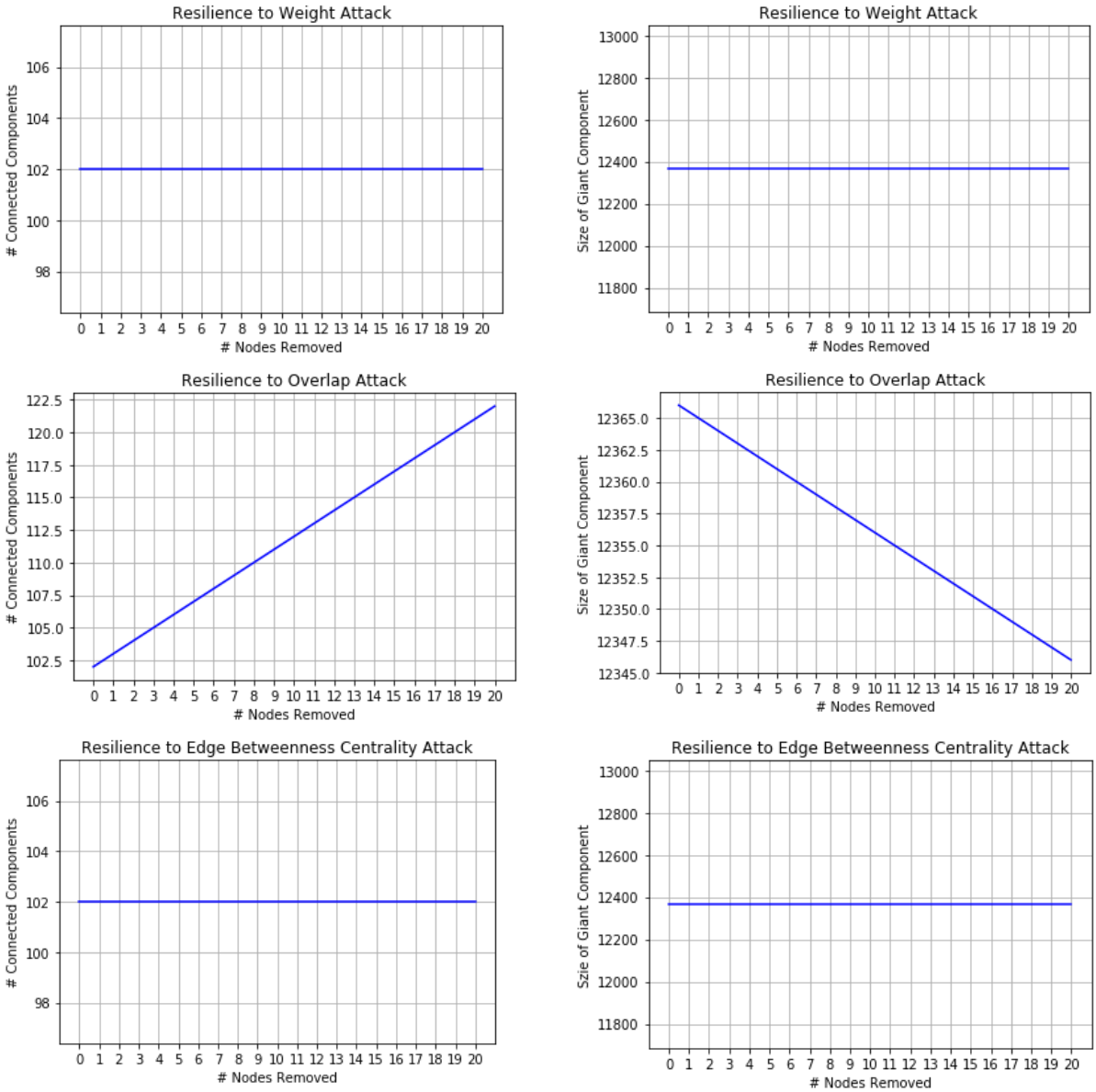


Figure 19. Edge Removal Scenarios

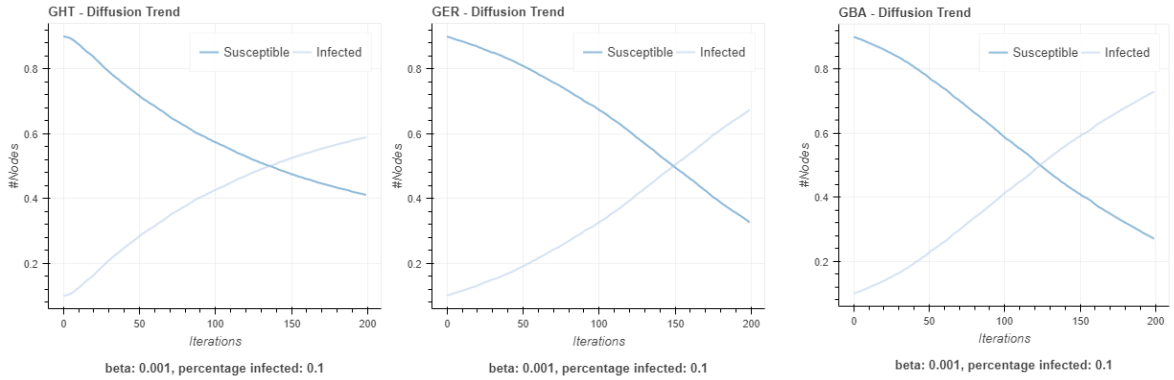
5. Spreading

In order to model the propagation of a phenomenon such as information or a virus in a network, spreading models are used. Actually, our network is not suitable for studying the spreading of information in twitter, but we applied the functionalities suggested by the Network Diffusion Library (NDlib) in python to model the spreading. Particularly, we investigated the dynamics of the three commonly used epidemic models, i.e., SI, SIS and SIR models.

5.1. SI Model

Based on this model, once a node is infected it will remain in this condition permanently. Finally, the whole network is going to be infected after a specific time. Additionally, in the early phases of the epidemic, the fraction of infected nodes increases exponentially. The graphical representation of the simulation of SI model on G_{HT}, G_{ER} and G_{BA} is provided in Figure 20.

The rapidness of the spread depends on the network's average degree and the probability (β) that an infected node will convey the pathogen to a susceptible one. At first, we set beta to 0,001 and as can be seen in the figures, the infection is very slow for three networks. Then, we set beta to 0.01 and the speed of infection increased rapidly. Particularly, for G_{HT} in less than 15 iterations, the number of infected nodes equals that of susceptible ones but it took a long time for infection to reach out the entire network. This is because there are a large number of low-degree nodes in G_{HT}. But for G_{ER} and G_{BA}, after less than 60 iterations, the network got completely contaminated.



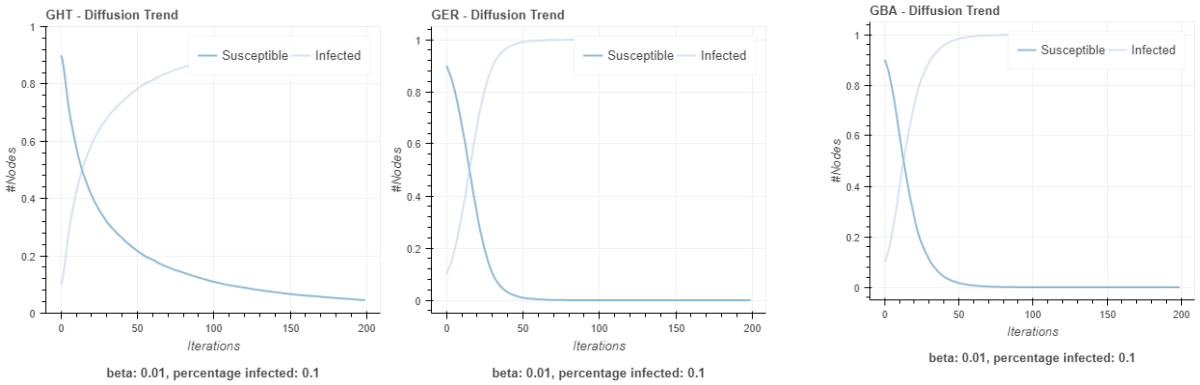
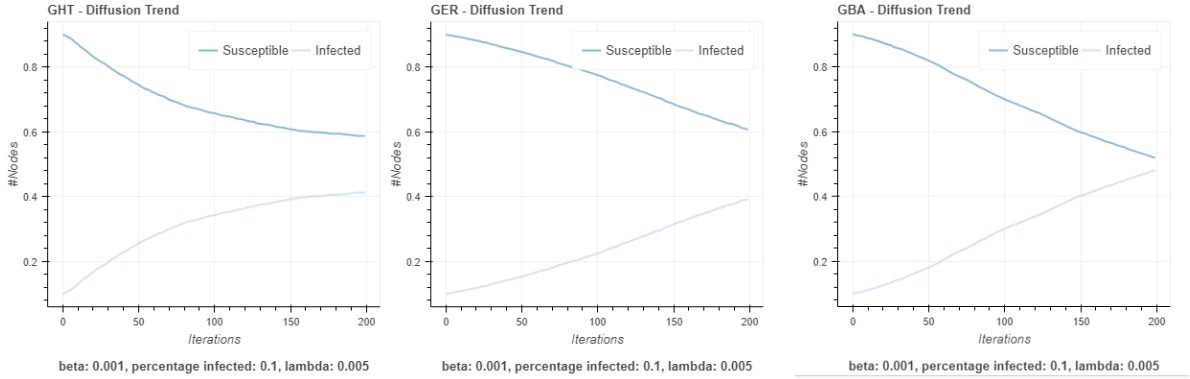


Figure 20. SI Model

5.2. SIS Model

The difference between the SI and the SIS model is that in the SIS model nodes may recover from the disease and become susceptible again. Thus, in addition to beta, which is the infection rate, we have a lambda which is the recovery rate. Depending on the recovery rate, there could be either an endemic (a finite fraction of the nodes is infected) or a disease-free state. With a constant beta, we changed lambda from 0.005 to 0.03. For lambda = 0.005 the system is in the endemic state and the infected fraction of individuals does not change with time after a while.
Then we changed lambda to 0.03, and we can observe that the system is in disease-free state and the infection disappears after some time. Figure 21 shows the results of experiments.



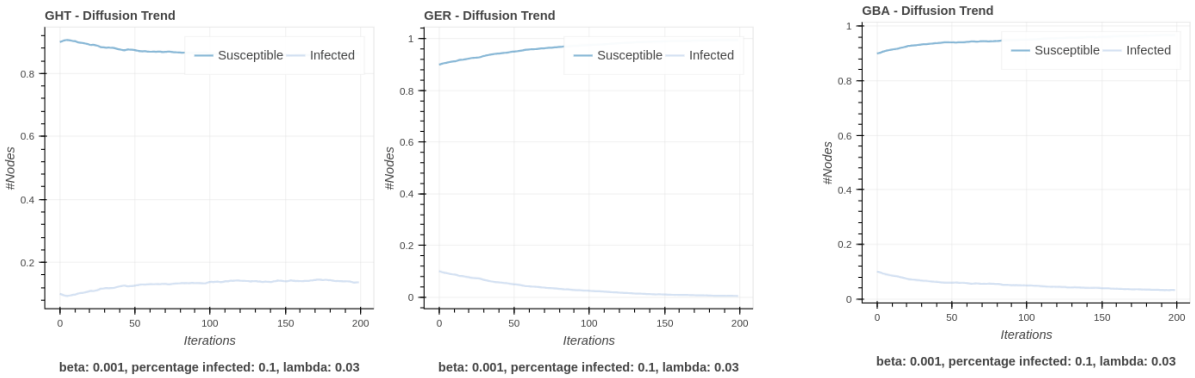


Figure 21. SIS Model

5.3. SIR Model

In this model, nodes develop immunity after they recover from the infection. Instead of returning to the susceptible state, they are “removed” from the population of nodes to be considered. Also, the fraction of infected nodes drops to zero. Figure 22 illustrates the simulation of SIR model on the crawled data and two random networks. With constant beta, we doubled gamma from 0.005 to 0.01. For gamma = 0.005, the infection starts to vanish around iteration 55, but by increasing the gamma, the vanishing starts earlier, around iteration 40.

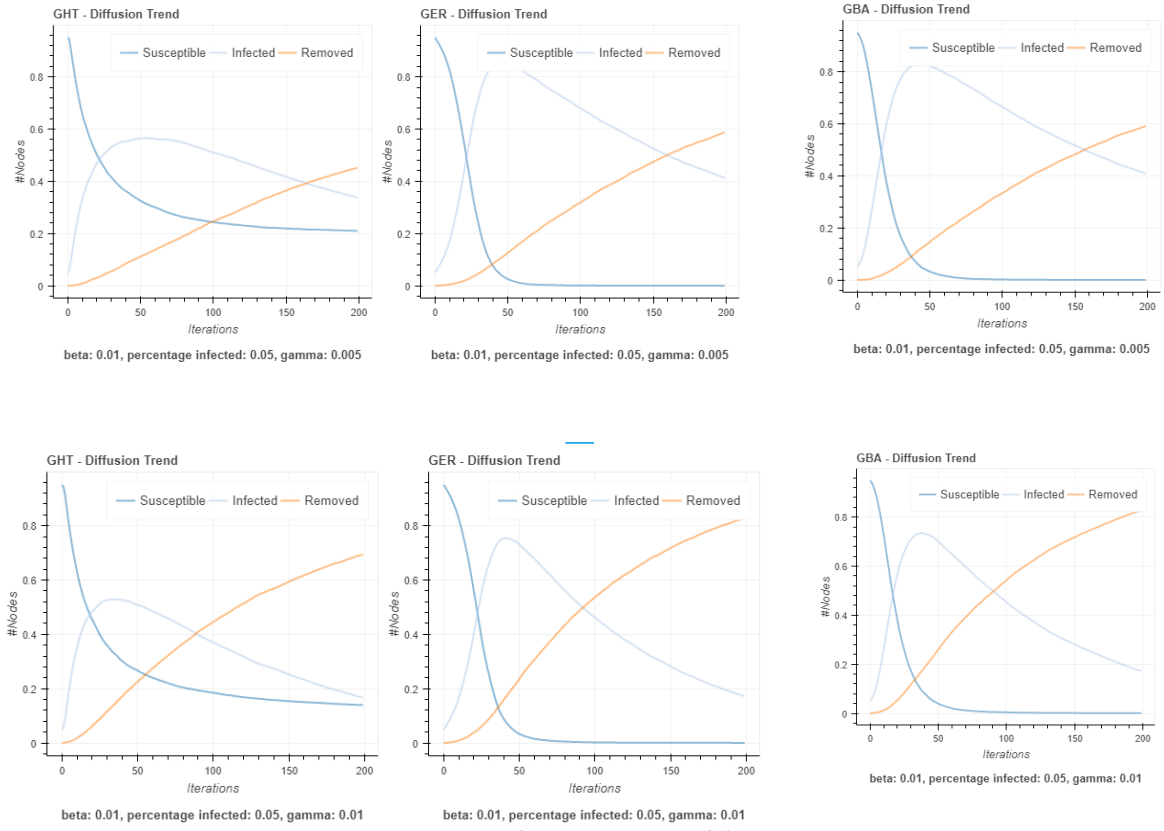


Figure 22. SIR Model