How to beat terrorism efficiently: identification of set of key players in terrorist networks

Marco Pietro Abrate, Natalie Bolón Brun, Shahow Kakavandy, Jangwon Park
École Polytechnique Fédérale de Lausanne, 2019

I. INTRODUCTION

Proliferation of terrorism in recent years has led people to believe it as a real threat to their livelihood. Vital to the success of such terrorist organizations are the cohesiveness and ability to communicate efficiently within their respective terrorist networks. To make these networks vulnerable, identifying sources of such properties is an imperative mission and hence becomes the focus of this report. More technically, we seek to develop an appropriate methodology to evaluate the importance of each terrorist to the effectiveness of the network as a whole, and identify an optimal set of key terrorists that one should target in order to debilitate it.

II. PROBLEM STATEMENT

The project aims to find points of vulnerability in the terrorist relations network that one can exploit to reduce its overall effectiveness. We define vulnerability in the sense of key terrorists in the network whose absence will fragment it as much as possible. Similarly, we define it also in the sense of key terrorists who, if fed with deliberate misinformation, are best positioned to spread it most quickly and widely. As such, we employ a key player approach which is broken into two separate problems:

1) **Fragmentation**: identify a set of key terrorists that best fragments the network when removed.
2) **Information flow**: identify a set of key terrorists who are best positioned in the network to spread false information most efficiently.

III. DATA PROCESSING & CLEANING

The original dataset is acquired from LINQS [1]. The network it provides encodes relations between terrorists. Its nodes represent relations (labeled as colleague, congregate, contact or family) while the edges represent names of terrorists. Given the nature of the edges, they are unweighted and undirected. The original network has 851 nodes and 8,592 edges and its largest connected component consists of 665 nodes and 6,552 edges.

The interpretation of this first representation suggested by the original analysis of the network [2] is arguably counter-intuitive. Therefore, the first data processing step is to invert the network such that the nodes represent the terrorists, between whom is an edge only if they are related in some way. However, disconnected components in the original network must be inverted and analyzed separately. In this project, we invert and analyze the largest component which is sufficiently large to represent all the major characteristics of the entire network (and hereby refer to the largest component of the original network simply as the 'original network'). The following describes the network inversion process:

1) Extract terrorist names from the unique ID of each node in the largest component of the original network.
2) Initialize an adjacency matrix whose size is equal to the number of unique terrorists found in step 1.
3) Set $a_{ij} = 1$ between terrorists i and j if they belonged to the same node in the original network.

The unique ID of each node in the original network is a URL with terrorist names embedded in it. We extract the names by parsing the URL on special characters (e.g. #). An important finding in this process is that each unique ID will contain at most two terrorist names. This implies that no matter how high the degree of the node in the original network is, all its edges represent at most two unique terrorists. When the unique ID is missing a terrorist name, we discovered that it always has a unique datetime string in its place. Therefore, we tentatively use the datetime string as the name for that particular terrorist.

The inverted network has a size of 244 nodes and 661 edges, a significant reduction from the original network. This verifies that many edges in the original network are duplicate terrorists. Additionally, 126 of the 244 unique terrorists did not have known names and were thus replaced with unique datetime strings.

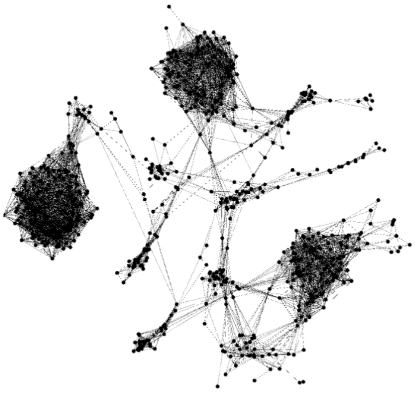


Fig. 1: Largest component of the original network

IV. EXPLORATORY DATA ANALYSIS

Figure 1 presents the largest component of the original network. It is characterized by many peripheral nodes and three hubs, which are rather loosely connected with each other. As we will shortly see with the inverted network, this is also a relatively dense network due to the same terrorist representing multiple edges. These observations partly justify the intuition behind pursuing network inversion.

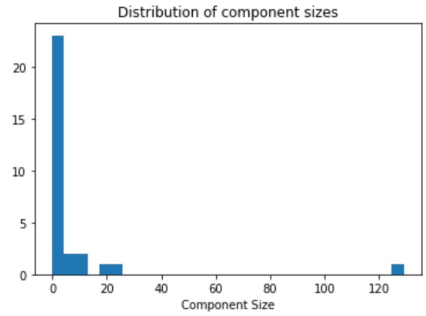


Fig. 2: Distribution of the sizes of 30 disconnected components in the inverted network

Inversion of the largest connected component of the original network may still result in several disconnected components. Nevertheless, for the purpose of fragmenting a network, it is interesting to work with the largest connected network here as well. However, we would only be justified in working solely with the largest component if it represented the majority of the network by far. To this end, we discovered that the inverted network consists of 30 components, many of which are either isolated nodes or very small components as shown in Figure 2. There is one component with 129 terrorists whose size exceeds all other components by a very wide margin. Therefore, we pursued further analysis with only this largest component in this project.

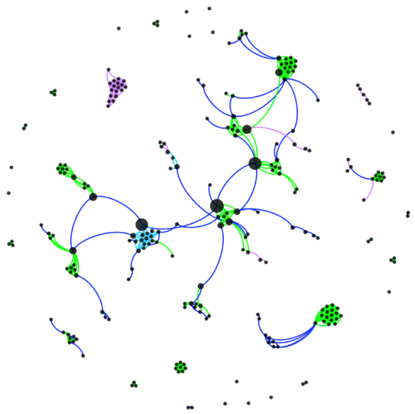


Fig. 3: Inverted network with disconnected components. Pink edges indicate family relations.

By taking the largest component, we notice some changes in the proportions of different types of relations as shown in Table I. The biggest change is seen with family relations, implying that terrorists who are families tend to be isolated from other networks. This is evident in Figure 3 where the family relations (pink) are mostly found in a specific part of the graph as an isolated community. Furthermore, we contrast the sparsity of the inverted network with the original network in Figure 1, and also notice that there is now a single prominent hub, both of which indeed make our analysis simpler and more intuitive.

TABLE I: Proportions of relation types in the inverted network

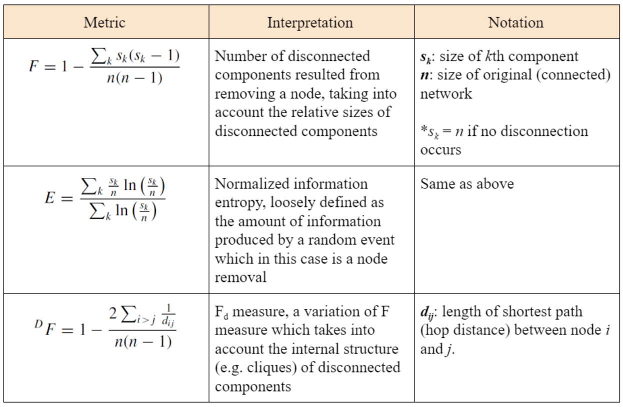
Relation	Entire Network	Largest Component	Δ
Colleague	69.59%	68.86%	-0.73%
Congregate	8.02%	14.86%	+6.84%
Contact	10.59%	13.71%	+3.12%
Family	11.8%	2.57%	-9.23%

V. METRICS

Before we can find an optimal set of terrorists to target, we need metrics to evaluate their relative importance. These metrics are developed separately according to the two distinct problems we identified in Section II ”Problem Statement” – fragmentation and information flow. To this end, we develop two sets of measures which respectively compute a score of each terrorist on how important he is for fragmenting the network when he is removed and spreading information efficiently given his position.

A. *Metrics for the Fragmentation Problem*

[3] proposes a number of metrics for the problem of finding the key players in a social network optimal for fragmentation. While there are many off-the-shelf metrics, such as centrality measures, for measuring the relative importance of each node, [3] argues that they are not optimal for the fragmentation problem for the compelling reason that optimality of centrality measures in identifying key players, whose removal leads to the greatest fragmentation of the network, is not guaranteed in certain cases. As such, we employ three more appropriate metrics which are summarized in Figure 4. For greater detail, we refer the reader to [3].



Metric	Interpretation	Notation
$F = 1 - \frac{\sum_k s_k(s_k - 1)}{n(n-1)}$	Number of disconnected components resulted from removing a node, taking into account the relative sizes of disconnected components	$\boldsymbol{s_k}$: size of kth component $\boldsymbol{n}$: size of original (connected) network $*s_k = n$ if no disconnection occurs
$E = \frac{\sum_k \frac{s_k}{n} \ln\left(\frac{s_k}{n}\right)}{\sum_k \ln\left(\frac{s_k}{n}\right)}$	Normalized information entropy, loosely defined as the amount of information produced by a random event which in this case is a node removal	Same as above
${}^{D}F = 1 - \frac{2\sum_{i>j} \frac{1}{d_{ij}}}{n(n-1)}$	$\mathrm{F_d}$ measure, a variation of F measure which takes into account the internal structure (e.g. cliques) of disconnected components	$\boldsymbol{d_{ij}}$: length of shortest path (hop distance) between node i and j.

Fig. 4: Descriptions of the three metrics used to measure relative importance of each node for fragmentation

B. *Metrics for the Information Flow Problem*

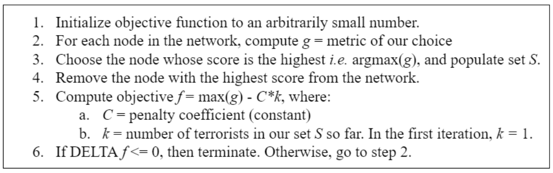
For this problem, we are interested in each terrorist's or a set of terrorists' ability to spread (mis)information to the rest of the network as efficiently as possible. A good example is the mean distance from the set of terrorists to all other terrorists in the network. However, such an average metric may overlook the presence of excessively long paths, which should ideally be penalized more strongly. To account for this, the longest shortest distance from the set to all other terrorists as well as its frequency are recorded. An optimal set of terrorists for spreading misinformation most efficiently would have the minimum values in these metrics.

VI. GREEDY OPTIMIZATION ALGORITHM

In this section, we present a greedy optimization algorithm which outputs an optimal set of key terrorists based on maximization or minimization of the metrics described in the previous section.

A. *Fragmentation Problem*

At first, it may seem plausible to conclude that the optimal set of k key terrorists is simply the set of the top k terrorists as evaluated by the metrics described in the previous section. However, this is not true since, in the context of the fragmentation problem for instance, the removal of the first key terrorist will change the structure of the network; the terrorist who used to be the second most important in the original network then may no longer be deemed important under the new configuration. Therefore, for the fragmentation problem, we develop a greedy optimization algorithm which sequentially finds the top key terrorist at each step as evaluated by a metric (any metric in principle), removes him from the network, and continues the same process. Figure 5 presents the pseudo code for this algorithm.



```
1. Initialize objective function to an arbitrarily small number.
2. For each node in the network, compute $g$ = metric of our choice
3. Choose the node whose score is the highest i.e. argmax($g$), and populate set $S$.
4. Remove the node with the highest score from the network.
5. Compute objective $f$ = max($g$) - $C$*$k$, where:
    a. $C$ = penalty coefficient (constant)
    b. $k$ = number of terrorists in our set $S$ so far. In the first iteration, $k$ = 1.
6. If DELTA $f$ <= 0, then terminate. Otherwise, go to step 2.
```

Fig. 5: Pseudo code of the greedy optimization algorithm for the fragmentation problem

While this algorithm is inspired by [3], it has a few notable adaptations which we believe to be improvements of the original. First, the algorithm in [3] finds a set of k players simultaneously for the fragmentation problem. Though the outcome of this is equal to Figure 5, it does not allow us to determine which one among the k players is the most important. In other words, we do not have any idea about the internal ranking of the key players. The sequential algorithm in Figure 5, on the other hand, allows us to quantify the relative ranking among the set of k key terrorists.

Secondly, [3] requires a predetermined value of k which is often an arbitrary choice. To address this, we include a regularization term which penalizes every additional terrorist to the set S scaled by the penalty coefficient C. Although the choice of C is also arbitrary, we have nevertheless introduced a method of controlling the trade-off between prioritizing the maximization of the metric vs. including more terrorists. Furthermore, the user is free to change the exact form of the regularization term to better accommodate the resource capacity and attitude of his/her organization (e.g. linear, exponential, etc.). In this project, we impose a linear penalty for its simplicity.

B. *Information Flow Problem*

Similarly, a greedy search has been implemented for the information flow problem with the aim to find a set of k key terrorists that collectively spread misinformation most efficiently throughout the network. In this case, the identification of the optimal set is done concurrently (as opposed to sequentially) and solved by the minimization of an objective function plus a penalization term that takes into account the size of the set. The objective function is presented as a weighted combination of the mean shortest distance from the set to all other nodes in the network and the maximum shortest distance multiplied by the number of nodes at this distance. The goal is then to minimize the time required to pass on information to the entire network and the number of people the information passes through. The second criterion is important as information tends to be distorted the more people it passes.

The search is performed by first creating all possible combinations of terrorists of a predetermined size k and evaluating their respective performance as information spreaders. Unlike the fragmentation problem, the search cannot be performed sequentially by adding new key terrorists to the previously found set. This is due to the fact that two nodes that achieve good results individually are not guaranteed to form the best pair collectively, as they can be redundant. Therefore, we look for the highest performing group rather that the group of highest performing individuals.

Finally, some external parameters are used as penalty terms. For instance, a penalty parameter is used to take into account the number of nodes located at the maximum shortest distance from the set. The default value is set to 0.1 as it allows a moderate influence on the final objective value without giving too much weight to this factor. On the other hand, a penalty term for the size of the set is also a free parameter that allows one to consider the cost of increasing the size of the set.

VII. RESULTS & DISCUSSIONS

In this section, we present the two optimal sets of key terrorists, found by the algorithm for each of the two problems that we set out to answer in the beginning of the report. As benchmarks, we compare our solutions with those found by simply selecting the top k terrorists based on degree, betweenness, and closeness centralities.

A. *Fragmentation problem*

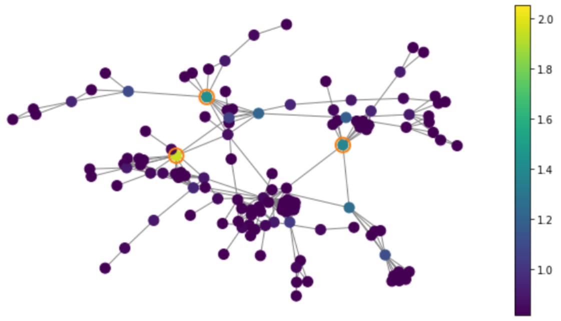


Fig. 6: Optimal set of 3 key terrorists for fragmentation found by the greedy optimization algorithm

Figure 6 presents the optimal set of key terrorists that will best fragment the network when removed. With a penalty coefficient C = 0.75, the algorithm finds three terrorists. As previously noted, increasing the magnitude of C will lead to smaller number of terrorists in the set while decreasing it allows the algorithm to find more. Among the terrorists, the node colored in yellow is the most important terrorist, followed by the node above it, and finally the node on the far right is the third most important. The color bar in Figure 6 indicates the combined measures of the three metrics presented in section Metrics, where higher values indicate growing importance. However, it is important to note that the distribution of this signal can change completely once a terrorist is removed from the network sequentially.

A natural action item from the result of this algorithm would be to prioritize arresting or assassinating these particular individuals. By removing them, the terrorist network is fragmented to the greatest extent than by any other group of three terrorists. Figure 7 shows the state of the network after these three individuals are removed. In stark contrast to Figure 6, there are many more distinct components (seven in total), suggesting the effectiveness of the algorithm. The scale of the color bar here is less interpretable due to many disconnected components.

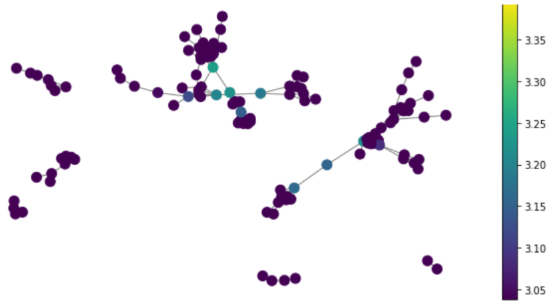


Fig. 7: Network after removing the optimal set of 3 key terrorists

To assess the effectiveness of the algorithm, we compare our solution to that obtained by simply choosing the top three terrorists based on the highest degree, betweenness, and closeness centralities respectively. In Table II, we can verify that the algorithm outperforms all other benchmarks in all aspects by far. The criteria include F measure and information entropy as described in Figure 4 as well as the number of disconnected components caused by the removal of terrorists.

TABLE II: Comparison of our optimal solution

Benchmark	F measure	Information Entropy	Components
Degree	0.279	0.516	4
Betweenness	0.497	0.701	3
Closeness	0.497	0.701	3
Our solution	**0.692**	**1.355**	**7**

B. *Information flow*

The optimal set of terrorists obtained by the optimization algorithm with a penalty term C set to 0.5 is composed of three nodes: (27, 63, 87). The average distance from this set to any node in the network is 2.46 hops, and there is only one node located at the maximum distance of 6 hops.

By varying the penalty term related to the size of the set, we can have sets of size two or four. Nevertheless, this will depend on the resource capacity of the organization. As we consider catching an extra person can come at a very high cost, a set of 3 people is considered reasonable.

We aim to assess the efficiency of our solution acting as a source compared to other sets of terrorists. Similar to the benchmarks used in the fragmentation problem, we generate three different sets of size 3 – our solution obtained via optimization *(S1)*, the group of highest performing terrorists according to betweenness centrality *(S2)*, and finally the group of highest performing terrorists according to their degree *(S3)*. These sets are then compared in terms of their distances to the rest of the network and diffusion of information.

Figure 8 presents the comparison of the sets' average and maximum distances to the rest of the network. Notice that *(S3)* yields much worse results and therefore is excluded from our discussion. On the other hand, *(S1)* and *(S2)* yield similar results. Nevertheless, *(S1)* still obtains a lower average distance and a greater number on nodes located at a maximum distance of 2 hops. In the context of this problem, these are interesting values as a greater number of nodes will receive the information in a relatively short period of time. The solution achieved by the algorithm tends to become much better for larger sets than three terrorists. For this analysis, we refer the reader to our code.

Finally, the process of how information diffuses is simulated as a heat diffusion process. The analysis is performed by applying an impulse signal on the terrorists in the set and filtering the signal with a heat filter to study the diffusion when the terrorists act as sources. Again *(S3)* is discarded by a clear lack of performance.

Results shown in Figure 9 illustrate the diffusion of the information considering different sets of sources. Here, a value of *-1* describes the maximum level of information a terrorist can possibly receive while *0* indicates no

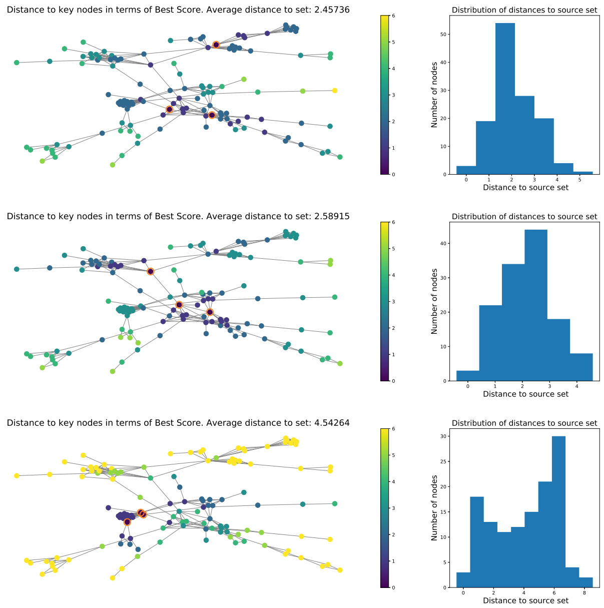


Fig. 8: Distance to the source set along the network for different source sets.

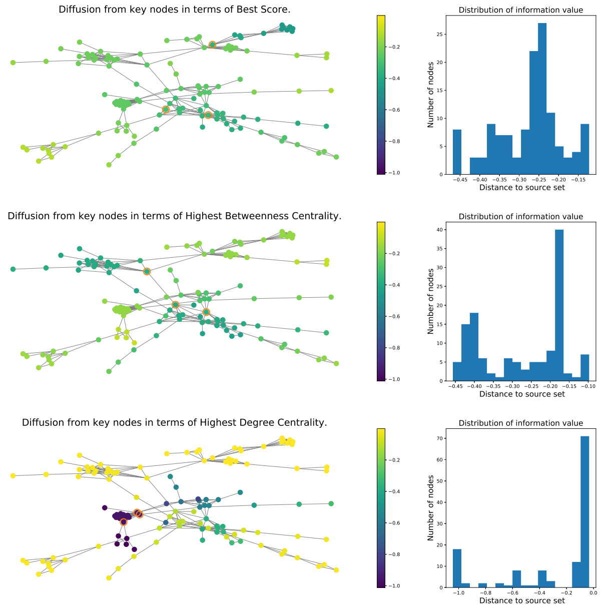


Fig. 9: Simulation of information diffusion along the network with different sets of nodes as sources

information is received at all. *(S1)* and *(S2)* present two different distributions of information values. For instance, *(S1)* presents a normal distribution with most of the nodes having received an amount of information above the average. Additionally, relatively fewer nodes miss out on the information (under 10% of the maximum) and none of them completely uninformed. On the contrary, *(S2)* produces two peaks in the distribution, leaving many terrorists at either tail of the distribution. For the purpose of spreading misinformation with the aim to create chaos and confusion, using *(S2)* as source of information will leave many nodes with little exposure to information, which is not useful for our intended goal. Nevertheless, if further details not known for the current analysis presented the nodes of *(S2)* as more valuable in other terms, its performance as information sources will still be quite close to the one of the set we have considered optimal.

VIII. Limitations & Conclusion

In this project, we defined vulnerability to be the set of terrorists that would best fragment the network when removed as well as those who are best positioned in the network to spread misinformation most efficiently. We treated these two problems separately by first developing a set of metrics and solving for the optimal set of terrorists via a greedy optimization algorithm. In the end, we verified that our methodology outperforms the traditional key-player approach in which the top k individuals are simply grouped together.

The two sets encountered as optimal with the two approaches both comprise of three key individuals but are not necessarily identical. With the fragmentation approach, the key set is formed by nodes $\{$ *38, 27, 22*$\}$ while the information approach yields as optimal the set $\{$ *27, 63, 87*$\}$ (identification by name or unique datetime string is avoided as it does not provide relevant information for the task). Node *27* being common in both solutions lets us consider it as the most important key piece to take into account when performing an attack over the network.

This project focuses more heavily on developing an appropriate methodology. As such, the project have limited data (total of 129 unique terrorists which amounts to a relatively small network). Future work can be done on gathering and incorporating more data from various sources to create a larger terrorist network.

The nature of the data also presents some limitation as the network is unweighted and therefore assumes all relations are valued equally. In reality, this may not be true. However, there is certainly the possibility to adapt our methodology for weighted networks by changing our functions for the metrics. Nevertheless, our work is still insightful and can be applicable to possibly all other terrorists networks as well as social networks.

References

[1] Getoor, L. (n.d.). Datasets — LINQS. [online] Linqs.soe.ucsc.edu. Available at: https://linqs.soe.ucsc.edu/node/236 [Accessed 11 Jan. 2019].

[2] Zhao, B., Sen, P. & Getoor, L. (2006). Entity and Relationship Labeling in Affiliation Networks.

[3] Borgatti, S. (2006). Identifying sets of key players in a social network. Computational and Mathematical Organization Theory, 12(1), pp.21-34.