GA-Based Q-Attack on Community Detection

Jinyin Chen, Lihong Chen, Yixian Chen, Minghao Zhao, Shanqing Yu, Qi Xuan, *Member, IEEE*, and Xiaoniu Yang

***Abstract*— Community detection plays an important role in social networks, since it can help to naturally divide the network into smaller parts so as to simplify network analysis. However, on the other hand, it arises the concern that individual information may be overmined, and the concept community deception has been proposed to protect individual privacy on social networks. Here, we introduce and formalize the problem of *community detection attack* and develop efficient strategies to attack community detection algorithms by rewiring a small number of connections, leading to privacy protection. In particular, we first give two heuristic attack strategies, i.e., Community Detection Attack (CDA) and Degree Based Attack (DBA), as baselines, utilizing the information of detected community structure and node degree, respectively. Then, we propose an attack strategy called "genetic algorithm (GA)-based Q-Attack," where the modularity Q is used to design the fitness function. We launch community detection attack based on the above three strategies against six community detection algorithms on several social networks. By comparison, our Q-Attack method achieves much better attack effects than CDA and DBA, in terms of the larger reduction of both modularity Q and normalized mutual information (NMI). In addition, we further take transferability tests and find that adversarial networks obtained by Q-Attack on a specific community detection algorithm also show considerable attack effects while generalized to other algorithms.**

***Index Terms*— Adversarial network, community detection, community detection attack, genetic algorithm, modularity, privacy protection, social network.**

I. Introduction

COMPLEX networks can well represent various complex systems in our daily life, such as social networks [1], [2], biological networks [3], power networks [4], and financial networks [5]. Network science has been a very active field because of its highly interdisciplinary nature [6], problems such as link prediction [7], node classification [8], network reconstruction [9], or community detection [10] have been widely studied. As one hot topic in this field, community detection has attracted great attention of researchers from various disciplines [11]–[13]. Detecting communities in the network has played an important role in getting a deep understanding of its organizations and functions.

However, with the rapid development of community detection algorithms in recent years [14], a new challenging problem arises: information overmined. People realize that some information, even their own privacy, is going to be over-mined by those social network analysis tools. The web we browse, the people we contact with and more things like that are all recorded through the Internet. Our interests, circle of friends, partners on business can be easily detected once these data are utilized [7], [15]. Community hiding or deception [16]–[18] is put forward to hide certain community. Still, community detection attack is launched on the network to change some of its structure thereby making the performance of related algorithms worse and the accuracy of results lower.

The problem of making communities in the network more difficult to detect through attack strategies against detection algorithm is worth studying because the good attack strategies can be a guideline for individuals or organizations to disguise their social circles and change the way they interact with others, lest some privacies get leaked [17]. In this paper, we focus on investigating the attack strategy and testing the effectiveness of these strategies through experiments. In particular, we propose two heuristic attack strategies to address the problem shown in the last paragraph. The heuristic strategy is vital because it is easy to implement and plays well sometimes. During the design process of heuristic attack strategies, we take community detection algorithms and some properties of complex networks into consideration. One question is about what kind of nodes to be chosen as targets may benefit the attack effect. Typically, node centrality is a common-used index to measure the importance of nodes in the network. It mainly includes degree, betweenness, and closeness. Compared with betweenness, whose calculation desires global information, degree is a quantity only based on local information with less computation [19]. Thus, we propose an attack strategy based on degree and take some tests.

Furthermore, we consider the design of attack strategy as an optimization problem. Due to the good performance in solving complex optimization problems, evolutionary algorithms, such

Manuscript received October 21, 2018; revised March 5, 2019; accepted April 7, 2019. Date of publication May 14, 2019; date of current version June 10, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61502423 and Grant 61572439, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LR19F030001 and Grant LY19F020025, in part by the Zhejiang Science and Technology Plan Project under Grant LGF18F030009, and in part by the Key Technologies, System and Application of Cyberspace Big Search, Major Project of Zhejiang Lab under Grant 2019DH0ZX01. *(Corresponding author: Qi Xuan.)*

J. Chen, S. Yu, and Q. Xuan are with the Institute of Cyberspace Security, College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China, and also with the Zhejiang Lab, Hangzhou 311121, China (e-mail: chenjinyin@zjut.edu.cn; yushanqing@zjut.edu.cn; xuanqi@zjut.edu.cn).

L. Chen, Y. Chen, and M. Zhao are with the Institute of Cyberspace Security, College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: coffeeclh@163.com; yichixchen@163.com; yzbyzmh1314@163.com).

X. Yang is with the Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 310023, China, and also with the Science and Technology on Communication Information Security Control Laboratory, Jiaxing 314033, China (e-mail: yxn2117@126.com).

Digital Object Identifier 10.1109/TCSS.2019.2912801

as genetic algorithm (GA), have been widely used in various disciplines, including complex networks [20]–[22]. Searching for attack schemes that make the performance of community detection algorithms decrease most under given attack cost is a typical combinatorial optimization problem. We propose an attack strategy based on GA, namely, *Q-Attack*, where the modularity Q is used to design the fitness function and show its effectiveness on different community detection algorithms and several real-world networks. The major contributions of our work are summarized as follows.

1) First, we introduce and formalize community detection attack problem, which is to implement global community deception and privacy protection.
2) Second, we propose two heuristic attack strategies and GA-based Q-Attack method to launch attack with negligible rewiring to cheat community detection algorithms.
3) Third, we conduct comprehensive experiments to compare the attack effects of different attack strategies against six community detection algorithms on several real-world networks.
4) Finally, the transferability of Q-Attack is validated and we think that community detection attack can also be a robust evaluation metric for anti-attack capacity comparison of community detection algorithms.

The rest of this paper is organized as follows. Section II reviews the related work on community detection and attacks on complex network. Section III presents our attack strategies for destroying community structure in the network, including heuristics and optimization methods. We demonstrate the experiments and results in Section IV and finally draw conclusions in Section V.

II. Related Work

A. Community Detection Algorithms

A large number of community detection algorithms have been proposed since the problem brought up [14]. Detection algorithms continue to spring up because of the diversity of networks in our real world. Here, we mainly give a brief introduction of some ones.

Girvan and Newman [23] first proposed a community detection algorithm based on the betweenness of edges and experiments showed that it did a good job in small networks. Some researchers made changes in this study and proposed new ones, e.g., Tyler *et al.* [24] got approximate betweenness using the fast algorithm of Brandes, Radicchi *et al.* [25] used clustering coefficient of edges instead so that only local information was required and the efficiency of the algorithm was improved. In addition, Newman [26] also proposed a greedy algorithm that mainly aimed at optimizing modularity. CNM was an improvement of Newman's method, which was proposed by Clauset *et al.* [27]. The main technology used in this algorithm was the data structures that called stacks. Futhermore, Blondel *et al.* [28] divided the modularity optimization problem into two levels and the computational complexity was essential linearly. Spectral clustering algorithms [29], [30] leveraged the eigenvalue spectrum of several graph matrices to detect the community structure in networks.

Except these methods based on modularity optimization, researchers also considered the problem of community detection from other perspectives. Label propagation algorithm (LPA) was proposed by Raghavan *et al.* [31], whose main idea was updating the label of each node according to its neighbors. Rosvall and Bergstrom [32] proposed an algorithm called Infomap, which was based on the information theory. This algorithm found out the partition for the network through optimizing the average length of description $L(M)$. Furthermore, Ronhovde and Nussinov [33] proposed a community detection algorithm based on the minimization of the Hamiltonian of Potts model.

B. Attacks on Complex Networks

Several earlier research studies have studied the influence of attacks on the network performance. Holme *et al.* [19] proposed four strategies to investigate the attack vulnerability of different kinds of networks, including *ID removal*, *IB removal*, *RD removal*, and *RB removal*, both for nodes and links. ID removal represented removing nodes (or links) according to the descending order of degrees in the initial network, whereas "B" represented betweenness and "R" represented recalculation. In this study, it was validated that the network structure changed and some performances degraded via attacks. Bellingeri *et al.* [34] found that node deletion according to betweenness centrality was most efficient while using the size of the largest connected component (LCC) as a measure of network damage. As for the community structure, Karrer *et al.* [35] studied the robustness of community structure in networks through perturbing networks in a specific way. They constructed a random network with the same number of nodes and links as the original one, then replaced the links in the original network with that in the random one with the probability of α. These studies shed lights on the further work about taking attacks on networks.

Adversarial attacks against some network algorithms are emerging in recent years. Yu *et al.* [36] developed both heuristic and evolutionary methods to add subtle perturbations to original networks, so that some sensitive links would be harder to predict. Dai *et al.* [37] proposed three methods to modify graph structure, including *RL-S2V*, *GradArgmax*, and *GeneticAlg*, aiming at fooling the classifiers based on graph neural network models. Moreover, Zügner *et al.* [38] proposed the first adversarial attack on networks using graph convolutional network, namely, *NETTACK*, and found that this method was effective on node classification task. Quite recently, Bojchevski and Günnemann [39], as well as Chen *et al.* [40], proposed the adversarial attacks on network embedding at the same time, which may promote the extensive discussion on the robustness of network algorithms, since network embedding establishes a bridge between network space and Euclidean space and thus facilitates many downstream algorithms, such as node classification and link prediction.

C. Researches on Antidetection of Community

As for privacy concerns aroused by community detection, some researchers started paying attention to this problem

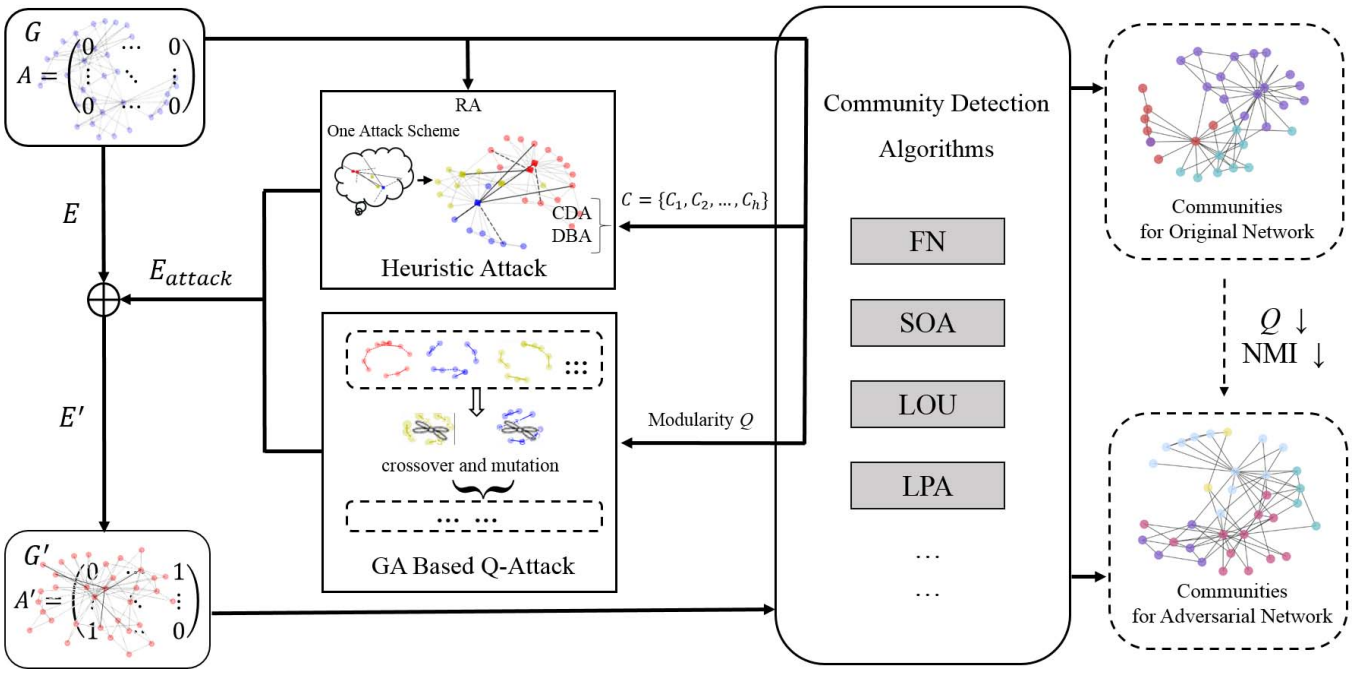


Fig. 1. Framework of community detection attack. We get the attack scheme E_{attack} combined with community detection algorithms. Two metrics Q and NMI are used to evaluate the attack effect.

during the past 2 years. Waniek *et al.* [17] came up with two heuristic algorithms remove one, add many and disconnect internally, connect externally (DICE) for hiding individuals and communities, respectively. DICE implemented through disconnecting d intralinks of given community C and connecting $b - d$ interlinks under budget b and a measure was proposed to evaluate the concealment. Fionda and Pirro [18] proposed the concept of community deception to cope with community detection. Safeness-based deception algorithm and modularity-based deception algorithm were mainly tested through amounts of experiments. Another job they did was defining the deception score to evaluate effects of proposed methods. Similar to the measure proposed in [17], deception score also took community spread and hiding into account and it was more comprehensive while reachability was also considered. Extensive distribution of members in C and less percentages of $C's$ for each $C_i \in \overline{C}$ showed a good hiding, where $\overline{C}$ was the community structure detected by specific community detection algorithm and C_i was one element in $\overline{C}$.

There is still little attention that has been paid on the problem of taking attacks on the network to make it harder for algorithms to detect the community structure. Both Waniek's and Finoda's research studies are dedicated to hide only a small set of nodes (community C in the broad sense) targetly. In this paper, we address the problem of information overmined in community detection from a global perspective. We aim at making the overall quantification of the communities detected in networks decrease significantly with subtle network changes, which shows the effectiveness of our attack strategies.

III. Methods

First, we briefly demonstrate the major problem we focus on in this paper, using related notations and definitions. Let $G = (V, E)$ be the original network, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $E = \{e_1, e_2, \ldots, e_m\}$ is the set of links. $E_{\text{attack}} = \{+\widetilde{e}_1, -\widetilde{e}_2, \ldots, +\widetilde{e}_{2T-1}, -\widetilde{e}_{2T}\}$ is the solution that comprised of a series of links, "+" represents adding the link to the network while "−" represents removing. Then, we have the adversarial network $G' = (V', E')$ defined as follows:

$$\begin{cases} V' = V \\ E' = E + E_{\text{attack}}. \end{cases} \tag{1}$$

We thus want to find out such E_{attack} to change some connections in G and get the adversarial network G'. For these $G's$, the community detection algorithms perform significantly worse, i.e., the quality of detection results decreases.

Fig. 1 shows the overview of our work. In the following, we give the details of attack strategies for community detection algorithms proposed in this paper, including heuristics and evolutionary ones.

A. Heuristic Attack Strategy

We have introduced that we take link attacks on the network to change some relations between nodes. In this paper, we choose *rewiring* attack to maintain the degree of target nodes, i.e., adding a link to the target node while deleting one from it. Such rewiring attack is of relatively less cost, i.e., only two nodes need to change their degree under a rewiring attack, whereas four nodes may change their degree if we choose to remove a link between two nodes and add a link between another two at the same time. In social networks, one rewiring attack means getting a false friend while hiding a real one for the person chosen as the target node, the total number of friends of this person keeps unchanged, such attack way can

Algorithm 1: RA

Input: G,K,T
Output: G'

$V_t \leftarrow RandomK(G, K)$;
$AttackNum = 0$;
for $AttackNum < T$ **do**
  $v_i \leftarrow RandomOne(V_t)$;
  $N_i, \overline{N_i} \leftarrow GetSet(G, v_i)$;
  if $N_i \neq \emptyset$ *and* $\overline{N_i} \neq \emptyset$ **then**
    $v_{del} \leftarrow RandomOne(N_i)$;
    $v_{add} \leftarrow RandomOne(\overline{N_i})$;
    remove $< v_i, v_{del} >$ from E and add $< v_i, v_{add} >$ to E;
    Update G;
    $AttackNum = AttackNum + 1$;
  end
end
Get the adversarial network G' under T rewiring attacks.

Algorithm 2: CDA

Input: G, K, T
Output: G'

$C = \{C_1, C_2, \ldots, C_h\} \leftarrow Detection(G)$;
$V_t \leftarrow RandomK(G, K)$;
$AttackNum = 0$;
for $AttackNum < T$ **do**
  $v_i \leftarrow RandomOne(V_t)$;
  $S_1, S_2 \leftarrow GetSet(G, v_i, C)$;
  if $S_1 \neq \emptyset$ *and* $S_2 \neq \emptyset$ **then**
    $v_{del} \leftarrow RandomOne(S_1)$;
    $v_{add} \leftarrow RandomOne(S_2)$;
    remove $< v_i, v_{del} >$ from E and add $< v_i, v_{add} >$ to E;
    Update G;
    $AttackNum = AttackNum + 1$;
  end
end
Get the adversarial network G' under T rewiring attacks.

maintain the person's status in the entire network, i.e., rewiring shows a good concealment of taking attacks on the network.

While thinking of attack strategies, we first come up with a random attack (RA) strategy, described in Algorithm 1, for comparison. Here, we randomly select K nodes from G to form a target node set V_t. We then randomly select a node $v_i \in V_t$ in each iteration and implement the following operation: deleting an existent link while adding nonexistent one for the target node. The neighbor set of the target node v_i is denoted by $N_i = \{v_j | < v_i, v_j > \in E\}$, while its nonneighbor set is denoted by $\overline{N_i}$. The number of rewiring attacks (or the number of iterations), denoted by T, is another parameter that represents the attack cost. The procedure of RA strategy is shown in following pseudocode.

A network with community structure often shows the characteristic of having a high density of intra-community links and a lower density of intercommunity links. Deleting links within community and adding links between communities, thus, can weaken the community structure in the network. Therefore, having some knowledge of the community structure in advance can help to design more effective attack strategies. Based on this, we propose a heuristic attack strategy combined with the community division results via specific community detection algorithm, called Community Detection Attack (CDA), the procedure of which is shown in Algorithm 2.

For target node v_i, we define its intracommunity neighbor set as $S_1 = C_i \cap N_i$, where C_i is the node set of the community that v_i belongs to. Similarly, we can get its intercommunity nonneighbor set $S_2 = \overline{C_i} \cap \overline{N_i}$, with $\overline{C_i} = V - C_i$.

Fig. 2 illustrates how the CDA strategy works graphically. For the target node, the removal of intracommunity links weakens its connection with the original organization and the addition of intercommunity links generates a force to drag it to the others.

It was found that many real-world networks followed power-law degree distribution [41], consistent with the classic 80/20 rule. This means, usually, a small number of nodes

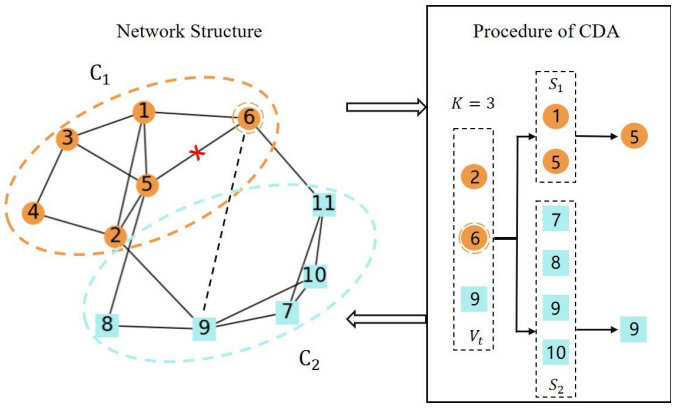


Fig. 2. C_1 and C_2 represent the communities detected by a specific community detection algorithm and different colors of nodes represent different communities they belong to. Node 6 is chosen from the target node set V_t as the target node under the current attack. S_1 and S_2 represent the intracommunity neighbor set and intercommunity nonneighbor set of the target node, respectively. The line with a red "×" represents the deleted link and the dashed line represents the added link, performed by the attack.

possess a large number of connections, whereas the rest only have a few in real-world networks. Some changes in these hub nodes of larger degree usually have a bigger impact on the whole network structure. For instance, statistically, people with more friends in social networks always plays more important roles in the circle and their absence seems to make the party less active. Therefore, we propose another heuristic attack strategy, namely, Degree Based Attack (DBA), aiming to attack the nodes of large degree in the network.

The only difference between DBA and CDA is that, in DBA, we choose the nodes of large degree as our target nodes, whereas in CDA, we do it randomly. Thus, next, we just explain how to choose the K target nodes in DBA based on node degree, since the rest attack part is totally the same as CDA. The procedure to choose the target nodes based on their degree in DBA is given in Algorithm 3. We get

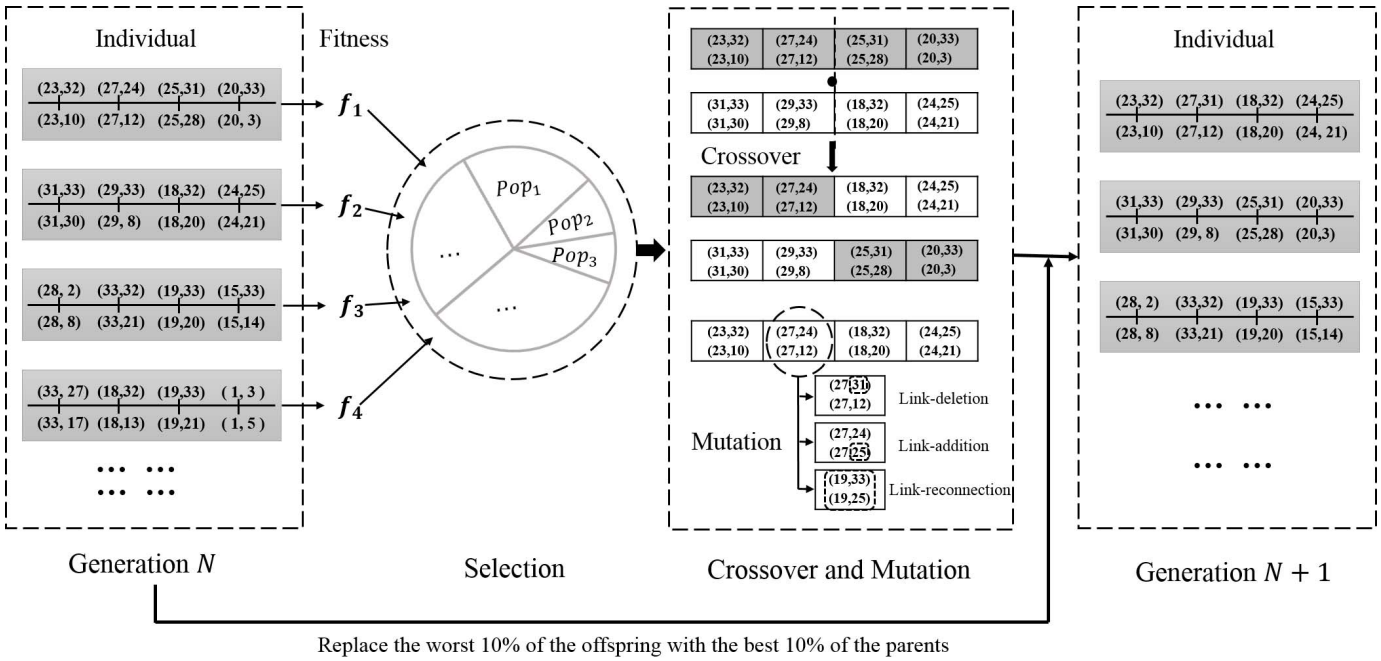


Fig. 3. Overview of one iteration of GA-based Q-Attack Strategy.

Algorithm 3: Target Nodes Chosen Based on Degree

```
1  C = {C_1, C_2, ..., C_h} ← Detection(G);
2  for len(KNode) < K do
3  |  v ← BiggestDegree(V);
4  |  add node v to KNode;
5  |  V = V − C_i, where v ∈ C_i;
6  |  if K > h then
7  |  |  update V every turn;
8  |  |  repeat from step BiggestDegree;
9  |  end
10 end
```

the community detection results $C = \{C_1, C_2, \ldots, C_h\}$ at the beginning of the algorithm. The function *BiggestDegree* (Line 3) gets the node with the biggest degree from node set V and the procedure in Line 5 means removing the nodes belonging to C_i from V. One turn is finished when we have iterated through all communities in C and gotten the node with the biggest degree in each community. The step *update V every turn* (Line 7) means removing the nodes that have been chosen as targets from the initial node set V, which consists of all the nodes in the network.

B. GA-Based Q-Attack

Evolutionary algorithms are a type of intelligent optimization technologies designed by simulating natural phenomena or leveraging various mechanisms of living organisms in nature [42]–[44]. GA is a typical one of such algorithms. Its basic model was first introduced and investigated by Holland [42], which mainly leveraged the natural selection mechanism to solve optimization problems. A population of chromosomes, which represent individuals with different characteristics, initialize at the beginning of the algorithm and then generate offspring according to the designed genetic operators, such as crossover and mutation. Individuals with better fitness have more chances to survive and multiply so that the population is able to evolve. Here, we propose an attack strategy based on GA to search effective rewiring attack schemes. Fig. 3 shows an overview of the evolutionary process.

Before the implementation of GA, we first give our encoding method and fitness function.

1) *Encoding:* Same as the attack strategies proposed in Section III-A, here we choose a rewiring attack as a gene, including a deleted link and an added link. The length of each chromosome represents the number of attacks.
2) *Fitness Function:* Modularity is an important evaluation to assess the quality of a particular partition for the network and it has been widely applied in the field of community detection (more details in Section IV-C). Here, we design our fitness function as follows:

$$f = e^{-Q} \tag{2}$$

indicating that individuals of lower modularity will have larger fitness. This is also the reason why we name this attack strategy as *Q-Attack*.

Now, we are going to find out individuals that can make the partition of an adversarial network under specific community detection algorithm have lower modularity. In the following, we will give a specific illustration about how GA operates in community detection attack.

1) *Initialization:* An initial population is randomly generated at a fixed size in GA. One thing we need to take care is to avoid link deletion or addition conflict,

which means deleting or adding a link repeatedly. Each individual in the population is a combination of *rewiring* attacks, representing a solution of attacks on the network.

2) *Selection:* Selection is a manifestation of the *survival of the fittest* mechanism. According to the laws of nature's evolution, the individuals of higher fitness have a greater chance to survive and multiply than those of lower fitness. We use a common selection operator, namely, *roulette wheel*. In this selection way, the probability for an individual to mating is proportional to its fitness, as represented by the following equation:

$$p_i = \frac{f(i)}{\sum_{j=1}^{n} f(j)}. \tag{3}$$

3) *Crossover:* Here, we adopt a single point crossover that is the easiest to achieve in GA, i.e., we generate a cut point at random and change the gene segments between parents. The probability for paired individuals to generate offspring through crossover operation is denoted by P_c. In order to avoid the conflicts on genes (as mentioned before, two identical deleted links or added links are not allowed to appear on one chromosome), we do a feasibility test before two individuals get crossed.
4) *Mutation:* To prevent the solution from falling into a local optimum, mutation operator is a necessary component in GA. Considering the network characteristics, here, we propose three kinds of mutation operators: link-deletion mutation, link-addition mutation, and link-reconnection mutation. Link-deletion or link-addition mutation indicates that the deleted or added link of the gene changes independently, whereas the link-reconnection mutation indicates that a gene is completely replaced. These three mutations occur with equal probability and the overall mutation rate is denoted by P_m. Such designs on mutation operators can promote the diversity of population. Conflicts detection is also considered while the genes are mutating.
5) *Elitism:* Over the course of evolution, individuals with high fitness in the parent may be destroyed. Elitism strategy has been proposed to solve this problem. In this paper, we replace the worst 10% of the offspring with the best 10% of the parent to maintain excellent genes.
6) *Termination Criteria:* We set the evolutionary generation as a constant and the algorithm stops when this condition is fulfilled.

In summary, GA-based Q-Attack mainly involves the encoding, fitness function, and the design of genetic operators. The pseudocode of this attack strategy is given in Algorithm 4.

Algorithm 4: Q-Attack Based on GA

Input: G,$PopSize$,$Generation$,P_c,P_m,T
Output: Pop

```
ParentPop ← Inatialization(G, PopSize, T);
while i < Generation do
    SelectedPop ← Selection(ParentPop, PopSize);
    CrossoverPop ← Crossover(SelectedPop, PopSize, T, P_c);
    MutationPop ← Mutation(G, CrossoverPop, PopSize, T, P_m);
    OffspringPop ← Elistism(MutationPop, ParentPop);
    ParentPop ← OffspringPop;
    i ← i + 1;
end
```

IV. Experiments

A. Data Sets

In order to evaluate the attack effect of our attack strategies, we test them on four real-world data sets, including Zachary's karate club network [45], bottlenose dolphin network [46], American college football network [23], and American political books network [29], which are commonly used for community detection with ground-truth partitions.

1) *Zachary's Karate Club Network:* It is a data set about the relationship in a karate club recorded by Zachary. The network consists of 34 nodes and 78 links, which is ultimately divided into two groups out of the conflicts between the instructor and the administrator.
2) *Bottlenose Dolphin Network:* This network is created by Lusseau, demonstrating the associations of the dolphins living off Doubtful Sound, New Zealand. The network consists of 62 nodes and 159 links. In reality, these dolphins are divided into two groups.
3) *American College Football Network:* This data set is about the college football matches in one season. There are 115 nodes and 613 links in the network. The nodes that represent the college football teams belong to 12 communities.
4) *American Political Books Network:* The network consists of 105 nodes and 441 links. Each node in the network represents a book published on political topics and the link between two books shows that they are purchased by the same consumer. These books are divided into three categories according to their political stands.

B. Community Detection Algorithms

Here, we introduce six well-known community detection algorithms that are attacked by our strategies proposed in Section III. Note that these algorithms are mainly used for detecting nonoverlapping community structures.

1) *Fast Newman (FN) Algorithm [26]:* It is a hierarchical agglomerative algorithm. Each node is considered as a community at the beginning and we merge communities in pairs repeatedly with the aim of optimizing Q, until all the nodes are merged into one community.
2) *Spectral Optimization Algorithm (SOA) [29], [30]:* Through analyzing the spectral properties of networks,

Newman gives a reformulation of modularity

$$Q = \frac{1}{4m}\boldsymbol{s}^T\left(A_{ij} - \frac{k_i k_j}{2m}\right)\boldsymbol{s} = \frac{1}{4m}\boldsymbol{s}^T\boldsymbol{B}\boldsymbol{s} \tag{4}$$

where m is the number of links, A_{ij} is the element of adjacency matrix A, $\boldsymbol{s}$ is a column vector with each element s_i representing the community label of node v_i, and k_i and k_j are the degrees of nodes v_i and v_j, respectively. $\boldsymbol{B}$ is called modularity matrix. The communities that nodes belong to depend on the signs of the elements in the leading eigenvector of the modularity matrix.

3) *Louvain Algorithm (LOU) [28]:* This algorithm considers the community detection as a multilevel modularity optimization problem. It starts with isolated nodes and repeats the process of removing the node to the community that obtains the maximum gain of modularity until no further improvement can be achieved. Then, it constructs a new network by merging the community into a supernode. These two steps are performed repeatedly until the algorithm is stable.
4) *Label Propagation Algorithm (LPA) [31]:* Each node is assigned a label at the beginning of the algorithm and it is updated as the most frequent label of the node's neighbors during the information propagation process. The algorithm terminates when no node changes its label anymore.
5) *Infomap Algorithm (INF) [32]:* This algorithm is based on network maps and coding theory. It detects communities by compressing the description of information flows on networks. In other words, the key of Infomap is to minimize the average description length $L(M)$ since more information is generated when random walks between communities occur.
6) *Node2vec + KMeans (Node2vec + KM) [47]:* Node2vec is one of the state-of-the-art embedding methods and it learns low-dimensional representation for nodes in the network. The K-means clustering algorithm is then used to detect communities when nodes are embedded into Euclidean space through node2vec.

C. Metrics for Community Structure

1) *Modularity:* It is widely used to measure the quality of divisions of a network, especially for the networks with unknown community structure, which was first proposed by Newman and Grivan [48] and is defined by

$$Q = \sum_i \left(e_{ii} - a_i^2\right) \tag{5}$$

where e_{ii} represents the fraction of links in the network with two terminals both in cluster C_i, $a_i = \sum_j e_{ij}$ represents the fraction of links that connect to the nodes in cluster C_i. In other words, modularity Q measures the difference between actual fraction of within-community links and the expected value of the same quantity with random connections. A reformulation of the modularity is represented by (4).

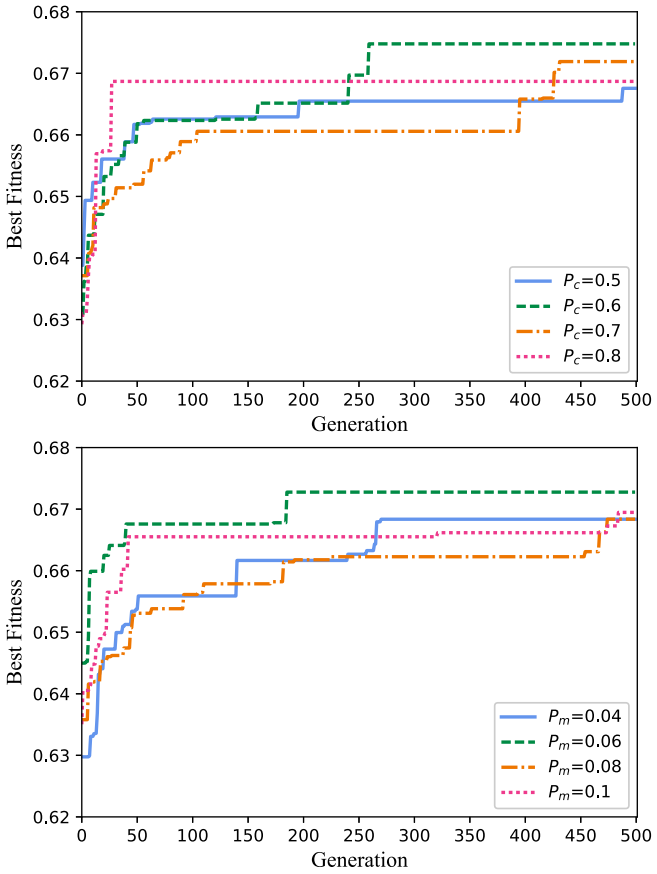


Fig. 4. Curves of best fitness for various values of parameters P_c and P_m when Q-Attack is applied to FN algorithm on Dolphins network.

2) *Normalized Mutual Information (NMI):* It is another commonly used criterion to assess the quality of clustering results in analyzing network community structure, which was proposed by Danon *et al.* [49]. For two partitions X and Y, the mutual information $I(X, Y)$ is defined as the relative entropy between the joint distribution $P(XY)$ and the production distribution $P(X)P(Y)$

$$\begin{aligned} I(X, Y) &= D(P(X, Y)||P(X)P(Y)) \\ &= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \end{aligned} \tag{6}$$

A noticeable problem for mutual information alone being a similarity measure is that subpartitions derived from Y by splitting some of its clusters into small ones would have same mutual information with Y. NMI, thus, is proposed to deal with this problem, which is defined by

$$I_{\text{norm}}(X, Y) = \frac{2I(X, Y)}{H(X) + H(Y)}. \tag{7}$$

The value of NMI indicates the similarity between two partitions, i.e., the larger value means more similar between the two. When NMI equals 1, partition X is identical to partition Y.

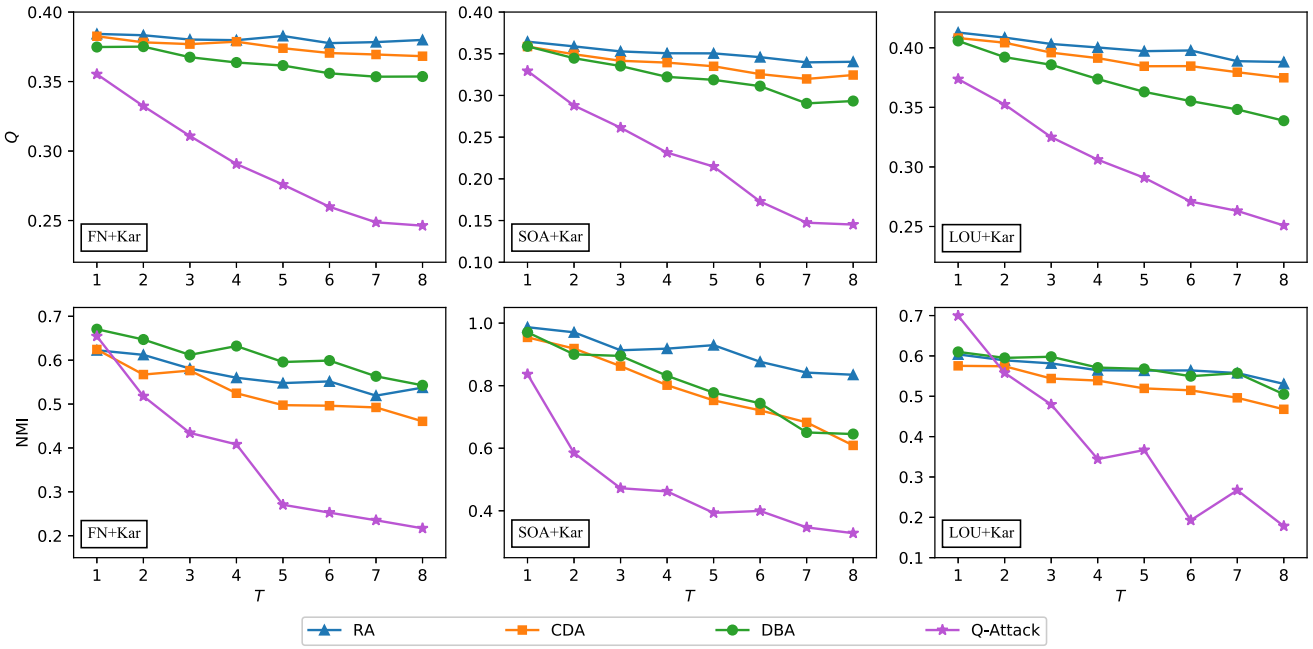


Fig. 5. Attack effects of the four attack strategies on three community detection algorithms and Karate network for various numbers of attacks.

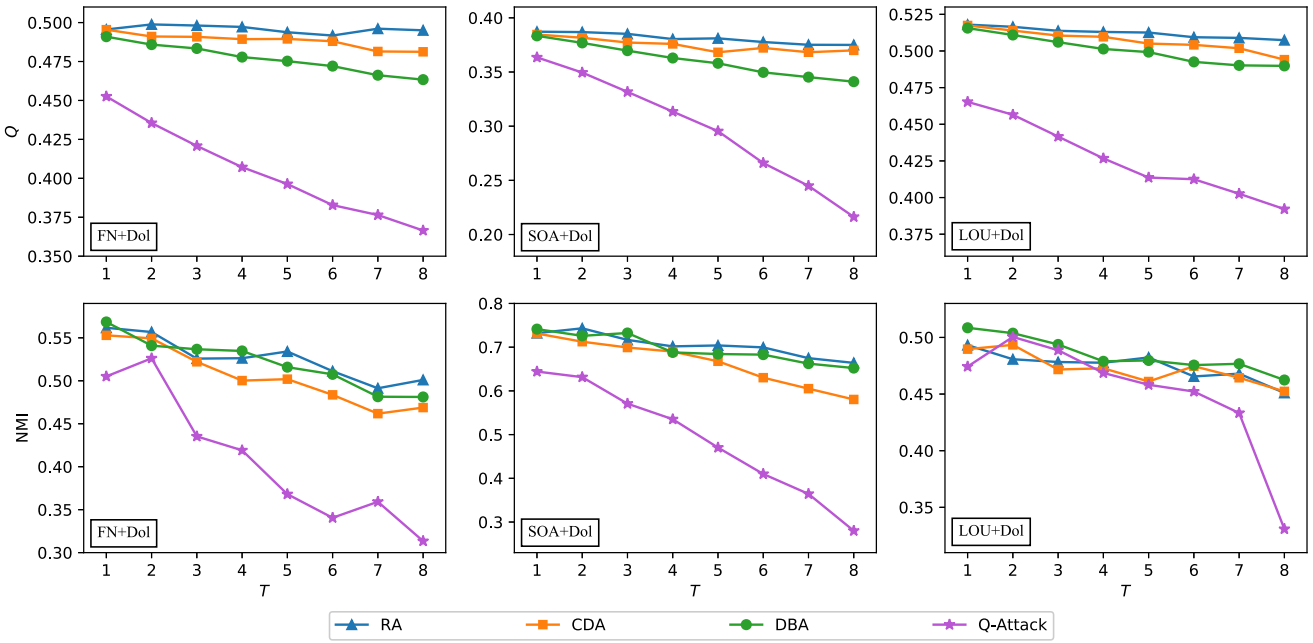


Fig. 6. Attack effects of the four attack strategies on three community detection algorithms and Dolphins network for various numbers of attacks.

TABLE I
OPTIMAL PARAMETERS FOR Q-ATTACK IN DIFFERENT CASES

	FN+Kar	FN+Dol	SOA+Kar	SOA+Dol	LOU+Kar	LOU+Dol
P_c	0.8	0.6	0.7	0.6	0.7	0.8
P_m	0.1	0.06	0.1	0.04	0.1	0.1

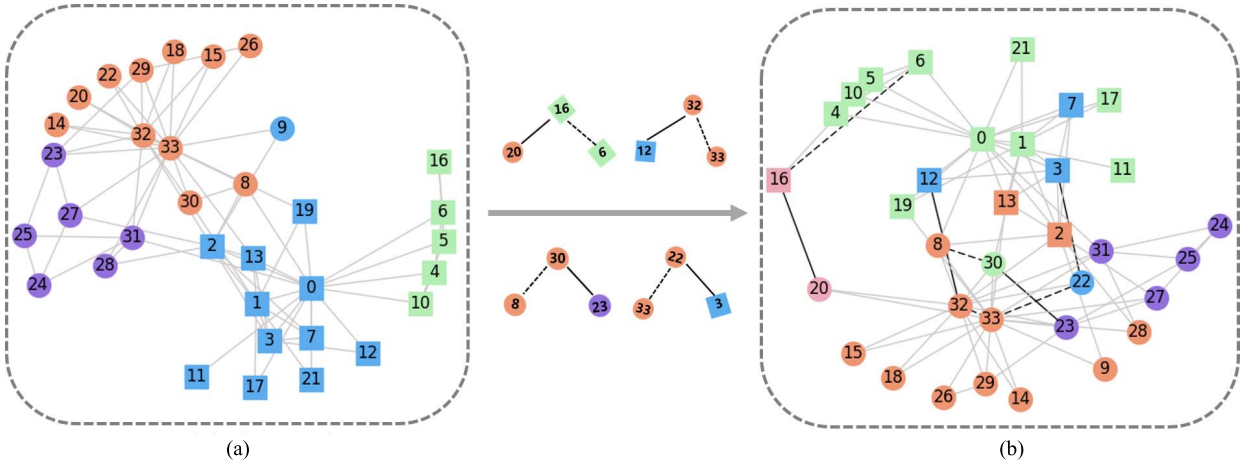
D. Experimental Results

Our experiments are performed on a PC i5 CPU 2.60-GHz and 4-GBs RAM. Our programming environment is python 2.7. We first test our attack strategies on three community detection algorithms and two networks with different attack number T. For our GA-based Q-Attack strategy, it concludes four parameters. We set the population size PopSize = 100 and the evolutionary generation Generation = 500 uniformly. Then, we tune the crossover rate P_c and the mutation rate P_m from four different values (0.5, 0.6, 0.7, 0.8 for P_c and 0.04, 0.06, 0.08, 0.1 for P_m), according to the results for parameter configuration. Fig. 4 shows the curves of best fitness for a set of experiments when Q-Attack is applied to FN algorithm on Dolphins, and we get the optimal

TABLE II

ATTACK RESULTS WHEN THE ATTACK NUMBER IS SET TO A FIXED PERCENTAGE OF LINKS IN THE NETWORKS. HERE, WE ONLY PRESENT THE RELATIVE REDUCTION OF Q AND NMI, COMPARED WITH THEIR VALUES IN THE ORIGINAL NETWORK WITHOUT ATTACK

Karate	Q							NMI						
	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average
Original	0.381	0.371	0.419	0.380	0.402	0.368	0.387	0.692	1	0.587	0.806	0.699	0.951	0.789
RA	0.24%	5.62%	4.43%	18.13%	3.64%	6.47%	6.42%	19.14%	8.16%	3.79%	27.58%	9.43%	13.30%	13.57%
CDA	0.53%	8.64%	6.56%	23.89%	12.93%	9.78%	10.39%	**24.23%**	**19.84%**	**8.14%**	**34.45%**	**23.76%**	**23.37%**	**22.30%**
DBA	**4.46%**	**13.24%**	**10.75%**	**26.48%**	**15.93%**	**14.53%**	**14.23%**	8.71%	16.84%	2.65%	33.12%	7.33%	19.22%	14.65%
Q-Attack	**23.64%**	**37.70%**	**26.92%**	**36.95%**	**74.28%**	**43.74%**	**40.54%**	**41.06%**	**53.81%**	**41.30%**	**35.54%**	**68.07%**	**63.92%**	**50.62%**
Dolphins	Q							NMI						
	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average
Original	0.495	0.390	0.519	0.506	0.524	0.379	0.469	0.573	0.753	0.511	0.607	0.553	0.889	0.648
RA	0.09%	3.81%	2.15%	9.83%	3.19%	3.23%	3.72%	12.52%	11.84%	**11.71%**	**5.28%**	13.47%	9.22%	10.67%
CDA	2.88%	5.10%	4.71%	17.74%	6.76%	4.20%	6.90%	**18.14%**	**22.94%**	11.45%	1.47%	**16.53%**	**19.41%**	**14.99%**
DBA	**6.49%**	**12.51%**	**5.53%**	**18.26%**	**8.72%**	**11.52%**	**10.50%**	15.98%	13.41%	9.48%	2.41%	15.80%	14.35%	11.91%
Q-Attack	**26.04%**	**44.53%**	**24.38%**	**22.63%**	**12.84%**	**19.59%**	**25.00%**	**45.27%**	**62.83%**	**35.24%**	**6.66%**	**15.97%**	**39.93%**	**34.32%**
Football	Q							NMI						
	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average
Original	0.550	0.493	0.605	0.603	0.601	0.598	0.575	0.698	0.699	0.890	0.888	0.924	0.922	0.837
RA	-0.05%	-3.26%	1.89%	4.59%	1.94%	1.91%	1.17%	-4.26%	-5.38%	1.94%	1.08%	0.00%	0.05%	-1.09%
CDA	**1.86%**	-0.65%	2.97%	5.31%	3.16%	3.25%	**2.65%**	-1.18%	**1.63%**	**2.64%**	**1.60%**	**0.12%**	0.24%	**0.84%**
DBA	0.56%	**-0.03%**	**3.16%**	**5.47%**	**3.22%**	**3.28%**	2.61%	**0.13%**	-1.49%	2.44%	1.49%	**0.11%**	**0.39%**	0.51%
Q-Attack	**20.36%**	**19.97%**	**11.02%**	**9.12%**	**3.43%**	**4.59%**	**11.42%**	**27.10%**	**34.23%**	**16.01%**	**6.93%**	-0.20%	**2.68%**	**14.46%**
Polbooks	Q							NMI						
	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average
Original	0.502	0.467	0.520	0.495	0.523	0.500	0.501	0.531	0.520	0.512	0.586	0.493	0.564	0.534
RA	1.26%	1.66%	1.76%	2.01%	1.59%	1.62%	1.65%	0.11%	0.71%	1.60%	4.74%	0.10%	1.41%	1.45%
CDA	2.95%	**3.96%**	2.75%	3.89%	2.64%	3.05%	3.21%	**2.89%**	1.01%	**3.50%**	7.94%	**1.07%**	**5.00%**	**3.57%**
DBA	**3.80%**	2.44%	**3.02%**	**5.30%**	**3.29%**	**3.31%**	**3.53%**	0.72%	**2.50%**	-1.17%	**9.10%**	-1.50%	2.20%	1.98%
Q-Attack	**22.17%**	**16.35%**	**17.92%**	**5.40%**	**7.48%**	**5.82%**	**12.52%**	**40.53%**	**21.29%**	**37.06%**	**13.02%**	**0.56%**	**7.97%**	**20.07%**



(a) (b)

Fig. 7. (a) Communities detected by Louvain algorithm on Karate network. (b) Communities detected by Louvain algorithm on the adversarial network. Different colors of nodes represent different communities they belong to, whereas different shapes represent different communities these nodes belong to according to the given label.

parameters $P_c = 0.6$ and $P_m = 0.06$ for this case, under $T = 4$, which is a medium value. Table I presents all the optimal parameters we use for Q-Attack in six different situations here. For heuristic attack strategies proposed in Section III-A, the numbers of target nodes for Karate and Dolphins are set to 5 (15% nodes) and 6 (10% nodes), respectively.

TABLE III

ANALYSIS ON THE TRANSFERABILITY OF Q-ATTACK. HERE, WE USE THE ADVERSARIAL NETWORKS, GENERATED BY Q-ATTACK ON PARTICULAR COMMUNITY DETECTION ALGORITHMS TO ATTACK OTHER ALGORITHMS

Dataset	Network	FN	SOA	LOU	LPA	INF	Node2vec+KM	Average	Average(-self)
Karate	Q-Attack(FN)	**23.64%**	**13.84%**	10.29%	36.91%	20.88%	18.33%	20.65%	**20.05%**
	Q-Attack(SOA)	1.00%	**37.70%**	4.52%	27.98%	14.49%	**25.00%**	18.45%	14.60%
	Q-Attack(LOU)	10.29%	12.72%	**26.92%**	31.57%	**34.67%**	15.14%	**21.89%**	**20.88%**
	Q-Attack(LPA)	1.19%	9.52%	7.59%	36.95%	6.11%	12.76%	12.35%	7.43%
	Q-Attack(INF)	**10.78%**	13.65%	**15.23%**	**43.69%**	**74.28%**	13.68%	**28.55%**	19.41%
	Q-Attack(Node2vec+KM)	0.35%	13.27%	6.04%	**40.50%**	11.14%	**43.74%**	19.17%	14.26%
Dolphins	Q-Attack(FN)	**26.04%**	10.55%	7.32%	**25.17%**	7.48%	9.48%	**14.34%**	**12.00%**
	Q-Attack(SOA)	3.92%	**44.43%**	5.12%	14.54%	6.25%	**24.35%**	**16.43%**	10.83%
	Q-Attack(LOU)	8.08%	10.85%	**24.96%**	19.82%	**10.24%**	9.89%	13.97%	11.78%
	Q-Attack(LPA)	5.61%	7.73%	7.06%	**22.63%**	9.15%	7.27%	9.91%	7.36%
	Q-Attack(INF)	**9.01%**	9.83%	**9.42%**	**22.63%**	**12.84%**	9.50%	12.20%	**12.08%**
	Q-Attack(Node2vec+KM)	-0.49%	**11.45%**	1.53%	21.90%	3.12%	**19.59%**	9.52%	7.50%
Football	Q-Attack(FN)	**20.36%**	-1.78%	**2.85%**	5.58%	2.87%	3.19%	**5.51%**	2.54%
	Q-Attack(SOA)	0.44%	**19.97%**	2.81%	**5.83%**	2.59%	3.06%	**5.78%**	**2.95%**
	Q-Attack(LOU)	**3.13%**	**-0.34%**	**11.02%**	4.54%	**3.08%**	3.38%	4.14%	**2.76%**
	Q-Attack(LPA)	0.62%	-9.52%	2.48%	**9.12%**	2.75%	3.27%	1.45%	-0.08%
	Q-Attack(INF)	1.69%	-9.65%	2.70%	3.99%	**3.43%**	**4.10%**	1.04%	0.57%
	Q-Attack(Node2vec+KM)	-0.69%	-5.69%	2.32%	5.25%	2.35%	**4.59%**	1.35%	0.71%
Polbooks	Q-Attack(FN)	**22.17%**	2.92%	2.28%	4.48%	2.76%	**3.13%**	**6.29%**	3.11%
	Q-Attack(SOA)	4.41%	**16.35%**	2.46%	3.74%	2.26%	2.70%	5.32%	3.11%
	Q-Attack(LOU)	3.49%	4.20%	**17.92%**	**5.58%**	**3.13%**	2.90%	**6.20%**	**3.86%**
	Q-Attack(LPA)	**8.98%**	3.79%	1.12%	**5.40%**	2.38%	3.05%	4.12%	**3.86%**
	Q-Attack(INF)	2.45%	**6.51%**	**5.10%**	4.67%	**7.48%**	2.53%	4.79%	**4.25%**
	Q-Attack(Node2vec+KM)	1.62%	2.44%	2.29%	4.38%	2.42%	**5.82%**	3.16%	2.63%

Modularity Q and NMI introduced in Section IV-C are taken as the metrics to evaluate effectiveness of attack strategies. The results are shown in Figs. 5 and 6. Corresponding to each specified T, we generate 50 adversarial networks for each heuristic strategy and record the mean values of Q and NMI. Each point on the curve of Q-Attack is the mean value over ten runnings with T changing from 2 to 8. While for $T = 1$, crossover operator described in GA is hard to implement. Here, there are two solutions for this problem: 1) we can perform an internal crossover for one *rewiring* attack, i.e., swapping out the deleted link or the added one and 2) we can get the best result by going through all possible situations. Here, we get the results according to the second solution, since the two networks here we use are both small and, thus, it only takes little time to get the results.

From Figs. 5 and 6, we can see that, in general, for almost all attack strategies, the values of Q and NMI decrease as the budget T increases, i.e., the attack effect is more significant when more links can be changed. As expected, Q-Attack behaves best to reduce the value of Q. This is not surprising since its goal is designed to minimize Q based on (2). Moreover, for the other metric NMI, we can still find that the curves for Q-Attack present sharper decline than the others, indicating its best attack effect on both Karate and Dolphins networks. Such results validate that our GA-based Q-Attack is, indeed, more effective in both weakening the strength of community structure and reducing the similarity between the community detection results and the true labels. In addition, we also note that, in most situations, DBA does a better job than CDA in reducing Q while we get the opposite results on the metric NMI. This seems reasonable by considering that nodes of large degree plays a vital role in maintaining the structure of network and thus attacks on such nodes may affect the network structure more significantly. However, on the other hand, such large-degree nodes always stay in the centers of communities and, thus, are relatively difficult to be grouped into wrong communities when they are under attack, which may lead to the less reduction of NMI for DBA than CDA.

To provide more comprehensive and quantitative comparisons, we take extensive experiments on six community detection algorithms and four networks totally. Crossover rate P_c and mutation rate P_m are set to 0.8 and 0.1, respectively. The parameter K, i.e., the number of nodes chosen as targets in heuristic attack strategies, is set to 11 (10% nodes) for Football and 10 (10% nodes) for Polbooks. The attack number T is set to 5% of total number of links in the network for Karate and Dolphins, 2% for larger networks Football and Polbooks. We use the default settings for the parameters in

node2vec and take the number of clusters in K-means the same as ground truth. Table II shows the relative reductions of Q and NMI under four attack strategies. For each column in the table, we mark the top two best results as bold. Again, we see that our GA-based Q-Attack has significantly better attack effects, in terms of larger relative reductions of both Q and NMI, than the three heuristic strategies in almost all cases except the reduction of NMI for Football while attacking on Infomap algorithm (INF). We also observe that the advantage of GA-based Q-Attack seems more obvious when we control the attack budget to a smaller value. The relative reduction of two metrics for Q-Attack is about 2 to 3 times those for the optimal heuristic attack strategy on average with attack number T limited to 5% for Karate and Dolphins while it increases to 5 times and even more when attack number T is limited to 2% for Football and Polbooks.

In Fig. 7, we visualize an example that Q-Attack is applied on the Louvain algorithm and Karate network. The community structure in the original network detected by Louvain algorithm is shown in Fig. 7(a). The modularity Q and NMI are 0.4188 and 0.5866, respectively. We also use Louvain algorithm to detect the community structure in the adversarial network obtained under four rewiring attacks, which is shown in Fig. 7(b). After attack, the modularity Q falls to 0.3053 while NMI falls to 0.3078 and the number of communities changes from 4 to 5. We can see that, indeed, the community detection result presents higher level disorder after attack, indicating the good attack effect of Q-Attack.

E. Transferability of Q-Attack

As we can see, Q-Attack is typically based on the modularity Q which depends on certain community detection algorithm. However, there are a large number of community detection algorithms in reality, it is hard for us to know which one has been applied. Q-Attack may be of less practicability if the adversarial network obtained according to a given community detection algorithm loses its effectiveness on attacking others. In order to address this concern, we design experiments to testify the transferability of Q-Attack. The relative reductions of modularity Q are presented in Table III. Again, for each case, we mark the top two best results as bold.

We find that the adversarial networks obtained by Q-Attack on specific community detection algorithms still show considerable attack effects for others in most cases. Another interesting finding is that adversarial networks obtained by taking attacks on Louvain and Infomap show significant transfer attack effects more frequently, while generalized to other algorithms. Meanwhile, Louvain and Infomap seem more robust since reductions of Q on these two algorithms are relatively small in most cases, except taking attacks on these two algorithms directly. The possible explanation for our finding is that adversarial networks obtained by taking attacks on algorithms with better performance will show a stronger transferability. It is consistent with the findings in the research studies guided by Lancichinetti and Fortunato [6] and Yang *et al.* [50], which show Infomap and Louvain have better performance through comparative analysis of different community detection algorithms.

V. CONCLUSION

In this paper, we propose several strategies, including CDA, DBA, and GA-based Q-Attack to attack a number of community detection algorithms for networks. We perform the experiments on four well-known social networks and find that these attack strategies are effective to make the community detection algorithms fail partly in most cases, in terms of decreasing modularity Q and NMI, by just disturbing a small number of connections. By comparison, Q-Attack behaves much better than CDA and DBA in almost all the considered white-box cases, where the target community detection algorithms are known in advance. Moreover, this attack strategy also presents certain transferability to make effective black-box attacks. That is, the adversarial networks, generated by Q-Attack on particular community detection algorithms, can still be used to effectively attack many others. These results indicate that it is feasible to utilize Q-Attack to protect people's privacy against community detection in reality.

With more and more attentions are focused on the problem of information overmined in social networks, many works are left to be studied. For instance, here, our Q-Attack method only takes modularity Q as the optimization objective due to its simplicity, methods based on multiobjective optimization are also worth studying and may present better attack effects. For wide applications on lager networks, designing attack strategies of higher efficiency by simplifying the fitness calculation and choosing the optimization algorithms with lower time complexity is one of the research emphases in the future. In addition, the community detection algorithms we have considered in the experiments are mainly used to detect nonoverlapping communities, but there are also many overlapping communities in the real world, and the attack strategies thus can be extended to such more general situations.

ACKNOWLEDGMENT

The authors would like to thank all the members in the IVSN Research Group, Zhejiang University of Technology for the valuable discussion about the ideas and technical details presented in this paper.

REFERENCES

[1] Q. Xuan, Z.-Y. Zhang, C. Fu, H.-X. Hu, and V. Filkov, "Social synchrony on complex networks," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1420–1431, May 2018.

[2] Q. Xuan, H. Fang, C. Fu, and V. Filkov, "Temporal motifs reveal collaboration patterns in Online task-oriented networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 91, no. 5, May 2015, Art. no. 052813.

[3] J. O. Garcia, A. Ashourvan, S. Muldoon, J. M. Vettel, and D. S. Bassett, "Applications of community detection techniques to brain graphs: Algorithmic considerations and implications for neural function," *Proc. IEEE*, vol. 106, no. 5, pp. 846–867, May 2018.

[4] Z. Chen, J. Wu, Y. Xia, and X. Zhang, "Robustness of interdependent power grids and communication networks: A complex network perspective," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 65, no. 1, pp. 115–119, Jan. 2018.

[5] S. Schiavo, J. Reyes, and G. Fagiolo, "International trade and financial integration: A weighted network analysis," *Quant. Finance*, vol. 10, no. 4, pp. 389–399, Jun. 2010.

[6] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 5, Nov. 2009, Art. no. 056117.

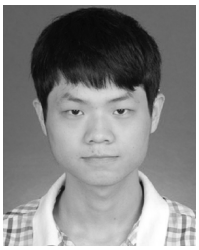
[7] C. Fu *et al.*, "Link weight prediction using supervised learning methods and its application to yelp layered network," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1507–1518, Aug. 2018.
[8] T. N. Kipf and M. Welling. (2016). "Semi-supervised classification with graph convolutional networks." [Online]. Available: https://arxiv.org/abs/1609.02907
[9] H.-F. Zhang, F. Xu, Z.-K. Bao, and C. Ma, "Reconstructing of networks with binary-state dynamics via generalized statistical inference," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 4, pp. 1608–1619, Apr. 2018.
[10] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.
[11] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, Jan. 2001.
[12] P. M. Gleiser and L. Danon, "Community structure in jazz," *Adv. complex Syst.*, vol. 6, no. 04, pp. 565–573, Dec. 2003.
[13] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2826–2841, Oct. 2007.
[14] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, Nov. 2016.
[15] Q. Xuan *et al.*, "Modern food foraging patterns: Geography and cuisine choices of restaurant patrons on yelp," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 2, pp. 508–517, Jun. 2018.
[16] S. Nagaraja, "The impact of unlinkability on adversarial community detection: Effects and countermeasures," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.* Berlin, Germany: Springer, Jul. 2010, pp. 253–272.
[17] M. Waniek, T. P. Michalak, M. J. Wooldridge, and T. Rahwan, "Hiding individuals and communities in a social network," *Nature Hum. Behav.*, vol. 2, no. 2, pp. 139–147, Jan. 2018.
[18] V. Fionda and G. Pirrò, "Community deception or: How to stop fearing community detection algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 4, pp. 660–673, Apr. 2018.
[19] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 1, May 2002, Art. no. 056109.
[20] M. Tasgin, A. Herdagdelen, and H. Bingol. (2007). "Community detection in complex networks using genetic algorithms." [Online]. Available: https://arxiv.org/abs/0711.0491
[21] H. Liu, X.-B. Hu, S. Yang, K. Zhang, and E. Di Paolo, "Application of complex network theory and genetic algorithm in airline route networks," *Transp. Res. Rec.*, vol. 2214, no. 1, pp. 50–58, 2011.
[22] S. Wang, H. Zou, Q. Sun, X. Zhu, and F. Yang, "Community detection via improved genetic algorithm in complex network," *Inf. Technol. J.*, vol. 11, no. 3, pp. 384–387, Mar. 2012.
[23] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.
[24] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman, "E-mail as spectroscopy: Automated discovery of community structure within organizations," *Inf. Soc.*, vol. 21, no. 2, pp. 143–153, Feb. 2005.
[25] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 9, pp. 2658–2663, Mar. 2004.
[26] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, Jun. 2004, Art. no. 066133.
[27] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, Dec. 2004, Art. no. 066111.
[28] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.
[29] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.
[30] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, Sep. 2006, Art. no. 036104.
[31] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.
[32] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 2, pp. 1118–1123, Jan. 2008.
[33] P. Ronhovde and Z. Nussinov, "Multiresolution community detection for megascale networks by information-based replica correlations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 1, Jul. 2009, Art. no. 016109.
[34] M. Bellingeri, D. Cassi, and S. Vincenzi, "Efficiency of attack strategies on complex model and real-world networks," *Phys. A, Stat. Mech. Appl.*, vol. 414, pp. 174–180, Nov. 2014.
[35] B. Karrer, E. Levina, and M. E. Newman, "Robustness of community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 77, no. 4, Apr. 2008, Art. no. 046119.
[36] S. Yu *et al.* (2018). "Target defense against link-prediction-based attacks via evolutionary perturbations." [Online]. Available: https://arxiv.org/abs/1809.05912
[37] H. Dai *et al.* (2018). "Adversarial attack on graph structured data." [Online]. Available: https://arxiv.org/abs/1806.02371
[38] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2018, pp. 2847–2856.
[39] A. Bojchevski and Günnemann. (2018). "Adversarial attacks on node embeddings."[Online]. Available: https://arxiv.org/abs/1809.01093
[40] J. Chen, Y. Wu, X. Xu, Y. Chen, H. Zheng, and Q. Xuan. (2018). "Fast gradient attack on network embedding." [Online]. Available: https://arxiv.org/abs/1809.02797
[41] A. L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.
[42] J. H. Holland *et al.*, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
[43] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
[44] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43.
[45] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.
[46] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecology Sociobiology*, vol. 54, no. 4, pp. 396–405, Sep. 2003.
[47] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.
[48] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.
[49] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech., Theory Exp.*, vol. 2005, no. 09, Sep. 2005, Art. no. P09008.
[50] Z. Yang, R. Algesheimer, and C. J. Tessone, "A comparative analysis of community detection algorithms on artificial networks," *Sci. Rep.*, vol. 6, Aug. 2016, Art. no. 30750.

Jinyin Chen received the B.S. and Ph.D. degrees from the Zhejiang University of Technology, Hangzhou, China, in 2004 and 2009, respectively.

She studied evolutionary computing with the Ashikaga Institute of Technology, Ashikaga, Japan, in 2005 and 2006. She is currently an Associate Professor with the College of Information Engineering, Zhejiang University of Technology, and also with the Zhejiang Lab, Hangzhou. Her current research interests include evolutionary computing, data mining, and deep learning algorithm.

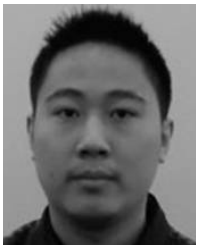
Lihong Chen received the B.S. degree from the Zhejiang University of Technology, Hangzhou, China, in 2018, where she is currently pursuing the M.S. degree with the College of Information Engineering.

Her current research interests include social network analysis and evolutionary computing.



Yixian Chen received the B.S. degree from the Zhejiang University of Technology, Hangzhou, China, in 2018, where he is currently pursuing the M.S. degree with the College of Information Engineering.

His current research interests include data mining and applications and intelligent computing.



Minghao Zhao received the B.S. degree from the Wuhan University of Technology, Wuhan, China, in 2014. He is currently pursuing the M.S. degree with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China.

His current research interests include social networks analysis and machine learning.

Shanqing Yu received the M.S. degree from the Graduate School of Information, Production and Systems, Waseda University, Tokyo, Japan, in 2008, and the Ph.D. degree from the School of Computer Engineering, Waseda University, in 2011.

She is currently a Lecturer with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China, and also with the Zhejiang Lab, Hangzhou. Her current research interests include intelligent computation, data mining and intelligent transport system.

Qi Xuan (M'18) received the B.S. and Ph.D. degrees in control theory and engineering from Zhejiang University, Hangzhou, China, in 2003 and 2008, respectively.

From 2008 to 2010, he was a Post-Doctoral Researcher with the Department of Information Science and Electronic Engineering, Zhejiang University. In 2010 and 2017, he was a Research Assistant with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. From 2012 to 2014, he was a Post-Doctoral Fellow with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently a Professor with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, and also with the Zhejiang Lab, Hangzhou. His current research interests include network-based algorithms design, social networks, data mining, cyberspace security, machine learning, and computer vision.

Xiaoniu Yang is currently a Chief Scientist with the Science and Technology on Communication Information Security Control Laboratory, Jiaxing, China. He is also a Fellow of the Chinese Academy of Engineering, Beijing, China, and the Chinese Institute of Electronics, Beijing. He is also an Adjunct Professor with Zhejiang University, Hangzhou, China, and a Ph.D. Supervisor with Xidian University, Xi'an, China. He has authored or coauthored the first software radio book in China *Software Radio Principles and Applications* [Publishing House of Electronics Industry, X. Yang, C. Lou and J. Xu, 2001 (in Chinese)]. His current research interests include software-defined satellite, big data for radio signals, and deep learning-based signal processing.