Information Sciences 584 (2022) 387–398

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A network generator for covert network structures

Amr Elsisy [a,b], Aamir Mandviwalla [a,b], Boleslaw K. Szymanski [a,b,c,*], Thomas Sharkey [b,d]

[a] Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
[b] Network Science and Technology Center, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
[c] Społeczna Akademia Nauk, Łódź, Poland
[d] Department of Industrial Engineering, Clemson University, Clemson, SC 29631, USA

ARTICLE INFO

Article history:
Received 14 June 2021
Received in revised form 12 September 2021
Accepted 25 October 2021
Available online 6 November 2021

Keywords:
Social networks
Random weighted network generator
Network structure stability
Covert networks
Hierarchical networks

ABSTRACT

We focus on organizational structures in covert networks, such as criminal or terrorist networks. Their members engage in illegal activities and attempt to hide their association and interactions with these networks. Hence, data about such networks are incomplete. We introduce a novel method of rewiring covert networks parameterized by the edge connectivity standard deviation. The generated networks are statistically similar to themselves and to the original network. The higher-level organizational structures are modeled as a multi-layer network while the lowest level uses the Stochastic Block Model. Such synthetic networks provide alternative structures for data about the original network. Using them, analysts can find structures that are frequent, therefore stable under perturbations. Another application is to anonymize generated networks and use them for testing new software developed in open research facilities. The results indicate that modeling edge structure and the hierarchy together is essential for generating networks that are statistically similar but not identical to each other or the original network. In experiments, we generate many synthetic networks from two covert networks. Only a few structures of synthetics networks repeat, with the most stable ones shared by 18% of all synthetic networks making them strong candidates for the ground truth structure.

1. Introduction

The randomized generation of synthetic networks is often used for testing which properties of a given network depend on its structure and features [3,10]. Such a use has become popular since Erdós and Rényi introduced the random network model [11]. It generates an edge between each pair of nodes with a fixed probability p. The resulting networks are highly random as expected. Yet, despite interesting mathematical properties of this model, later research discovered that random networks rarely arise in nature or engineering practice.

The more advanced models proposed later include the scale-free network model [2] and its variants that represent the structures of many social and naturally arising networks. Another one is the Stochastic Block Model (SBM) [18] which extends the random network model by grouping nodes into blocks. The probability of an edge between a pair of nodes is determined by the probability of connection between the blocks to which the nodes belong. This model produces block structures resembling those arising in real networks but limits the differences among degrees of individual nodes located

[*] Corresponding author at: Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA.
E-mail address: szymab@rpi.edu (B.K. Szymanski).

https://doi.org/10.1016/j.ins.2021.10.066
0020-0255/© 2021 The Author(s). Published by Elsevier Inc.

in the same block. This weakness is addressed by SBM variants, such as the degree planted SBM [20] or the degree-corrected planted partition model [25].

The Lancichinetti-Fortunato-Radicchi (LFR) benchmark [21] generates synthetic networks with the desired heterogeneity of the node degrees and block size distribution. The generated networks are customized using such parameters as power law exponents for the node degrees, the block size, and the density of edges within the same block to mimic real networks. These networks are often used to test community detection methods. A model presented in [34] generates synthetic networks by rewiring edges in real-life networks. The more edges are rewired, the blurrier the blocks become, allowing only the increasingly more stable blocks to prevail.

The Hidden Community Detection (HICODE) model [17] identifies hidden communities by first performing a standard community detection method and then intentionally weakening the detected communities by removing or reducing the weight of edges. This allows weaker communities to be detected. Unlike our approach, the proposed method neither accounts for node hierarchy, nor preserves total edge connectivity in a rewired network. Another generative model creates hierarchical multi-layer networks that are often used to represent the hierarchical management structures, but does not consider reliability of a group or edge structure of the network [28].

In general, we are interested in on-line social networks that have been supported by the growing number of computer platforms and companies. The activities in such networks generate massive amounts of data, to which, however, access is increasingly limited due to privacy concerns. We are particularly interested in covert (a.k.a. hidden) networks, such as terrorist and criminal networks. The common trait of such networks is that their members try to hide their participation and illegal activities, as well as the network's structure. Often essential interactions of participants are covert, and their non-essential interactions overly visible, making essential ones difficult to observe. Law enforcement in most of the countries is limited in the means of collected data by privacy laws protecting citizens from unrestricted surveillance. As a result, the data about covert networks is often incomplete and partially incorrect. This creates a challenge in interpreting or discerning the structure and activities of such networks. An additional challenge arises from the inaccessibility to researchers of data about networks under investigations. Only for a fraction of networks whose members were prosecuted in the court, such data becomes publicly available.

The main contribution of the network generator introduced here is creating synthetic networks that are structurally similar to real networks, but with anonymous nodes that are interconnected or clustered differently than nodes in the original network. The direct use of such anonymized networks is to provide a safe but statistically equivalent substitute for real networks for research on analytical tools for covert networks. A large number of such networks generated from a single covert network can serve as a set of alternative interpretations of the structure of that network. The distribution of frequencies of these structures enables analysts to quantify statistically expected outcomes of operations on the covert networks. A high frequency of presence of a particular synthetic structure among synthetic networks makes it a likely candidate for the ground truth structure.

The rest of the paper is organized as follows. In Section 2, we overview the organizational aspects of covert networks and describe the ways in which our generative model accounts for them. In Section 3, we present the data sets defining two covert networks discussed in the paper and explain how these sets differ from data sets defining open social networks. The detailed description of the proposed method is presented in Section 4. The results and the methods used to compare the generated networks to the corresponding original network are discussed in Section 5. The conclusion is discussed in Section 6.

2. Modeling organizational aspects of covert networks

Discovery and monitoring of covert networks often rely on getting access to information flows among nodes suspected to be involved in network activities. This flow may involve wiretapped telephone interactions, message exchanges, recorded conversations and meetings, or copies of written documents. We refer to an *organization* represented by such a network as *covert* and to the nodes as *members*. Using community detection, we discover what we call *groups* of members who interact among themselves more often than with other members. We prefer this term over *communities*, which usually denotes the nodes having casual relationships, *clusters*, which refer to subsets of nodes with similar values for selected attributes, *blocks* that are used in the context of SBM model.

In small organizations, groups could be *independent* of the organizational structure of the network, but more often they have a *hierarchical* structure for management nodes. The number of hierarchy levels depends on the organization size. In covert networks, the hierarchical structure is important since law enforcement interdiction success vitally depends on correct recognition of the leadership roles in a crime organization. For this reason, our model explicitly assigns to each node its place in the organizational hierarchy. In addition, each group is represented by two parameters, one defining edge densities inside a group, and the other from this group to any other group. This extends also to densities of edges from a group leader to its subordinates, and, separately, to other nodes. Given an original network, the generator individualizes it by randomly rewiring edges of its groups and its management hierarchy [28]. As a result, our model combines two network models: SBM for groups and hierarchical network for higher level management structure.

Most of the previously proposed synthetic network generators rely only on one network generative model. For example, in [30] the authors propose a network generator that creates synthetic multi-layered networks. A network generator presented in [16] implements small-world social networks with the desired high clustering coefficient, while [23] presents a generator of a dynamic scale free network with the given exponent γ with the minimal divergence from the scale-free model at each size of its evolution. Another network generator described in [4] creates dynamic networks with the prescribed group structure.

3. Data and metrics

To present our generator in action, we use the Caviar and Ciel datasets that are publicly available [24]. The former defines a network representing the drug smuggling West End Gang in Montreal, Canada, using data collected from 1994 to 1996. This gang was active in trafficking hashish and cocaine. During the investigation, police repeatedly confiscated shipments of drugs, but made no arrests until the investigation ended. The collected data mainly consists of the lists of phone calls made among gang members. Every two months, investigators were creating a snapshot of the network, in which each edge represents the calls made between two nodes during the corresponding two months and the edge weight is equal to the number of these calls. In total 11 snapshots were collected. The snapshots allow us to observe the network reactions to the shipment confiscations, and to the changes in positions of members in the network occurring between snapshots.

In the case of the Caviar network, we know which nodes had some management roles from the publicly released court proceedings. For example, node N1 (see Fig. 1) was identified as the leader of the hashish group, and node N12 as the leader of the cocaine group. Fig. 1 shows more nodes in management roles in the Caviar groups. Still, the community detection algorithm alone did not assign many low degree nodes to any group [1]. We also present the results obtained with the Ciel dataset [24], which defines a network representing an illicit drug transportation crime organization that was engaged in smuggling hashish from Jamaica into Montreal. The data were collected from May 1996 to June 1997 and defines a network with weighted edges representing the volumes of nodes' interactions obtained from the records of telephone calls and con-

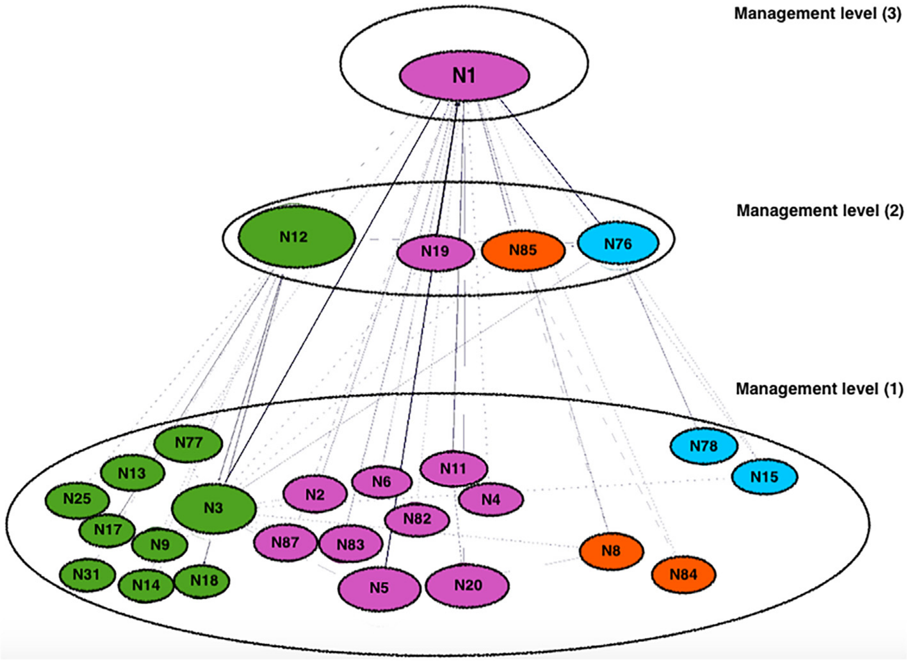

Fig. 1. Illustration of our model integration of the stochastic block model (SBM) with the hierarchical multi-layer network model in the Caviar network. Nodes marked with the same color belong to the same group. Nodes placed at the same hierarchy levels play the same roles in the management of this network.

Table 1
The relative betweenness centralities of the management nodes of the Caviar and Ciel networks reveal their different priorities in managing a criminal organization.

Network	Caviar				Ciel		
Node identifier	N1	N3	N12	N76	N1	N2	N10
RBC score	0.430	0.180	0.303	0.078	0.591	0.641	0.015
Rank of the score	1	3	2	4	2	1	7

versation surveillance. Only key managers of the network groups are identified. Nodes N1, N2, and N10 all took part in leading the network. None of the ground truth groups were listed.

For both networks, we use the Louvain community detection algorithm [6] to find groups and the relative betweenness centrality that we use to identify nodes involved in managerial roles. Since criminal networks often have sparse connectivity because of attempts to hide network activities, we execute a version of the Louvain algorithm for graphs with undirected edges. For this execution, we temporally transform directed edges of the generated networks into undirected ones, summing their weights for pairs of nodes which have two opposing edges connecting them.

Covert networks may prioritize either efficiency or security, but not both. The betweenness centrality of a node measures a fraction of the shortest paths of information flow between all pairs of nodes passing through this node [32]. A normalized version of this metric, called relative betweenness centrality, limits its range to [0,1]. We use it for ease of comparison of the results. We conclude that the structures of the Caviar and Ciel networks are fundamentally different. The Caviar network prioritized efficiency. This is indicated by the highest relative betweenness centrality scores of the management nodes, which are N1, leader, and N3, N12, N76, managers, among all nodes in the Caviar network (cf. Table 1). The implied easy access to these nodes from others supports high efficiency of information flow in the network, but not security for the managing nodes. In contrast, in the Ciel network, the leader has limited connections to direct subordinates. Consequently, the leader, N10, has a very low relative betweenness centrality (RBC) when compared to the manager nodes, N1 and N2, (see Table 1) even though some management role was plausible for him [24].

We chose the Caviar and Ciel datasets for our study because they were intensively analyzed using public data released during court proceedings [1,5,31]. We show that synthetic networks generated by our generator for both networks are similar to the original ones. This is important because only pieces of the ground truth are available for covert networks under investigation.

4. Implementation of the BWRN generator

4.1. Overview

The process of running the BWRN generator on a given network requires two inputs. The first assigns to each node its place in the management hierarchy, and if a node has a management position, a list of its direct subordinates, and its membership in a group. Nodes with a management position may not have peers, in which case it will be considered a single node group. The second input assigns to each edge identities of the groups to which its endpoints belong. This step creates proto-edges that are randomly assigned nodes from the relevant groups in the process of a synthetic network generation.

4.2. Applications of BWRN generator

The first application of our generator is to enable sharing data in sponsored research, where a sponsor collects and owns sensitive or proprietary data while the sponsored researchers need realistic networks to develop efficient tools for network analytics. Thus, data sharing requires proper separation and abstraction of structural information about the network from the sensitive personal and operational information. A data owner initiates this process by creating three lists of nodes, groups, and nodes' management roles. Then, the generator further transforms node ids into numeric, randomly assigned ids, clusters nodes into groups, and computes statistics about edges and their weights within and across those clusters. Hence, the description of the original networks passed to researchers is succinct and void of any personal data. The output of the generated networks follows the same format as is required for the input.

The generator may enable researchers to develop and test software for covert networks using the generated synthetic networks instead of inaccessible to them real data covert networks because of security and privacy concerns.

The most novel and important use of the generator is as a tool for network analytics on the original network. Studying the generated synthetic networks that are statistically similar to a given covert network is useful in two ways in new network analytics. Synthetic network structures that are infrequent among synthetic networks are unlikely to be close to the ground truth structure for which the frequent ones are good candidates. However, a large number of such candidates indicates the need for more data to decrease ambiguity about the analyzed covert network.

4.3. Generating groups in covert networks

Generating weighted networks has received a lot of attention over the last decade. Many of the proposed models aim to produce weighted networks that have properties similar to real social networks. In [36,33,19], weighted network models, based on Barabási's preferential attachment model [2], are proposed with the goal of generating weighted networks with power-law degree distribution, positive degree correlation, and high clustering coefficient. In [23], a growth model that preserves the power-law degree distribution with nodes dynamically joining and leaving the network is proposed. Another model that generates weighted networks that meet user defined criteria, such as symmetry, clustering, and positive degree correlation, is proposed in [29]. Rather than user defined criteria, generated networks can be based on real input networks, to generating weighted networks that maintain the power-law degree, degree correlation, and high clustering coefficient of the

Table 2
Notation table.

$B(n, p)$	Process of n Bernoulli trials with success probability p
d	Node degree
E	Number of directed edges
E^g	Maximum number of directed edges among nodes in group g
$E^{i,j}$	Number of directed edges from g_i to g_j
g_i	Group i
g_i^s, g_i^e	Groups with starting and ending nodes of edge i
$\neq gi$	The number of all nodes not in a group g_i, and hierarchy $< s(i)$
k	Weight of generated edge
n	Number of nodes in the network
$ns(n, p)$	Random number of successes in $B(n, p)$
$n(d)$	Number of nodes with d degree
p_a	$w - p_B w_B$
p_B	Probability defining the variance of the generated weights
$s(i)$	Node directly superior to node i
w, w_i	Weight of a single edge, weight of edge i
w_B	$\lfloor w/p_B \rfloor$
W	Vector of w_i's of a weight of edge i
W_S, W_U	Sum of weights of edges, sum of unique weights of edges
$W^{i,j}$	Sum of weights of edges from g_i to g_j

real networks [22,8]. In [13], Fortunato provides an authoritative review of the state of the art in community detection. The review covers many generative models and provides a more in-depth explanation of them for interested readers. It also demonstrates the poor performance of these models in replicating the community structure of the original network. All the discussed above models do not inform their rewiring processes about the hierarchy or group structure of the original network. Information about both of these aspects of a network enables the BWRN generator to rewire edges preserving internal connection densities within groups and between leaders and their subordinates. Hence, the synthetic networks created by the BWRN generator not only have an edge structure statistically similar to that of the original network, but also their group structures and organizational hierarchies are similar.

Social networks often have directed weighted edges to represent intensity of interactions measured in frequency of calls, messages, or meetings. Before our work, random weighted graphs were generated by WRG model [15]. Let W_S denote the sum of weights of all edges and E^g the maximum number of edges we can generate among subsets of nodes. In WRG, the edges' weights are generated by running Bernoulli trials with probability $p = \frac{W_S}{W_S + E^g}$. The first failed trial stops the runs. The number of successful trials before this failure defines the weight of the generated edge. This process gives rise to the geometric distribution of edge weights.

We introduce here a new approach, called *Bernoulli Weighted Random Network* (BWRN) model, see Algorithm 1 for pseudo code, and Table 2 for notation used. It uses two parameters. One is a vector of the weights w_i's of edges in the original graph sorted in descended weight order, and the other is the probability p_B that controls the variance of the generated weights. The process of generating edges starts with the first, heaviest weight and progresses down the vector toward the smallest weight. Given the currently processed weight w_i, an associated weight is computed as $w_B = \lfloor w_i/p_B \rfloor$. For each edge weight w_i in the original graph, we randomly select a weight in the range $[0, \lceil w/p_B \rceil]$.

This design yields a distribution of weights with probability of choosing weight k, where $0 \leqslant k \leqslant \lceil w/p_B \rceil$ defined as:

$$p_w(k, p_B) = \begin{cases} \frac{(w_B)!}{(w_B - k)!k!} p_B^k (1 - p_B)^{w_B - k} & \text{if } \mathrm{k} \leqslant \mathrm{w_B} \\ w - p_B w_B & \text{for } \mathrm{k} = \mathrm{w_B} + 1\mathrm{if} \mathrm{p_B} \mathrm{w_B} < \mathrm{w} \end{cases} \tag{1}$$

As the result, the average sum of weights of all edges created by this process is the same as in the original network. Indeed, the expected weight from Eq. 1 is $p_B w_B + (w - p_B w_B) = w$.

The parameter p_B defines probability that the edge of weight w will not be generated, which is $(1 - p_B)^{w_B}$ if $w = p_B w_B$ and $(1 - p_B)^{w_B}(1 - w + p_B w)$ otherwise, so it quickly decreases with increase of w and p_B. With $p_B > 0.9$ even edges with weight 1 have a low chance (< 1%) to be dropped. An interesting trade-off arises for slightly lower values of p_B. For example, with $p_B = 0.875$, which we used for computational experiments here, about 10% of such edges will not appear in the generated network but a similar fraction of edges will increase their weight to 2, strengthening cohesiveness of some groups.

The variance of the distribution of the weights generated for an edge with weight w in the original data is $w(1 - p_B) + p_a(p_B - p_a) \approx w(1 - p_B)$[12], so it grows with increase of w but decays with increase of p_B[1]. Thus, selecting large p_B will make generated synthetic networks more similar to the original one, while decreasing it would have opposite effects. This feature of the model is essential for covert networks, which have both missing and incorrect edges (e.g., edges connecting a member of a Crime Organization (CO) to a person outside of it). Analysts involved in CO investigations can use an estimate of

[1] Indeed $\sum_{k=0}^{w_B} k^2 p_k (1 - p_a) + k^2 \ * \ p_k p_a + (2k + 1) p_k p_a = w(1 - p_B) + w^2 + p_a(p_B - p_a)$. Hence $Var(p_B, w) = w(1 - p_B) + p_a(p_B - p_a)$

levels of this deception and select a value of p_B that would rewire enough edges to add the hidden connections and remove the incorrect ones, but not enough to disassemble the crime groups. For example, analysts can monitor whether the known CO members are losing connections to other members, which would indicate that p_B is too low.

Our model allows edges to be weighted to accommodate differentiation of edge densities within groups at the same level of hierarchy (e.g., among peers) than across the hierarchy (e.g., between the group leader and a subordinate). This enables modeling flexibly information flow intensities among managers and subordinates, to prioritize information flow efficiency among peers, or opposite to prioritize safety of managers.

4.4. Implementation of BWRN model

.

Algorithm 1:BWRN model

```
Input: vector $W$; group $g_i^s, g_i^e$; int $E$, real $p_B \in (0,1]$;
function $ns(n,p)$; process $B(n,p)$;
Comment: all symbols are defined in Table 2
Sort proto-edges in the order of groups they connect and their descending weights
  for $i$ = 1 to $E$ do
    $w_B = \lfloor w_i/p_B \rfloor$
    randomly select a pair of not yet connected nodes in groups $g_i^s, g_i^e$
    $w_i^g$ = ns($w_B, p_B$)
    if $p_B w_B < w$ then increase $w_i^g$ by ns(1, $w - p_B w_B$)
end if
end for
```

We consider two possible methods of computing a function $ns(w_B, p_B)$. The first method selects randomly a pair of not yet connected nodes and runs up to w_b Bernoulli trials with probability of success being p_B until the first failure. If w_b trials are successful and $p_B w_B < w$, the method runs one more Bernoulli trial with probability $p_a = w - p_B w_B$. The weight from such a run is equal to the number of consecutive successes in those trials. The edge is not created when the first Bernoulli test fails. The complexity is $O(E + W_S)$, where W_S is the sum of all weights in the network and E accounts for the trial that yields the first failure. This could be costly for large networks that are also dense.

The second method is to precompute the probabilities of the number of successes with Bernoulli trials. This needs to be done once for the given number of trials and the selected probability of a success. Since both of these values may be reused for several networks, the cost of precomputing may end up amortized over such networks. Its complexity is $O(E + W_U/p_B)$, where W_U is the sum of all unique weights in the network and E accounts for trials arising from inequality $0 \leqslant \lceil w_i/p_B \rceil - w_i/p_B < 1$. Most often, $O(W_U/p_B) \leqslant O(W_S)$. Yet in practical terms, in social networks only edges with low degrees repeat frequently, so the advantage may not be large. More gain can be expected from amortization. Given the precomomputed probabilities of success and the value selected uniformly randomly from the range of $(0, 1]$, a binary search for the number of successes corresponding to this random value has the complexity $O(E + \sum_W \log_2(w_i))$, where E arises because $0 \leqslant \lceil \log_2(w_i) \rceil - w_i < 1$. This is significantly lower than the complexity $O(W_S + E)$ of the first method.

4.5. Detecting groups in the generated networks

To test the accuracy of our synthetic network generator, we ran the Louvain community detection algorithm [6] on all the generated networks and compared the generated groups to the groups in the original network. We also tested the performance of our approach using another algorithm called SpeakEasy [14]. It yielded similar results, indicating that the results do not depend much on the selection of the specific community detection algorithm.

We observe that in the Caviar and Ciel networks, the original and generated groups often do not fully match. There are several reasons for such differences. First, datasets for covert networks are incomplete and may not have data about many undetected edges. Second, important nodes are often so highly interconnected that they may belong to several different groups. For example, in Caviar, nodes N1 and N3 belong to the same group in the original graph, but N3 has so many connections to other groups that the generator occasionally assigns it to them. For similar reasons such miss-assignments may happen to the managers. At the same time, such a miss-assignment may signal diverse roles that such a node plays in the network. Hence, comparative analysis of groups discovered in the synthetically generated networks may shed a new light on the operations of the original covert network.

We also detect the network hierarchy levels to reveal the structural properties of the generated networks. We use the relative betweenness centrality, which is easy to compute, as a hierarchy level measure because in networks there is a strong correlation between the hierarchy measures and betweenness centrality scores [27]. Comparing the nodes with high relative betweenness centrality in the generated network and such nodes in the original network enables us to measure how well the generated networks preserve leadership hierarchy.A Combined Score (CS) measures the overall similarity of generated networks to the original one, by taking the product of the Group NMI Score (GS) and Jaccard Leadership Score (LS), as defined in Eq. 2.

$$\mathrm{CS} = \mathrm{GS} * \mathrm{LS} \tag{2}$$

4.6. Steps for generating synthetic networks

The processing of the input data for our generator starts with computation of probabilities of existence and weights of edges for the generated network. We use two models to assign randomized weights to the weighted edges. Both methods classify the edges into several classes and generate edges and weights for each class separately. The first class includes internal edges of a group g_i of size $|g_i|$ at the same level of management hierarchy, so including members but not superiors of the neighbors. In such a case, there are $E^i = |g_i|(|g_i| - 1)$ directed edges. We denote the sum of their weights as W^i. The second class includes edges across the members of two different groups g_i, g_j. There are $E^{i,j} = |g_i||g_j|$ such edges from g_i to g_j and $E^{j,i}$ in the opposite direction. We denote the sums of their weights as $W^{i,j}, W^{j,i}$. The next class includes edges from a superior, $s(i)$, to the members of its group, g_i, and from the group members to this superior. There are $E^{s(i),i} = E^{i,s(i)} = |g_i|$ of such edges in each direction. Their sums of weights are denoted as $W^{s(i),i}, W^{i,s(i)}$. Let $| \neq gi|$ denote the number of all nodes not in a group g_i, and with lower hierarchy than s(i). We define the class of edges from the superior of group i to $\neq gi$ nodes as $E^{s(i),\neg i} = | \neq gi|$ in each direction, with the sum of weights denoted $W^{s(i),\neg i}$.

Next, the management roles are assigned to the nodes. Our model allows for an arbitrary number of management hierarchy levels. Yet, given that both Caviar and Ciel networks are composed of small groups, we set here the limit for their hierarchy levels at three. The number of nodes assigned to each hierarchy level higher than one depends on the total number of groups at the immediately lower level. The third level of management consists of the highest authority node in the local network. Recall that we refer to such a node as a *leader*, and to second level nodes as *managers*, while the first level of hierarchy is composed of the remaining nodes, called members, organized into groups. Managers serve as intermediaries between the leader and the members. The review of the literature reveals that the majority of small companies have a ratio of four employees per manager [9]. This is a bit smaller than in case of covert networks, such as Caviar, where this ratio is close to six. The plausible reasons for this difference include self-motivation of the members for doing their tasks, as well as keeping a fraction of the organization members interacting with the leader small for safety reasons.

Not all groups have to be supervised through hierarchical management. We refer to such a group as *independent*. A group of small size and a low fraction of reciprocal connections is likely to be independent. An example is the money-laundering group in the Caviar network that might be regarded as independent. Its members have only outgoing edges targeting the outside nodes. Thus, members of this group do not have incoming edges from any other node in the network.

4.7. Baseline network generator

To get a better baseline for network accuracy than the Erdös-Rényi random network, we built a straightforward extension of SBM by supporting weighted directed edges with integer weight. It takes as input lists of network nodes, edges with integer weights, and groups in the original network. The first step is to count, for each group, the sum of weights of all edges inside this group, and then edges of this group leading to and from each other group. These sums become probabilities by dividing them by the sum of weights of all the network edges. Using the built-in numpy [26] random choice method, the baseline generator picks a pair of groups, including those in which the source and target groups are the same. Then, a node from the source group and a node from the target group are selected repeatedly until the selected nodes are different, thus avoiding self-loops. Using the created set of probabilities, we select the probability for the selected group and execute a Bernoulli trial with this probability. On success of the trial, the weight of connection between these two nodes is increased by one. This entire process is repeated until the total weight of all edges becomes the same as in the input network. The last step is to run Louvain community detection on the resulting network and compare the output with the original network groups.

4.8. Flowchart and time complexity of the BWRN generator

Fig. 2 summarizes the process of generating synthetic networks using the proposed generator. Although we start with a single real network, the synthetic networks generated from it will all be unique. Some generated networks may have the same group structure, but they will all have a unique set of generated edges, so after node anonymization, it would be difficult to recover the original input network from the single synthetic network. Thus, the anonymized synthetic networks can be shared with researchers that operate in open research laboratories for testing and validating software on networks that are similar to real covert networks.

The time complexity of executing the BWRN generator presented in Fig. 2 can be established as follows. The input data consists of the management hierarchy, group structure, and the list of proto-edges of the network to be rewired and may be

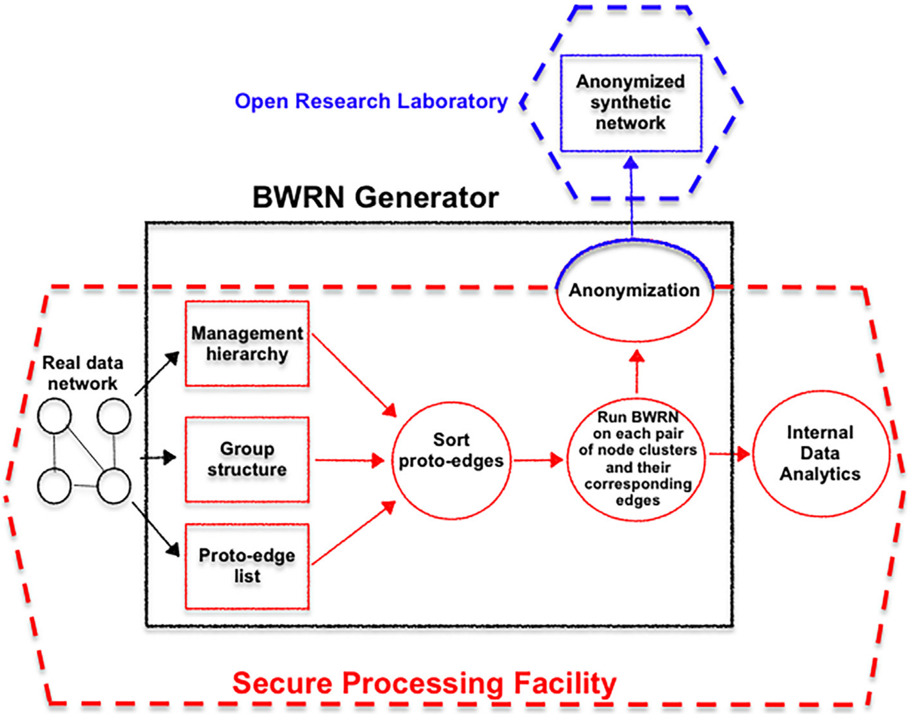

Fig. 2. The graphic framework showing the process of generating networks using the proposed generator. Circles represent the processing stages of the generator, rectangles stand for datasets, and hexagons show computational centers. The red color denotes secure processes and data with access limited to within the secure processing facility, while the blue color denotes open access data and facilities.

already included in the data associated with the real network to be rewired. Alternatively, the management hierarchy can be uncovered using the relative betweenness centrality with complexity of $O(n(n\log n + E)$. This is by factor of n the most computationally intensive part of the BWRN generator execution, but it can be avoided if ground truth about the management hierarchy is uncovered by investigation. The group structure could be found by Louvain (or any other community detection) algorithm with complexity $O(n\log n)$.

The network rewiring process consists of two main steps. The first sorts the network proto-edges according to groups they connect and in descending order of their weights. The complexity of sorting is $O(E\log E)$. The second step runs the BWRN generator, on every proto-edge of the rewired network. Since each proto-edge is rewired once, the complexity of this step is $O(E + \sum_{i\in W}\log(w_i))$, where w_i denotes the weight of edge i.

5. Results

We test the network generator by generating $T = 1000$ synthetic networks and taking the average scores to avoid any statistically insignificant outliers. We use the Louvain community detection algorithm to find the group structure of each generated network. Then, using the Normalized Mutual Information (NMI) metric [7], we compare groups detected in the generated networks to groups in the corresponding original network.

5.1. Results with caviar and ciel datasets

To fully evaluate how similar are the synthetic networks to the original network used as input to the generator, we measure not only similarity of group structure, but also similarity of identified management nodes. We find all management nodes by their high relative betweenness centrality in all compared networks. We use as threshold 90% of the relative betweenness centrality metric of the node whose rank among the highest values of this metric is equal to the number of management nodes in the original network. For both datasets, we used weighted edge generators to create three sets of synthetic networks. For the first set we used our Bernoulli Weighted Random Network model, for the second set, the Weighted Random Graph model, while for the last set, the weighted SBM described in Section 4.7.

Table 3 contains results of measurements of the first aspect of similarity that synthetic networks generated by BWRN model are by factor of two more similar to original group structures than groups in networks generated with WRG model. The BWRN and weighted SBM baseline models performed close to each other.

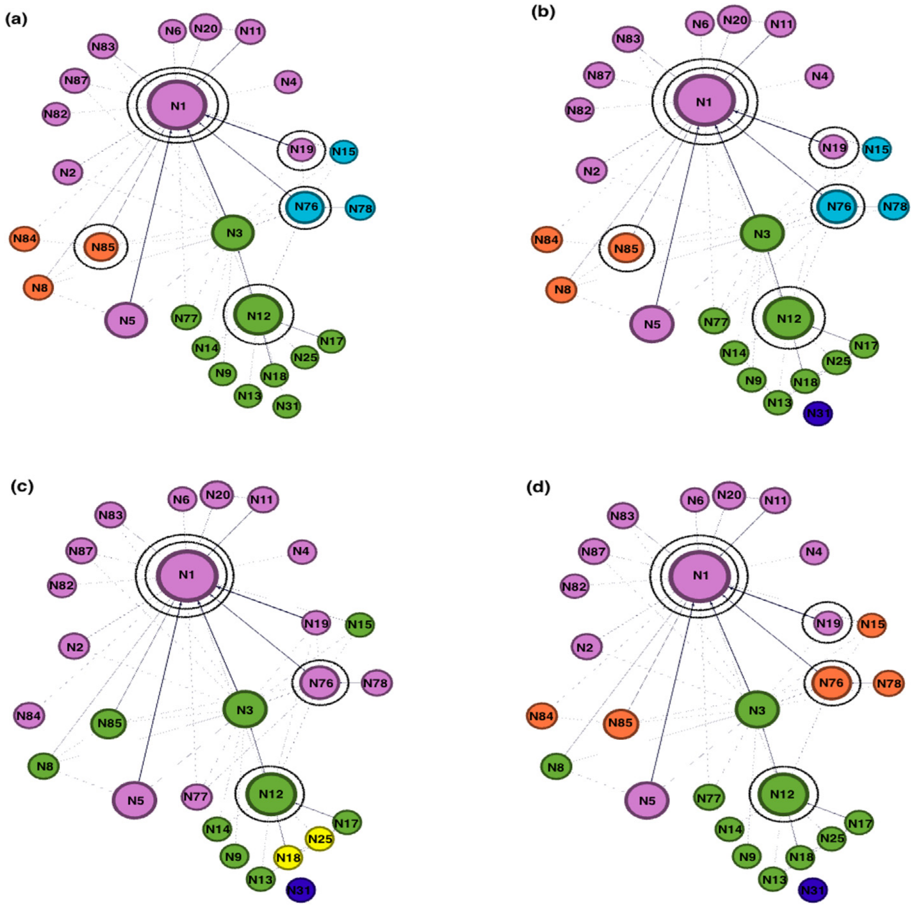

Fig. 3. The presented Caviar networks depict a) the groups in the original network and those detected in the synthetic networks with similarity that is (b) highest, (c) lowest, and (d) average.

Table 3
Results show NMI scores from comparing the groups in networks generated from the Caviar and Ciel networks to the groups in the original networks. The results show performance of BWRN generator, and generators using the Weighted Random Graph model and the weighted SBM baseline model.

Original Network	Caviar			Ciel		
Generator	Generator		Weighted	Generator		Weighted
	BWRN	WRG	SBM	BWRN	WRG	SBM
Mean	0.815	0.356	**0.850**	**0.800**	0.513	0.761
Median	**0.839**	0.365	0.739	**0.883**	0.567	0.751
Min	0.415	0.088	0.358	0.734	0.288	0.551
Max	0.953	0.565	0.883	1.000	0.692	1.000

This means that the management structure is a differentiating factor in comparing these two generators. It is evaluated in Table 4 that demonstrates the importance of leadership detection for keeping the synthetic distinct but close to the original network. The results obtained using Jaccard metric [35] show that the networks generated using our Bernoulli Weighted Random Network model preserves the group structure and leadership hierarchy twice as well as the baseline generated networks and nearly four times better than those using Weighted Random Graph model. Moreover, the unweighted SBM generator performs much worse than the baseline with the weighted SBM. In summary, our BWRN generator better differentiates between roles of leaders and members within groups while performing at least slightly better than others on community detection. These results verify the strength of our generator in preserving groups and hierarchy within the generated networks.

Table 4
The first row of the results shows group structure similarity from Table 3, the second the leadership similarity, and the last row shows the combined Score that is the product of the first two scores. In all three rows, the best score is shown in bold font.

Metric	BWRN	WRG	Weighted SBM
Group NMI median (GS)	**0.839**	0.365	0.739
Jaccard Leadership (LS)	**0.681**	0.402	0.308
Combined Score (CS)	**0.571**	0.146	0.281

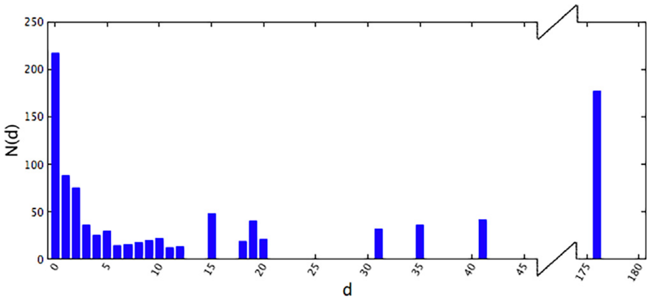

Fig. 4. The histogram of meta-graph degree distribution for exact matching of groups in the generated networks. The x-axis defines the node degree d, while the y-axis shows $n(d)$, the number of nodes with d degree.

5.2. Stability of group structures of covert networks

Generating networks statistically similar to the investigated real network enables us to analyze how stable is the structure of this network. These networks represent sets of small perturbations of the original network interconnections. If many generated synthetic networks are structurally similar to the original network, the latter network is stable and resistant to perturbations. On the other hand, if the structure of the original network is not stable, only a few synthetic networks will be structurally similar to the real network.

We apply this analysis to the Caviar network, by generating $G = 1000$ random synthetic networks, and comparing their structures to each other, and to the structure of the Caviar original network. We start this analysis by creating a meta-graph, nodes of which are the generated networks, so the size of the meta-graph is $G = 1000$ nodes. For each node, we draw an undirected edge between this node and any other node in the meta-graph with a matching group structure. We create two versions of the meta-graph. In one, edges represent an exact matching. In another, the edges show flexible matching, which allows up to one node difference in each group for drawing an edge. Fig. 4 shows the resulting distribution of the node degrees in meta-graph with exact matching. We find that the original network's structure repeats ten times for exact matching among 1000 generated synthetic networks. Thus, this structure is not very stable. It is sensitive to small perturbations in node connectivity in the generated networks. In contrast, we also find that the most frequent group structure occurs 177 times with exact matching, and 310 times for flexible matching. It also happens to be the most similar to the group structure of the Caviar graph. The synthetic network with this structure is shown in Sub-Fig. 3(b). Moreover, each of the top ten most frequently occurring structures repeats at least 20 times for exact matching, and at least 206 times for flexible matching. The remaining generated networks have either unique structures, or structures that were similar to only a few generated networks.

From a practical perspective, the identification of stable groups in a criminal organization is important. It will allow analysts to concentrate on group structures that arise frequently and thus represent plausible interpretations of data collected about the network. Using the meta-graph of generated networks, analysts can find such frequent structures.

6. Conclusions

We first introduce the *BWRN* model capable of dealing with weighted and directed edges. It generates networks whose total weight of edges is close to that weight in the original graph, and yet their numbers of edges can differ. It creates a theoretical basis for *BWRN* generator whose novelty includes the ways we generate synthetic networks, and the ways we use them for network anonymization and new network analytics. The generator uses the Stochastic Block Model to create group structures alternative to those existing in the original network. To preserve the organizational aspects of the original network, our generator uses hierarchical network model.

We thoroughly tested the generator on two real covert networks, Caviar and Ciel. To measure the similarity between the generated networks and the original one, we use the well-known Louvain community detection algorithm. Applying the NMI and the Jaccard metrics, we measured the results, which demonstrate the high levels of similarity among generated networks and the original one.

We conclude that accounting for groups and a management hierarchy of a network is essential for generating synthetic networks that are statistically similar to the original network.

In future work, we plan to add the fourth step of network generation going beyond the original graph to enable its *hierarchical network expansion*. One way to accomplish it is to regenerate, using BWRN generator, any part of the original network multiple times at any level of hierarchy and provide additional levels of management hierarchy if needed. Such an extension will allow the researchers and analysts to study the evolution of covert networks.

CRediT authorship contribution statement

Amr Elsisy: Conceptualization, Methodology, Software, Investigation, Writing - original draft, Writing - review & editing. **Aamir Mandviwalla:** Methodology, Software, Investigation, Writing - original draft, Writing - review & editing. **Boleslaw K. Szymanski:** Conceptualization, Methodology, Writing - original draft, Writing - review & editing, Supervision, Project administration. **Thomas Sharkey:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the U.S. Department of Homeland Security under Grant Award Number 2017-ST061-CINA01, the Office of Naval Research (ONR) Grant N00014-15-1-2640, the Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO) under Contract W911NF-17-C-0099, and the Army Research Office (ARO) Grant W911NF-16–1-0524. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security or U.S. Department of Defense.

References

[1] A. Bahulkar, B.K. Szymanski, N.O. Baycik, T.C. Sharkey, Community detection with edge augmentation in criminal networks, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 1168–1175.

[2] A.L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509–512.

[3] A.L. Barabási, R. Albert, H. Jeong, Scale-free characteristics of random networks: the topology of the world-wide web, Physica A 281 (2000) 69–77.

[4] O. Benyahia, C. Largeron, B. Jeudy, O.R. Zaïane, Dancer: dynamic attributed network with community structure generator, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer (2016) 41–44.

[5] G. Berlusconi, Do all the pieces matter? assessing the reliability of law enforcement data sources for the network analysis of wire taps, Glob. Crime 14 (2013) 61–81.

[6] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech.: Theory Exp. 2008, P10008..

[7] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech.: Theory Exp. 2005, P09008..

[8] M.C. Davis, Z. Ma, W. Liu, P. Miller, R. Hunter, F. Kee, Generating realistic labelled, weighted random graphs, Algorithms 8 (2015) 1143–1174.

[9] B. Davison, Management span of control: how wide is too wide?, J Bus. Strategy 22 (2003) 22–29.

[10] A. Elsisy, B.K. Szymanski, J.A. Plum, M. Qi, A. Pentland, A partial knowledge of friends speeds social search, PLoS ONE 16 (2021) e0255982.

[11] P. Erdös, A. Rényi, On random graphs. I, Publ. Math. 6 (1959) 290–297.

[12] W. Feller, An Introduction to Probability Theory and Its Applications, 3 ed., Willey, New York, NY, 1968.

[13] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (2010) 75–174.

[14] C. Gaiteri, M. Chen, B. Szymanski, K. Kuzmin, J. Xie, C. Lee, T. Blanche, E. Chaibub Neto, S.C. Huang, T. Grabowski, T. Madhyastha, V. Komashko, Identifying robust communities and multi-community nodes by combining top-down and bottom-up approaches to clustering, Sci. Rep. 5 (2015) 16361.

[15] D. Garlaschelli, The weighted random graph model, New J. Phys. 11 (2009) 073005.

[16] W. Guo, S.B. Kraines, A random network generator with finely tunable clustering coefficient for small-world social networks, in: 2009 International Conference on Computational Aspects of Social Networks, IEEE, 2009, pp. 10–17.

[17] K. He, Y. Li, S. Soundarajan, J.E. Hopcroft, Hidden community detection in social networks, Inf. Sci. 425 (2018) 92–106.

[18] P.W. Holland, K.B. Laskey, S. Leinhardt, Stochastic blockmodels: first steps, Soc. Netw. 5 (1983) 109–137.

[19] B. Hu, X.Y. Jiang, J.F. Ding, Y.B. Xie, B.H. Wang, A weighted network model for interpersonal relationship evolution, Physica A 353 (2005) 576–594.

[20] B. Karrer, M.E. Newman, Stochastic blockmodels and community structure in networks, Phys. Rev. E 83 (2011) 016107.

[21] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.

[22] P. Li, J. Yu, J. Liu, D. Zhou, B. Cao, Generating weighted social networks using multigraph, Physica A 539 (2020) 122894.

[23] X. Lu, E. Bulut, B. Szymanski, Towards limited scale-free topology with dynamic peer participation, Comput. Netw. 106 (2016) 109–121.

[24] C. Morselli, Inside criminal networks, vol. 8, Springer, 2009.

[25] M.E. Newman, Equivalence between modularity optimization and maximum likelihood methods for community detection, Phys. Rev. E 94 (2016) 052315.

[26] T.E. Oliphant, A guide to NumPy, vol. 1, Trelgol Publishing USA, 2006.

[27] S. Rajeh, M. Savonnet, E. Leclercq, H. Cherifi, Interplay between hierarchy and centrality in complex networks, IEEE Access 8 (2020) 129717–129742.

[28] E. Ravasz, A.L. Barabási, Hierarchical organization in complex networks, Phys. Rev. E 67 (2003) 026112.

[29] D. Shanafelt, K. Salau, J. Baggio, Do-it-yourself networks: a novel method of generating weighted networks, R. Soc. Open Sci. 4 (2017) 171227.

[30] K.K. Shang, B. Yang, J.M. Moore, Q. Ji, M. Small, Growing networks with communities: a distributive link model, Chaos 30, 041101, 2020..

[31] D.B. Skillicorn, Q. Zheng, C. Morselli, Spectral embedding for dynamic social networks, in: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2013, pp. 316–323.

[32] S.K.R. Unnithan, B. Kannan, M. Jathavedan, Betweenness centrality in some classes of graphs, Int. J. Comb. 2014 (2014) 241723.

[33] G. Wen, Z. Duan, G. Chen, X. Geng, A weighted local-world evolving network model with aging nodes, Physica A 390 (2011) 4012–4026.

[34] J. Xiao, H.F. Ren, X.K. Xu, Constructing real-life benchmarks for community detection by rewiring edges, Complexity 2020, 2020..

[35] M.J. Zaki, W.J. Meira, Data Mining and Machine Learning: Fundamental Concepts and Algorithms, second ed., Cambridge University Press, Cambridge, U.K., 2020.

[36] Y.B. Zhou, S.M. Cai, W.X. Wang, P.L. Zhou, Age-based model for weighted network with general assortative mixing, Physica A 388 (2009) 999–1006.