Discovering and leveraging communities in dark multi-layered networks for network disruption

Ryan Miller, Raluca Gera
Department of Applied Mathematics
Naval Postgraduate School, CA, USA
{remiller1, rgera}@nps.edu

Akrati Saxena
Department of CSE
Indian Institute of Technology Ropar, India
akrati.saxena@iitrpr.ac.in

Tanmoy Chakraborty
Department of CSE
IIIT Delhi, India
tanmoy@iiitd.ac.in

Abstract—In this paper we introduce a methodology to identify communities in dark multilayered networks, taking into account that the main challenges of these networks are incompleteness, fuzzy boundaries, and dynamic behavior. To account for these characteristics, we create knowledge sharing communities (KSC) that determine the community detection. KSC is driven by weighing the edge attributes as desired for the application that the communities are used. We provide an interactive algorithm that allows the operator to decide on the weights and the thresholds applied to create the communities. By choosing these variables, our results quantitatively outperform community detection on the collapsed monoplex network.

Index Terms—community detection, multi-layered network, dark networks, interactive algorithm.

I. INTRODUCTION AND MOTIVATION

Fear is not a new concept, yet organizations whose purpose is to spread fear remain difficult to fully comprehend. These groups are known by many names, such as terrorists, insurgents, or simply criminal organizations. In social network analysis, they are referred by names such as dark networks [1]. The complex structure of dark networks challenges researchers to develop more precise analytical methods to model and enhance our understanding of these networks.

This paper seeks a general-purpose algorithm for multilayered dark networks that is detailed enough to detect meaningful communities as well as flexible enough to be applied to a variety of networks. We propose a new algorithm that sorts the layers of terrorist networks into aggregate weighted categories to enhance network data integrity and ultimately, to detect more meaningful communities. Our method allows the user to choose the appropriate community detection algorithm and the threshold that produces the most relevant community partition. We show that this new algorithm enhances network data integrity, resulting in more analytically meaningful partitioned communities for dark networks than the current layer aggregation methods. Flexibility is achieved by engaging the user at multiple stages throughout the methodology implementation process, but also by offering a default. User input develops detailed and meaningful communities within the context of the user’s analytical goals.

We consider real network data sets and extend the methodology for general purposes. The communities from this proposed algorithm have the potential to enhance current understanding of multiplex networks. When this increased understanding is specifically applied to dark networks, it has the potential to aid analysts in network disruption and consequently, to restore safety and stability to terror inflicted regions.

There are many difficulties associated with mapping and analyzing dark networks. Krebs states that incompleteness is a huge factor since criminal networks do not want to be discovered [2]. As a consequence, network mapping and analysis are limited by the availability of relevant and accurate data, which we address in our methodology. Krebs clarifies the term fuzzy boundaries by connecting it to the process of data filtration. Clearly identifying which relationships are important and which are not directly impacts the process of modeling and analyzing a network.

In this paper, we examine three dark multiplex network case studies [3] to test our algorithm. The Noordin Top Network [4] provides the inspiration for our method and is discussed in great detail. The Fuerzas Armadas Revolucionarias de Colombia (FARC) [5], and Boko Haram [6] Terrorist Networks are used as further verification of our methodology. For each network, we validate our algorithm by determining the adjusted conductance, cluster adequacy, adjusted normalized cut, and normalized permanence of the resultant communities. To demonstrate the utility of finding communities for network disruption purposes, we built a network flow shortest path interdiction model. The model determines the optimal strategy, given a finite number of attacks, to disrupt the flow of information from a set of supply sources to a set of demand destinations. We enhance the optimal solution strategy for this model by examining the properties of the detected communities in the Noordin Top Network. The goal of this enhancement is to achieve similar optimal solutions while increasing the algorithm performance efficiency.

In Section II we present the related work and in Section III we present the description of our data, the baseline algorithm for community detection, metrics to evaluate the proposed algorithm against the baseline, and background on the validation of our methodology as a network interdiction model. Section IV we provide an overview and justification for our methodology for dark networks. In Section V we present our community detection results for different threshold values in
related work. We conclude with Section VI, in which we
discuss weaknesses of this research and recommend some potential extensions
of this research to improve our detection algorithm and our
approach to disrupting networks using community properties.

II. RELATED WORK

We first introduce the existing community detection work
and dark networks.

A. Community detection

A multilayered network \( M \) is defined to be
\[
M = (V_M, E_M, V, L),
\]
where \( V \) is the total number of vertices in the Network,
\( L = \{L_a\}_{a=1}^d \) for each aspect \( a \in \{1, \ldots, d\} \),
\( V_M \subseteq V \times L_1 \times \ldots \times L_d \) that contains only
the node-layer combinations in which a node is present in
the corresponding layer, and \( E_M \subseteq V_M \times V_M \) [7]. For our
research there are no edges between layers.

When all the layers are collapsed into a single network with
parallel edges or single edges with weights, we can further
classify the network as monoplex. Kivelä et al. [7] define
a monoplex network, \( O \), as: the aggregation of all of the
layers of a multiplex network into a single weighted layer.
Aggregation is achieved by defining edge weights, \( m \), between
vertices in each layer and expressing the final weight as a
linear combination of \( m \) from each layer. Thus, the underlying
data of the network can be used as a multilayer network or a
monoplex.

Our goal is to perform community detection on the multilay-
ered networks, as they are richer in data than their associated
monoplex networks as we will show in this research. One of
the difficulties of community detection is that a detailed and
comprehensive definition of community does not currently ex-
ist in network science as it potentially constrains the creativity
and applicability of the development of community detection
algorithms [7], [8]. For this work, the community definition
developed by Radicchi et al. [9] is used as a foundation: A
community is a set of vertices within the graph such that
connections between vertices within the community are denser
than connections with the rest of the network.

Some of the more popular community detection algorithms
are centered around the modularity function [7]. Newman
defines network modularity as the difference between the
actual number of edges in a partitioned group and the expected
number of edges in a partitioned group for a similar network
with the same number of vertices, where the edges are
randomly placed [10].

The Louvain method is a modularity based algorithm that
has demonstrated its utility in applied network science. This
method is an extension of the fast greedy algorithm using a
two-phased approach. During the first phase, communities
are initially identified using the fast greedy optimization process.
The second phase constructs an entirely new network by
replacing all of the vertices that belong to each community
with a single vertex. The fast greedy is then applied to this
new network and repeated till modularity reaches its maximal
value. The communities in the original network are then
created by collecting all of the original vertices that were
identified into a vertex in the subsequent iterations of the fast

B. Terrorist networks

There are many difficulties associated with mapping and
analyzing dark networks, that differentiates dark networks
from other social networks. Krebs [2] uses the September
11th 2001 terrorist attack in the United States as a case
study in dark network mapping and analysis. Krebs describes
three challenges previously identified by Sparrow [12] that are
specifically associated with mapping and analyzing criminal
social networks. Krebs identifies these challenges as incom-
pleteness, fuzzy boundaries, and dynamic behavior as we
explain below. Incompleteness is a huge factor since criminal
networks do not want to be discovered. As a consequence,
network mapping and analysis are limited by the availability
of relevant and accurate data.

One of the main purposes of dark network analysis is
network disruption. Many dark network analysts take the
approach of using high degrees of centrality to identify key
actors. However, Everton [13] believes this tactic may not
always be effective, and he encourages analysts to examine
more of the topographical characteristics of the network. He
believes this holistic approach identifies more appropriate
targets for disruption purposes. Research shows that this is
due to the incomplete nature of dark networks and the high
sensitivity of centrality computations based on small changes
to the network, especially for small networks [2].

Strategies for disrupting terrorist networks found in the
literature are (a) through information aggregation and knowl-
edge sharing where the key vertices to target in the network
are vertices with unique skills and vertices that have deep-
rooted trust relationships with other groups [13], or (b) by
focusing on the detection of covert communities in a dark
network over a time period (which was evaluated using a True
Positive Rate that labels one of the communities as covert, and
calculates a performance ratio based on covert and background
population members found in the community) [14]. Our data
set includes some temporal information, and we focus on
detecting multiple communities in the network that can be
validated using these strategies.

In the absence of a physical customer for attack-defense, this
research uses the Joint Improvised Explosive Device Defeat
Organization (JIEDDO) Attack the Network (ATN) philosophy
to infer the customer objectives. The United States Joint Forces
Command [15] summarizes JIEDDO’s ATN objectives to:
1) Identify key leaders in the network
2) Understand influence and relations
3) Identify and capitalize on vulnerabilities
4) Disrupt activities
5) Eliminate the ability for the network to function.

Understanding these customer objectives provides context
and focus for our purpose-driven communities definition.
JIEDDO essentially wants to learn more about the terrorist network for the purposes of disrupting its ability to function, and so our purpose-driven communities will thus be knowledge sharing communities (KSC). We define **Knowledge Sharing Communities**: Given the Noordin Network and JIEDDO, the KSC is the intersection of Trust, Lines of Communication, and Knowledge communities based on the need to disrupt intra-organizational coordination in the Noordin Network [3].

### III. Experimental Setup

We present a succinct overview of each of our data sets, metrics, and a validation metric using network interdiction.

#### A. Data Description

We use three data sets for this research: 1. Noordin Top, 2. FARC, and 3. Boko Haram.

The Noordin Top Network dataset contains the relationship information of 139 terrorists that belong to five major parent terrorist organizations operating in Indonesia based on the networks from [3]. The network is named after the key broker, Noordin Top, who was known for coordinating between terrorist organizations for training and operations. This network was primarily developed from the information provided by an article published by the International Crisis Group in 2006, *Terrorism in Indonesia: Noordin’s Networks* [13]. Roberts et al. [4] used this information to construct a possible total of 36 relationship types and attributes used to build the layers of the Noordin Network.

The FARC Terrorist Network dataset includes the relationship information of 142 terrorists known as the Revolutionary Armed Forces of Colombia that primarily operates in Colombia and Venezuela since 1964. The network is sparse for most layers but has a well-documented hierarchical structural layer due to social media [16]. This network dataset was created by Cunningham et al. [5] using a variety of open source documents. We re-organized the available relationship data into edge lists to build 10 separate layers for the case study on the FARC Terrorist Network.

The Boko Haram Terrorist Network is another dark network. This dataset contains the relationship information of 44 terrorists that belong to an Islamic sect that primarily operates in northern Nigeria since 2002. This network dataset was also created by Cunningham [6] using a variety of open source documents. We re-organized the available relationship data into edge lists to build 9 separate layers for the case study on the Boko Haram Terrorist Network.

Each network has the different relationship types connecting one vertex (person) to another vertex (person). The attributes are the properties assigned to each vertex, used to create edge labels. The vertex attributes will be used to measure community effectiveness and resilience. A detailed description of the data can be found in [3].

#### B. Baseline Algorithms

The baseline algorithm used in our experiments was introduced in Subsection II-A. It is the standard Louvain community detection [11] applied to the aggregated monoplex network.

### C. Metrics

We use five metrics to compare the quality of the community partitions obtained against their respective control algorithms:

- **Cluster adequacy** [17], $Q'$, as a normalized modularity

  \[
  Q' = \frac{\text{Modularity}}{1 - \frac{\# \text{ of communities}}{\# \text{ of nodes}}}
  \]

  where the modularity was defined in Subsection II-A

- **Adjusted Conductance**, $\phi'(S)$, as a normalized conductance [18]

  \[
  \phi'(C_i) = 1 - \phi(C_i),
  \]

  where the conductance, $\phi$, of a set, $C_i$, in a set of $k$ disjoint communities $C_1, C_2, \ldots, C_k$ as:

  \[
  \phi(C_i) = \frac{\text{vol}(C_i, \overline{C_i})}{\min(\text{vol}(C_i), \text{vol}(\overline{C_i}))}.
  \]

  Here, $\text{vol}(C_i, \overline{C_i})$ is the number of edges between the hypothesized community $i$ and its complement. Also $\text{vol}(C_i)$ is the number of edges between the nodes within the community $C_i$, and $\text{vol}(\overline{C_i})$ is the number of edges between the nodes outside the community $C_i$. The conductance of $G$ [18] is the minimum conductance of any subset of vertices:

  \[
  \phi(G) = \min_{C_i \subseteq V} \phi(C_i).
  \]

- **Adjusted Cut ratio**: is $1 - \psi$, where $\psi$ is the Cut ratio [19] that is defined below. Given a network $G(V, E)$ and a set of $k$ disjoint communities $C_1, C_2, \ldots, C_k$, we define cut ratio as

  \[
  \phi = \frac{1}{k} \sum_{i=1}^{k} \frac{\text{cut}(C_i, \overline{C_i})}{|C_i|},
  \]

  where $\text{cut}(C_i, \overline{C_i})$ indicates the number of edges between community $C_i$ and rest of the network.

- **Adjusted Normalized cut**: This is computed as $1 - \pi$, where $\pi$ is the normalized cut ratio [20] defined below. It is same as cut ratio with the difference in the denominator:

  \[
  \pi = \frac{1}{k} \sum_{i=1}^{k} \frac{\text{cut}(C_i, \overline{C_i})}{\text{vol}(C_i)},
  \]

  where $\text{vol}(C_i)$ is the sum of degree of all the nodes inside $C_i$.

- **Normalized Permanence**: This is computed as $\frac{\text{Perm}(v)}{\max(\text{Perm}(v))}$ [21]. Given a network $G(V, E)$ and a set of $k$ disjoint communities $C_1, C_2, \ldots, C_k$, we define permanence [21] of a node $v$ as

  \[
  \text{Perm}(v) = \frac{I(v)}{D(v)} \times \frac{1}{E_{max}(v)} - (1 - C_{in}(v))
  \]
where $I(v)$ is the number of internal neighbors of $v$, $D(v)$ is the degree of $v$, $E_{max}$ is the maximum external connections of a node, and $C_{in}(v)$ is the internal clustering coefficient of $v$. Then the permanence of the entire graph $G$ is $\text{Perm}(G) = \frac{1}{|V|} \sum_{v \in V} \text{Perm}(v)$. It ranges from $-1$ to $1$.

Higher values for each of these metrics result in better quality communities.

### Input:
Aggregate Simple Graph $G$ from Multilayer Network $M$.

**Step 1:**
Layer Selection

* Category$_{n}$, $n \in \{1, 2, \ldots, k\}$

* Category$_{n}$, $n \in \{1, 2, \ldots, t\}$

**Step 2:**
Weighted Category Sorting

Category$_n$, $n \in \{1, 2, \ldots, m\}$

**Step 3:**
Community Detection Algorithm

**Step 4:**
Convert communities into cliques. Remove clique interconnections. Assign category weight to all clique edges.

**Step 5:**
Aggregate by LOC to build weighted graph $W$.

**Step 6:**
Nodes in the components of $W$, give communities.

**Output:**
Plot resulting communities in $G$.

Fig. 1: Algorithm overview (general case).

### IV. METHODOLOGY OVERVIEW

In this section, we introduce the process, or algorithm used to transform the data from a multiplex network into meaningful partitioned communities according to user-based analytical goals and objectives. Figure 1 provides an overview of our methodology and Algorithm 1 presents the pseudocode for implementing this algorithm. This method takes layers of a multiplex network $M$ as an input and produces threshold controlled communities as an output.

Given the edge attributes, the layers of the $M$ are sorted into weighted categories ($\text{Category}_{w_i}$). Each $\text{Category}_{w_i}$ becomes a layer in our multilayered networks (which can be given by each attribute if desired), and we use them for individual category community detection. For each layer, the inter-community edges are removed as they are noise for our methodology. This is because they connect the graph by connecting the communities rather than making the communities tighter.

These communities in each category are converted into weighted cliques based on the assumption that edges are missing as minimal information is usually captured on terrorist networks and most of the direct links are obfuscated by the terrorists to secure their connecting information. The aggregation of all of these cliques results in the weighted graph $W$. Choice of a threshold $\varepsilon$ results in components in $W_\varepsilon$ that create the final communities, which we then identify in $G$. The vertices in $G$ are partitioned into the recently identified communities in $W_\varepsilon$. The algorithm is designed to identify communities of order two and larger. Any vertex that is not sorted into a community is placed in the Misfit Community by default.

The proposed method contains six steps which are briefly explained below.

**Step 1: Layer Selection**

This first step is focused on preparing and selecting the network data that is most appropriate based on the user’s goals. First, we examine the user’s goals to understand the motivation behind identifying communities. From this, we introduce the concept of the purpose-driven communities (PDC).

The selection of these layers is entirely dependent on the user’s analytical goals. The detection algorithm’s success and subsequent depth of the community property analysis are also dependent on the available relationship data. Once the layers have been selected, they will be sorted into one of $m$ categories ($m \leq n$) as described in Step 2. The aggregation of similar layers into categories reduces sparseness and increases network density for more accurate community detection.

**Step 2: Layer Sorting into Weighted Categories ($\text{Category}_{w_i}$)**

Now that we have identified the $n$ layers, each layer is placed into exactly one of $m$ weighted categories

$$\text{Category}_{w_i}, \forall i \in \{1, 2, \ldots, m\}.$$ 

Following our user-based philosophy, categories are chosen based on their relevance to the user’s analytical goals. Weights are assigned to each category based on the degree of importance and associated contribution toward forming the PDCs. As a default, all categories are assigned a weight value of one. If a foundational category is identified, then the respective weight of the foundational category should be chosen to be greater than the summation of the remaining categories.

$$w_{\text{foundation}} > \sum_{i=1}^{m} w_i - w_{\text{foundation}}.$$ 

We tested several sets of cases on the Noordin Network using different weights. During one case study, the Trust category was given the highest weight of $w_1 = 4$, followed by LOC with $w_2 = 2$ and Knowledge with $w_3 = 1$. This weighting system was applied based on the reasoning that Trust is the foundational category required to build KSCs. The dynamic nature of the dark network allows two people that
are only connected by the trust to potentially develop LOC and Knowledge and ultimately build a KSC. The combined weight of LOC and Knowledge is intentionally less than Trust to further establish Trust as a foundational category \((w_1 > w_2 + w_3)\).

**Step 3: Community Detection on Categories**

All of the layers of each \(\text{Category}_{\epsilon}^{i}\) are aggregated to form a sub-monoplex network \(\mathcal{O}_{\text{Category}_{\epsilon}}\). Now we apply Louvain community detection method \([11]\) on each of the three obtained layers.

**Step 4: Community to Clique Conversion**

The resultant communities for each category are converted into cliques. Each community within the categories is represented as a complete graph to emphasize the edge relationship of belonging to the same community. The edges within each category are given the same respective categorical weight.

In this step, we also remove the edges between the communities in step 3. The reason is that the inter-community edges are mostly noise for our methodology as they connect the graph together by connecting the communities rather than making the communities tighter.

**Step 5: Build the Weighted Graph \(W\)**

This step combines the resultant clique communities from all of the categories into an aggregate weighted graph \(W\). The edge weight, \(\epsilon_{w_{jk}}\), between any two vertices \(v_j\) and \(v_k\) in \(W\) is the summation of the edge weights between \(v_j\) and \(v_k\) from each category \(m\):

\[
e_{w_{jk}} = \sum_{i=1}^{m} w_{jk}^i, \forall jk \in \text{Category } i, \quad (9)
\]

where some edges may have zero weight.

**A. Step 6: Communities Through Tolerance \(\epsilon\) Selection**

The final step of this method is again user-driven to determine an acceptable threshold tolerance, \(\epsilon\). Choosing different thresholds creates a constraint on the graph that limits the amount of data considered to build communities. A choice of \(\epsilon = \sum_{i=1}^{m} w_i\) carries the strongest meaning and true intersection of communities across the \(m\) categories. The user is left to decide an acceptable value for \(\epsilon\) and may want to experiment with different \(\epsilon\) values to produce the desired meaningful communities. The sum of all of the category weights serves as a logical upper-bound for \(\epsilon\). If a foundational category exists, then the weight of the foundational category is recommended as a lower bound for \(\epsilon\). This prevents the other categories from forming communities without including the foundational category.

The threshold selection of \(\epsilon\) results in partitioning \(W\) into communities. These components of \(W\) are the PDCs. Any components that contain only one vertex are placed into a misfit community. As a final output, the algorithm plots the resultant PDCs’ nodes onto \(O\) to observe inter-community relations in the network to create the final communities plotted in \(O\).

Algorithm 1 is applied to all three dark network case studies and its results are presented in Section V.

**V. RESULTS AND ANALYSIS**

We now focus on our experiment design, displaying and analyzing our results. Three case studies are considered for each dark network.

Each case study represents a different selection of weight values for \(w_1\), \(w_2\), and \(w_3\) as described in Step 3 of our methodology. We then studied nine subcases for each case that corresponds to different choices for epsilon as described in Step 6 of the methodology. Figure 2 depicts the organization of the different weight cases, and threshold subcases studied in this chapter, and they are referred in our results plots.

We grouped the subcases in sets of three based on \(\epsilon\) values corresponding to: Case \(1\) \(\epsilon \leq w_1 + w_2 + w_3\), \(w_1 + w_2\), and \(w_1 + w_3\); Case \(2\) \(\epsilon = w_1 + w_2 + w_3\), \(w_1 + w_2\), and \(w_1 + w_3\); and Case \(3\) \(\epsilon \geq w_1 + w_2\), \(w_1 + w_3\), and \(w_1\) respectively.
For display purposes, we focused on plotting the individual subcase community results for the Noordin Network.

We compare each plot against the established control cases. Before conducting this experiment, we established our hypothesis research for the quality of communities that are produced by the different cases and subcases. Qualitatively, we believe that Case 3 will produce the most meaningful communities based on our definition of KSC. Case 3 provides the necessary category weight distribution based on the trust attribute, to dominate the remaining community information from the other categories. Krebs and Everton established the importance of trust to the functionality and resilience of dark network. Forcing trust to be included as the lower bound for epsilon choices, in this case, is consistent with their convictions.

Quantitatively, our intuition is that for a given value, v, subcases that involve \( \epsilon \leq v \) will result in a small number of large sized communities. We believe this threshold will be too relaxed of a choice for \( \epsilon \). The subcases for \( \epsilon = v \) will produce many communities that are very small as we exclude particular relationships from enforcing an equality in the threshold versus inequality. Also, it prevents vertices from being neighbors in certain categories in order to achieve equality, which doesn’t seem to be realistic, but we consider them for completeness. Consequently, these cases may be too restrictive of a choice for \( \epsilon \). Finally, we find the subcases for \( \epsilon \geq v \) will produce better communities since these subcases are more relaxed than \( \epsilon = v \), yet more restrictive than \( \epsilon \leq v \), requiring that vertices are friends in at least one category, but possibly more.

A. Noordin Results and Analysis

In this section, we display the results and analysis of the Noordin Network. The uniform distribution of weight values \( w_1 = 1, w_2 = 1, \text{ and } w_3 = 1 \) serves as a default if the user is unable to determine a logical ordering for category importance. This threshold was chosen as the default since it represents equal importance amongst all categories.

Besides the uniform case, we also consider case 2: \( w_1 = 3, w_2 = 2, \text{ and } w_3 = 1 \) and case 3: \( w_1 = 4, w_2 = 2, \text{ and } w_3 = 1 \). The difference between them is the fact that one layer can trump the combination of the other two, based on the belief that in dark network the trust layer is essential.

Noordin is the largest of the three terrorist networks according to edge count, with a relatively equal distribution of edges among the three categories. We summarize the results of all three Noordin cases in Figure 3. The evidence from the Noordin cases supports the observation that as the average community size increases, the average conductance and cluster adequacy increases. We also consistently observe that \( \epsilon = w_1 + w_2 + w_3 \) produced a high volume of small and qualitatively poor communities. In general, the \( \epsilon = v \) threshold choices produced the poorest quality communities.

For subcases 1.1, 1.2, 1.3, and 1.9, the adjusted conductance, adjusted permanence, adjusted normalized cut, and the cluster adequacy values outperform the control case of Louvain method on the monoplex.

Recall that the control case also represents an equal dis-
tribution in weights amongst all of the different layers while collapsed in the monoplex. Thus, our results from case 1 indicate that we can increase community quality by employing our methodology. We also notice that case 3 slightly outperforms case 2. The highest adjusted conductance value when plotted in the weighted graph results from subcase 3.2. This suggests that heavily weighting a particular category, such as trust, potentially produces better quality communities as we see in the next two networks as well.

We now present a similar and concise analysis of the other two dark networks.

B. Boko Haram Results and Analysis

The Boko Haram Network is much sparser and more disconnected than the Noordin Network. However, each category contains a relatively equal amount of edges.

In Figure 4 we observe similar trends as the Noordin Network. We continue to observe that \( \varepsilon = w_1 + w_2 + w_3 \) provides the poorest quality communities as the intersection of the communities in all three categories. We observe that the cases where \( \varepsilon \leq \text{weight}_{\text{choice}} \) generally produce the best quality communities. However, we observe more variance in the ordering of the subcases between the cases. The general trend of community quality increasing with average community size continues for Boko Haram.

C. FARC Results and Analysis

In FARC network, we observe that some threshold cases did not result in communities as shown in Figure 5. This is a product of the edge distribution in the three categories. Since there are no communities in \( \varepsilon = w_1 + w_2 + w_3 \), this means that the same edge relationship does not exist in all three categories. We also observe a much more dramatic shift in the quality of communities as we transition from \( \varepsilon \geq \text{subcase} \) to \( \varepsilon \leq \text{subcases} \). The domination of the LOC category makes it difficult to produce quality communities when LOC is not included.

VI. Conclusions and Future Work

We introduced a purpose-driven community detection algorithm for multiplex networks that is user-engaged at multiple steps to develop analytically useful communities. The algorithm focuses on a user-defined goal, which directs the algorithm to select and combine layers appropriately in support of that goal. To test our algorithm, we used three dark network datasets from the NPS Common Operational Research Environment Lab, with a user-defined goal of network disruption. We specifically tailored the algorithm to reduce the effects of incomplete information on dark network analysis.

In total, we explored 81 subcases from our dark networks that included different weights and information threshold choices. Through our analysis of the three terrorist networks, we identified the following common observations. The different community quality metrics were relatively consistent in their evaluation of each subcase. This consistency indicates that the metrics could provide substantial evidence in determining the quality of the communities in the absence of ground truth. The control cases metric values were typically very high when compared to most subcases, yet several subcases still performed better.

Generally, as the average size of the communities increased, the values of our metrics increased as well. This indicates that fewer communities of larger size are optimal for the dark networks we have studied. However, the size of the community becomes irrelevant if the community is a component of the network, as it is for us with many singleton nodes. Communities that are also characterized as components have no external edges to the community, which results in perfect adjusted conductance.

The research shows that indeed \( \varepsilon = v \) produces many small and poor quality communities, as supported by the results of all three networks based on the metrics used. Placing this high restriction on the community development forced the communities to remain small, revealing their poor quality. The external connections that were ignored during the community development process of applying thresholds become very important in determining community quality.
In most cases, the subcases with the highest average adjusted conductance and cluster adequacy came from the threshold choice of $\varepsilon \leq v$, which was the most relaxed of all the thresholds and did not contain any misfit vertices. Community quality instead increased as the restriction of the threshold cases was relaxed. Under these relaxed conditions, every vertex was assigned to a community and no vertices were labeled misfits.

Based on our metrics, case 1 generally performed the best, yet the best overall subcase for Noordin was subcase 3.2. However, intuition still points to subcase 3.9 because it provides the necessary bias for the trust foundation category to dominate the remaining community information from the other categories.

Extensions to this algorithm can be considered for weighted networks, where the weight of each edge gets multiplied by the weight chosen for the whole layer in Step 4 of the algorithm.

ACKNOWLEDGMENTS

RM and RG thank the DoD for partially sponsoring the current research.

REFERENCES