Hindawi
Security and Communication Networks
Volume 2021, Article ID 6631247, 13 pages
https://doi.org/10.1155/2021/6631247

Research Article

Towards Revealing Parallel Adversarial Attack on Politician Socialnet of Graph Structure

Yunzhe Tian,[1] **Jiqiang Liu**,[1] **Endong Tong**,[1] **Wenjia Niu**,[1] **Liang Chang**,[2] **Qi Alfred Chen**,[3] **Gang Li**,[4] **and Wei Wang**[1]

[1]*Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, China*
[2]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China*
[3]*University of California, Irvine, CA, USA*
[4]*Australia Centre for Cyber Security Research and Innovation, Deakin University, Geelong, Australia*

Correspondence should be addressed to Endong Tong; edtong@bjtu.edu.cn and Wenjia Niu; niuwj@bjtu.edu.cn

Received 16 October 2020; Revised 25 January 2021; Accepted 26 February 2021; Published 9 March 2021

Academic Editor: Zhe-Li Liu

Socialnet becomes an important component in real life, drawing a lot of study issues of security and safety. Recently, for the features of graph structure in socialnet, adversarial attacks on node classification are exposed, and automatic attack methods such as fast gradient attack (FGA) and NETTACK are developed for per-node attacks, which can be utilized for multinode attacks in a sequential way. However, due to the overlook of perturbation influence between different per-node attacks, the above sequential method does not guarantee a global attack success rate for all target nodes, under a fixed budget of perturbation. In this paper, we propose a parallel adversarial attack framework on node classification. We redesign new loss function and objective function for nonconstraint and constraint perturbations, respectively. Through constructing intersection and supplement mechanisms of perturbations, we then integrate node filtering-based P-FGA and P-NETTACK in a unified framework, finally realizing parallel adversarial attacks. Experiments on politician socialnet dataset Polblogs with detailed analysis are conducted to show the effectiveness of our approach.

1. Introduction

With the development of Internet and IT technology, an emerging cyber space [1], which refers to the global network of interdependent information technology infrastructures, telecommunications networks, and computer processing systems, is covering most aspects of our daily life nowadays. In such space, as a highly important and detailed representation, various emerging social networks (e.g., Facebook, Twitter, WeChat, and TikTok) are greatly pushing the new revolution of network interconnection and interdependence, as well as the social relations and information propagation [2, 3].

Social network is called socialnet in short. Due to the popularity, billions of socialnet users share their personal data and connect with friends and family through various devices and applications. Since the socialnet can be abstracted to a simple kind of graph with features of nodes and edges, many researchers have contributed their efforts to study socialnet and corresponding services based on graph- and workflow-related approaches [4–8]. One of the most frequently applied tasks on graph data is node classification, the goal of which is to predict the labels of the remaining nodes when given a single large graph and the class labels of a few nodes [9]. For example, we can utilize node classification to predict the political labels of politician such as Liberals and Conservatives, according to their socialnet interactions.

For node classification in recent years, the graph convolutional network (GCN) [10, 11], a kind of graph neural network (GNN) [12, 13] based on deep learning, has shown a great potential. Unfortunately, such GCN also opens a new door for cyber attacks. Adversarial attacks against GCN are discovered, through few edge perturbations of addition or

deletion, and they are uneasy to notice [14]. Furthermore, automatic attack methods are developed to explore effective perturbations including constraint and nonconstraint perturbations. Constraint perturbation refers to the edge perturbation satisfying specific requirements such as node degree distribution of graph [15]. Nonconstraint perturbation means a free perturbation. Accordingly, fast gradient attack (FGA) [16] and NETTACK [17] are typical nonconstraint and constraint methods, respectively. The above methods enable per-node attacks, as well as multinode attacks in a sequential way. However, due to the overlook of perturbation influence between different per-node attacks, the above sequential method does not guarantee a global attack success rate for all target nodes, under a fixed budget of perturbation. Figure 1 shows the differences between sequential and parallel attacks in a motivating example. For the No. 1, 2, and 3 nodes, the attack goal is to change their class labels through changing graph structure with perturbations, including edge addition and edge deletion. We can see that, due to removing the edges from efforts of edge addition in attack of No. 1 node, No. 2 and No. 3 node attacks of sequential attack waste edges perturbations and cause No. 1 node attack to fail with a global attack success rate of 2/3, while the parallel attack considers perturbation influence and has a higher attack success rate with a lower budget.

In this work, we are the first to perform multinode attack in a parallel way by integrating two methods P-FGA and P-NETTACK in a unified attack framework. Based on nonconstraint FGA, we redesign a new loss function in P-FGA, which employs CW-loss [18] to replace CE-loss. For P-NETTACK, we utilize the maximum sum of surrogate loss as new objective function to support parallel attack. Moreover, we apply a node filtering mechanism to P-NETTACK and P-FGA, which filters out those nodes that are successfully attacked from target node set. After extracting common perturbations, we also provide a random supplement of perturbations to fill the budget.

We experiment on politician socialnet dataset Polblogs [19] of 1222 nodes and 16714 edges, showing the effectiveness of our approach. We find that our approach can achieve a high attack success rate (*ASR*) of 71.5% at the lowest perturbation budget of 1/5 d_{sum} (d_{sum} is the sum of the degrees of all target nodes), that is over 15% higher than that of NETTACK or FGA, still keeping a satisfied test statistic of 0.005. The filtering mechanism can greatly improve *ASR*, with nearly 20% average increment. We summarize our contributions as follows:

(1) We give the very first attempt to propose a multinode parallel adversarial attack framework on node classification in socialnet of graph structure, based on considering perturbation influence between per-node attacks.

(2) Node filtering-based nonconstraint P-FGA and node filtering-based constraint P-NETTACK are proposed, and we integrate them into a unified multinode parallel attack framework, through constructing intersection and supplement mechanisms of perturbation.

(3) We evaluate our approach empirically on real dataset of politician socialnet Polblogs. Based on parallel attacking on the graph of 1222 nodes and 16714 edges, we reveal and verify the effectiveness of our approach compared to sequential attacks in terms of attack strength and attack stealthiness.

The rest of the paper is structured as follows: Section 2 introduces the preliminaries and problem definition. Section 3 proposes a multinode parallel attack framework. Section 4 reports our experiments and evaluations on the politician socialnet dataset Polblogs. In Section 5, we discuss the related works. Finally, Section 6 concludes the work of this paper.

2. Preliminaries and Problem Definition

2.1. Graph Structure of Socialnet. In real socialnet, one person can have an interaction with others by operations like the following: commenting, reposting, etc. Such interaction can be quantified and qualified, varying from different measurements. For simplicity, we assume that we just use an undirected unweighted edge to denote an interaction existence, constructing graph structure of socialnet (See Figure 2). Moreover, we simply assume that one node only has one classification label, and we do not focus on multiple free-label user profile or granularity-based hierarchical user profile [20] in attack scenarios of this paper. Thus, for a socialnet graph, we have a triple $G = (V, C, \mathbf{A})$ including node set V, label set C, and adjacent matrix $\mathbf{A}$, in which $V = \{v_1, v_2, \ldots, v_n\}$, $C = \{c_1, c_2, \ldots, c_n\}$ ($|V| = |C| = n$), and $\mathbf{A}$ is shown as follows:

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}, \quad a_{ij} \in \{0, 1\}. \tag{1}$$

2.2. Graph Convolutional Network. As a kind of GNN, GCN is an extremely powerful neural network architecture for deep learning on graphs to produce useful feature representations of nodes in networks. Given a $G = (V, C, \mathbf{A})$, we can partially delete some node's label (c_i = null) to obtain a new $G' = (V, C', \mathbf{A})$. The goal of node classification is to learn a function $\mathbf{Z}$, which maps each node $v \in V$ to one class ($|\{c_i = \text{null}\}|=0$).

We use a two-layer GCN to approximate the function $\mathbf{Z}$:

$$\mathbf{Z} = f_{\boldsymbol{\theta}}(\mathbf{A}, \mathbf{X}) = \text{softmax}\left(\widehat{\mathbf{A}}\sigma\left(\widehat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(1)}\right)\mathbf{W}^{(2)}\right), \tag{2}$$

where $\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-(1/2)}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-(1/2)}$, $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_{\mathbf{N}}$ is the adjacent matrix $\mathbf{A}$ of the input graph G' with self-loops added via the identity matrix $\mathbf{I}_{\mathbf{N}}$, $\widetilde{\mathbf{D}}_{\mathbf{ii}} = \sum_j \widetilde{a}_{ij}$ is the degree matrix of $\widetilde{\mathbf{A}}$, and $\mathbf{X}$ is a matrix of node feature vectors. For the graph whose nodes do not have feature attributes, $\mathbf{X}$ can be set to an identity matrix $\mathbf{I}_{\mathbf{N}}$. $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ are the trainable weight matrices of the first and second layers, respectively, and $\sigma(\cdot)$ is a ReLU activation function. For the semisupervised node classification, the optimal parameters $\boldsymbol{\theta} = \left\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}\right\}$ are

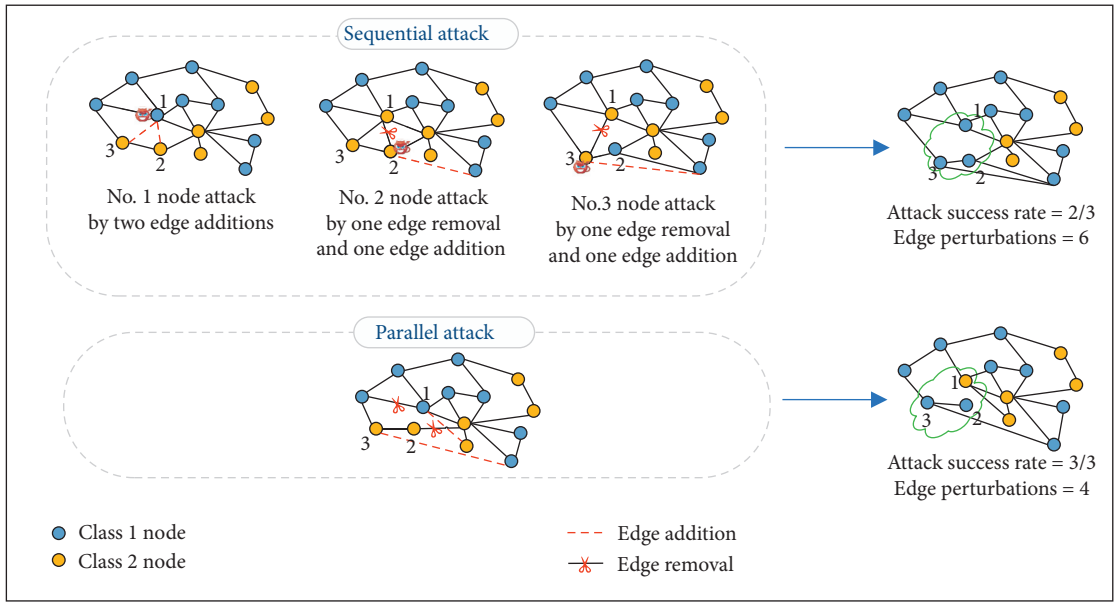

Figure 1: Differences between sequential and parallel attacks in a motivating example.

learnt by minimizing the cross-entropy loss over all labeled examples:

$$L(\boldsymbol{\theta}; \mathbf{A}, \mathbf{X}) = -\sum_{v \in V_l} \ln \mathbf{Z}_{v,c_v}, \tag{3}$$

where $V_l \subseteq V$ is the set of nodes with labels, namely, training set, c_v is the given true label of node v, and $\mathbf{Z}_{v,c_v}$ is the probability of assigning class c_v to node v.

2.3. Problem Definition. Given the attack target set $V_t \subseteq V$ in G' and perturbation budget Δ, multinode attack on GCN can be regarded as the following optimization problem:

$$\mathbf{A}^* = \arg\max_{\mathbf{A}^*} \sum_{v \in V_t} \left(\operatorname{sign}\left(\max_{c \neq c_v} \mathbf{Z}_{v,c} - \mathbf{Z}_{v,c_v} \right) \right), \tag{4}$$

s.t.

$$\sum_{u<v} |a_{uv} - a^*_{uv}| \leq \Delta, \tag{5}$$

$$u \in V_t \vee v \in V_t, \quad \text{where } a^*_{uv} \neq a_{uv}, \tag{6}$$

where $\max_{c \neq c_v} \mathbf{Z}_{v,c} - \mathbf{Z}_{v,c_v} > 0$, $\operatorname{sign}(\cdot) = 1$, else $\operatorname{sign}(\cdot) = 0$.

Formula (4) shows the objective function, aiming to find the optimal adjacent matrix. When the sum of misclassification for all target nodes is maximum, it means a most successful multinode attack. Formulas (5) and (6) show the constraints that should be satisfied. Formula (5) requires that the number of edge perturbations be no more than Δ (a predefined constant). Formula (6) has the constraint that any edge perturbation must be linked to a target attack node.

3. Multinode Parallel Attack Framework

Our multinode parallel attack framework is shown in Figure 3. Firstly, given an original graph $G' = (V, C', \mathbf{A})$ as defined in Section 2.1, we train a GCN for node classification task, and we obtain C, in which all nodes are labeled with prediction, and record G into testing result as ground truth. Then, given a target node set $V_t \subseteq V$, we utilize P-FGA and P-NETTACK to perturb the original graph, attacking target nodes in V_t. In each iteration of nonconstraint P-FGA method, based on GCN-gradient information of the adjacent matrix $\mathbf{A}$, we select the pair of nodes (v_i, v_j) of maximum absolute value of gradient to perform perturbation (edge deletion or edge addition), generating a new adversarial graph $G^{\text{adv}}_{P\text{–FGA}}$ by the generator. In each iteration of constraint P-NETTACK, to ensure keeping the perturbations unnoticeable and preserving the important structural characteristics, we firstly compute the candidate perturbation set C_{pert} to ensure the similar node degree distribution after perturbation execution. Then, according to our redesigned objective function, from candidate perturbation set, we greedily select the optimal perturbation (v_m, v_n), which obtains the highest objective score, generating a new adversarial graph $G^{\text{adv}}_{P\text{–NETTACK}}$ by the generator.

In the filtering mechanism, after each perturbation of P-FGA or P-NETTACK, the new predicted labels should be compared with the testing result to determine the attack effect. For those nodes that are successfully attacked, such mechanism filters them out from target node set V_t to form a new target node set V'_t, ignoring those nodes in the next gradient/objective function computation and perturbation selection. Such process is repeated until the perturbation budget Δ is reached. $\mathbf{D}_{P\text{–FGA}}$ and $\mathbf{D}_{P\text{–NETTACK}}$ are perturbation sets based on P-FGA and P-NETTACK, respectively. To integrate $\mathbf{D}_{P\text{–FGA}}$ and $\mathbf{D}_{P\text{–NETTACK}}$ and generate unified perturbations, we provide an intersection mechanism to extract common perturbations and a perturbation supplement mechanism to fill the perturbation budget Δ. Finally, the integrated perturbation set $\mathbf{D}_{\text{comb}}$ is used to realize effective multinode parallel adversarial attacks.

3.1. P-FGA Method. In our P-FGA, to adapt to the multinode attack, we redesign a new loss function L_{multi} for attack target set V_t, which employs CW-loss [18] to replace CE-loss and takes all target nodes into consideration (see equation (7)).

$$L_{\text{multi}} = \sum_{v \in V_t} \left\{ \max_{c \neq c_v} \ln Z_{v,c} - \ln Z_{v,c_v} \right\}. \tag{7}$$

Following the gradient-based idea of original FGA, based on the new loss function L_{multi} for multinode attack, we firstly calculate the partial derivatives with respect to the element a_{ij} of adjacency matrix $\mathbf{A}$ and further obtain gradient matrix $\mathbf{GM}$, and its element g_{ij} can be calculated by

$$g_{ij} = \frac{\partial L_{\text{multi}}}{\partial a_{ij}}. \tag{8}$$

Considering that the adjacency matrix is symmetric and its gradient matrix should also be symmetric, thus, we have

$$\hat{g}_{ij} = \hat{g}_{ji} = \begin{cases} \dfrac{g_{ij} + g_{ji}}{2}, & i \neq j, \\ 0, & i = j, \end{cases}$$
$$\widetilde{g}_{ij} = \hat{g}_{ij} \times \left(-2 \cdot a_{ij} + 1\right), \tag{9}$$

where $\widetilde{g}_{ij}$ forms $\widetilde{\mathbf{GM}}$. A bigger value of multinode loss function L_{multi} corresponds to worse prediction results for the target nodes in V_t. And edge perturbations along the direction of the gradient can make the loss increase more faster locally. That is, for a positive gradient $\widehat{g}_{ij}$, adding the edge between the pair of nodes (v_i, v_j) can increase the loss. Similarly, for a negative gradient $\widehat{g}_{ij}$, deleting the edge also increases the loss.

However, since the adjacent matrix $\mathbf{A}$ is binary discrete and $a_{ij} \in \{0, 1\}$, not all edges can be perturbed along the direction of the gradient. For example, for a pair of nodes (v_i, v_j) who have positive/negative gradient (i.e., $(\widehat{g}_{ij} > 0 / \widehat{g}_{ij} < 0)$) and meanwhile are connected/disconnected (i.e., $(a_{ij} = 1 / a_{ij} = 0)$), we cannot further add/delete the edge along the direction of the gradient. Thus, we design equation (9); for a positive gradient $\widehat{g}_{ij}$, when $a_{ij} = 0$, $\widetilde{g}_{ij}$ is positive; when $a_{ij} = 1$, $\widetilde{g}_{ij}$ is negative. Similarly, for a negative gradient $\widehat{g}_{ij}$, when $a_{ij} = 1$, $\widetilde{g}_{ij}$ is positive; when $a_{ij} = 0$, $\widetilde{g}_{ij}$ is negative. Only the positive $\widetilde{g}_{ij}$ enables the addition/deletion of the edge along the direction of the gradient. Then, for edge addition or deletion, we pick the optimal edge (v_m, v_n), $v_m \in V_t \vee v_n \in V_t$, with the maximum $\widetilde{g}_{mn}$, and the adjacent matrix $\mathbf{A}$ is updated to $\mathbf{A}'$ by changing the corresponding value (a_{mn} and a_{nm}) to a different binary value (see equation (10)).

$$a'_{mn} = a'_{nm} = 1 - a_{mn}. \tag{10}$$

The pseudocode for P-FGA is given in Algorithm 1.

3.2. P-NETTACK Method. In constraint P-NETTACK, we use test statistic $\Lambda(G', G^{\text{adv}})$ to determine whether our generated adversarial graph $G^{\text{adv}} = (V, C', \mathbf{A}')$ and original graph $G' = (V, C', \mathbf{A})$ have similar node degree distribution of pow-law distribution $p(x) \propto x^{-\alpha}$, in which $p(x)$ denotes the probability of certain degree x, and α refers to scaling parameter. The test statistic Λ can be calculated based on the following formulas.

$$\alpha_G = 1 + \left|\mathbf{D}_{G'}\right| \cdot \left[\sum_{d_i \in \mathbf{D}_{G'}} \log \frac{d_i}{d_{\min} - (1/2)} \right]^{-1}, \tag{11}$$

$$l(\mathbf{D}_{G'}) = \left|\mathbf{D}_{G'}\right| \cdot \log \alpha_{G'} + \left|\mathbf{D}_G\right| \cdot \alpha_{G'} \cdot \log d_{\min} + (\alpha_{G'} + 1) \sum_{d_i \in \mathbf{D}_{G_t}} \log d_i, \tag{12}$$

$$\Lambda\left(G', G^{\text{adv}}\right) = -2 \cdot l\left(\mathbf{D}_{G_t} \bigcup \mathbf{D}_{G^{\text{adv}}}\right) + 2 \cdot \left[l(\mathbf{D}_{G'}) + l(\mathbf{D}_{G^{\text{adv}}})\right]. \tag{13}$$

In equation (11), $d_{\min}$ is the minimum degree that a node has to be considered in the power-law test, and $\mathbf{D}_{G'} = \left\{d_v^{G'} | v \in V, d_v^{G'} \geq d_{\min}\right\}$ is the multiset containing the list of node degrees, where $d_v^{G'}$ is the degree of node v in G' [21]. Equation (12) is used to evaluate the log-likelihood $l(\mathbf{D}_{G'})$ for the sample $\mathbf{D}_{G'}$ [22]. Then, we can get final test statistic by equation (13). Similar to NETTACK, we only accept adversarial graph G^{adv} whose degree distribution fulfils $\Lambda(G', G^{\text{adv}}) < 0.004$ and thus obtain the candidate perturbation set C_{pert}. In our P-NETTACK, the edge perturbations in C_{pert} must be linked to an attack target node.

To efficiently select the optimal perturbation from C_{pert}, NETTACK utilizes a linear surrogate model $\mathbf{Z}'$ to approximate the nonlinear GCN model $\mathbf{Z}$ by removing the activation function $\sigma(\cdot)$. $\mathbf{Z}'$ is calculated as follows:

$$\mathbf{Z}' = \text{softmax}\left(\hat{\mathbf{A}}\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(1)}\mathbf{W}^{(2)}\right) = \text{softmax}\left(\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}\right). \tag{14}$$

In our P-NETTACK, given an attack target set V_t, we utilize the sum of single surrogate losses for each $v \in V_t$ as the new surrogate loss to support multinode attack:

$$L_{\text{multi}}(\mathbf{A}; \mathbf{X}, \mathbf{W}, V_t) = \sum_{v \in V_t} \left\{ \max_{c \neq c_v} \left[\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}\right]_{v,c} - \left[\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}\right]_{v,c_v} \right\}, \tag{15}$$

where $[\hat{\mathbf{A}}^2\mathbf{X}\mathbf{W}]_{v_t,c}$ is the value of class c given to the node v_t by the surrogate model. The multinode scoring function that evaluates the multinode surrogate loss obtained after adding/deleting an edge $e = (i, j) \in C_{\text{pert}}$ is defined as

$$s_{\text{multi}}(e; \mathbf{A}, V_t) \coloneqq L_{\text{multi}}(\mathbf{A}'; \mathbf{X}, \mathbf{W}, V_t), \tag{16}$$

where $\mathbf{A}$ is updated to $\mathbf{A}'$ by $a'_{ij} = a'_{ji} = 1 - a_{ij}$. Following the greedy approximate scheme in NETTACK, during each iteration, we select the optimal perturbation that has the highest value of multinode scoring function from the candidate perturbation set C_{pert} to execute. The above processes including candidate perturbation computation, determining

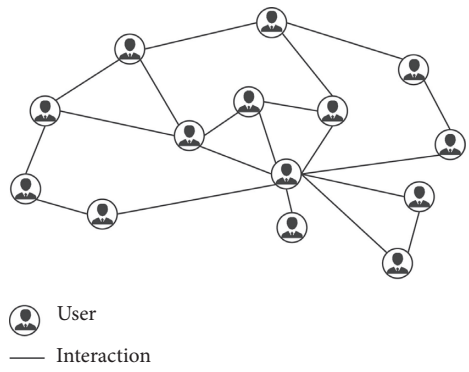

Figure 2: Illustration of socialnet graph.

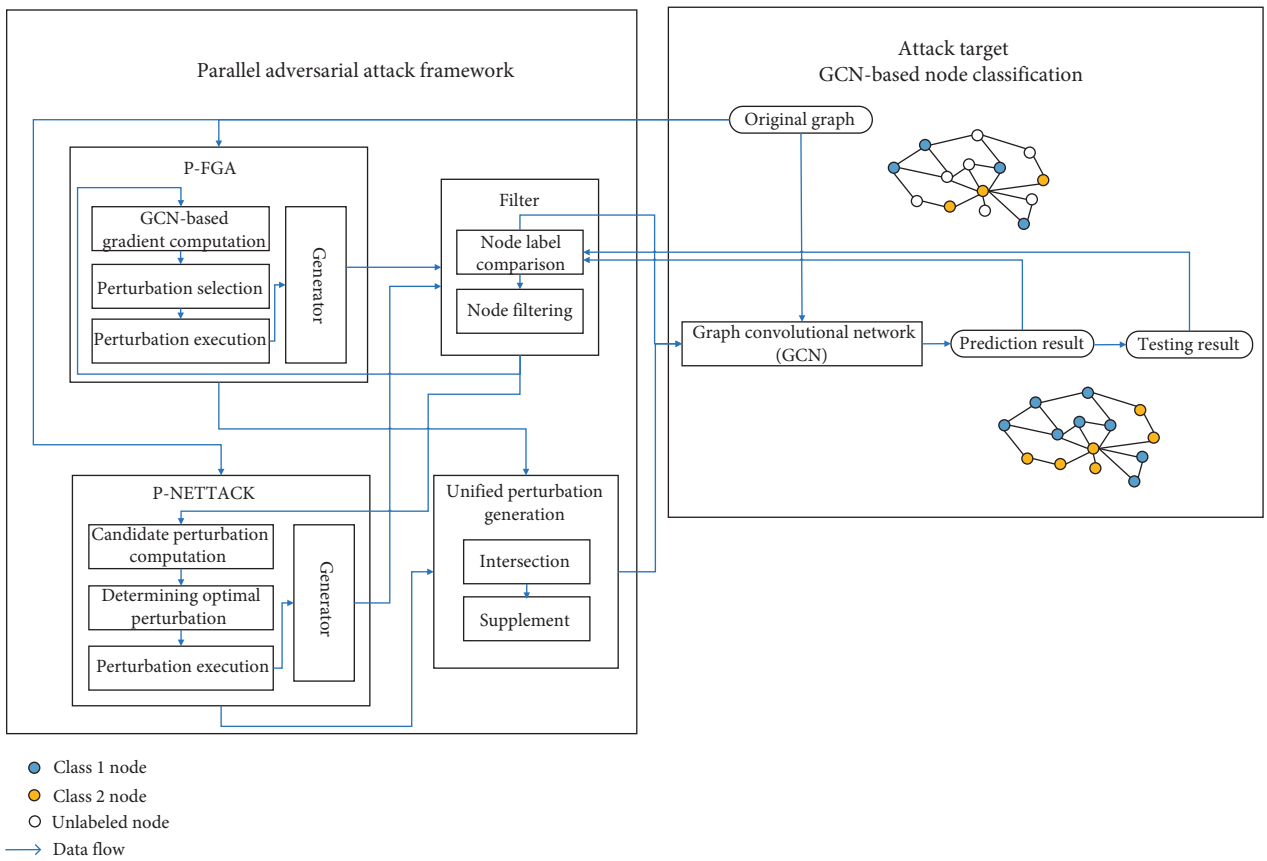

Figure 3: The parallel adversarial attack framework against GCN-based node classification.

optimal perturbation, and perturbation execution are repeated until the perturbation budget Δ is reached. The pseudocode for P-NETTACK is given in Algorithm 2.

3.3. Filtering Mechanism. In this part, we propose a filtering mechanism that filters out target nodes that are successfully attacked from the target node set. After each perturbation, by the filtering mechanism, we obtain a filtered attack target set V_t', which is used in the next iteration. If there are no nodes in V_t', which means that all target nodes have been attacked successfully, and we reset the attack target set V_t' to the original attack target set V_t. The pseudocode for filtering mechanism is given in Algorithm 3.

3.4. Intersection and Supplement Mechanism. In this section, we construct intersection and supplement mechanism of perturbations. Given the perturbation sets $\mathbf{D}_{P-\mathrm{FGA}}$ and $\mathbf{D}_{P-\mathrm{NETTACK}}$ under a fixed perturbation budget Δ, we first utilize intersection mechanism to extract their common perturbations $\mathbf{D}_{\mathrm{comb}}$. In general, the number of common perturbations is less than perturbation budget Δ. Thus, we should provide a perturbation supplement mechanism to fill the budget.

Input: $G' = (V, C', \mathbf{A})$, attack target set V_t, perturbation budget Δ
Output: perturbation set $D_{P-\mathrm{FGA}}$
(1) Train the GCN model $\mathbf{Z}$ on original graph G'
(2) Initialize $\mathbf{A}^{(0)} = \mathbf{A}$
(3) Initialize perturbation set $D_{P-\mathrm{FGA}}$
(4) **for** $h = 1$ to Δ **do**
(5) //GCN-based Gradient Computation
(6) Calculate multi-node target loss function $L_{\mathrm{multi}} = \sum_{v \in V_t} \left\{ \max_{c \neq c_v} \ln \mathbf{Z}_{v,c} - \ln \mathbf{Z}_{v,c_v} \right\}$
(7) Construct $\widetilde{\mathbf{GM}}^{(h-1)}$ based on the $\mathbf{A}^{(h-1)}$:
$g_{ij}^{(h-1)} = (\partial L_{\mathrm{multi}} / \partial a_{ij}^{(h-1)})$, $\widehat{g}_{ij}^{(h-1)} = \widehat{g}_{ji}^{(h-1)} = \begin{cases} g_{ij}^{(h-1)} + g_{ji}^{(h-1)}/2 & i \neq j \\ 0 & i = j \end{cases}$,
$\widetilde{g}_{ij}^{(h-1)} = \widehat{g}_{ij}^{(h-1)} \times (-2 \cdot a_{ij}^{(h-1)} + 1)$
(8) //Perturbation Selection
(9) Select $e^* = (v_m, v_n)$ where $v_m \in V_t \vee v_n \in V_t$, having the maximum $\widetilde{g}_{mn}^{(h-1)}$
(10) //Perturbation Execution
(11) Obtain the adjacency matrix $A^{(h)}$ by $a_{mn}^{(h)} = a_{nm}^{(h)} = 1 - a_{mn}^{(h-1)}$
(12) Generate a new adversarial graph $G^{(h)} = (V, C', \mathbf{A}^{(h)})$
(13) Add e^* to $D_{P-\mathrm{FGA}}$
(14) **end**
(15) **return** $D_{P-\mathrm{FGA}}$

Algorithm 1: Parallel fast gradient attack (P-FGA).

Input: $G' = (V, C', \mathbf{A})$, attack target set V_t, perturbation budget Δ
Output: perturbation set$D_{P-\mathrm{NETTACK}}$
(1) Train the surrogate model $\mathbf{Z}$' on original graph G' to obtain $\mathbf{W}$
(2) Initialize $\mathbf{A}^{(0)} = \mathbf{A}$
(3) Initialize perturbation set $D_{P-\mathrm{NETTACK}}$
(4) **for** $h = 1$ to Δ **do**
(5) Construct the valid candidate perturbations set C_{pert}
$\Lambda(G', G^{(h)}) < 0.004$ and $v_i \in V_t \vee v_j \in V_t$
where $(v_i, v_j) \in C_{\mathrm{pert}}, G^{(h)} = (V, Ct, \mathbf{A}^{(\mathbf{h})})$ and $a_{ij}^{(h)} = a_{ji}^{(h)} = 1 - a_{ij}^{(h-1)}$
(6) Select $e^* = (v_m, v_n)$ of the maximum multi-node scoring function value in C_{pert}
$e^* = (v_m, v_n) = \arg\max_{e \in C_{\mathrm{pert}}} s_{\mathrm{multi}}(e; \mathbf{A}^{(h-1)}, V_t)$
(7) Obtain the adjacency matrix $\mathbf{A}^{(\mathbf{h})}$ by $a_{mn}^{(h)} = a_{nm}^{(h)} = 1 - a_{mn}^{(h-1)}$
(8) Generate a new adversarial graph $G^{(h)} = (V, C', \mathbf{A}^{(h)})$
(9) Add e^* to $D_{P-\mathrm{NETTACK}}$
(10) **end**
(11) **return** $D_{P-\mathrm{NETTACK}}$

Algorithm 2: Parallel NETTACK (P-NETTACK).

We denote $\mathbf{D}'_{P-\mathrm{NETTACK}}$ as the set consisting of the perturbations in $\mathbf{D}_{P-\mathrm{NETTACK}}$ but not in $\mathbf{D}_{\mathrm{comb}}$. Similarly, $\mathbf{D}'_{P-\mathrm{FGA}}$ contains the perturbations in $\mathbf{D}_{P-\mathrm{FGA}}$ but not in $\mathbf{D}_{\mathrm{comb}}$. Δ' is the difference between Δ and the number of $\mathbf{D}_{\mathrm{comb}}$. Besides, we use a supplementary factor k to control the proportion of supplementary perturbations from $\mathbf{D}'_{P-\mathrm{NETTACK}}$. Specially, we randomly select $[k \cdot \Delta']$ and $\Delta' - [k \cdot \Delta']$ perturbations from $\mathbf{D}'_{P-\mathrm{NETTACK}}$ and $\mathbf{D}'_{P-\mathrm{FGA}}$, respectively, and add them to the $\mathbf{D}_{\mathrm{comb}}$, forming the final unified perturbation set. The pseudocode for intersection and supplement mechanism of perturbations is given in Algorithm 4.

4. Experiments

4.1. Dataset and Environment. We use the well-known politician socialnet Polblogs [19] as our experimental dataset to evaluate our methods. The basic statistics are summarized in Table 1, and only the largest connected component is considered. We randomly choose 20% nodes in the dataset as the labeled nodes for training. The testing set consists of the rest of the unlabeled nodes.

We also give our experimental environment configuration in Table 2.

4.2. Target Parameters and Baselines. Our GCN as an attack target is constructed based on the program on the Github (https://github.com/tkipf/gcn). We train all models for a maximum of 200 epochs using Adam [23] with a learning rate of 0.01. We initialize weights using the initialization described in Glorot and Bengio [24] and accordingly (row-) normalize input feature vectors.

We compare our proposed attack method with comprehensive state-of-the-art adversarial attack methods

Input: perturbed graph $G^{adv} = (V, C, \mathbf{A}')$, attack target set V_t, node classification model $\mathbf{Z}$
Output: filtered attack target set V_t'
(1) Initialize $V_t' = V_t$
(2) **for** each $v \in V_t$ **do**
(3) Predict the label of v in G^{adv} by $\mathbf{Z}$
(4) **if** c_v of ground truth is not equal to prediction result **then**
(5) Remove v from V_t' //filtering
(6) **end**
(7) **if** $|V_t'| == 0$ **then**
(8) $V_t' = V_t$ //reset
(9) **return** V_t'

Algorithm 3: Filtering mechanism.

Input: $D_{P-\mathrm{FGA}}$, $D_{P-\mathrm{NETTACK}}$, supplementary factor k, perturbation budget Δ
Output: D_{comb}
(1) Execute the intersection of $D_{P-\mathrm{FGA}}$ and $D_{P-\mathrm{NETTACK}}$ to obtain D_{comb}
$D_{\mathrm{comb}} = D_{P-\mathrm{FGA}} \cap D_{P-\mathrm{NETTACK}}$
(2) **if** $|D_{\mathrm{comb}}| < \Delta$ **then**
(3) Obtain $D'_{P-\mathrm{NETTACK}} = D_{P-\mathrm{NETTACK}} - D_{\mathrm{comb}}$
(4) Obtain $D'_{P-\mathrm{FGA}} = D_{P-\mathrm{FGA}} - D_{\mathrm{comb}}$
(5) Calculate $\Delta' = \Delta - |D_{\mathrm{comb}}|$
(6) Randomly add $[k \cdot \Delta']$ perturbations from $D'_{P-\mathrm{Nettack}}$ to D_{comb}
(7) Randomly add $[\Delta' - k \cdot \Delta']$ perturbations from $D'_{D-\mathrm{FGA}}$ to D_{comb}
(8) **return** D_{comb}

Algorithm 4: Intersection and supplement mechanism.

Table 1: Dataset statistics of Polblogs.

Nodes	Edges	Classes	Maximum degree	Minimum degree	Average degree
1222	16714	2	351	1	27.4

including FGA and NETTACK. We use codes of the baselines provided by their authors.

(i) **FGA** [16] extracts the gradient of pairwise nodes based on the adversarial network and then selects the pair of nodes with maximum absolute link gradient to realize the attack and update the adversarial network.

(ii) **NETTACK** [17] designs adversarial attacks based on a static surrogate model and greedily selects the optimal perturbation through preserving the key structural features of a graph.

(iii) **Random attack** randomly perturbs the edges related to target nodes.

5. Evaluations

5.1. Evaluation Metric

5.1.1. Attack Success Rate (ASR). ASR is the ratio of the number of successfully attacked nodes to the total number of target nodes, which can be calculated as follows:

$$\mathrm{ASR} = \frac{n_{\mathrm{succ}}}{|V_t|}, \tag{17}$$

where n_{succ} denotes the number of successfully attacked nodes and V_t is the attack target set.

5.1.2. Average Attack Speed (AAS). AAS refers to average running time of each attack, and it can be calculated as follows:

$$\mathrm{AAS} = \frac{t_{\mathrm{total}}}{\Delta}, \tag{18}$$

where t_{total} denotes the total attack time on target set V_t, and Δ is the perturbation budget.

5.1.3. Test Statistic Λ. Test statistic Λ is used to evaluate attack stealthiness (see equation (13)), which measures the structural difference between original graph and adversarial graph. A smaller Λ means that the degree distribution of the adversarial graph is more similar to the original graph's, and thus, the perturbations are more unnoticeable.

Table 2: Experimental environment configuration.

Experimental environment	Environmental configuration
Operating system	Windows 10
CPU	2.4 GHz intel core i5
Memory	16 GB
Hardware	500G
Software	Python 3.6

5.2. ASR Analysis. In our experiments, each attack target set consists of five target nodes, and all of them are from the test set that has been classified correctly in original graph. We divided the perturbation budgets into five levels according to the sum of degrees of all target nodes in the attack target set V_t, i.e., $\Delta \in \{(1/5)d_{\text{sum}}, (2/5)d_{\text{sum}}, (3/5)d_{\text{sum}}, (4/5)d_{\text{sum}}, (5/5)d_{\text{sum}}\}$.

As we can see from Table 3, for each Δ, we compare ASR among P^*, P-NETTACK ($k = 1$), P-FGA ($k = 0$), NETTACK, FGA, and Random Attack, in which P^* is the best value of our unified approach. From Algorithm 4, we know that if $k = 1$, our unified method can be simplified as P-NETTACK; and if $k = 0$, our unified method can be simplified as P-FGA. $P*$ has the highest ASR values of 0.715, 0.880, 1, and 1 at Δ_1, $\Delta_2, \Delta_4, \Delta_5$, respectively. When there is a quite low budget Δ_1, the ASR of P^* is over 15% higher than that of NETTACK or FGA. P-NETTACK ($k = 1$) and P-FGA ($k = 0$) have extremely close values for all budget Δ. Figure 4 shows the visual comparison in Table 3.

In Table 4, we can see that, for $\Delta_1, \Delta_2, \Delta_3$, our approach achieves highest *ASR* values of 0.715 ($k = 0.5$), 0.880 ($k = 0.7$), and 0.987 ($k = 0.8$), respectively. At Δ_4, Δ_5, for many k settings, ASR values can reach 1. For example, at Δ_4, ASR = 1 when $k = 0.1, 0.2, 0.3, 0.4, 0.5, 0.7$. Figure 5 shows the detailed *ASR* variation along with k increment.

5.3. Test Statistics Λ Analysis. As we can see from Table 5, P-NETTACK ($k = 1$) has the lowest Λ values of 0.003, 0.005, 0.005, and 0004 at $\Delta_1, \Delta_2, \Delta_4, \Delta_5$, respectively. Although P-NETTACK ($k = 1$) and NETTACK have the same constraint mechanism, theΛ values of P-NETTACK ($k = 1$) are always lower than those of NETTACK. For P-FGA ($k = 0$) and FGA, which have not enforced the constraint, the Λ values are extremely higher and continue increasing with the increment of Δ. Figure 6 shows the visual comparison in Table 5.

In Table 6, we can see that, for all $\Delta \in \{\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5\}$, with the increment of k, the test statistics Λ keep decreasing, towards better results. Figure 7 clearly shows the Λ variation along with k increment.

5.4. AAS Analysis. As we can see from Table 7, P-NETTACK is the most time-consuming adversarial attack method, with an average of 11.17s of each attack. Since the candidate perturbation set of P-NETTACK is larger than that of NETTACK, the *AAS* of P-NETTACK is much higher than that of NETTACK. Instead, P-FGA and FGA have extremely close AAS values, 0.17s and 0.14s, respectively.

5.5. Filtering Mechanism Analysis. In Table 8, we can see that, for all $\Delta \in \{\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5\}$, the filtering mechanism can greatly improve ASR, with nearly 20% average increment. And for P-FGA, the ASR values at Δ_2 are higher than those of P-FGA (without filtering) at Δ_5. Thus, we can see that the filtering mechanism plays a quite important role for P-NETTACK and P-FGA.

6. Related Work

6.1. Politician Socialnet Analysis. In the last few years, social media has become an important political communication channel, attracting a lot of studies. Adamic and Glance [19] analyzed the political blogosphere over the period of two months preceding the US Presidential Election of 2004, measuring the degree of interaction between liberal and conservative blogs and revealing many interesting differences between the two communities such as linking patterns and discussion topics. Caton et al. [25] presented a Social Observatory, which focused on public Facebook profiles of 187 German politicians from five federal parties, observing how they interacted with constituents, measuring sentiment difference between the politicians and their followers, and analyzing online speech patterns of different parties. Stieglitz and Dang-Xuan [26] proposed a social media analytics framework in political context, aiming at continuously collecting, storing, monitoring, analyzing, and summarizing politically relevant user-generated content from different social media to gain a deeper insight into political discourse in social media.

However, few studies focus on the security analysis of politician socialnet including politician label classification from the perspective of adversarial graph attack. In comparison, we focus on studying security issues of politician socialnet based on graph structure, targeting a GCN model for politician label classification. Interestingly, politician socialnet is highly vulnerable, and the attack cost is quite cheap only by deleting few existing interactions or adding few new interactions. As an important communication bridge between politicians and citizens, the security analysis of politician socialnet should be highly valued.

6.2. Adversarial Attack on Graphs. Recently, some studies have investigated the adversarial attack on neural networks for graph structure. Zügner et al. [17] first revealed the existence of adversarial attack against GCN in node classification task, by slightly modifying graph structure or node attributions to lead to misclassification of a target node. Dai et al. [27] studied test-time nontargeted adversarial attacks on both node classification task and graph classification [28] task based on reinforcement learning. In addition to white-box attack scenario, they also extended their attack method into practical black-box and restricted black-box attack scenarios. Zhang et al. [29] systematically investigated the vulnerability of knowledge graph embedding for the first time. By adding or deleting facts in the knowledge graph, they destroyed the relation prediction model based on representative knowledge graph embedding methods

Table 3: ASR comparison between P^*, P-NETTACK, P-FGA, NETTACK, FGA, and Random Attack.

Perturbation budget	P^*	P-NETTACK ($k=1$)	P-FGA ($k=0$)	NETTACK	FGA	Random attack
$\Delta_1 = 1/5d_{sum}$	**0.715 ($k=0.5$)**	0.710	**0.715**	0.453	0.480	0.0
$\Delta_2 = 2/5d_{sum}$	**0.880 ($k=0.7$)**	0.867	0.874	0.857	0.877	0.024
$\Delta_3 = 3/5d_{sum}$	0.987 ($k=0.8$)	0.963	0.953	**0.995**	0.985	0.025
$\Delta_4 = 4/5d_{sum}$	**1**	0.993	**1**	0.992	0.992	0.012
$\Delta_5 = 5/5d_{sum}$	**1**	0.992	**1**	**1**	**1**	0.01

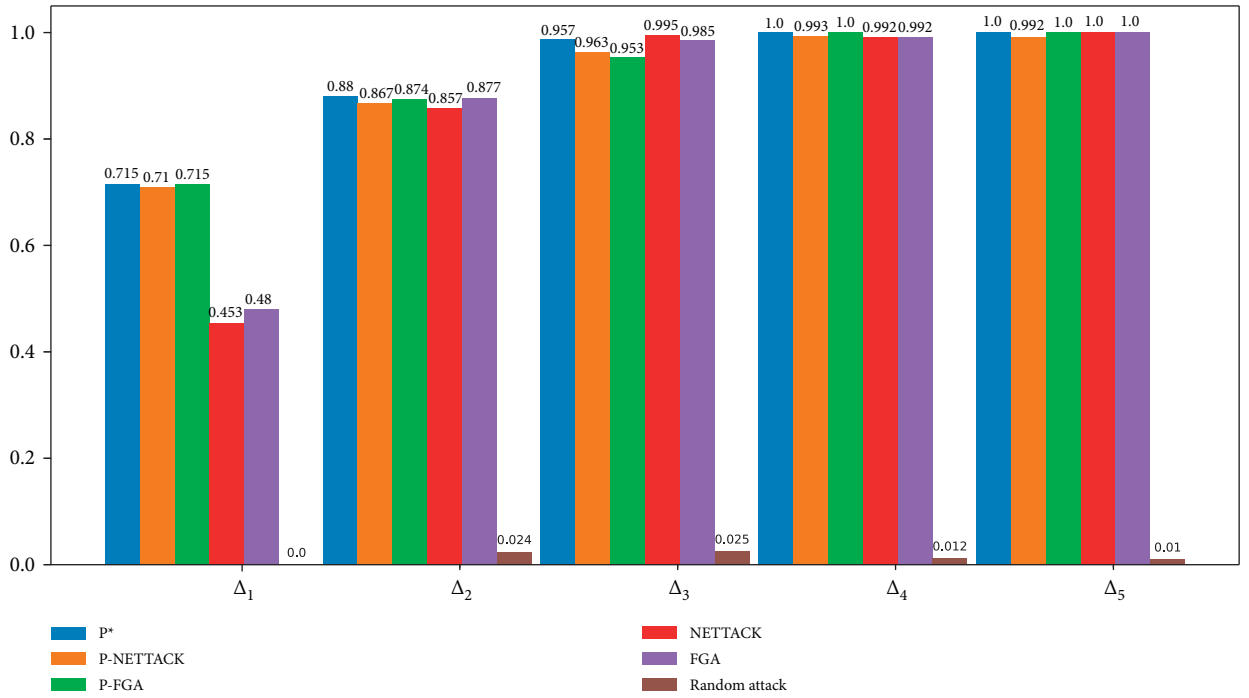

Figure 4: ASR comparison under different perturbation budget Δ.

Table 4: ASR variation under different supplementary factor k.

Perturbation budget Δ	$k=0.1$	$k=0.2$	$k=0.3$	$k=0.4$	$k=0.5$	$k=0.6$	$k=0.7$	$k=0.8$	$k=0.9$
$\Delta_1 = 1/5\ d_{sum}$	0.705	0.713	0.706	0.711	**0.715**	0.703	0.695	0.698	0.697
$\Delta_2 = 2/5\ d_{sum}$	0.864	0.874	0.878	0.865	0.86	0.879	**0.880**	0.867	0.878
$\Delta_3 = 3/5\ d_{sum}$	0.958	0.967	0.962	0.978	0.982	0.983	0.973	**0.987**	0.978
$\Delta_4 = 4/5\ d_{sum}$	**1**	**1**	**1**	**1**	**1**	0.997	**1**	0.997	0.997
$\Delta_5 = 5/5\ d_{sum}$	**1**	**1**	**1**	**1**	**1**	**1**	0.997	0.998	0.998

including TransE [30] and RESCAL [31], which is also the first investigation on adversarial attack for heterogeneous graph. Chen et al. [16] explored the adversarial attack on both node classification task and community detection task [32] based on GCN-based gradient information.

However, most works about adversarial attack on node classification only focus on the per-node attack, aiming to achieve misclassification for a target node. Although, for those per-node attack methods, the multinode attack can be performed in a sequential way, the perturbation influence of different per-node attacks is overlooked. In comparison, our parallel attack method, which considers all target nodes and perturbation influence at the same time, is better for multinode attack. In addition, as the first to propose the parallel attack on graph structure, our work can provide an inspiration for adversarial attack on other tasks in a parallel way, such as parallel adversarial attack on prediction of multiple links.

In addition to the benefits mentioned above, the main drawback of our method is that it is time-consuming, especially the P-NETTACK (see Table 7), due to the reason that, at each iteration, more candidate perturbations are taken into computation compared with sequential per-node attack. One of the solutions is developing more computationally efficient test statistic function and scoring function. On the other hand, proposing a perturbation filtering mechanism to reduce the size of multinode candidate perturbations set is also an effective way. In addition, our method does not consider the constraints of attributed graphs [33], such as attribution-based node similarity

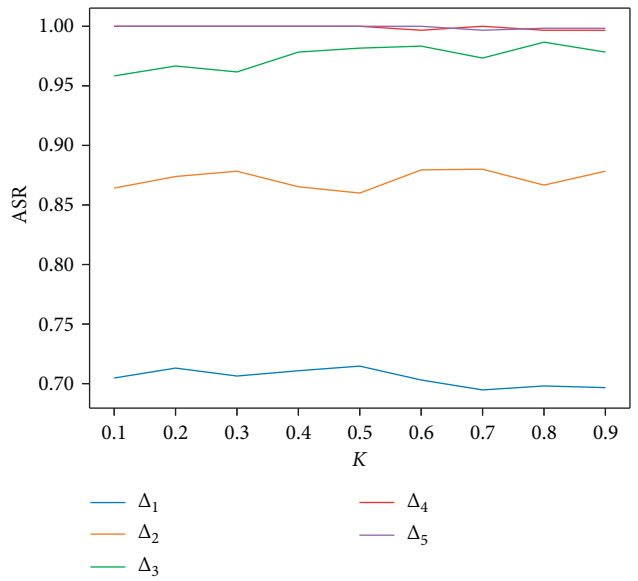

Figure 5: k influence on ASR value among different perturbation budgets Δ.

Table 5: Test statistics Λ comparison between P*, P-NETTACK, P-FGA, NETTACK, and FGA.

Perturbation budget Δ	P^*	P-NETTACK ($k=1$)	P-FGA ($k=0$)	NETTACK	FGA
$\Delta_1 = 1/5\ d_{\text{sum}}$	0.005	**0.003**	0.035	0.008	0.022
$\Delta_2 = 2/5\ d_{\text{sum}}$	0.006	**0.005**	0.100	0.013	0.091
$\Delta_3 = 3/5\ d_{\text{sum}}$	**0.005**	0.007	0.147	0.017	0.126
$\Delta_4 = 4/5\ d_{\text{sum}}$	**0.005**	**0.005**	0.166	0.016	0.134
$\Delta_5 = 5/5\ d_{\text{sum}}$	0.006	**0.004**	0.204	0.014	0.156

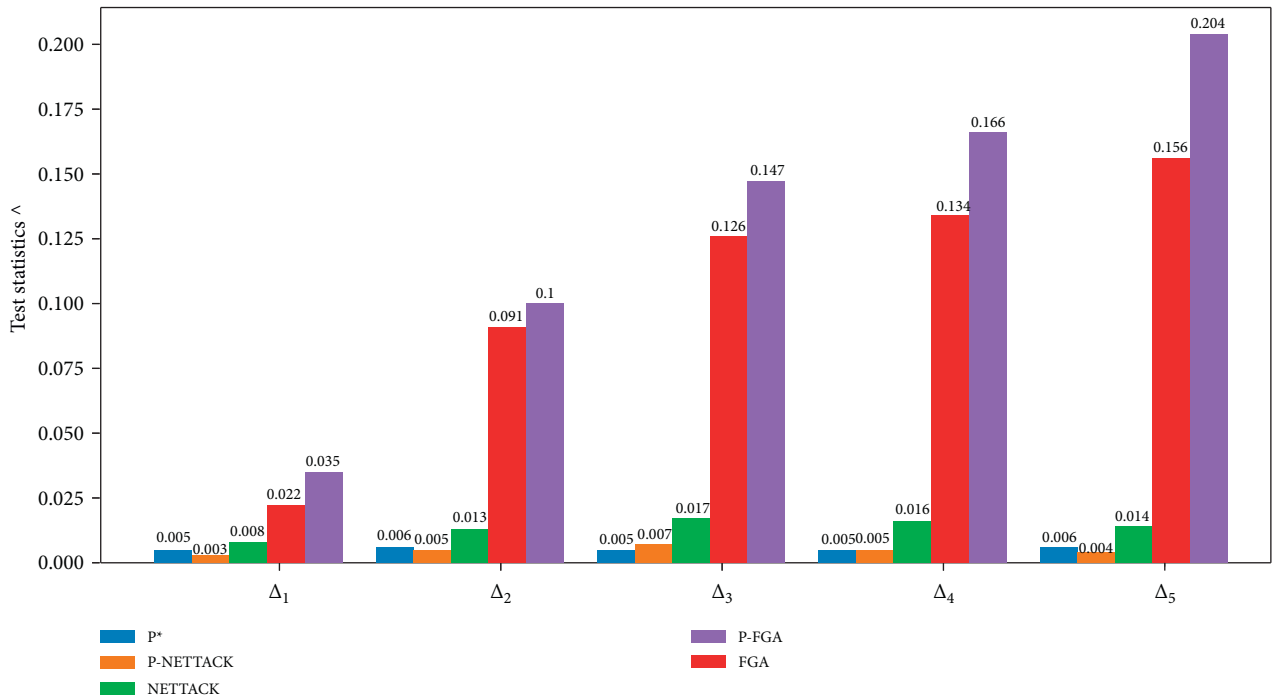

Figure 6: Test statistics Λ comparison under different perturbation budgets Δ.

Table 6: Test statistics Λ variation under different supplementary factors k.

Perturbation budget Δ	$k=0.1$	$k=0.2$	$k=0.3$	$k=0.4$	$k=0.5$	$k=0.6$	$k=0.7$	$k=0.8$	$k=0.9$
$\Delta_1 = 1/5\ d_{\text{sum}}$	0.033	0.026	0.018	0.017	0.014	0.011	0.008	0.006	0.005
$\Delta_2 = 2/5\ d_{\text{sum}}$	0.092	0.059	0.050	0.035	0.026	0.020	0.010	0.008	0.006
$\Delta_3 = 3/5\ d_{\text{sum}}$	0.124	0.098	0.081	0.045	0.033	0.021	0.019	0.006	0.005
$\Delta_4 = 4/5\ d_{\text{sum}}$	0.138	0.110	0.077	0.066	0.035	0.027	0.017	0.011	0.005
$\Delta_5 = 5/5\ d_{\text{sum}}$	0.160	0.120	0.101	0.080	0.051	0.028	0.019	0.013	0.006

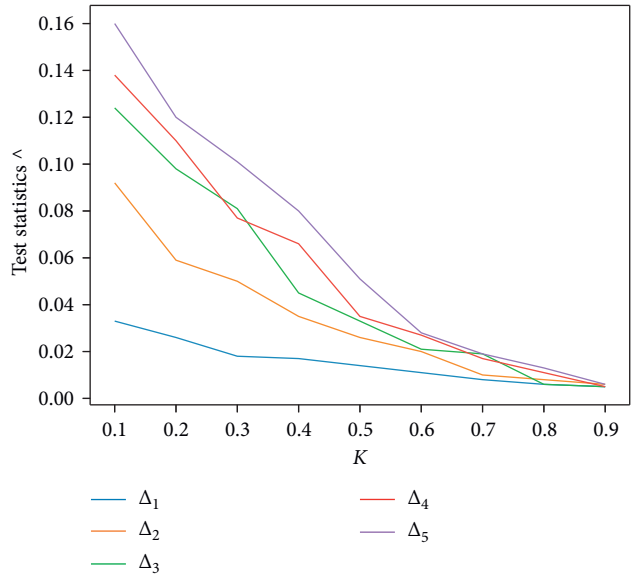

Figure 7: k influence on test statistics Λ value among different perturbation budgets Δ.

Table 7: AAS comparison between P-NETTACK, P-FGA, NETTACK, and FGA.

Methods	AAS (second)
P-NETTACK	11.17
P-FGA	0.17
NETTACK	1.30
FGA	0.14

Table 8: Filtering influence on *ASR* of P-NETTACK and P-FGA.

Perturbation budget Δ	P-NETTACK	P-NETTACK (without filtering)	P-FGA	P-FGA (without filtering)
$\Delta_1 = 1/5\ d_{\text{sum}}$	0.710	0.45	0.715	0.551
$\Delta_2 = 2/5\ d_{\text{sum}}$	0.867	0.662	0.874	0.653
$\Delta_3 = 3/5\ d_{\text{sum}}$	0.963	0.736	0.953	0.721
$\Delta_4 = 4/5\ d_{\text{sum}}$	0.993	0.818	1	0.759
$\Delta_5 = 5/5\ d_{\text{sum}}$	0.992	0.887	1	0.813

constraint [34] and attribution cooccurrence constraint [17]. Parallel multinode adversarial attack on attributed graph and Heterogeneous Information Network (HIN) [35] still needs further exploration.

7. Conclusions

In this paper, we propose a multinode parallel adversarial attack framework on node classification in socialnet of graph structure, based on considering perturbation influence between per-node attacks. Through redesigning new loss function and objective function for nonconstraint and constraint perturbations, respectively, and constructing intersection and supplement mechanisms of perturbation, we integrate nonconstraint P-FGA and constraint P-NETTACK into a unified attack framework. Based on politician socialnet Polblogs of 1222 nodes and 16714 edges, we evaluate attack success rate, test statistics, and average attack speed for our approach. Our approach shows a high attack success rate of 71.5% at the lowest perturbation budget of 1/5 d_{sum}, keeping a satisfied test statistic of 0.005.

This work severs as a first step to take security analysis on multinode parallel adversarial attack in politician socialnet. It is expected to inspire a series of follow-up studies, including but not limited to (1) adversarial attack on prediction of multiple links; (2) more concrete defense design and implementation.

Data Availability

The dataset of Polblogs can be obtained from http://networkrepository.com/polblogs.php.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61972025, 61802389, 61672092, U1811264, and 61966009), the National Key R&D Program of China (2020YFB1005604 and 2020YFB2103800), Fundamental Research Funds for the Central Universities of China (2018JBZ103 and 2019RC008), and Guangxi Key Laboratory of Trusted Software (KX201902).

References

[1] L. Strate, "The varieties of cyberspace: problems in definition and delimitation," *Western Journal of Communication*, vol. 63, no. 3, pp. 382–412, 1999.

[2] J. Li, J. Li, and X. Chen, "MobiShare+: security improved system for location sharing in mobile online social networks," *Journal of Internet Services and Information Security*, vol. 4, no. 1, pp. 25–36, 2014.

[3] A. Branitskiy, D. Levshun, N. Krasilnikova et al., "Determination of young generation's sensitivity to the destructive stimuli based on the information in social networks," *Journal of Internet Services and Information Security*, vol. 9, no. 3, pp. 1–20, 2019.

[4] D. F. Nettleton, "Data mining of social networks represented as graphs," *Computer Science Review*, vol. 7, pp. 1–34, 2013.

[5] M. Kolomeets, A. Benachour, D. El Baz et al., "Reference architecture for social networks graph analysis tool," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 4, pp. 109–125, 2019.

[6] M. Kolomeets, A. Chechulin, and I. V. Kotenko, "Social networks analysis by graph algorithms on the example of the VKontakte social network," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 2, pp. 55–75, 2019.

[7] W. Niu, G. Li, H. Tang, X. Zhou, and Z. Shi, "CARSA: a context-aware reasoning-based service agent model for AI planning of web service composition," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1757–1770, 2011.

[8] W. Niu, G. Li, Z. Zhao, H. Tang, and Z. Shi, "Multi-granularity context model for dynamic Web service composition," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 312–326, 2011.

[9] S. Dabhi and M. Parmar, "Nodenet: a graph regularised neural network for node classification," 2020, https://arxiv.org/abs/2006.09022.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, https://arxiv.org/abs/1609.02907.

[11] S. Zhang, H. Tong, and J. Xu, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, p. 11, 2019.

[12] F. Scarselli, M. Gori, and A. C. Tsoi, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[13] Z. Wu, S. Pan, and F. Chen, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[14] D. Zügner and S. Günnemann, "Certifiable robustness of graph convolutional networks under structure perturbations," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery\& Data Mining*, pp. 1656–1665, Virtual Event, CA, USA, July 2020.

[15] L. Sun, Y. Dou, and C. Yang, "Adversarial attack and defense on graph data: a survey," 2018, https://arxiv.org/abs/1812.10528.

[16] J. Chen, Y. Wu, and X. Xu, "Fast gradient attack on network embedding," 2018, https://arxiv.org/abs/1809.02797.

[17] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2847–2856, London, UK, August 2018.

[18] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, pp. 39–57, IEEE, San Jose, CA, USA, May 2017.

[19] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 US election: divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43, New York, NY, USA, August 2005.

[20] Y. Zhang and J. Koren, "Efficient bayesian hierarchical user modeling for recommendation system," in *Proceedings of the 30th Annual International Acm Sigir Conference on Research and Development in Information Retrieval*, pp. 47–54, Amsterdam, The Netherlands, July 2007.

[21] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[22] A. Bessi, "Two samples test for discrete power-law distributions," 2015, https://arxiv.org/abs/1503.00643.

[23] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, Sardinia, Italy, May 2010.

[25] S. Caton, M. Hall, and C. Weinhardt, "How do politicians use facebook? an applied social observatory," *Big Data & Society*, vol. 2, no. 2, 2015.

[26] S. Stieglitz and L. Dang-Xuan, "Social media and political communication: a social media analytics framework," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1277–1291, 2013.

[27] H. Dai, H. Li, and T. Tian, "Adversarial attack on graph structured data," 2018, https://arxiv.org/abs/1806.02371.

[28] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1024–1034, Long Beach, CA, USA, December 2017.

[29] H. Zhang, T. Zheng, and J. Gao, "Towards data poisoning attack against knowledge graph embedding," 2019.

[30] A. Bordes, N. Usunier, and A. Garcia-Duran, "Translating embeddings for modeling multi-relational data," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2787–2795, Lake Tahoe, CA, USA, December 2013.

[31] M. Nickel, V. Tresp, and H. P. Kriegel, "A three-way model for collective learning on multi-relational data," *ICML*, vol. 11, pp. 809–816, 2011.

[32] K. Allab, L. Labiod, and M. Nadif, "A semi-NMF-PCA unified framework for data clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 2–16, 2016.

[33] G. Cui, J. Zhou, and C. Yang, "Adaptive graph encoder for attributed graph embedding," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 976–985, San Diego, CA, USA, August 2020.

[34] H. Wu, C. Wang, and Y. Tyshetskiy, "Adversarial examples on graph data: deep insights into attack and defense," 2019, https://arxiv.org/abs/1903.01610.

[35] Y. Lu, Y. Fang, and C. Shi, "Meta-learning on heterogeneous information networks for cold-start recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1563–1573, San Diego, CA, USA, August 2020.