Network Analysis of the Madrid Train Bombing: Investigating the Terrorist Connections

Lucrezia Ceresa, 839050

July 15, 2024

Contents

1 Introduction

The objective of this paper is to analyse the graph representing the terrorist network involved in the Madrid bombings of March 11, 2004, also known as 11-M. These attacks, of Islamic origin, were perpetrated against several local trains in the Spanish capital, causing 192 deaths (177 of them in the immediate aftermath of the attacks) and 2,057 injuries. The attacks are considered among the most serious against civilians in Europe since the Second World War, along with the attacks in Paris on 13 November 2015.

1.1 Description of attacks

On the morning of Thursday, March 11, 2004, three days before the general elections, ten backpacks filled with explosives exploded on four regional trains in Madrid, at four different stations. The explosions occurred during rush hour, between 7:36 and 7:40, at the Madrid stations of Atocha (3 bombs), El Pozo (2 bombs), Santa Eugenia (1 bomb), and on a fourth train near Téllez Street (4 bombs) on the tracks leading to Atocha from the south.

The police found two additional unexploded devices, which were immediately detonated by bomb squads for safety reasons. Another bag containing 500 grams of explosives, shrapnel, a detonator, and a timer based on a modified mobile phone was found unexploded among the objects and luggage collected from the attack sites. This bag, transported to a police station and later to a trade fair center (IFEMA) along with the victims, was crucial for the investigation. This artifact, initially placed on the Vallecas train, quickly led to the first concrete hypotheses and arrests on March 13.

There were doubts about responsibility for the attacks. The two organizations suspected were ETA (Basque terrorist organization) and Islamic terrorist cells linked to Al Qaeda. On July 17, 2008, the Supreme Court confirmed the Islamist matrix of the terrorist group.

1.2 The Graph

The Train Bombing network contains contacts between suspected terrorists involved in the train bombing of Madrid on March 11, 2004 as reconstructed from newspapers. A node represents a terrorist and an edge between two terrorists shows that there was a contact between the two terrorists [Mor17]. The edge weights (from 1 to 4) denote how "strong" a connection was.

1. Trust–friendship (contact, kinship, links in the telephone center);
2. Ties to Al Qaeda and to Osama Bin Laden;
3. Co-participation in training camps and/or wars;
4. Co-participation in previous terrorist Attacks.

It can be seen in the Figure 1 that the weights distribution is unbalanced. It seems that the weight 1 is the most present. To be precise, the percentages of the weights are shown in the pie chart 2.

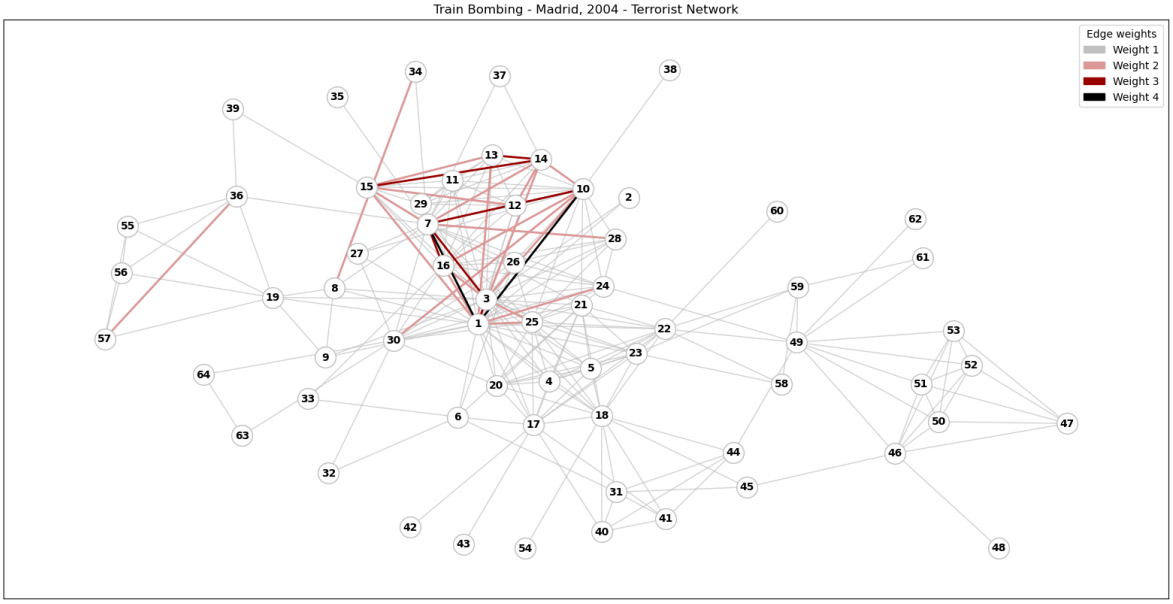

Figure 1: Graph of the terrorists involved in the Madrid attack on March 11, 2024.

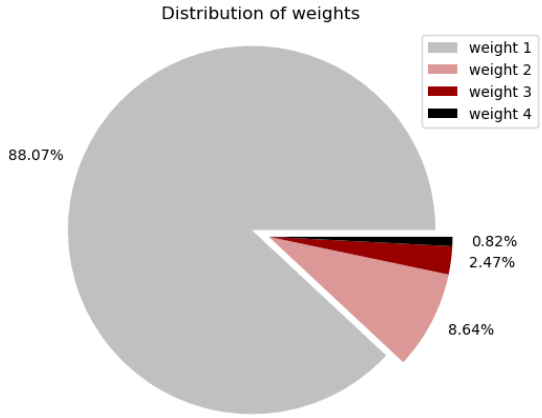

Figure 2: Distribution of weights.

2 Measurements

In this section, we present and describe the most important measures of the network. To begin, we summarize them in the Table 1.

Measure	Value
Number of nodes	64
Number of edges	243
Graph is connected	True
Maximum degree	29
Average degree	7.593750
Number of singletons	0
Density	0.120536
Max strength	43
Average strenght	8.810000
Number of cycles	180
Average path length (APL)	2.690972
Diameter	6
Number of triangles	527
Clustering coefficient	0.561036

Table 1: Network measures.

2.1 Distances

The distance distribution (Figure 3) resembles a Gaussian distribution, characterized by the mean and standard deviation of the vector of all distances (excluding zero distances).

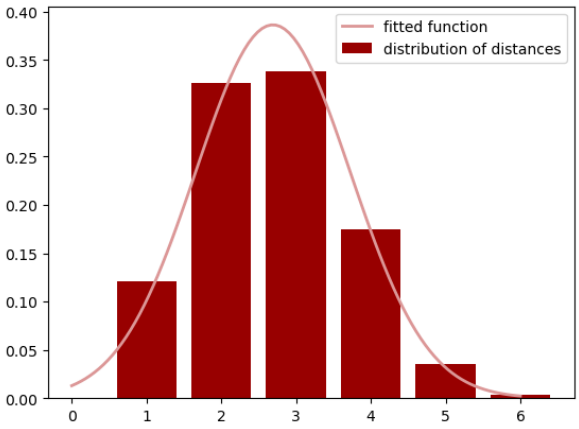

Figure 3: Distance distribution.

The graph's average path length is approximately 2.7, as indicated in Table 1. We describe a graph as having short paths (or exhibiting a "small-world" phenomenon) if the average path length grows very slowly with the size of the network, say logarithmically. For our network with $N = 64$, where N represents the number of nodes, $\log N \approx 4.16 > 2.7$. Now, we consider one node at a time, adding it to a new empty graph. Each time a node is added, its associated edges are also included. This process increases the graph size by 1 at each iteration, and the average path length is recalculated. The plot in Figure 4 shows that the increase in the average path length, relative to the network size, does not grow faster than the logarithm of the size. Therefore, we can conclude that the network exhibits the small-world phenomenon.

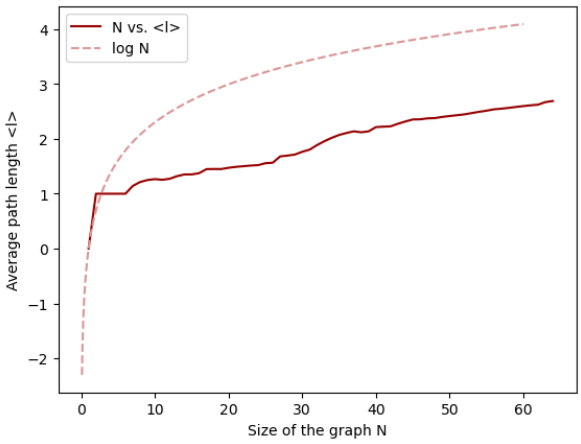

Figure 4: $\log N$ grows faster than the average path length

2.2 Degree and Strength

Since the average degree is approximately 7.6 and the maximum degree is 29 (as shown in Table 1), it indicates the potential presence of hubs within the network. For this reason we plot the degree distribution in the Figure 5.

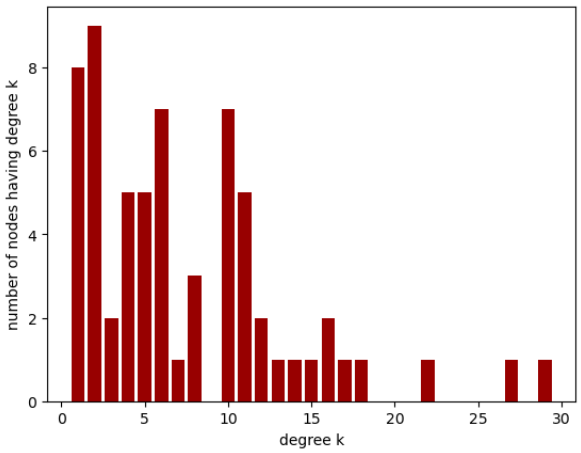

Figure 5: Degree distribution.

The degree distribution indicates that there are three degree values significantly higher than the other. Therefore, we aim to identify the labels of the nodes with these degrees, as they can be considered hubs.

The following Python code identifies the nodes with the highest degrees.

```
In [1]:
# Determine the three highest degrees
max_1 = max(dict(G.degree).values())
max_2 = max([d for d in dict(G.degree).values() if d<max_1])
max_3 = max([d for d in dict(G.degree).values() if d<max_2])
three_max = [max_1, max_2, max_3]
# Display the dictionary where keys represent the degree and values are lists of
# nodes corresponding to that degree
{m : [n for n in G.nodes if G.degree(n) == m] for m in three_max}

Out [1]: {29: [1], 27: [3], 22: [7]}
```

The provided output indicates that the main hubs in the network are nodes 1, 3, and 7 with degree 29, 27 and 22 respectively. Let's examine them closely in the graph: Figure 6 illustrates that the edges connected to these three hubs span a significant portion of the central area of the graph.

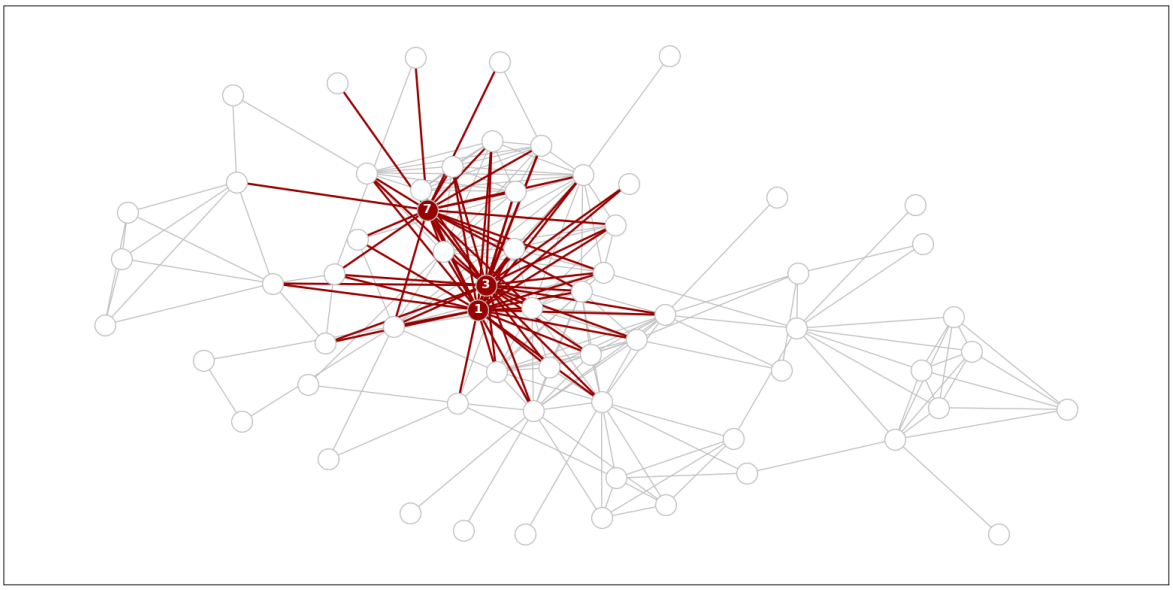

Figure 6: Three nodes with maximum degree (hubs)

The mean and the variance of the degree distribution are respectively:

$$< k >= \frac{\sum_{i \in N} k_n}{N} = 7.6 \qquad \text{and} \qquad < k^2 >= \frac{\sum_{i \in N} k_n^2}{N} = 95.66$$

Then the heterogeneity parameter is:

$$\frac{< k^2 >}{< k >^2} \approx 1.66$$

In this case, the heterogeneity parameter slightly exceeds 1, indicating that the graph is not particularly heterogeneous. Indeed, the March 2024 attack in Madrid was carried out by a local cell inspired by al-Qaeda, but which had not received direct orders from (and so had not direct contacts with) the organization's leaders. The structure of the terroristic network during the period of the attack is changing and is increasingly based on a network of independent cells, disengaged from any form of hierarchy, to avoid tracking al-Qaeda activities. This means that within a terrorist cell the contacts between individuals are rather homogeneous [Taf].

Since our graph is weighted, we compare the degree distribution with the strength distribution. The strength distribution, depicted in Figure 7, follows a similar trend to the degree distribution, but with higher values on the y-axis. To facilitate comparison, let's normalize both distributions and plot them together in the same figure (Figure 8).

In the Figure 9, three power laws with exponents 1.5, 2, and 3 are also plotted. None of these power laws appears to fit the degree distribution, which aligns with our finding of a heterogeneity parameter close to 1. Typically, power laws are indicative of very heterogeneous networks, so the absence of a match supports the observation that our network is relatively homogeneous.

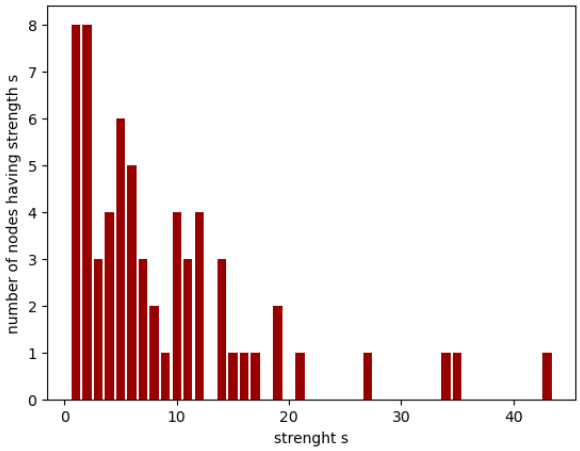

Figure 7: Strength distribution.

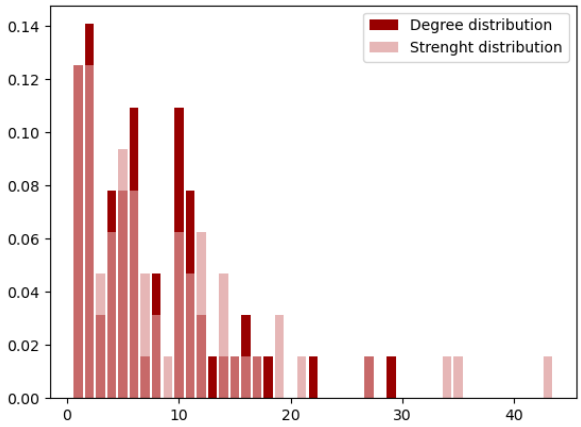

Figure 8: Comparison between degree and strength distribution.

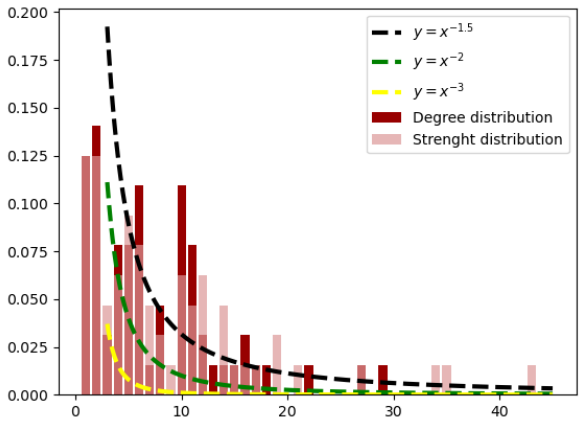

Figure 9: Power laws do not fit the degree distribution.

2.3 Clustering Coefficient

The clustering coefficient gives information about triangles involved by each node. The formula is:

$$C(i) = \frac{\tau(i)}{\binom{k_i}{2}} = \frac{2\tau(i)}{k_i(k_i - 1)}$$

Globally there are 527 triangles, as the Table 1 indicates. Let's look at another table (Table 2), listing the clustering coefficients for each node.

Node i	C(i)	Node i	C(i)	Node i	C(i)	Node i	C(i)
1	0.332512	17	0.466667	33	0.000000	57	1.000000
2	1.000000	18	0.450000	34	1.000000	49	0.218182
3	0.367521	19	0.428571	35	0.000000	58	1.000000
4	1.000000	20	0.872727	36	0.400000	59	0.500000
5	1.000000	21	0.787879	37	1.000000	60	0.000000
6	0.190476	22	0.538462	38	0.000000	63	1.000000
7	0.367965	23	0.727273	39	0.000000	64	1.000000
8	0.600000	24	0.672727	40	0.800000	46	0.500000
9	1.000000	25	0.602941	41	0.800000	47	1.000000
10	0.529412	26	0.822222	42	0.000000	48	0.000000
11	1.000000	27	1.000000	43	0.000000	50	0.933333
12	1.000000	28	1.000000	44	0.600000	51	0.933333
13	1.000000	29	1.000000	45	0.333333	52	0.933333
14	0.836364	30	0.448718	54	0.000000	53	0.933333
15	0.818182	31	0.466667	55	1.000000	61	1.000000
16	0.616667	32	0.000000	56	1.000000	62	0.000000

Table 2: Table of clustering coefficients.

We observe that many values are equal to 1, indicating that every pair of neighbors of a node is connected, forming a triangle. This is not surprising, as triadic closure is very common in social networks (where nodes represent individuals). The Figure 10 shows the distribution of the clustering coefficient and we can notice the prevalence of values equal to 1.

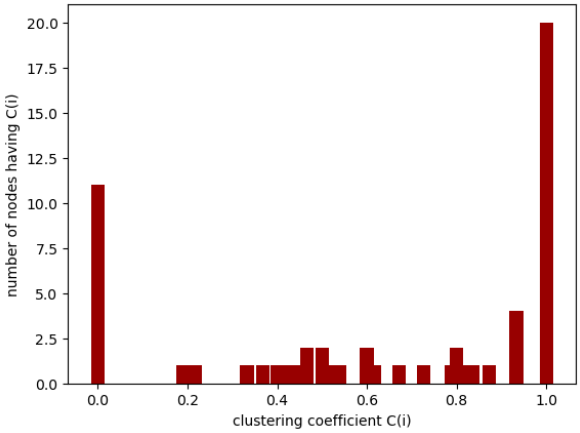

Figure 10: clustering coefficient distribution.

The clustering coefficient of the network reflects the balance between the high occurrence of values equal to 1 and those equal to 0. It is calculated as:

$$C = \frac{\sum_{i \in N} C(i)}{N} \approx 0.56$$

2.4 Connected Components

The graph is connected, meaning the largest connected component is the graph itself. However, what happens if some edges are removed? Will the graph remain connected? Understanding this is crucial, especially since our graph represents a terrorist network. Identifying the most critical edge or node could be a key strategy for law enforcement to disrupt communication and prevent future attacks.

Let's compare two methods to assess the network's robustness: Random Failure and Targeted Attack. In the Random Failure scenario, we remove nodes randomly, one at a time. In the Targeted Attack scenario, we start by removing the nodes with the highest degree. The results are displayed in the plot 11. The Targeted Attack disconnects faster the graph, then it is crucial to knock out terrorist with the highest number of contacts.

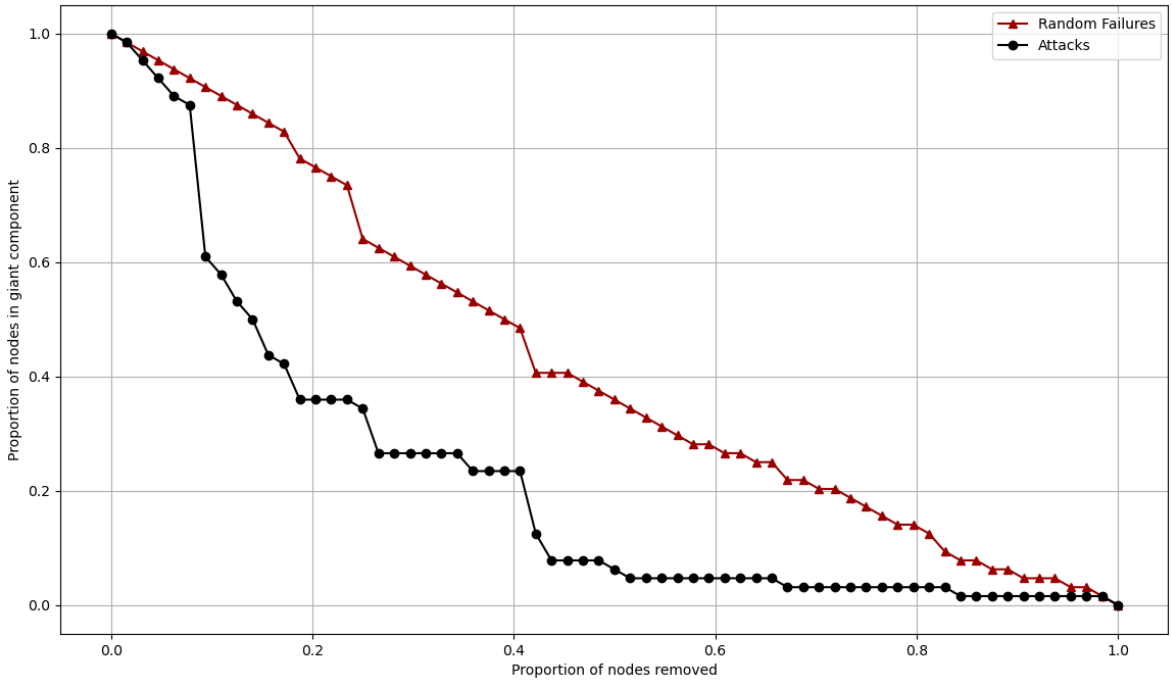

Figure 11: Target Attack disconnects faster the graph.

The methods are based on the idea of removing nodes, how does the graph change if we interrupt edges (that represent the strength of contacts)? First of all, let's identify edges that connect two or more distinct groups of individuals within the network. These critical connections could be key targets for law enforcement to interrupt, as disrupting them might prevent the formation or operation of cohesive groups within the network. This kind of edges are the so-called local bridges.

To look for local bridges, we need to compute the neighborhood overlap. This measure indicates how many neighbors are shared between the two endpoints of an edge. If (A, B) is an edge, where $N(A)$ and $N(B)$ are the set of neighbors of A and B, respectively, then the formula is:

$$O_{AB} = \frac{|N(A) \cap N(B)|}{|N(A) \cup N(B)|}$$

The neighborhood overlap takes a value between 0 and 1. If it is 0, it means there are no shared neighbors, making the edge a local (or possibly global) bridge. If it is 1, it means the two nodes share all their neighbors, indicating they are at the center of a single, tightly-knit group.

While neighborhood overlap indicates the count of edges that are local bridges, embeddedness (the numerator of O_{AB}) simply indicates whether an edge is a local bridge or not (that is, if it is 0 or greater than 0).

The following Python code generates a list of local bridges in the graph.

```
In [2]:
# neighborhood overlap
def neighborhood_overlap(G, a, b):
    neighbors_a = set(G.neighbors(a))
    neighbors_b = set(G.neighbors(b))
    neighbors_a.discard(b)
    neighbors_b.discard(a)
    intersection = neighbors_a & neighbors_b
    union = neighbors_a | neighbors_b
    if not union:
        return 0
    overlap = len(intersection) / len(union)
    return round(overlap, 3)

# local bridge
def find_local_bridges(G):
    local_bridges = []
    for edge in G.edges:
        if neighborhood_overlap(G, edge[0], edge[1]) == 0:
            local_bridges.append(edge)
    return local_bridges

local_bridges = find_local_bridges(G)
local_bridges

Out [2]: [(6, 31), (6, 32), (6, 33), (7, 35), (7, 36),
 (10, 38), (15, 39), (16, 33), (17, 42), (17, 43),
 (18, 54), (22, 60), (24, 49), (30, 32), (36, 39),
 (44, 49), (45, 46), (49, 62), (46, 48)]
```

In addition to local bridges, nodes known as 'structural holes' are also very important. These nodes act as connectors for many local bridges, enabling them to facilitate communication between multiple terrorist groups. In the output of the previous Python code, we observe that certain nodes appear in multiple local bridges, indicating their role as structural-hole nodes. Specifically, nodes 6 and 49 fulfill this role.

The overlap increases with the strength of the nodes, which aligns with the observation that nodes with more neighbors typically have higher neighborhood overlap and strength (since most links in our network are weighted by 1). However, when the strength is very high, the overlap may decrease. This is because only a few nodes have very high strength, and these nodes can have high-weight connections but a low number of neighbors. The plot in Figure 12 show exactly this trend.

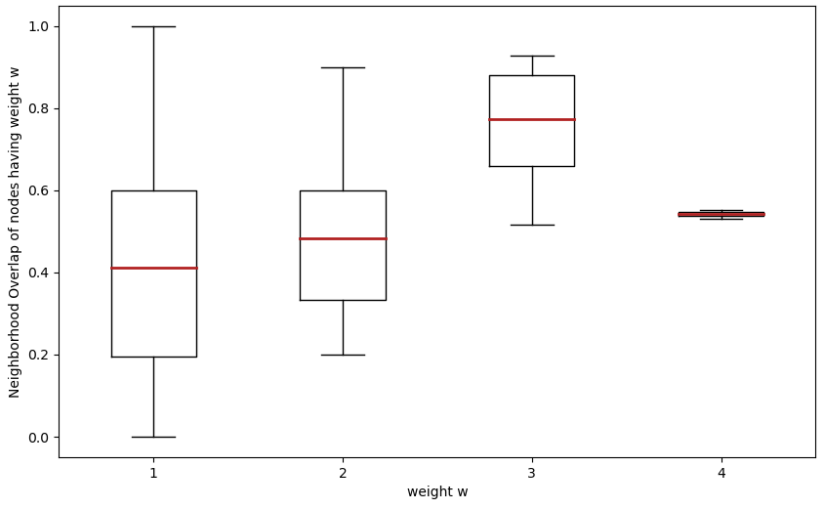

Figure 12: Overlap vs. weights.

2.5 Degree correlation

The objective of this paragraph is to understand the general structure of the graph, which can exhibit either a core-periphery or a hub-and-spoke structure. Observing the graph (Figure 1), it appears that the prevalent structure is core-periphery, although the external nodes also exhibit substantial connectivity.

Assortativity provides a metric to distinguish between these structural types. It measures the correlation between a node's degree and the average degree of its neighbors, denoted as $k_{nn}(k)$ where k is the degree. This relationship is visually represented by plotting $(k, k_{nn}(k))$ in Figure 13.

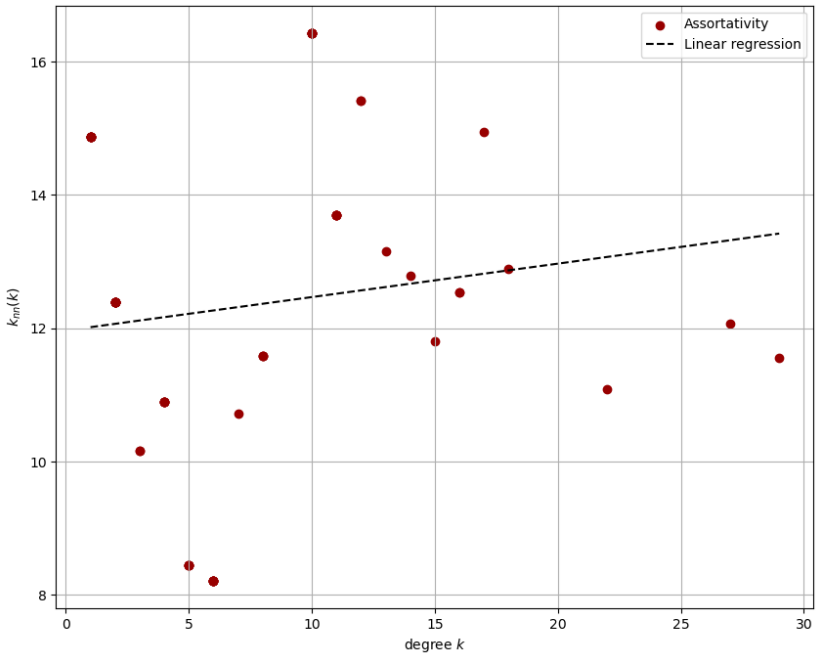

Figure 13: Average of the neighbors of nodes of degree k vs. k

The fitted line if Figure 13 has a slope of 0.05, indicating a very low positive correlation. This suggests that the graph exhibits behavior more similar to core-periphery, even if it is approaching a neutral degree correlation. Indeed, smaller social networks, such as the one we are analyzing, are characterized by a large number of triadic closure and, as a result, also numerous connections between nodes. Moreover, this structure is in line with that of terrorist cells, in which members are very closely linked.

2.6 Communities

Communities are groups of nodes that exhibit high cohesion within the group and significant separation from nodes outside the group. It is important to determine whether the graph analyzed in this paper exhibits such community structures. They can represent terrorist cell nuclei.

Let's begin with the Kernighan-Lin algorithm. This algorithm generates a bisection with a local minimum cut size (the number of edges connecting the two communities). Since this minimum is local, it is necessary to run the algorithm multiple times to increase the likelihood of selecting the partition with the lowest cut size. The Figure 14 shows nine possible outcomes of the Kernighan-Lin algorithm. The cut sizes are printed above each trial and we can observe that the minimum found is 30.

The Girvin-Newman is a divisive hierarchical clustering. In each iteration the edges with the highest betweenness are removed until all the nodes are disconnected and each one forms a cluster. The

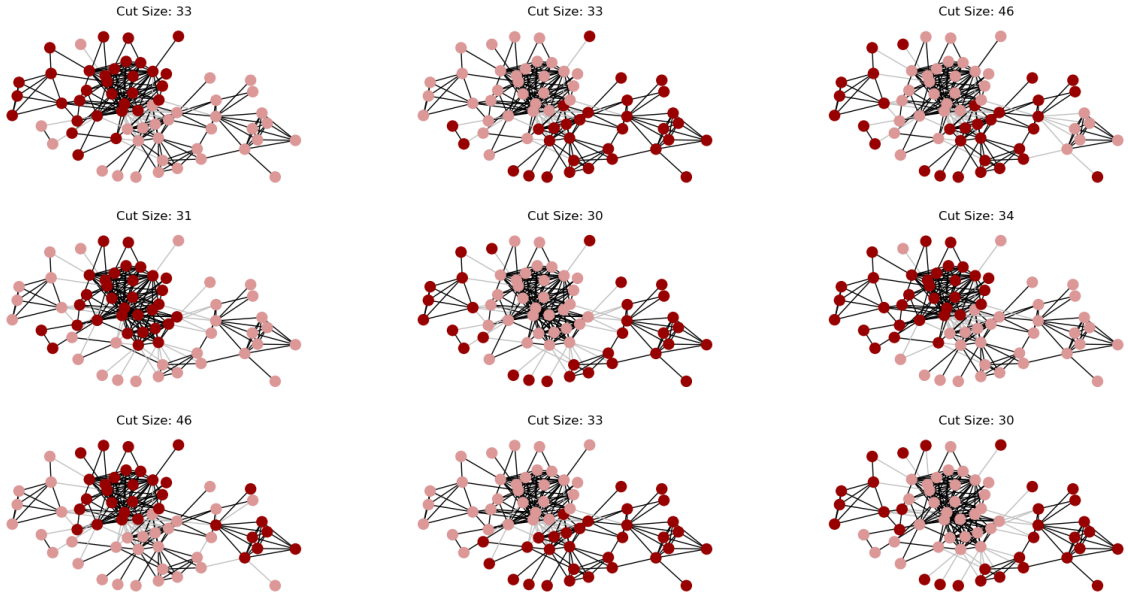

Figure 14: Kerningham-Lin is performed nine times and the cut size is printed

output is the dendrogram, shown in Figure 15.

To determine the optimal number of clusters, we can plot the modularity. Modularity is a quality measure that compares the number of links (or the strength in the case of our weighted graph) within the communities found by an algorithm (the Girvan-Newman one in this case) to those in a random network (one without communities). Modularity ranges from negative values to less than one, with a value of zero indicating the entire graph forms a single community. Our goal is to find the maximum modularity value between 0 and 1. The plot in Figure 16 illustrates the variation of modularity with respect to the number of clusters. The maximum modularity is achieved when the graph is divided into 15 clusters.

The principles of modularity and the Girvan-Newman algorithm are combined in Newman's greedy algorithm, whose goal is to merge the pairs of groups of nodes that yields the highest increase (or lowest decrease) of the modularity. Now, we aim to compare the differences between the results of these methods. The two graphs shown in Figures 17 and 18 are quite similar. The fifteen clusters are subsets of the six larger clusters. The optimal partition is the one with six clusters, as the partition with fifteen clusters includes many clusters that are too small.

Let's compare the six-cluster partition with the results from the Louvain and label propagation algorithms. The Figure 19 shows the clustering obtained using the Louvain algorithm, which results in only four clusters. We can observe that these four clusters encompass the six clusters from the previous partition. The main difference with the label propagation output, shown in Figure 20, is the merging of the clusters previously colored blue and green into a single green cluster.

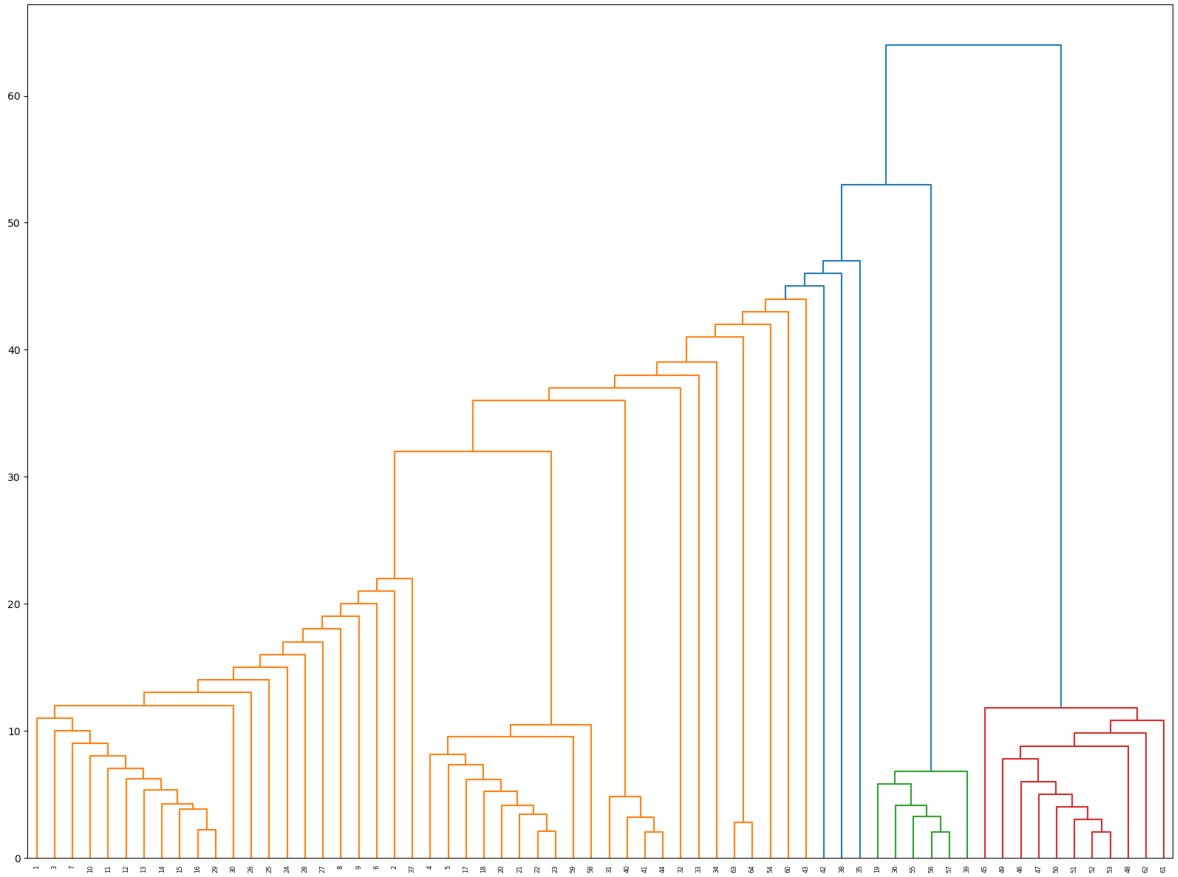

Figure 15: Dendrogram of the Girvin-Newman output.

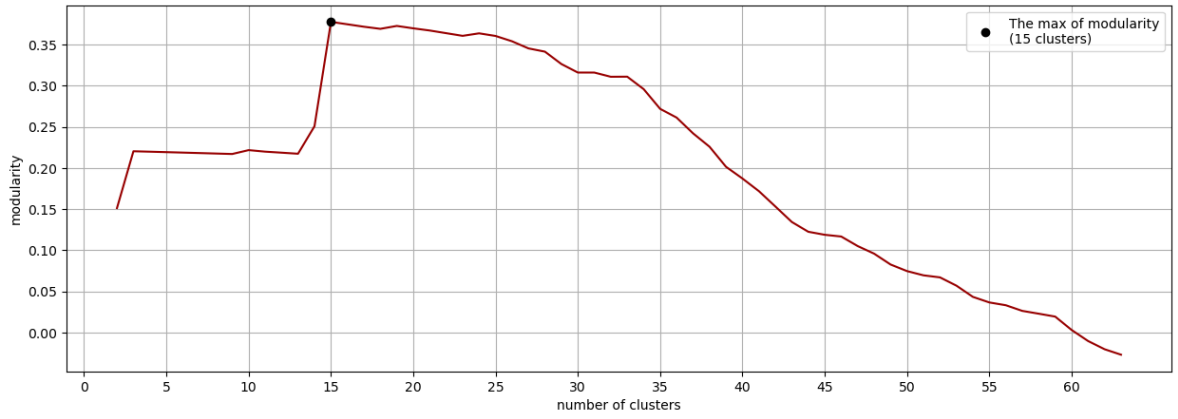

Figure 16: Modularity vs. number of clusters.

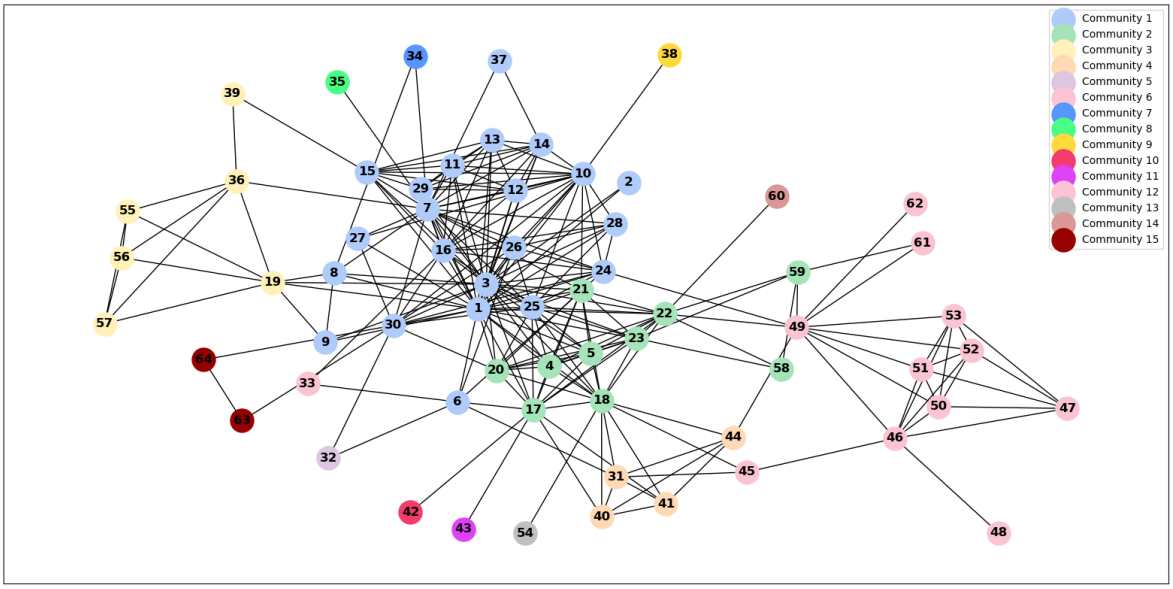

Figure 17: Output of Girvan-Newman optimizing the modularity.

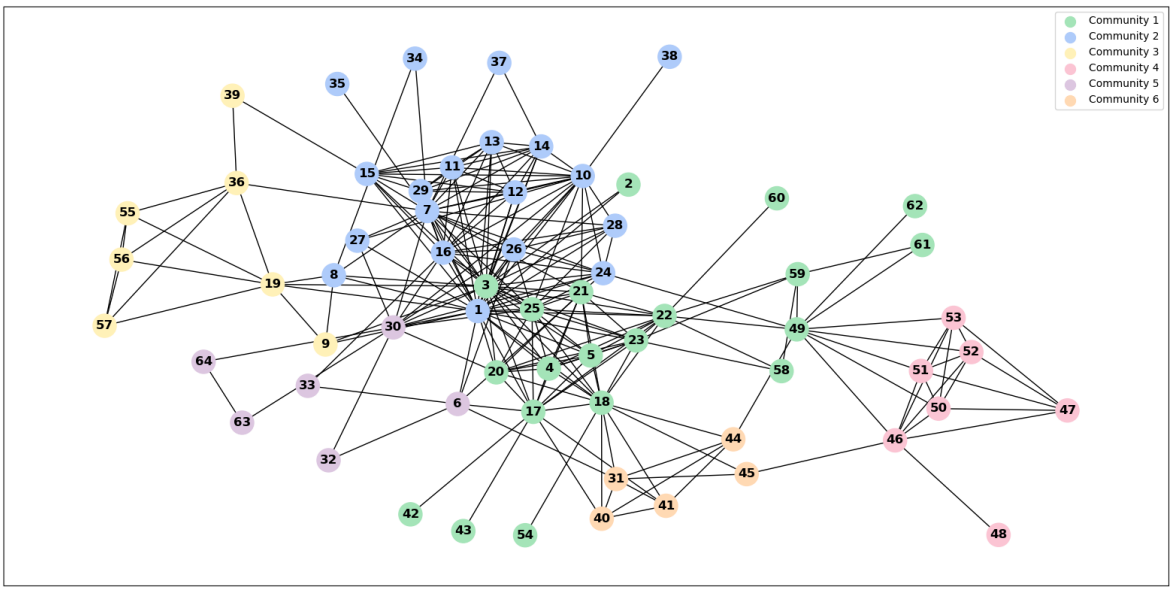

Figure 18: Output of Newman's greedy algorithm.

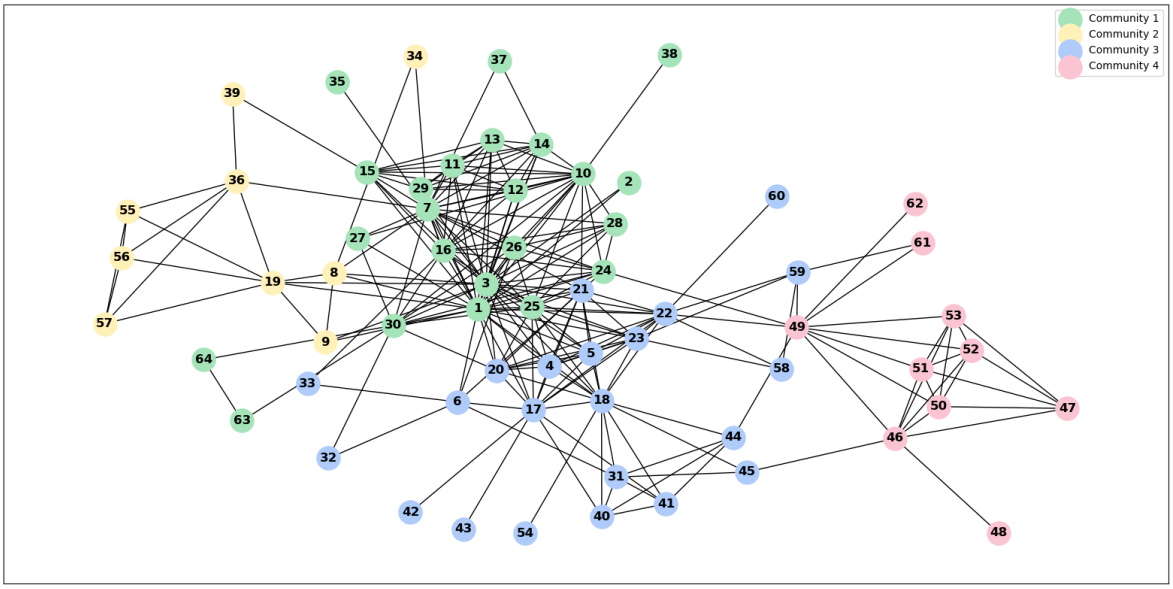

Figure 19: Output of Louvain algorithm.

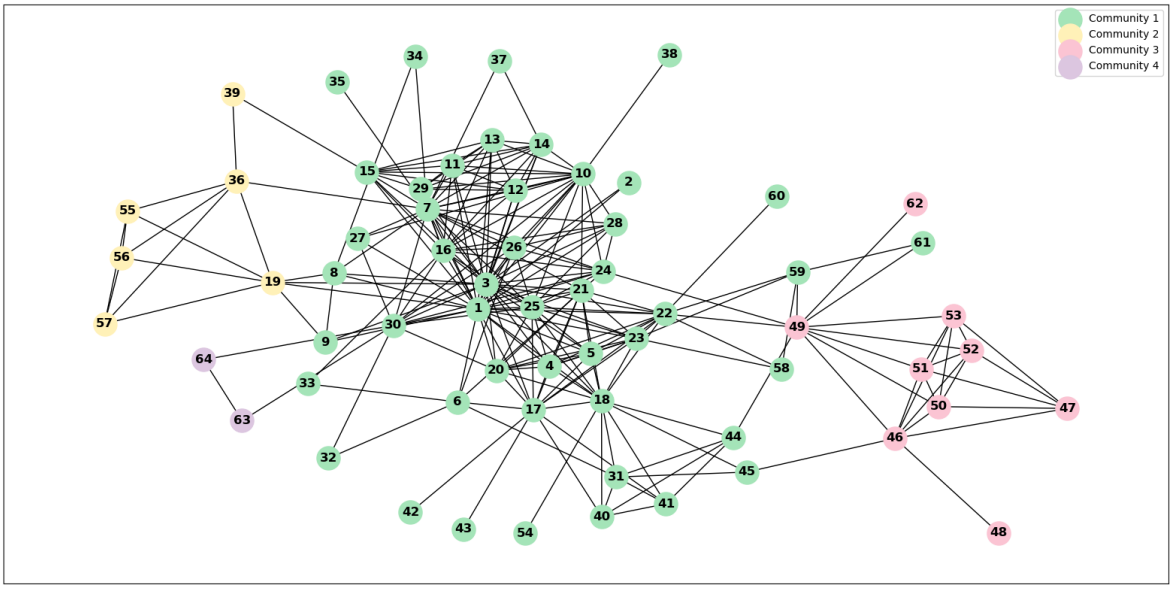

Figure 20: Output of label propagation algorithm.

2.7 Centralities

We have previously demonstrated in Figure 6 that nodes 1, 3, and 7 can be considered hubs due to their highest strength values. This measure indicates how "central" a node is in a graph. Besides strength, there are other ways to compute the importance of a node, such as closeness and betweenness.

Closeness centrality is based on the idea that a node is more central if it is, on average, closer to all other nodes. The closeness g_i of a node i is calculated as follows:

$$g_i = \frac{1}{\sum_{j \neq i} l_{ij}}$$

where l_{ij} is the distance (the shortest path length) between nodes i and j. The Figure 21 illustrates the closeness centrality for each node. and the distribution of the shortest path length.

Betweennes centrality is based on the idea that a node is more central the more often it is crossed by paths. The betweennes b_i of a node i is calculated as follows:

$$b_i = \sum_{h \neq j \neq i} \frac{\sigma_{hj}(i)}{\sigma_{hj}}$$

where σ_{hj} is the number of shortest paths from h to j and $\sigma_{hj}(i)$ is the number of shortest paths from h to j running though i. The Figure 22 illustrates the betweenness centrality for each node. and its distribution.

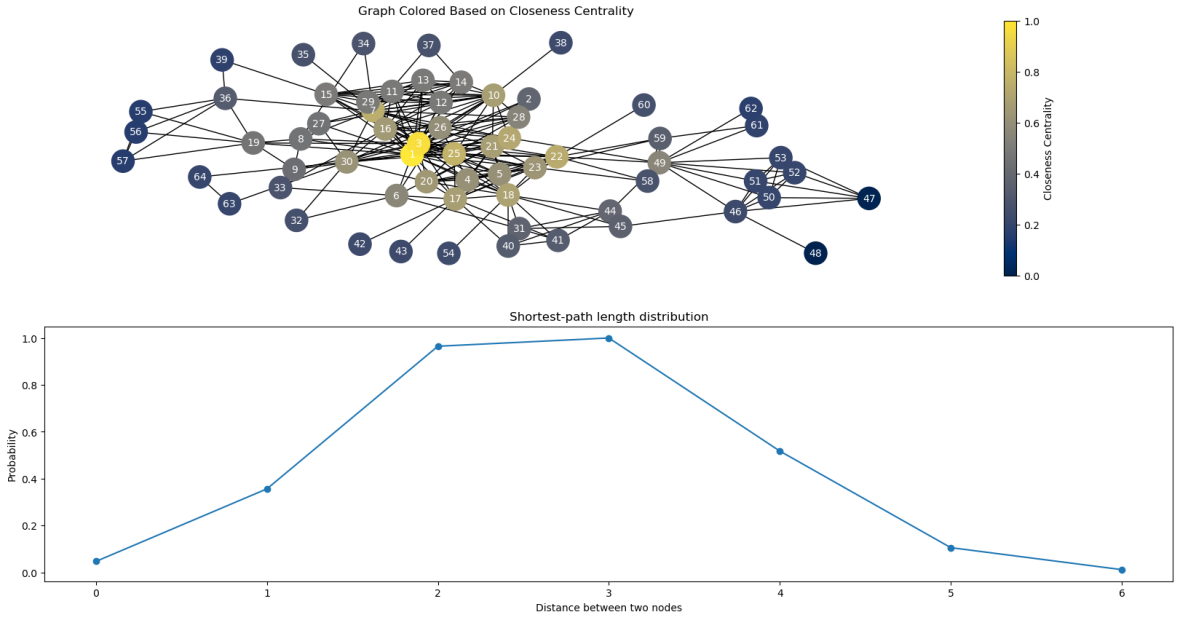

Figure 21: Closeness centrality distribution.

It is evident that closeness centrality generally exhibits higher values. The node 1 is the most central due to its high strength, closeness, and betweenness scores. It may represent a kind of point of reference known to many other terrorists. Interestingly, node 49 appears prominently only in the betweenness centrality visualization. This suggests that the individual represented by node 49 might serve as a critical bridge between multiple groups of terrorist cells. This observation aligns with the drowing in Figure 23, generated from the link betweenness centrality calculation, which clearly indicates that the link connecting node 49 to the central part of the graph is the most traversed by various paths.

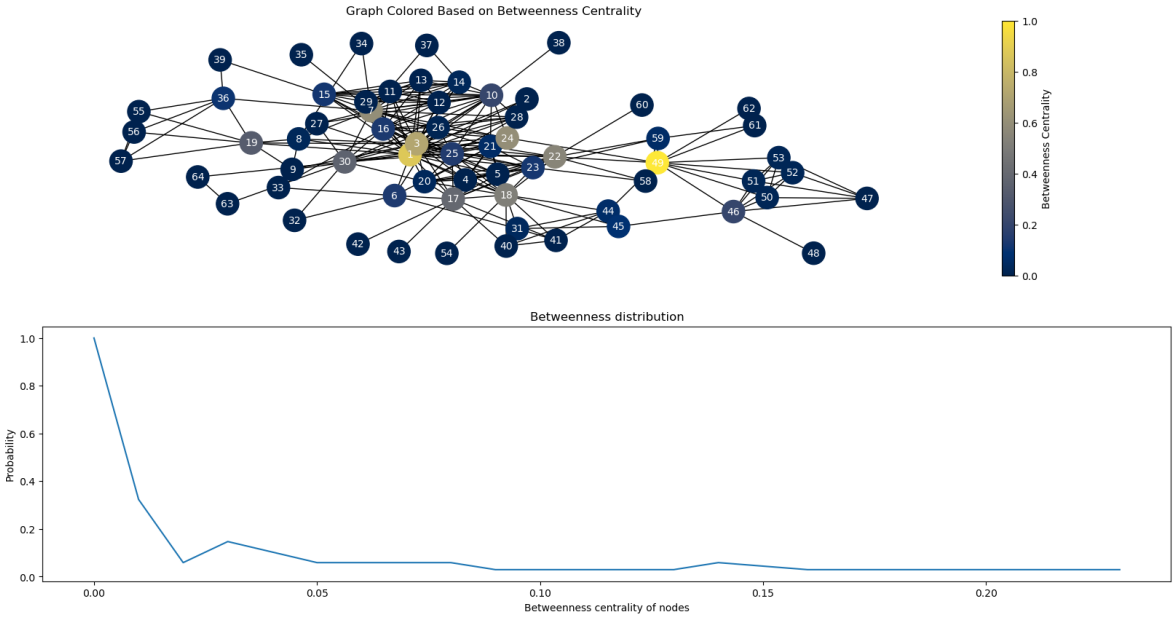

Figure 22: Betweenness centrality distribution.

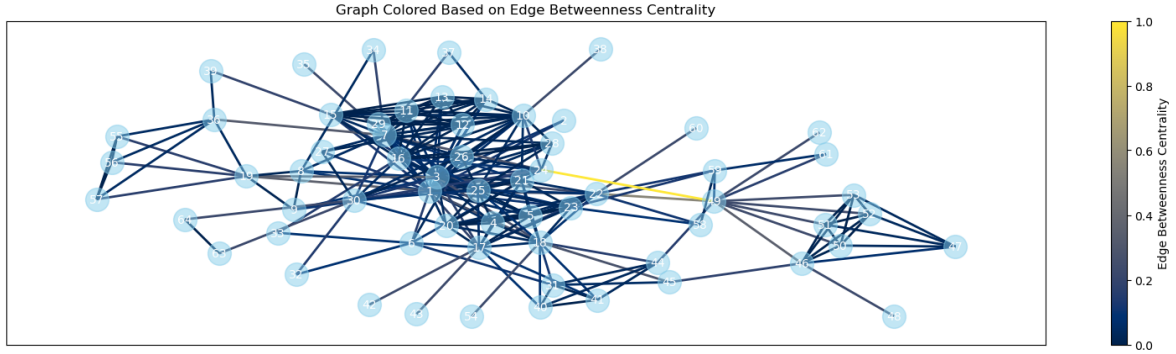

Figure 23: Link betweenness centrality distribution.

2.8 Core Decomposition

To identify the densest parts of the terrorist network and gain another perspective on the most critical groups of individuals for law enforcement intervention, we can perform a core decomposition. This algorithm iteratively removes the least connected nodes, revealing the nodes with the highest connectivity in the final stages. The figure 24 all the steps are shown. It is clear that the densest part is the central one, in according on the result of assortativity.

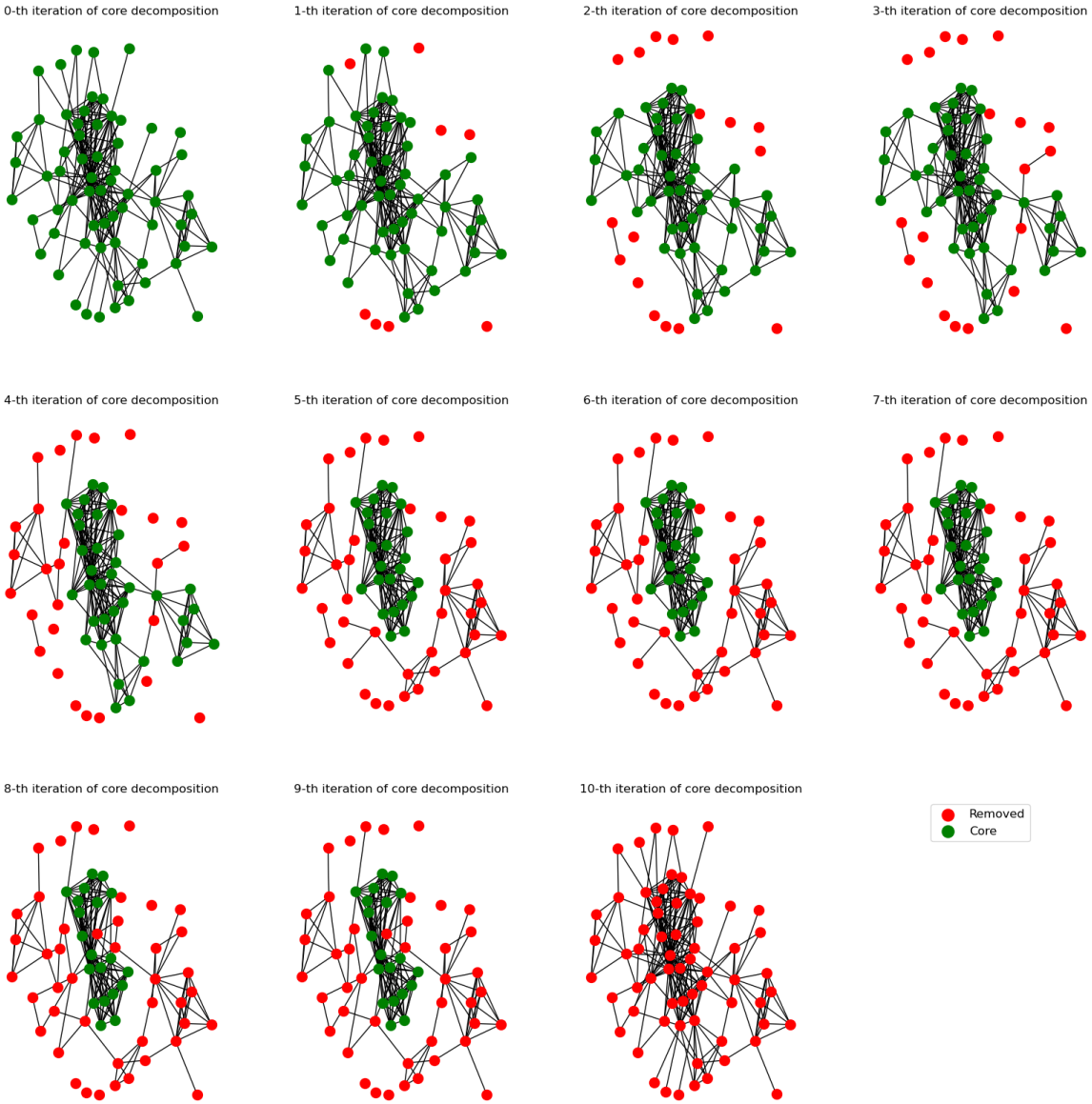

Figure 24: Iterations of core decomposition of the graph.

2.9 Homophily

According to the homophily principle, individuals belonging to a terroristic cell are more likely to trust each other and connect when they share similar characteristics, such as a common ideology, program and goal. Given their secretive nature and inability to openly recruit new members, extremists are likely to display patterns of homophily by prioritizing ties based on shared characteristics. Terrorist networks are likely to form close-knit groups. These are dense small clusters of individuals who are all connected and act as gatekeepers, admitting only those who they think they can trust [AS23].

In this paragraph, we aim to test for homophily in the graph under analysis using a homophily test. We utilize groups of nodes generated by the label propagation algorithm. A lower number of cross-group edges indicates a stronger signal of homophily. Then, in the following Python code, for each pair of clusters, we count the expected number of cross-edges and the actual number of cross-edges. These values are then normalized and compared.

```
In [3]:
p = [len(cluster) / G.number_of_nodes() for cluster in lpa_partition]
for i in range(len(lpa_partition)):
    for j in range(i+1,len(lpa_partition)):
        # actual crossing-edges
        ac = len([e for e in G.edges if (e[0] in lpa_partition[i])
                    and (e[1] in lpa_partition[j])]) / G.number_of_edges()
        # expected crossing-edges
        ex = 2 * p[i] * p[j]
        if ac < ex:
            print(f'{round(ac,3)} < {round(ex,3)}:
                    Homophily between clusters {i+1} and {j+1}')
        else:
            print(f'{round(ac,3)} > {round(ex,3)}:
                    No homophily between clusters {i+1} and {j+1}')

Out [3]:
0.082 < 0.191: Homophily between clusters 1 and 2
0.016 < 0.082: Homophily between clusters 1 and 3
0.029 < 0.055: Homophily between clusters 1 and 4
0.016 < 0.137: Homophily between clusters 1 and 5
0.008 < 0.027: Homophily between clusters 1 and 6
0.008 < 0.041: Homophily between clusters 2 and 3
0.0 < 0.027: Homophily between clusters 2 and 4
0.0 < 0.068: Homophily between clusters 2 and 5
0.0 < 0.014: Homophily between clusters 2 and 6
0.0 < 0.012: Homophily between clusters 3 and 4
0.0 < 0.029: Homophily between clusters 3 and 5
0.0 < 0.006: Homophily between clusters 3 and 6
0.004 < 0.02: Homophily between clusters 4 and 5
0.0 < 0.004: Homophily between clusters 4 and 6
0.0 < 0.01: Homophily between clusters 5 and 6
```

3 Comparison with artificial random network

In the following section the terrorist graph if compared with random graphs generated by four kind of models: the Erdos-Renyi model, the Watts-Strogats model, the Compound model and the Barabasi-Albert model.

3.1 Erdos-renyi model

The model of Erdos-Renyi takes in input the number of nodes $N = 64$ and the number of links $L = 243$. Links are placed at random, independently of each other. This implies that distances between pair of nodes are short and it is a shared property with the terrorist graph. The Figure 25 showhs that the two distributions are similar.

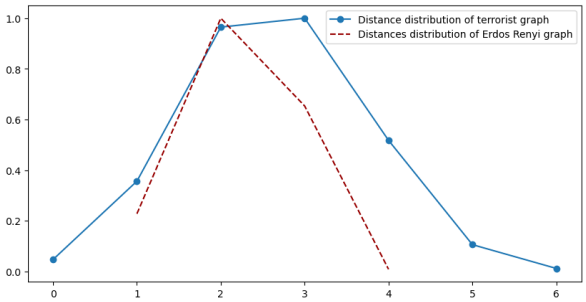

Figure 25: Distances distribution comparison.

Since the degree distribution is concentrated on the mean (as it is shown in Figure 26), then the clustering coefficient will be low. Indeed, the clustering coefficient of terrorist graph is 0.561, while the clustering coefficient of Erdos-Renyi graph is 0.131.

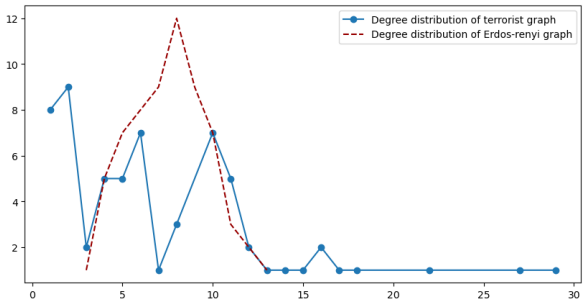

Figure 26: Degree distribution comparison.

3.2 Watts-Strogatz model

The input parameters of the Watts-Strogatz model are as follows: the number of nodes $N = 64$, the number of nearest neighbors each node is initially connected to in the ring, $k = 8$, and the probability of random rewiring of the links. The parameter k is set to 8 to approximate the same number of links as in the terrorist network graph. The rewiring probability is chosen to be 0.1, as this value strikes a balance between maintaining a short average path length and a high clustering coefficient, as it is shown in Figure 27.

The average path length of the Watts-Strogatz graph is 2.64, which is very close to the average path length of the terrorist graph, which is 2.69. Similarly, the clustering coefficients of the two graphs

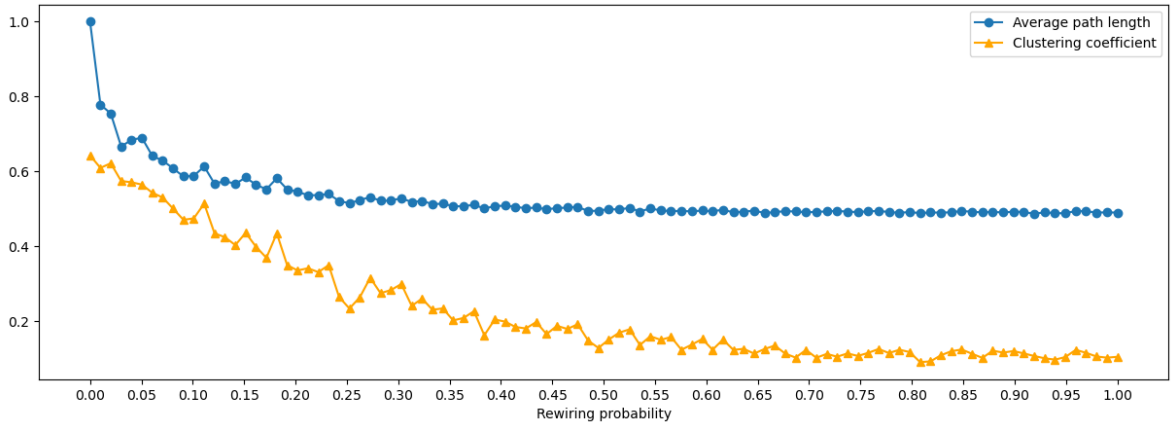

Figure 27: This plot display the average path length and the clustering coefficient variation with respect to the rewiring probability growth that comes out when a Watts-Strogatz model is used to generate a random graph.

are also comparable. The clustering coefficient of the terrorist graph is 0.561, while that of the Watts-Strogatz graph is 0.509. These similarities indicate that the Watts-Strogatz model effectively captures this two structural properties of the terrorist network.

In Figures 28 and 29, we compare the distance distributions and the degree distributions of the two graphs. As expected, the distance distribution does not vary significantly between the two models. However, the degree distribution is markedly different. The Watts-Strogatz model does not allow for the presence of hubs, resulting in nodes having approximately the same degree. In contrast, the actual terrorist network contains hubs, leading to a more heterogeneous degree distribution.

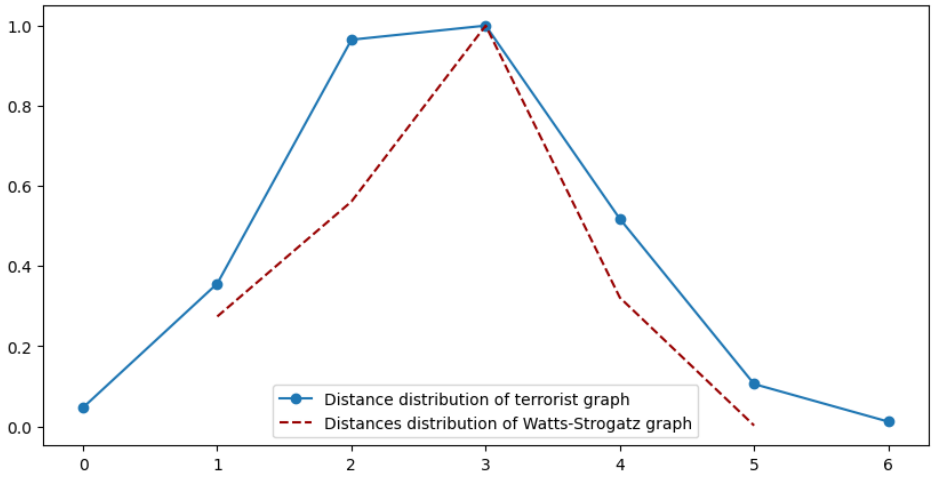

Figure 28: Distance distribution comparison between terrorist graph and Watts-Strogatz random graph.

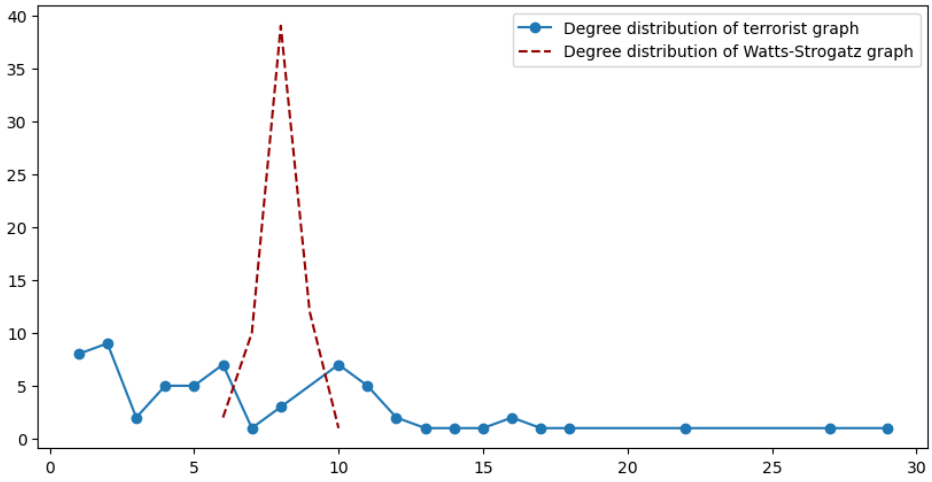

Figure 29: Degree distribution comparison between terrorist graph and Watts-Strogatz random graph.

3.3 Configuration model

Since the previous models did not maintain a degree distribution similar to that of the graph under consideration, we propose a new model based on a predefined sequence of degrees. This model takes as input the degree list of the nodes in the terrorist graph, ensuring that it retains the same number of nodes and edges. Consequently, this approach preserves the degree distribution (as the Figure 31 shows) and better reflects the structure of the terrorist network.

The new model also preserves other properties of the graph. As shown in Figure 30, the distance distribution and average path length are very similar between the two graphs. Specifically, the average path length of the new model is 2.45, while that of the terrorist graph is 2.69. Furthermore, the clustering coefficients are closely matched: the clustering coefficient of the terrorist graph is 0.561, and that of the new model is 0.509. This demonstrates that the new model effectively captures both the structural properties and degree distribution of the terrorist network.

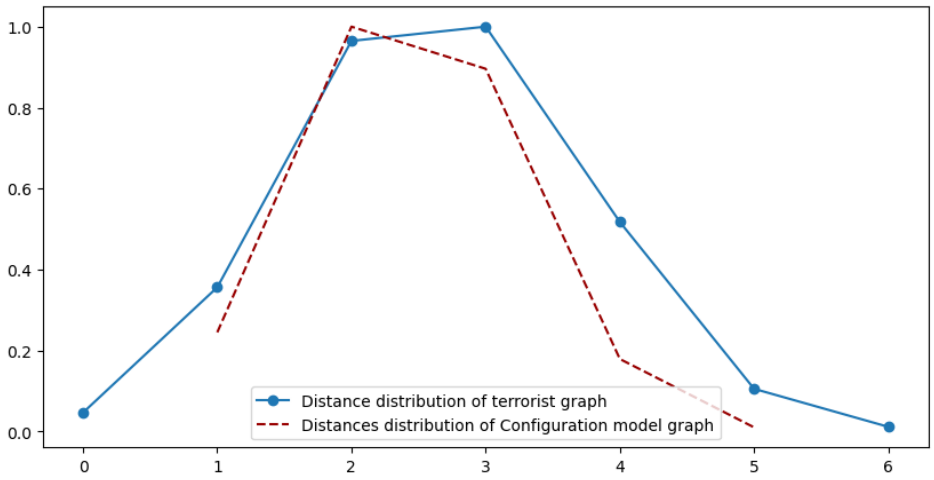

Figure 30: Distance distribution comparison between terrorist graph and random graph generated by the Configuration model.

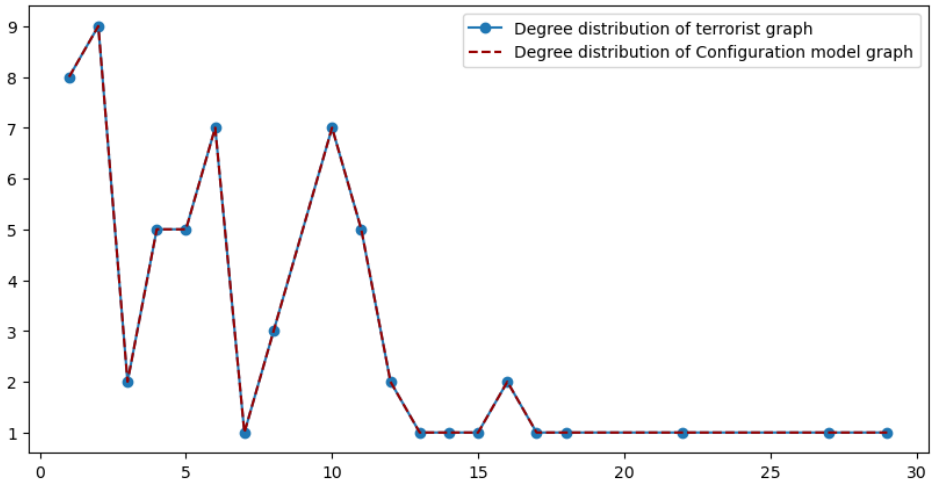

Figure 31: Degree distributions of terrorist graph and random graph generated by the Configuration model are exactly the same.

3.4 Barabasi-Albert model

We also want to compare the graph under consideration with a model based on preferential attachment: the Barabási-Albert model. This model is particularly relevant because real-world networks are dynamic, and the Barabási-Albert model can capture properties that other static models may overlook. In this model, one node is added at a time, each with a certain number of stubs (edges), and it is more likely to attach to nodes with a higher degree. Consequently, this model generates hubs, which are a common feature in real-world networks. The underlying idea is to represent the tendency of individuals to imitate the actions of others, leading to a network with a few highly connected nodes and many nodes with fewer connections.

By comparing the terrorist network graph with the graph generated by the Barabási-Albert model, we observe that the distance and degree distributions are quite similar (Figures 32 and 33). The average path lengths are also close: the average path length of the Barabási-Albert model graph is 2.196, while the average path length of the terrorist graph is 2.69. However, the clustering coefficient differs substantially between the two graphs. The clustering coefficient of the terrorist graph is 0.561, whereas the clustering coefficient of the Barabási-Albert model graph is 0.153. This discrepancy arises due to the "rich-get-richer" phenomenon inherent in the Barabási-Albert model, where many nodes tend to connect to a few highly important nodes (hubs). Consequently, nodes with lower importance (i.e., neighbors of hubs) are less frequently connected to each other, leading to a lack of triadic closures. In contrast, social networks, including the terrorist network graph we are analyzing, exhibit a strong presence of triadic closures.

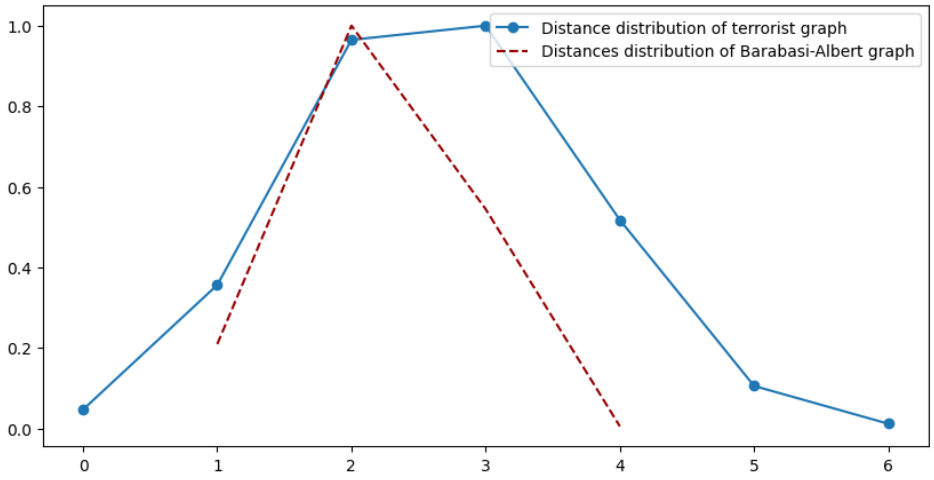

Figure 32: Distance distribution comparison between terrorist graph and random graph generated by the Barabasi-Albert model.

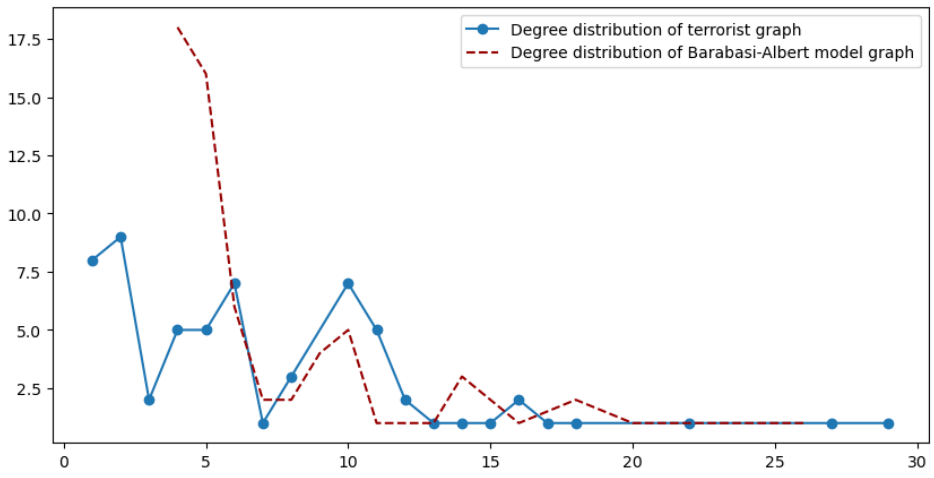

Figure 33: Degree distribution comparison between terrorist graph and random graph generated by the Barabasi-Albert model.

References

[AS23] Michael Jensen Anina Schwarzenbach. Extremists of a feather flock together? community structure and patterns of homophily in the us islamist co-offending network. Master's thesis, Department of Criminology and Criminal Justice, University of Maryland, Maryland, US, 2023.

[Mor17] Moreno. Train bombing network dataset – KONECT, oct 2017.

[Taf] Guglielmo Taffini. Organizzazione per cellule del terrorismo jihadista. *Question Giustizia*.