Targeted Shilling Attacks on GNN-based Recommender Systems

Sihan Guo
Beijing University of Posts and Telecommunications
Beijing, China
guosihan@bupt.edu.cn

Ting Bai*
Beijing University of Posts and Telecommunications
Beijing, China
baiting@bupt.edu.cn

Weihong Deng
Beijing University of Posts and Telecommunications
Beijing, China
whdeng@bupt.edu.cn

ABSTRACT

GNN-based recommender systems have shown their vulnerability to shilling attacks in recent studies. By conducting shilling attacks on recommender systems, the attackers aim to have homogeneous impacts on all users. However, such indiscriminate attacks suffer from a waste of resources because even if the target item is promoted to users who are not interested, they are unlikely to click on them. In this paper, we conduct targeted shilling attacks in GNN-based recommender systems. By automatically constructing the features and edges of the fake users, our proposed framework AutoAttack achieves accurate attacks on a specific group of users while minimizing the impact on non-target users. Specifically, the features of fake users are generated based on a similarity function, which is optimized according to the features of target users. The structure of fake users is learned by conducting spectral clustering on the target users based on their graph Laplacian matrix, which contains the degree and adjacency information that provides guidance to the edge generation of fake users. We conduct extensive experiments on four real-world datasets in different GNN-based RS and evaluate the performance of our method on the shilling attack and recommendation tasks comprehensively, showing the effectiveness and flexibility of our framework.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Targeted Shilling Attacks, Recommender Systems, Graph Neural Networks

ACM Reference Format:
Sihan Guo, Ting Bai*, and Weihong Deng. 2023. Targeted Shilling Attacks on GNN-based Recommender Systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3615073

* Ting Bai (baiting@bupt.edu.cn) is the corresponding author.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615073

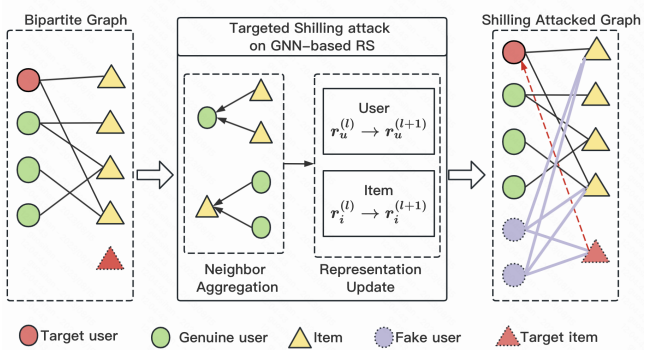

Figure 1: The process of targeted shilling attack on GNN-based recommender systems. The red circle represents the target user that we attempt to attack, the green circles represent genuine users and the purple circles are the fake users we inject. Items are marked by yellow triangles, while the target item is marked by a red triangle. The purpose of targeted shilling attacks is to promote the target item to the target user (marked by the red line). We generate fake interaction edges marked by purple lines for each fake user, which can be adopted in the GNN-based recommender systems.

1 INTRODUCTION

With the rapid development of graph neural networks, GNN-based recommender systems have achieved remarkable success in the past few years [8, 28, 35, 40]. Although GNN-based recommender systems have shown superior performance, the vulnerability of GNNs raises pressing concerns about the security issues in recommender systems [24, 42]. There have been a large number of research conducted on shilling attacks against GNN-based recommender systems [21, 34, 37]. However, most of the works focus on untargeted shilling attacks which tend to have homogeneous impacts on all users. Such indiscriminate attacks can be considered a waste of resources because even if the target item is promoted to users who are not interested, they are unlikely to click on them.

To address this issue, some works proposed targeted shilling attacks, which attempt to impact a certain user group [2, 4, 5, 17]. Targeted shilling attacks, which are called segment attacks [2] in some research, the attackers adopt the KNN algorithm to select a target user segment and popularize the target items among them. Through targeted shilling attacks, we can promote the target items specifically to users who have a genuine interest in them. It could significantly reduce attack costs and improves attack efficiency. Meanwhile, since it does not cause a significant impact on the overall recommendation performance, it can make the attacks more

covert. Furthermore, targeted shilling attacks can also be used to have a positive influence on recommendation systems. In practical terms, it can be used to precisely guide target users towards specific content or opinions in advertising recommendations and public opinion management. Recent researches [21, 27] have added fake nodes to attack GNN-based recommender systems, however, few of them have conducted shilling attacks in GNN-based recommender systems with targeted purpose.

In this paper, we take an initial attempt to propose an automatically generative framework for targeted shilling attacks in GNN-based recommender systems (termed as **AutoAttack**). The aim of targeted shilling attacks is to recommend the target items to a set of target users. The attacker can manipulate the recommendation results by injecting fake user nodes, whose features and interaction edges are automatically generated in the learning process. Following the collaborative principle in GNN-based recommender systems: similar users may have similar preferences toward items, we need to ensure that both the features and structures of fake nodes be similar to the target users, so that their interactions with target items may have collaborative impacts on the target users. To achieve this goal, the features of fake users are generated based on a similarity function, which is optimized according to the features of target users. The structure of fake users are learned by conducting spectral clustering on the target users based on their graph Laplacian matrix, which contains the degree and adjacency information that provides guidance to the edge generation of fake users. Our proposed targeted shilling attack framework AutoAttack is an end-to-end learning method. It completes the injection process of fake nodes by automatically generating their features and interaction edges. The contributions of our work are summarized as follows:

- We study a novel generative targeted shilling attacks framework (AutoAttack) in GNN-based recommender systems. Our framework is able to automatically generate the features and interaction edges of fake users and achieves accurate attacks on a specific group of target users while minimizing the impact on non-target users.
- In order to improve the precision of targeted shilling attacks, we consider both the features and structure information of fake nodes. To ensure the structural similarity of target users and fake users, we propose to conduct spectral clustering based on the graph Laplacian matrix, which is further utilized to generate the interaction edges.
- Extensive experiments are conducted on four real-world datasets, i.e., Gowalla, LastFM, Amazon-book, and Yelp, demonstrating that our proposed framework outperforms the SOTA method comprehensively (i.e., better performance on both shilling attack task and recommendation task). Our framework can be adopted into different types of GNN-based RS models, showing its flexibility.

2 PRELIMINARY

In this section, we formally define the problem of targeted shilling attacks in GNN-based recommender systems (RS).

2.1 Shilling Attacks in RS

Shilling attacks in recommender systems refer to recommending target items to users as much as possible by injecting fake users to influence the users' preferences. In the GNN-based recommendation model, the interactions between users and items can be modeled in a bipartite graph $G = (U \cup I, E)$, which consists of node sets of users U and items I, and interaction edges set E. To conduct shilling attacks, attackers will elaborately construct a set of fake users U_f and their interaction edges E_f to the items. The interaction graph after shilling attacks is $\hat{G} = (U^* \cup I, E^*)$, where $U^* = \{U \cup U_f\}$ and $E^* = \{E \cup E_f\}$.

The aim of shilling attacks is to recommend the target items I_t to all users by constructing the interaction edges to the fake users U_f. We formally define the optimization objective function of shilling attacks as:

$$\max_{u \in U, i_t \in I_t} GNN_{Rec}(u, i_t | U^*, I, E^*, I_t, \theta), \tag{1}$$

where $GNN_{Rec}(.)$ is the recommendation function to predict the probability that I_t is recommended to user u, and θ is the learning parameters in recommendation model.

2.2 Targeted Shilling Attacks in RS

Targeted shilling attacks aim to recommend target items to the target users, which is also called segment attack [2] in some studies. Different from shilling attacks, targeted shilling attacks promote the target items specifically to users who have a genuine interest in them, which could significantly reduce attack costs and improves attack efficiency, as well as cause less impact on the overall performance of recommender systems. The process of targeted shilling attacks on GNN-based recommender systems is illustrated in Fig. 1. Given a set of target users U_t, the purpose of targeted attacks is to influence the recommendation results of user $u \in U_t$ towards the target item $i \in I_t$ precisely while minimizing the impact on non-target users U_n. The optimization objective function of targeted shilling attacks can be defined as:

$$\begin{aligned} &\max_{u_t \in U_t, i_t \in I_t} GNN_{Rec}(u_t, i_t | U^*, I, E^*, I_t, U_t, \theta_t), \\ &\min_{u_n \in U_n, i_t \in I_t} GNN_{Rec}(u_n, i_t | U^*, I, E^*, I_t, U_n, \theta_n), \end{aligned} \tag{2}$$

where $GNN_{Rec}(.)$ is the recommendation function to predict the probability that I_t is recommended to target user u_t and no-target user u_n, and θ_t and θ_n are the learning parameters in recommendation model.

3 THE PROPOSED METHOD

To conduct targeted shilling attacks in GNN-based recommender systems, we propose a generative model termed as AutoAttack to automatically construct the profiles of fake nodes, as well as their interaction edges. The details of the model architecture are introduced in the following section.

3.1 A General Framework

The overview architecture of AutoAttack is shown in Fig. 2. It consists of two modules: the fake user generation module and the

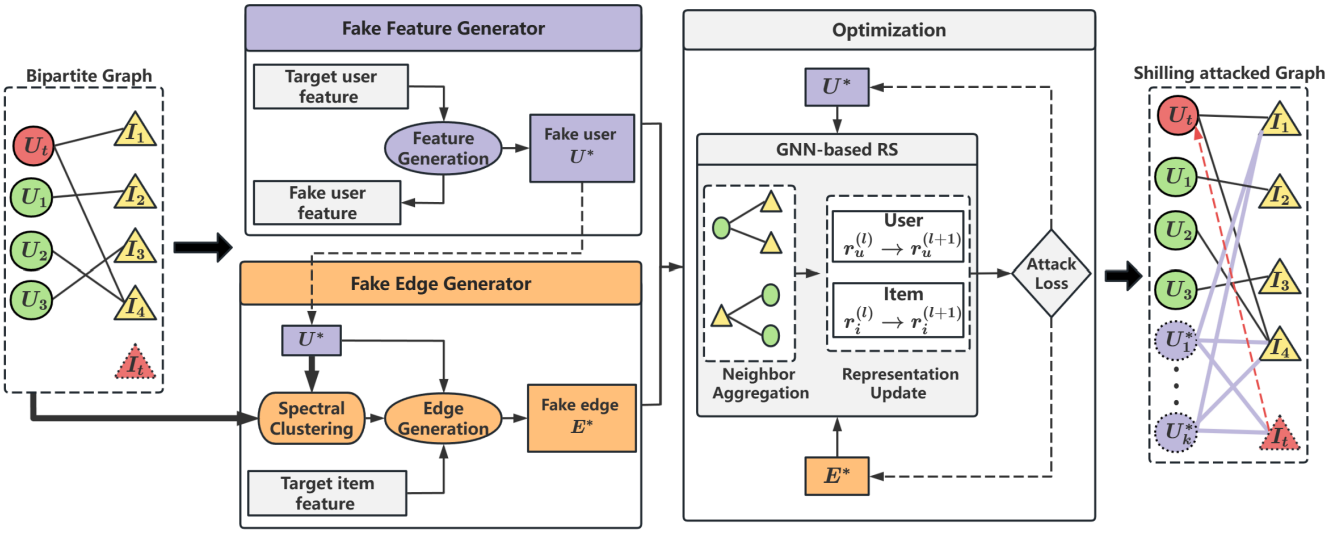

Figure 2: The framework of AutoAttack. It consists of a fake feature generator and a fake edge generator. By optimizing the attack loss function, we generate fake user features and interaction edges automatically to promote the target item to target users.

joint optimization module. Our aim is to promote target items to a group of specific users in the GNN-based recommender systems. We need to ensure that the features and structure of fake nodes be similar to the target users, so that their interactions with target items may have collaborative impacts on the target users. To achieve this goal, two generators are designed in the fake user generation module, which constructs the features and generates interaction edges for fake user nodes by imitating the features and structure of target user nodes respectively. Then the joint modeling module optimizes the attack impacts on both target and non-target users to achieve the precise attack goal.

3.2 Fake Users Generation

The generation of fake users considers both the feature and structural similarities of target users.

3.2.1 Fake Features Generation. To precisely impact the target user, the generated feature representations of fake users are supposed to closely resemble the feature distribution of the target users to enhance the precision of the shilling attacks. Specifically, for a target user $u \in U_t$ and its feature representation $\mathbf{f}_u \in \mathbb{R}^{1\times d}$, where d is the dimension of node features. The feature representation $\mathbf{f}_{u^*} \in \mathbb{R}^{1\times d}$ of fake user node $u^* \in U_f$ is generated by optimizing of the similarity function:

$$\begin{aligned} L_{sim} &= \sum_{u\in U_t, u^*\in U_f} similar(u, u^*), \\ &= \frac{1}{\sqrt{|U_t||U_f|}} \sum_{u\in U_t, u^*\in U_f} |\mathbf{f}_u - \mathbf{f}_{u^*}|. \end{aligned} \tag{3}$$

Considering that excessively high similarity between the features of fake users and target users would make the attack intention too obvious, we introduce a regularization term to constrain the generated features of fake nodes. Consequently, the features are generated with the constrained optimization function:

$$L_{sim} = \frac{1}{\sqrt{|U_t||U_f|}} \sum_{u\in U_t, u^*\in U_f} |\mathbf{f}_u - \mathbf{f}_{u^*}| - \lambda \sum_{u^*\in U_f} \frac{1}{|\mathbf{f}_u^*|}. \tag{4}$$

where λ is the constraint parameter.

3.2.2 Fake Interaction Generation. In GNN-based recommender systems, the preference of a node is aggregated by the information from its neighborhood. Hence, in addition to keeping the feature similar between fake nodes and target nodes, we also need to ensure the structural similarity between them. Based on this consideration, we conduct spectral clustering on the target users and fake nodes based on their graph Laplacian matrix, which contains the structure information (i.e., degree and adjacency information). Putting the fake users and the target users into the same cluster ensures structure similarity, which provides guidance to edge generations.

(1) *The spectral clustering operation.* We conduct spectral clustering on the graph G to obtain the category of each node. Spectral clustering is a graph-based method that classifies nodes by analyzing the eigenvectors of the graph Laplacian matrix. Assuming the adjacency matrix of G is $\mathbf{A}$, the Laplacian matrix $\mathbf{M}_{Lap}$ is calculated by:

$$\mathbf{M}_{Lap} = \mathbf{D} - \mathbf{A}, \tag{5}$$

where $\mathbf{D}$ is the degree matrix, in which each diagonal element represents the degree of nodes. We utilize the eigendecomposition $\mathbf{M}_{Lap} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ to compute the eigenvalues and eigenvectors of $\mathbf{M}_{Lap}$, represented by $\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$ and $V = \{\mathbf{v}_{\lambda_1}, \mathbf{v}_{\lambda_2}, ...\mathbf{v}_{\lambda_n}\}$ respectively. The eigenvectors represent the embedded features of the nodes in a low-dimensional space.

Then, we select the k eigenvectors corresponding to the k smallest eigenvalues and form a matrix $\mathbf{M}_V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$. Each row of $\mathbf{M}_V$ represents a node. To divide these nodes into k categories, we perform K-means clustering[10] on them. To initialize the clusterings, we randomly select K nodes as the cluster centers $C = \{c_1, c_2, ...c_k\}$ for each clustering. Then we calculate the Euclidean distance between other nodes and cluster centers. For node i and cneter j the Euclidean distance of them is $d_{ij} = \|\mathbf{v}_i - \mathbf{v}_{c_j}\|_2$. According to the Euclidean distance, we assign each non-centriod node to the category of the nearest center. After that, we update the cluster centers of each category by calculating the mean of all nodes assigned to it:

$$c_j = \frac{1}{|S_j|} \sum_{u \in S_j} u, \tag{6}$$

where $|S_j|$ is the number of nodes assigned to category S and c_j is the updated center of S. Then we repeat to update the cluster centers and the category assignment of nodes, until there is no significant change or the maximum number of iterations is reached. Finally, we can obtain k clustering of nodes $S = \{S_1, S_2, ..., S_k\}$. If two nodes are in the same clustering, we can consider them to have a high degree of structural similarity.

(2) *The generation of edges based on spectral clustering.* The interaction edges are automatically generated based on the structural similarity of the target nodes. In order to precisely influence the target user, we aim for a high degree of structural similarity between the fake users and the target users. We design an objective function to evaluate the structure similarity by spectral clustering, which is formulated as follows:

$$L_{cluster} = -\sum_{u \in U} y_u \log p_u + (1 - y_u) \log(1 - p_u), \tag{7}$$

where y_u is 1 if u is the target user, otherwise y_u is 0. p_u represents the ratio of fake users belonging to the same category as user u. Assuming that u belongs to category S_u, we calculate the number of fake users belonging to S_u as N_s. Then we can obtain p_u:

$$p_u = \frac{N_s}{N_f}, \tag{8}$$

where N_f is the total number of fake users. Since p_u is not continuous, we utilize the Gumbel-Softmax relaxation to compute the gradients. Through optimizing $L_{cluster}$, we can gradually approximate the structure of fake user nodes to that of the target user nodes.

(3) *The discretization of edges.* Considering that the edges are discrete, we can not directly optimize the generation of edges but instead learn the connection probability between item nodes and fake user nodes. For each fake user, we calculate the connection probability between them and each item by a two-layer neural network. For user u and item i, the connection probability $c_{u,i}$ of them can be formulated as follows:

$$c_{u,i} = \sigma(|\mathbf{v}_u||\mathbf{v}_i|W_1 + b_1)W_2 + b_2, \tag{9}$$

where σ is the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$, $|\mathbf{v}_u|$ and $|\mathbf{v}_i|$ are the structure characteristics of u and i obtained by spectral clustering, W_1, W_2, b_1, b_2 are learning parameters.

Initially, the connection probability $c_{u,i}$ of the target user to the target item is set to 1, while the others are set to 0. We optimize the connection probability for all fake users to minimize $L_{cluster}$ (see in Eq.7). Then we adopt the Gumbel-Softmax method to discretize the connection probability $c_{u,i}$ of fake users into edges $e_{u,i} \in E_f$. The Gumbel distribution G_i can model the distribution of sample extremum. Formally, $G_i = -\log(-\log(U_i))$ where $U_i \sim Uniform(0, 1)$. The discretized connection probability $c_{u,i}$ of the edge in the Gumbel distribution is:

$$Gumbel - Softmax(c_{u,i}) = \frac{\exp(c_{u,i} + G_i)}{\sum_{j=1}^{|I|} \exp(c_{u,j} + G_j)}. \tag{10}$$

If $Gumbel - Softmax(c_{u,i})$ is greater than the threshold (set to 0.5), we generate an edge between the user u and item i.

3.3 GNN-based Recommendation

We have generated the features and interaction edges of fake user nodes in the GNN-based recommendation model. Given the shilling attacked graph $\hat{G}$, we can obtain the prediction score $\hat{r}_{u,i}$ of user u and item i by their feature representations. Specifically, the hidden representation of a node is derived by integrating its own original features with the iteratively aggregated features of its neighbors. We adopt LightGCN [11], which is a simplified and powerful graph convolution network, as the GNN-based recommendation model. the aggregating process of LightGCN is formulated as follows:

$$\begin{aligned} \mathbf{h}_u^{(k+1)} &= \sum_{i \in N_u} \frac{1}{\sqrt{|N_u||N_i|}} \mathbf{h}_i^{(k)}, \\ \mathbf{h}_i^{(k+1)} &= \sum_{u \in N_i} \frac{1}{\sqrt{|N_u||N_i|}} \mathbf{h}_u^{(k)}, \end{aligned} \tag{11}$$

where $\mathbf{h}_i^{(k)}$ and $\mathbf{h}_u^{(k)}$ are the hidden representation of item i and user u at layer k. To form the final representations $\mathbf{h}_u$ and $\mathbf{h}_i$ for user u and item i, we combine the embeddings obtained at each layer:

$$\mathbf{h}_u = \sum_{k=0}^{H} \omega_k \mathbf{h}_u^{(k)}; \mathbf{h}_i = \sum_{k=0}^{H} \omega_k \mathbf{h}_i^{(k)}, \tag{12}$$

where ω_k is a hyperparameter denoting the importance of layer k, and H is the number of aggregation layers.

The prediction probability $r_{u,i}$ is defined as the inner product of user and item representations:

$$\hat{r}_{u,i} = \mathbf{h}_u \mathbf{h}_i^T. \tag{13}$$

3.4 Joint Optimization Function

Targeted shilling attacks aim to promote the target items specifically to the users who have a genuine interest in them. Hence for the target items I_t, we maximize the preference probability of target users U_t, the optimization function for the targeted shilling attack can be formulated as:

$$L_{tar} = -\sum_{u \in U_t, i \in I_t} \log \hat{r}_{u,i} \tag{14}$$

where $\hat{r}_{u,i}$ is the prediction probability of user u and item i obtained by the GNN-based recommender system.

As we conduct targeted shilling attacks, our method needs to cause less impact on the overall performance of recommender systems. Hence we introduce a constraint function to maintain the overall recommendation performance:

$$L_{rec} = -\sum_{u \in U, i \in I} r_{u,i} \log \hat{r}_{u,i} + (1 - r_{u,i}) \log(1 - \hat{r}_{u,i}), \tag{15}$$

where $r_{u,i}$ is the interactions of user u and item i in the initial bipartite graph G, $\hat{r}_{u,i}$ is the prediction probability of user node u and item node i based on the shilling attack graph $\hat{G}$ with fake nodes injection.

Taking all the above optimization objects and constraint conditions into account, the loss function to make jointly optimization is:

$$L = L_{tar} + \alpha L_{sim} + \beta L_{cluster} + \gamma L_{rec} \tag{16}$$

where α, β, and γ are hyper-parameters that control the impact of each optimization objective on the final attack effects.

4 EXPERIMENTS

In this section, we evaluate the shilling attack effects of our proposed method, as well as its impacts on the recommendation task.

4.1 Experimental Settings

4.1.1 Datasets. We conduct experiments on four real-world datasets, namely Gowalla, LastFM, Amazon-book, and Yelp. The dataset statistics are shown in Table. 1. Our experiments are conducted in an implicit recommendation scenario. For the interactions with rating scores, we classify the scores into 0 and 1 (if rating score > 3).

- Gowalla[1]: is a social networking dataset concluding friend relationships and check-in records of users.
- LastFM[2]: is a popular music dataset consisting of user listening histories and tagging records.
- Amazon-book[3]: is a widely used dataset in recommender systems, which contains the users' rating records for books collected from the Amazon online bookstore.
- Yelp[4]: is a popular dataset used in business data analysis.It primarily consists of user ratings for businesses and also includes some user-commodity interactions.

Table 1: The statistics of datasets.

Dataset	#Users	#Items	#Interactions
Gowalla	29,858	40,981	1,027,370
LastFM	1,892	17,632	186,479
Amazon-book	52,643	91,599	2,984,108
Yelp	31,831	40,841	1,666,869

[1]http://snap.stanford.edu/data/loc-gowalla.html
[2]https://grouplens.org/datasets/hetrec-2011/
[3]https://snap.stanford.edu/data/amazon/
[4]https://www.kaggle.com/yelp-dataset/yelp-dataset

4.1.2 Baselines. We compare the AutoAttack to five different shilling attack algorithms.

- Random Attack[14]: Attackers randomly choose a user from the user set as the profile template and construct the features of all fake users based on the characteristics of the template. For each fake user, a certain number of items are selected randomly from the item set as the interaction items along with the target item. It requires minimal prior knowledge of recommender systems.
- Popular Attack [22]: is similar to the random attack. The difference is that popular attacks only select the well-known popular items as the interaction items of fake users. This way, the target item will be associated with the popular items and promoted to more users.
- Vote Attack: is a popular attack-based method designed to attack a group of specific users. All target users vote for their interaction items, and a certain number of items with the highest number of votes (i.e., the most popular items in the target users' group), along with the target item, are selected as the interaction items for the fake users.
- Segment Attack [2]: is a method to conduct shilling attacks on a set of users with similar tastes. The attacker selects a set of similar users interested in the target item as the target segment and constructs the fake user profiles based on their characteristics and interaction behaviors.
- Greedy-GAN Attack [27] is a classical method to perform fake node attacks on graph convolutional networks. It injects fake nodes and corresponding fake edges into the graph and updates the features and links one by one.

Among the above attack methods, the random attack, popular attack, and Greedy-GAN attack are untargeted shilling attack methods, while vote attack and segment attack are the targeted shilling attack methods. The ways to construct the profiles of fake users in the vote attack and segment attack are mutually designed according to the profiles of target users. The most similar work to AutoAttack is Greedy-GAN Attack, which is a GNN-based shilling attack method for all users. Different from the above methods, our proposed AutoAttack is a targeted shilling attack framework. The impacts of shilling attacks on target users and non-target users should be considered simultaneously in the attack process. AutoAttack is an end-to-end learning approach that completes the injection process of fake nodes by automatically generating their features and interaction edges in the GNN-based recommender systems.

4.1.3 Evaluation Metrics. In order to quantitatively evaluate the impacts of AutoAttack, we design four evaluation metrics covering the performances of shilling attacks and recommendations. The effects of shilling attacks are evaluated by the access rate, overflow rate, and effective time. The impact of shilling attacks on recommendations is evaluated by the hit ratio of clicked items to the original users.

(1) **Access Rate.** Targeted shilling attacks aim to promote the target item I_t to the specific target user group U_t. The attack effect can be evaluated intuitively by the ranking of I_t in the recommendation lists for U_t. we use AccessRate@k to evaluate the I_t appears

Table 2: The performance of shilling attacks on different attack methods on four real-world datasets. The underline highlights the best-performing results in the compared baselines.

Attack	Amazon			Gowalla			LastFM			Yelp		
	Access↑	Over↓	Time↑	Access↑	Over↓	Time↑	Access↑	Over↓	Time↑	Access↑	Over↓	Time↑
Random	0.316	3.427	3	0.342	2.873	5	0.379	4.503	4	0.385	2.311	4
Popular	0.327	3.067	3	0.395	2.633	6	0.403	4.325	4	0.418	2.037	5
Greedy-GAN	0.529	1.522	8	0.614	1.212	14	0.537	1.775	12	0.547	1.36	9
Vote	0.576	1.127	9	0.632	0.983	9	0.607	1.27	8	0.618	1.042	10
Segment	0.769	0.592	12	0.736	0.592	13	0.814	0.48	15	0.805	0.425	15
AutoAttack	**1.0**	**0.015**	**20**	**1.0**	**0.02**	**20**	**1.0**	**0.03**	**20**	**1.0**	**0.02**	**20**

in the top K of the recommendation lists for U_t, defined as:

$$AccessRate@k = \frac{\#(U_t, I_t)}{|U_t|}, \tag{17}$$

where $\#(U_t, I_t)$ is the number of target users that the target items ranked in the top k of the recommendation lists, and $|U_t|$ is the total number of target users. To make accurate evaluations, the target users are randomly selected from the group of users whose original recommendation lists do not include target items. Access rate calculates the proportion of target users who are affected by the shilling attacks, which can directly reflect the effectiveness of AutoAttack.

(2) **Overflow Rate.** To achieve precisely targeted attacks, it is necessary to minimize the impact on non-target users U_n. We use overflow rate to evaluate the impact of the targeted attack on non-target users, which is defined as:

$$OverflowRate = \frac{AccessRate(U_n)}{AccessRate(U_t)}, \tag{18}$$

where $AccessRate(U_n)$ is the ratio of non-target users that are impacted by the attacks and $AccessRate(U_t)$ is the ratio of target users that are affected successfully.

(3) **Effective Times.** To evaluate the long-term effects of shilling attacks, we adopt effective times to evaluate the lasting effects in the inference process, which is defined as the number of inference epochs in which the target item I_t appears in the ranking lists of target users U_t after one round of attacks. In our experiments, we set the observation window to 20 epochs. If the effective duration is 20, it suggests that a single round of attacks can continuously exert its influence throughout the entire inference process. On the contrary, the effective duration is 0 indicating that the attack fails.

(4) **Hit Ratio.** Since the attack goal is to precisely intervene with target users, it is important to ensure that the overall performance of the recommender system does not significantly decline. We adopt the Hit ratio at rank k (Hit@k) to evaluate the recommendation performance.

4.1.4 Parameter settings. For evaluating the attack performance, we randomly divide the data into three parts, 80% for training, 10% for testing, and 10% for validation. The number of target users is set to 20. The injected number of fake users accounted for 1% of the total users, and the number of interaction edges for each fake user is set to be the average number of degrees of user nodes in the interaction bipartite graph. The embedding size, training epoch, and learning rate are set to 64, 200, and 0.05 respectively. The code will be publicly available after the review process.

4.2 Main Results

We adopt the LightGCN [11] as the GNN-based recommendation model and report the results from two aspects: the shilling attack performance and recommendation performance.

4.2.1 The Performance of Shilling Attacks. We evaluate the performance of shilling attacks with access rate, overflow rate, and effective time. As shown in Table. 2, we can observe that:

(1) For the access rate, we report the results that the target item appears in the top 20 of the recommendation lists for target users. We can see that our proposed framework AutoAttack achieves the 100% access rate, which indicates that the target items had been recommended to the top-20 recommended list of each target user. The improvement of AutoAttack is approximately an average of 28% on four datasets compared to the SOTA baseline-Segment Attack, showing the precise intervention ability of our proposed framework.

(2) The overflow rate estimates the intervention on non-target users. We can see that, the overflow rate of AutoAttack is much lower than other shilling attack methods, indicating the precise attacking ability of our method. The overflow rate in other baselines (except for segment and vote attack methods) is larger than 1, indicating the effect on non-target users is larger than on target users, making them difficult to achieve targeted shilling attacks.

(3) In terms of the long-term impacts, AutoAttack also exhibits outstanding shilling attack effects. The effective times of AutoAttack can reach 20 over all datasets, which is approximately 42.8% higher than the maximum effective times in the baseline. It indicates that AutoAttack is capable to achieve a long-term effective impact on the recommendation results.

4.2.2 The Performance of Recommendations. In addition to the performance of shilling attacks in recommender systems, it is also important to evaluate their influence on the recommendation results. The targeted shilling attacks should not decrease the overall recommendation performance, but rather selectively influence the recommendation results for the target users. The decrease in recommendation performance will affect user experience, resulting in user attrition, which makes the shilling attacks meaningless. We utilize the Hit ratio at rank 10 (i.e., HR@10) to evaluate the performance of the recommender systems on the original testing labels after being subjected to attacks. As shown in Table. 3, compared

to other baselines, AutoAttack achieves a minimal reduction of the recommendation performance, showing less influence on the original recommendation results. This ensures that the attacks remain unnoticeable and less likely to be discovered.

Table 3: The performance of recommendations after being attacked on four real-world datasets.

HR@10	Amazon	Gowalla	LastFM	Yelp
Random	0.498	0.503	0.664	0.562
Popular	0.518	0.569	0.692	0.589
Vote	0.466	0.578	0.687	0.592
Segment	0.479	0.507	0.599	0.597
Greedy-GAN	0.518	0.553	0.607	0.538
AutoAttack	**0.587**	**0.661**	**0.754**	**0.683**
BeforeAttack	0.825	0.873	0.941	0.907

4.3 Experimental Analysis

In the previous sections, we have verified the effectiveness of our proposed framework AutoAttack. In this section, we will conduct in-depth analyses of our approaches, including the ablation study, the flexibility demonstration, and the hyperparameter analysis.

4.3.1 Ablation Study. We conduct ablation experiments to investigate the impact of different modules on the attack performance. This helps in understanding the contribution of each module in the overall attack framework. To explore the role of each optimization constraint in the fake user generation process, we create several variants by removing the corresponding parts from the framework.

- AutoAttack-Target: this variant removes the preference prediction of target users. Specifically, the constraint of targeted attack L_{tar} (see in Eq. 14) is removed.
- AutoAttack-Feature: we use a random generation of fake user features to replace the constraint of feature similarity function L_{sim} (see in Eq. 3).
- AutoAttack-Structure: we directly select the most popular items as the interaction items of fake users, without considering the structure of target user nodes. The proposed spectral clustering module $L_{cluster}$ (see in Eq. 7) is removed.

Table 4: The attack performance of different variations.

Variation	AccessRate@20	OverflowRate
Ours-Target	0.642	0.359
Ours-Feature	0.579	0.724
Ours-Structure	0.401	1.391
AutoAttack	**1.0**	**0.03**

We report the results on the LastFM dataset in Table 4, we can see that: The optimization constraint of target attack L_{tar} is indeed crucial during the generation process as it determines the access rate and precision of the targeted attacks. After removing the corresponding part, the access rate of attack significantly decreases and the overflow rate substantially increases. Without the constraint of target attacks, the attacker will attempt to have a wide-ranging impact on all users. Besides, the similarities of features and structures to the target users are useful in generating the profiles of fake nodes. Without consideration of each of them, the performance of targeted shilling attacks decreases. The reduction of the variant Ours-Structure is more than Ours-Feature, indicating the importance of incorporating the structure information.

4.3.2 The Verification of Model Flexibility. In this section, we explore the flexibility of our proposed method. Note our proposed AutoAttack is a targeted shilling attacks framework in GNN-based recommender systems. Our framework can be adopted into various kinds of GNN-bassed RS. In addition to the LightGCN used in our experiments, we also adopt three typical GNN-based recommendation algorithms: NGCF [28], AGCN [35] and HashGCN [25] to verify the effectiveness of our framework. The experiment results on the LastFM dataset are shown in Table. 5. We can see that our framework with different recommendation algorithms achieves comparable performance on both shilling attacks and recommendations, showing the good flexibility of our framework in the GNN-based recommender systems, which enables it to be applied flexibly in the downstream applications.

Table 5: The transferability of AutoAttack across different GNN-based recommender systems.

Victim-RS	AccessRate	EffectiveTime	Overflow	HR@10
LightGCN	1.0	20	0.03	0.754
NGCF	0.932	20	0.061	0.738
AGCN	0.884	20	0.129	0.641
HashGCN	0.875	20	0.147	0.637

4.3.3 The Impact of Target Users' Preference. Considering that the preference of users toward target items may influence the effect of shilling attacks, we study the impact of the target user's preferences. We classify the users into three categories based on their interaction times with the target item: highly interested users, low interested users, and neutral users, represented as $U_{positive}$, $U_{negative}$, and $U_{neutral}$ respectively. For different user groups, we conduct targeted attacks separately. As shown in Fig. 3, we can see that the level of interest that target users have in the target item is positively correlated with the effectiveness of the shilling attacks on all the attack approaches. Our proposed AutoAttack achieves precise attacks even when the user has neutral interests in target items.

4.3.4 The Impact of Fake Nodes' Profiles. We explore the effects of the number of fake users and the number of fake edges on the shilling attack performance. To inject different numbers of fake users, we use the injection rate to vary the number of fake users, which refers to the proportion of fake users to the total number of users. The injection rate varies within the range of $\{0.1\%, 0.5\%, 1\%, 2\%, 5\%, 10\%\}$. For the fake edges of each fake user, we set six different degrees: $\{50\%, 80\%, 100\%, 120\%, 160\%, 200\%\}$ based on the average node degree in the graph. The attack performances with different fake users and fake edges are shown in Fig. 4. We can see that: as the number of fake users increases, the access rate of target items are increasing, but the performance of the recommendation system is decreasing. When the injection rate of fake users

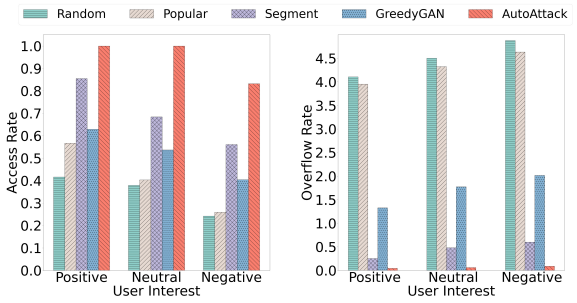

Figure 3: The correlations between attack performance and target user interests.

exceeds 1%, the increase in shilling attack performance is slight, but the degradation of recommendation performance is more serious. Besides, as for the fake edges, the access rate improves with the increasing number of fake edges at the beginning. However, when the edge number reaches 120%, the access rate starts to decline. The possible reason is that injecting too many fake interactions may introduce noise and lead to a decrease in user preference for target items. In addition, the recommendation performance of the victim model continuously deteriorates.

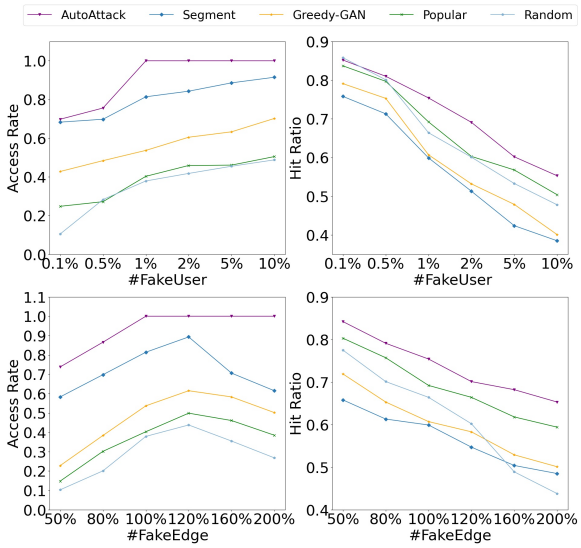

Figure 4: The impact of fake user profiles: the number of fake users and fake edges.

5 RELATED WORK

In this section, we introduce the concepts related to our study briefly. We first discuss the GNN-based recommender systems, and then we introduce the targeted shilling attack methods.

5.1 GNN-based Recommender System

Recommender systems based on Graph Neural Networks (GNNs) have exhibited superior performance [7, 11, 36, 39, 40], due to the ability of GNNs to learn the latent collaborative signal of user-item interactions [8]. GNN-based recommendation algorithms [13, 18, 30, 30, 38] model the user-item interaction matrix into a bipartite graph and convert the recommendation problem into a link prediction problem. To integrate the bipartite graph structure into the embedding process, Wang et al. [28] proposed Neural Graph Collaborative Filtering (NGCF), which exploits the high-order connectivity of users and items. Based on NGCF, He et al. [11] designed a simplified GCN-based RS (LightGCN) which can achieve better performance by linearly propagating the embeddings. However, GNN-based RS are not robust enough and are susceptible to shilling attacks[21, 29, 34]. By studying these attack methods, we can enhance the security and trustworthiness of recommendation systems, leading to a better user experience. Additionally, through appropriate interventions, we can leverage attacks to achieve beneficial effects, such as targeted recommendation, enhancement of recommendation diversity, and exploration of new content. In this paper, we attempt to explore the targeted shilling attack methods against the GNN-based recommender systems.

5.2 Untargeted Shilling Attacks

Previous work has proved that recommender systems are vulnerable to shilling attack [9, 19]. Shilling attacks can manipulate the recommendation results towards the attacker's desire by the injection of carefully-crafted fake users. The existing attack methods mostly focus on untargeted attacks. In early works, untargeted attacks were conducted using model-agnostic heuristic methods which directly constructed fake users based on prior knowledge, such as random attack [14], popular attack [22] and love-hate attack [19]. The heuristic methods are convenient and quick, but the effectiveness of the attacks is often insufficient.

With the development of neural networks, the generative methods of fake users have evolved from heuristic approaches to optimization-based methods [3, 12, 20, 23, 34]. The optimization-based attack methods are specially designed for the particular recommender systems, which means that they require full knowledge about the targeted systems [26]. For example, Li et al. [16] proposed a gradient-based optimization method for untargeted shilling attacks on factorization-based systems. They adopted the stochastic gradient Langevin dynamics optimization method to mimic the genuine users. Zhang et al. [41] designed an adversarial attack approach that can handle data incompleteness and perturbation by incorporating context-specific heuristic rules. Considering the superiority of graph structures in recommendation systems, recent works have also started to explore untargeted attack methods against recommender systems based on graph and GNN[6, 21, 29, 34, 37]. Nguyen Thanh et al. [21] firstly develop the generative attack method towards GNN-based recommender systems. The fake users and interactions are generated by a sequential attack framework GSPAttack, and optimized by a surrogate model. During the generation process, a Generative Adversarial Network (GAN) is used to craft unnoticeable fake users. And the fake edges are generated based on the prior knowledge of item popularity so that the target items can be promoted to more users.

5.3 Targeted Shilling Attacks

Untargeted shilling attacks tend to have an equal impact on all users, ignoring the differences in user characteristics. Indiscriminate attacks on all users would increase the cost of attacks and decrease their efficiency. In order to address this limitation, some works have proposed targeted shilling attack methods specifically designed for certain user groups. For instance, Burke et al. [2] proposed a segment attack method that aimed to impact a targeted set of users with similar tastes. Firstly, a set of similar users interested in the target item is selected as the target market segment. After that, the fake user profiles are constructed based on the characteristics of the target segment. Segment attacks can be successful on both user-based and item-based collaborative filtering. Similarly, Cheng and Hurley [4] designed an obfuscated attack method on model-based recommender systems, applying k-means clustering to identify the target user segments. It can achieve high diversity attack which is obfuscated to avoid PCA-based detection. What's more, some researchers [31, 32] utilize power users as the target user group. Power users can exert considerable influence over the recommendation results to other users, identified by in-degree centrality.

There are also some optimization-based methods for targeted shilling attacks. Lin et al. [17] presented an augmented shilling attack framework (AUSH) implemented by GAN. AUSH can target a specific user group by incorporating a shilling loss which increases the attack impact on users interested in the selected items. More recently, Fang et al. [5] designed an influence function based poisoning attack method for matrix-factorization-based recommender systems. The influence function is used to select a set of users who are influential to the recommendation results. The optimization problem of generated profiles is solved based on the influential users to improve the performance of model training. However, there is no existing target attack method that can be applied to GNN-based recommender systems. To withstand the impact of shilling attacks, some studies have focused on identifying approaches to build robust recommender systems. Wu et al. [33] proposed an adversarial poisoning training method that utilized injecting fake users to minimize empirical risk and build a robust system. Additionally, there are also some methods that can effectively detect targeted attacks. For example, Lee and Zhu [15] adopted a multidimensional scaling approach to detect distinct behaviors and discriminated attack users by clustering-based methods. Bilge et al. [1] designed a detection method via bisecting k-means clustering for particularly specific attacks.

6 CONCLUSION

In this paper, we proposed AutoAttack, an automatically generative targeted shilling attack framework designed for GNN-based recommender systems. AutoAttack promotes target items to a specific target user group by injecting a set of fake users, which features and structures are automatically generated according to the imitation of target users. AutoAttack is a general targeted shilling attack framework that could be equipped with different types of GNN-based recommendation algorithms, making it possible to be flexibly utilized in downstream applications.

In this work, we focus on proposing an effective targeted shilling attack approach. In the future, we will attempt to consider the detection of shilling attacks and design an unnoticeable shilling attack method.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant No.62102038. This work was supported in part by the National Natural Science Foundation of China under Grant No. 62236003.

REFERENCES

[1] Alper Bilge, Zeynep Ozdemir, and Huseyin Polat. 2014. A novel shilling attack detection method. *Procedia Computer Science* 31 (2014), 165–174.
[2] Robin Burke, Bamshad Mobasher, Runa Bhaumik, and Chad Williams. 2005. Segment-based injection attacks against collaborative filtering recommender systems. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 4–pp.
[3] Liang Chen, Yangjun Xu, Fenfang Xie, Min Huang, and Zibin Zheng. 2021. Data poisoning attacks on neighborhood-based recommender systems. *Transactions on Emerging Telecommunications Technologies* 32, 6 (2021), e3872.
[4] Zunping Cheng and Neil Hurley. 2009. Effective diverse and obfuscated attacks on model-based recommender systems. In *Proceedings of the third ACM conference on Recommender systems*. 141–148.
[5] Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. 2020. Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of The Web Conference 2020*. 3019–3025.
[6] Minghong Fang, Guolei Yang, Neil Zhenqiang Gong, and Jia Liu. 2018. Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference*. 381–392.
[7] Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. 2022. Graph neural networks for recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1623–1625.
[8] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2021. Graph neural networks for recommender systems: Challenges, methods, and directions. *arXiv preprint arXiv:2109.12843* (2021).
[9] Ihsan Gunes, Cihan Kaleli, Alper Bilge, and Huseyin Polat. 2014. Shilling attacks against recommender systems: A comprehensive survey. *Artificial Intelligence Review* 42, 4 (2014).
[10] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.
[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
[12] Hai Huang, Jiaming Mu, Neil Zhenqiang Gong, Qi Li, Bin Liu, and Mingwei Xu. 2021. Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644* (2021).
[13] Zan Huang, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. 2002. A graph-based recommender system for digital library. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*. 65–73.
[14] Shyong K Lam and John Riedl. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*. 393–402.
[15] Jong-Seok Lee and Dan Zhu. 2012. Shilling attack detection—a new approach for a trustworthy recommender system. *INFORMS Journal on Computing* 24, 1 (2012), 117–131.
[16] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. 2016. Data poisoning attacks on factorization-based collaborative filtering. *Advances in neural information processing systems* 29 (2016).
[17] Chen Lin, Si Chen, Hui Li, Yanghua Xiao, Lianyun Li, and Qian Yang. 2020. Attacking recommender systems with augmented user profiles. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 855–864.
[18] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Fairmatch: A graph-based approach for improving aggregate diversity in recommender systems. In *Proceedings of the 28th ACM conference on user modeling, adaptation and personalization*. 154–162.
[19] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. 2007. Toward trustworthy recommender systems: An analysis of attack models and

algorithm robustness. *ACM Transactions on Internet Technology (TOIT)* 7, 4 (2007), 23–es.

[20] Praveena Narayanan et al. 2021. Hybrid CNN and RNN-based shilling attack framework in social recommender networks. *EAI Endorsed Transactions on Scalable Information Systems* 9, 35 (2021).

[21] Toan Nguyen Thanh, Nguyen Duc Khang Quach, Thanh Tam Nguyen, Thanh Trung Huynh, Viet Hung Vu, Phi Le Nguyen, Jun Jo, and Quoc Viet Hung Nguyen. 2023. Poisoning GNN-based recommender systems with generative surrogate-based attacks. *ACM Transactions on Information Systems* 41, 3 (2023), 1–24.

[22] Michael P O'mahony, Neil J Hurley, and Guenole CM Silvestre. 2004. An evaluation of neighbourhood formation on the performance of collaborative filtering. *Artificial Intelligence Review* 21, 3 (2004), 215–228.

[23] Dazhong Rong, Qinming He, and Jianhai Chen. 2022. Poisoning Deep Learning based Recommender Model in Federated Learning Scenarios. *arXiv preprint arXiv:2204.13594* (2022).

[24] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. 2020. Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach. In *Proceedings of the Web Conference 2020*. 673–683.

[25] Qiaoyu Tan, Ninghao Liu, Xing Zhao, Hongxia Yang, Jingren Zhou, and Xia Hu. 2020. Learning to hash with graph neural networks for recommender systems. In *Proceedings of The Web Conference 2020*. 1988–1998.

[26] Jiaxi Tang, Hongyi Wen, and Ke Wang. 2020. Revisiting adversarially learned injection attacks against recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 318–327.

[27] Xiaoyun Wang, Minhao Cheng, Joe Eaton, Cho-Jui Hsieh, and Felix Wu. 2018. Attack graph convolutional networks by adding fake nodes. *arXiv preprint arXiv:1810.10751* (2018).

[28] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[29] Yongwei Wang, Yong Liu, and Zhiqi Shen. 2022. Revisiting Item Promotion in GNN-based Collaborative Filtering: A Masked Targeted Topological Attack Perspective. *arXiv preprint arXiv:2208.09979* (2022).

[30] Ziqi Wang, Yuwei Tan, and Ming Zhang. 2010. Graph-based recommendation on social networks. In *2010 12th International Asia-Pacific Web Conference*. IEEE, 116–122.

[31] David C Wilson and Carlos E Seminario. 2013. When power users attack: assessing impacts in collaborative recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*. 427–430.

[32] David C Wilson and Carlos E Seminario. 2014. Evil twins: Modeling power users in attacks on recommender systems. In *User Modeling, Adaptation, and Personalization: 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings 22*. Springer, 231–242.

[33] Chenwang Wu, Defu Lian, Yong Ge, Zhihao Zhu, Enhong Chen, and Senchao Yuan. 2021. Fight fire with fire: towards robust recommender systems via adversarial poisoning training. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1074–1083.

[34] Fan Wu, Min Gao, Junliang Yu, Zongwei Wang, Kecheng Liu, and Xu Wang. 2021. Ready for emerging threats to recommender systems? A graph convolution-based generative shilling attack. *Information Sciences* 578 (2021), 683–701.

[35] Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, Yanjie Fu, and Meng Wang. 2020. Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 679–688.

[36] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[37] Zih-Wun Wu, Chiao-Ting Chen, and Szu-Hao Huang. 2022. Poisoning attacks against knowledge graph-based recommendation systems using deep reinforcement learning. *Neural Computing and Applications* (2022), 1–19.

[38] Kaige Yang and Laura Toni. 2018. Graph-based recommendation system. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 798–802.

[39] Ruiping Yin, Kan Li, Guangquan Zhang, and Jie Lu. 2019. A deeper graph neural network for recommender systems. *Knowledge-Based Systems* 185 (2019), 105020.

[40] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[41] Hengtong Zhang, Changxin Tian, Yaliang Li, Lu Su, Nan Yang, Wayne Xin Zhao, and Jing Gao. 2021. Data poisoning attack against recommender system using incomplete and perturbed data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2154–2164.

[42] Daniel Zügner, Oliver Borchert, Amir Akbarnejad, and Stephan Günnemann. 2020. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 5 (2020), 1–31.