Gray-Box Shilling Attack: An Adversarial Learning Approach

ZONGWEI WANG, School of Big Data and Software Engineering, Chongqing University
MIN GAO*, School of Big Data and Software Engineering, Chongqing University
JUNDONG LI, Department of Electrical and Computer Engineering, Department of Computer Science, and School of Data Science, University of Virginia
JUNWEI ZHANG, School of Big Data and Software Engineering, Chongqing University
JIANG ZHONG, College of Computer Science, Chongqing University

Recommender systems are essential components of many information services, which aim to find relevant items that match user preferences. Several studies have shown shilling attacks can significantly weaken the robustness of recommender systems by injecting fake user profiles. Traditional shilling attacks focus on creating hand-engineered fake user profiles, but these profiles can be detected effortlessly by advanced detection methods. Adversarial learning, emerged in recent years, can be leveraged to generate powerful and intelligent attack models. To this end, in this paper, we explore potential risks of recommender systems and shed light on a gray-box shilling attack model based on generative adversarial networks, named GSA-GANs. Specifically, we aim to generate fake user profiles that can achieve two goals: unnoticeable and offensive. Towards these goals, there are several challenges that we need to address: (1) learn complex user behaviors from user-item rating data; (2) adversely influence the recommendation results without knowing the underlying recommendation algorithms. To tackle these challenges, two essential GAN modules are respectively designed to make generated fake profiles more similar to real ones and harmful to recommendation results. Experimental results on three public datasets demonstrate that the proposed GSA-GANs framework outperforms baseline models in attack effectiveness, transferability, and camouflage. In the end, we also provide several possible defensive strategies against GSA-GANs. The exploration and analysis in our work will contribute to the defense research of recommender systems.

CCS Concepts: • **Recommder system → Robustness analysis**.

Additional Key Words and Phrases: shilling attack, adversarial learning, GANs

1 INTRODUCTION

With the rapid growth of online services, recommender systems have attracted an increasing amount of attention [8, 39, 42, 44] and been widely deployed to recommend items (e.g., videos, movies, and news articles) due to their strong capability to find relevant items that match users' preference. These systems usually make recommendations by analyzing the historical behavior data generated by users represented as a user-item rating matrix. Collaborative filtering (CF) [3, 12], as one of the most popular methods, focuses on the similarity between

*Corresponding author

Authors' addresses: Zongwei Wang, zongwei@cqu.edu.cn, School of Big Data and Software Engineering, Chongqing University; Min Gao, gaomin@cqu.edu.cn, School of Big Data and Software Engineering, Chongqing University; Jundong Li, jundong@virginia.edu, Department of Electrical and Computer Engineering, Department of Computer Science, and School of Data Science, University of Virginia; Junwei Zhang, jw.zhang@cqu.edu.cn, School of Big Data and Software Engineering, Chongqing University; Jiang Zhong, zhongjiang@cqu.edu.cn, College of Computer Science, Chongqing University.

2157-6904/2022/3-ART $15.00
https://doi.org/10.1145/3512352

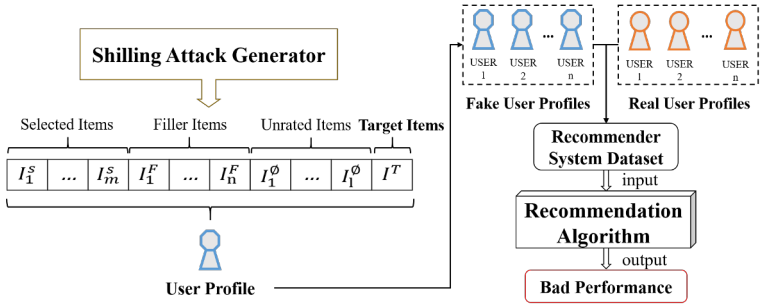

Fig. 1. An illustration of fake profiles and the process of shilling attacks.

users, or alternatively, between items for recommendation. In particular, user-based CF targets at finding similar users for target users and combines their preferences over items to fullfil the recommendation task. This strategy is consistent with the intuition that users are more likely to choose the items that their friends like. Similarly, item-based CF aims to recommend items that are similar to the ones that target users have bought or liked before. Both of these methods are valid for the recommendation task.

In fact, CF-based recommendation algorithms have achieved immense success in various real-world applications. Despite that, it is difficult to guarantee the robustness of these systems, mainly because of their vulnerability to shilling attacks (a.k.a. data poisoning attacks) [15, 33, 36, 43]. Specifically, shilling attacks aim to inject fake profiles (a.k.a. attack profile) into the user-item rating matrix to affect the process of finding similar users/items, and then the recommendation results will be adversely influenced as attackers desire, e.g., increase the popularity of some targeted items (a.k.a. push attack) or decrease the popularity of some other items (a.k.a. nuke attack) [32]. Several studies [11, 28] have demonstrated that injecting 1% fake profiles is sufficient to seriously affect the recommendation results. As a result, the user experience will be severely jeopardized while it is often regarded as one of the most crucial factors in modern recommender systems design. An illustration of the injected fake profiles and the process of shilling attack is shown in Fig. 1. Typically, the fake profile can be defined as ratings for four sets of items [1, 31]: the *selected items* are used to form the characteristics of the attack, the *filler items* that are used to camouflage the detection of the attack, the *unrated items* for which the attacker does not provide any ratings, and the *target items* that the attacker intends to manipulate. Specifically, the attacker rates the above four types of items according to a fixed rating strategy. Later on, the recommendation algorithms may be jeopardized due to the injected fake profiles in the training process.

The analyses of attacking models can help us understand the vulnerability of the recommendation algorithms, which may further enhance the robustness of recommender systems against the attacks. Over the past decade, a plethora of shilling attack models have been proposed, and researchers have successfully demonstrated their effectiveness in attacking recommendation models [22, 33]. Yet, there is a serious problem in shilling attack research. The problem is that the fake profiles are usually generated based on fixed rating strategies (e.g., random rating or follow Gaussian distribution), thus these fake profiles can be easily detected by advanced detection (a.k.a. defending) algorithms. Though researchers have investigated advanced attack models with flexible rating strategies that can adjust the ratings by themselves and generate adaptive attack profiles according to the recommendation results [7, 23, 24], these models belong to white-box attacks that need to know the recommendation algorithm of the underlying system in advance. Typically, white-box attacks have poor transferability such that the designed attack model for a particular recommendation algorithm is often ineffective for other algorithms. Therefore, gray-box shilling attacks have attracted a surge of research interests recently in which the attackers know real user ratings but have no knowledge of the underlying recommendation algorithms. To this end, in this paper, we focus on investigating gray-box shilling attacks for recommender systems.

Recently, an increasing amount of intelligent learning methods, like reinforcement learning [37] and adversarial learning [13], have been proposed. One natural assumption is that these intelligent learning algorithms could be potentially helpful in developing more advanced shilling attack models such that the generated attack profiles can obstruct the detection of defending algorithms. However, it is a nontrivial task mainly because of the following two reasons: (1) The number of ratings for each user is often limited and the rating behaviors of different users are quite diverse. Thus, the ratings cannot be simply modeled by common data distributions while the shilling attack models need to know the data distribution to create fake profiles to deceive the detection algorithms. However, to the best of our knowledge, few intelligent algorithms can accurately learn the data distributions from sparse and diverse user ratings. (2) Shilling attack models often need to know the predicted ratings of the underlying recommendation algorithms for real users. The reason is that in order to generate fake profiles to influence the recommendation algorithms as attackers desire, the predicted ratings need to be an integral part of the attack model's loss function. Without them, it is difficult to use intelligent learning algorithms to optimize the loss function. However, since gray-box shilling attacks often lack prior knowledge of the used recommendation algorithms, they cannot easily get the predicted ratings to design an effective attack model.

Fortunately, the recent popularity of generative adversarial networks (GANs) [13, 14] provides us great opportunities to tackle the above mentioned two challenges. Different from other data generation methods that must set a prior distribution in advance, GANs can learn complex data distribution by making a generator to compete with a discriminator, and the equilibrium is reached when the generator well approximates the genuine data distribution. Thus, GANs can be leveraged to tackle the first challenge by creating fake profiles to mimic the rating behaviors of real users. The core of the second challenge centers around predicting the ratings of real users without knowing the underlying recommendation algorithms. To tackle the second challenge, we can also make use of GANs to generate reliable predicted ratings based on the historical data, which has demonstrated to be successful in the literature [5, 38]. Motivated by this, in this paper, we propose a novel gray-box shilling attack model based on generative adversarial networks – named GSA-GANs. The proposed GSA-GANs consists of two essential GAN modules that serve different purposes. Specifically, the first GAN is employed to facilitate the generation of fake profiles by making them in line with real users; while the second GAN aims to predict missing ratings to enable better shilling attacks on the recommendation. As a summary, these two modules reinforce each other toward building a more powerful shilling attack model that can not only affect the recommendation result but also escape the detection of advanced defending algorithms. The main contributions of this paper are summarized as follows:

- We analyze the limitations of existing shilling attack models and evince the need for a new gray-box shilling attack model that can inject fake profiles by mimicking the rating behaviors of real users.
- We propose a novel gray-box attack model based on GANs, in which the rating strategy for the fake profile is adaptive and can be updated by the competition between the generative model and the discriminative model without knowing the underlying recommendation algorithms.
- We conduct extensive experiments to evaluate the effectiveness of the proposed shilling attack model in attacking the recommender systems and provide insights on how to defend against the proposed attacks.

The rest of this paper is organized as follows. Section 2 introduces the related work, including the general form of fake profiles in shilling attack and generative adversarial networks. In Section 3, we introduce the proposed intelligent gray-box shilling attack model GSA-GANs in detail. In Section 4, we perform experiments on three public datasets, analyze experimental results, and provide some insights on potential defending strategies. At last, we conclude the whole paper and vision future work in Section 5.

2 RELATED WORK

In this section, we will briefly review the related work about shilling attack and generative adversarial networks.

2.1 Shilling Attack

There are a wide variety of shilling attacks that require different levels of prior knowledge and serve different attack purposes [19]. For instance, primal sample attack, random attack, average attack [22], and love/hate attack [36] only require little knowledge of the underlying systems (e.g., average ratings of items), thus their attack effects are limited. To generate more powerful attacks, bandwagon attack makes use of the Zipf's law of user-item relations to select parts of items to strengthen the connections between the target items and the prevalent items [15]; and segment attack [15] takes advantage of item similarity to select parts of items to recommend target items to potential consumers [36]. Besides, relation attack [43] not only focuses on items but also relations in the generated fake profiles to attack social recommendation algorithms. To better disguise the generated fake profiles, unorganized malicious attack [33] combines the strengths of different attack strategies instead of relying on a fixed attack model. Conventional shilling attacks often have a general form (as illustrated in Fig. 2), including the ratings for target items, selected items, filler items, and unrated items. In traditional shilling attack models, filler items and selected items are used to attack and prevent detection, respectively, but in our models, we use a group of items to achieve these two goals because an item can be both a filler item and a selected item. If we separate items into two types of items rigidly, it will limit the capability of attack models to construct more flexible user profiles. We name the union of the selected items and the filler items as the *GAN items*, which are often determined together. According to the purpose of attackers, shilling attack can be divided into two different categories: push attack and nuke attack. The former one attempts to promote the ratings of some target items, while the latter aims to degrade the ratings of some other items. In fact, push attack and nuke attack have similar characteristics. Without loss of generality, we focus on the push attack in this paper.

2.2 Generative Adversarial Networks

The recent breakthrough of Generative Adversarial Networks (GANs) [13, 14, 34, 40] has impacted a number of different fields, such as computer vision [27], natural language processing [26], and recommendation [37]. GAN consists of two essential components: a generative model G and a discriminative model D. The generative model G aims at generating fake data that is similar to the real data to fool the discriminator D, while the discriminator D tries to distinguish whether the data comes from the real data or the fake data generated by G. In the training process, the generator G and D are playing a minimax game and the objective function of GAN is shown as below:

$$\min_G \max_D \mathcal{L}_{GAN} = E_{x \sim P_{data}}[\log D(x)] + E_{\hat{x} \sim P_G}[\log(1 - D(\hat{x}))], \tag{1}$$

where x is the data from real data distribution P_{data}, $\widehat{x}$ is the fake data from the data distribution P_G of generative model G, and $D(x)$ denotes the estimated probability that the discriminator takes x as real data. As GAN is very difficult to train in practice, i.e., G or D can hardly converge in the training process, WGAN [14] was proposed to make use of Wasserstein distance instead of Kullback-Leibler divergence adopted in the original GAN paper to measure the difference between the generated data distribution and the real one.

GANs have also been widely applied in the attack domain. Christakopoulou et al. [7] propose an adversarial attack model based on GANs to generate adversarial user profiles targeting subsets of users or items, or generally the top-K recommendation quality on an oblivious recommender. Furthermore, GANs can also be used to solve the recommendation tasks. IRGAN [37] uses GANs to fuse two schools of thinking – the generative retrieval focusing on predicting relevant documents given a query and the discriminative retrieval focusing on predicting relevancy of a given query-document pair to fulfill the recommendation task. DASO [9] applies the minimax game to dynamically guide the informative negative sampling process to contribute to a GAN based social recommendation model. GAN-HBNR [4] uses GANs to integrate the heterogeneous bibliographic network structure and vertex content information into an unified framework for citation recommendation. UGAN [40]

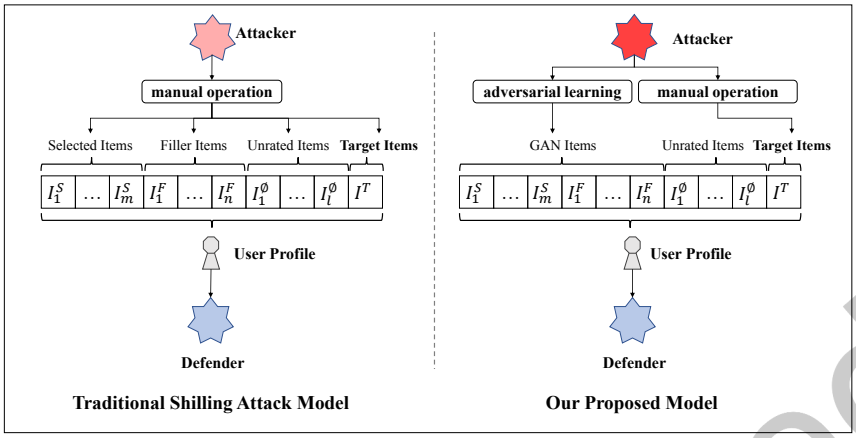

Fig. 2. A illustration about the difference between traditional shilling attack model and our proposed attack model. In traditional shilling attacks, defenders can explore potential unchangeable patterns because of the fixed item selection strategies of attackers. While in our proposed attack models, attackers combine filler items and selected items into GAN items, which are determined by adversarial training.

uses GANs' ability of learning data similarity to generate reliable ratings for user profiles to alleviate the data sparsity issue. CFGAN [6] combines collaborative filtering and GANs based on well-designed regularization items to generate information of ratings, and RAGAN [5] uses a similar method for data enhancement.

3 THE PROPOSED FRAMEWORK: GSA-GANS

In this section, we propose an intelligent gray-box shilling attack framework GSA-GANs. As mentioned before, traditional shilling attack models create fake profiles by providing ratings for four sets of items, including target items, selected items, filler items, and unrated items. However, these models adopt a predefined item selection strategy for the selected items I_s or the filler items I_F, thus their generated fake profiles can be easily detected by defenders. Different from previous research efforts, in this work, we develop a sophisticated learning algorithm to sample selected items and filler items (these two types of items are also called GAN items), and provide ratings for them. The difference between our model and existing models is shown in Fig. 2.

As can be observed in Fig. 3, there are three essential components of the proposed shilling attack model GSA-GANs: a generator G, a similarity discriminator D_S and a attack discriminator D_A. The relationships of the components of Fig. 3 are shown in Table 1.

The **generator** G tries to generate fake profiles to achieve two goals: the first goal is to generate fake user profiles that can approximate the genuine data distribution as much as possible; and the second goal is to ensure that the generated fake profiles can adversely influence the recommendation results. The **similarity discriminator** D_S tries to distinguish the generated fake profiles from real user profiles. Additionally, the **attack discriminator** D_A is used to evaluate the utility of the generated fake profiles – whether they can effectively manipulate the recommendation.

The training process of the proposed GSA-GANs can be divided into two main parts. The first part is to improve the similarity between generated fake profiles and real user profiles, and the second part is to enable better shilling attacks on recommendation.

The first part is to train D_S and G. Both the generated fake profiles and real user profiles are put into D_S, then D_S calculates the probability of that the fake profile is taken as real profile, and returns the results to the

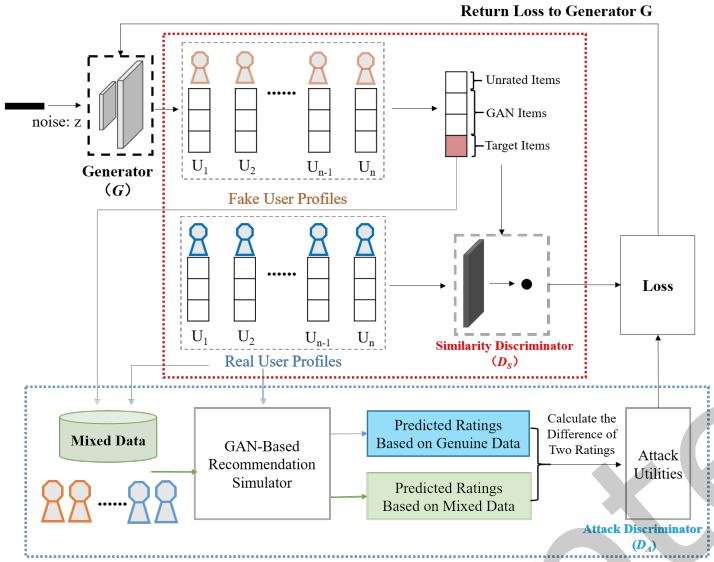

Fig. 3. An illustration of the proposed gray-box shilling attack framework GSA-GANs.

Table 1. The relationships of the components in Fig.3.

Component	Function	Part	
		Part 1	Part 2
		G and D_S train together to make fake user profiles similar to genuine ones	G and D_A train together to make fake user profiles have attack effectiveness
Generator G	Generate fake user profiles	✓	✓
Discriminator D_S	Ensure the generated fake profiles can approximate the genuine data distribution	✓	
Discriminator D_A	Ensure the generated fake profiles can influence the recommendation results		✓

generator G. We train D_S with the generator G until D_S cannot discriminate the generated fake profiles from the real profiles, thereby fake profiles will have high similarity to the genuine ones and cannot be easily detected by defenders. This is the first GAN that is employed to facilitate the generation of fake profiles by making them inline with real user. More details of this part will be elaborated in Section 3.1.

The second part is to train D_A and G, which is used to evaluate the utility of the generated attacks. The generator G generates fake profiles, and the target items are specified with designed ratings (e.g., the highest

rating in a 1-5 rating system). Then we inject the fake profiles into genuine training data for a mixed data, and use GANs based recommendation simulator, which is the second GAN in our framework, to predict ratings of target items based on the real user profiles and the mixed data. After that, we calculate the attack utility based on the difference of two ratings under different circumstances. Specifically, we train D_A with the generator G to maximize the attack utility. More details of the recommendation simulator and attack utility will be discussed later in Section 3.2.

3.1 Training for the Generation of Fake Profiles

We first discuss the training process for fake profile generation (the first GAN in Fig. 3 – generator G and discriminator D_S). In the beginning, we generate random noise vector z and feed it as input into the generator G. After G generates fake rating vectors and put them into D_S with real rating vectors, the discriminator D_S distinguishes whether the rating vectors are real or not and sends the feedback to the generator G. As a result, we will obtain well-trained generator G to generate fake rating vectors through the competition between G and D_S. The first GAN can be leveraged to generate fake profiles that are similar with real user profiles.

It should be noted that there is a problem to be solved in the above model. Recommender systems usually have millions of items, but normal users could only browse and rate a small amount of items. If we merely use the rated items to train G and D_S, we will lose a lot of useful information of unrated items. The unrated items can be divided into two types: items that users do not like and items that users have not rated. To consider the effects of items that users do not like, we take a negative sampling strategy for the unrated items and take the ratings of the selected unrated items as zero in the training process of G and D_S. Specifically, We randomly remove a portion of unrated items for each user and only consider the rest in training process. An example illustrates results of training with sampling zero-value items, in Fig 4.

The objective functions of G and D_S are formulated in Equation 3 and Equation 4. Our task is to minimize $\mathcal{L}_G$ to maximize the estimated probability that the discriminator D_S takes fake data as real data, making discriminator more likely to consider fake users as real ones. In contrast, we maximize $\mathcal{L}_{D_S}$ to maximize the estimated probability of real data and minimize the estimated probability of fake data, enabling that discriminator D_S can better distinguish fake users from real ones.

$$\min_G \max_D \mathcal{L}_{GAN} = \sum_u D(r_u) + \sum_u (1 - D(\hat{r}_u)), \tag{2}$$

$$\min \mathcal{L}_G = \min \sum_u \left(1 - D\left(\hat{r}_u\right)\right), \tag{3}$$

$$\max \mathcal{L}_{D_S} = \min(-\sum_u D\left(r_u\right) - \sum_u \left(1 - D\left(\hat{r}_u\right)\right)), \tag{4}$$

where r_u is a real user rating vector, and $\hat{r}_u$ is a generated user rating vector.

3.2 Training for Effective Attack

Although the fake profiles generated by the generator G in Section 3.1 can mimic the behaviors of real users, they cannot guarantee the effectiveness of the shilling attack. Thus, we use attack discriminator D_A that contains a recommendation simulator to evaluate attack utilities of the generated fake profiles meanwhile. Based on Equation 3, we further add a term $\mathcal{U}_{attack}$ into $\mathcal{L}_G$ and design the loss function as Equation 5, to ensure the generated fake profiles can lead to effective attacks.

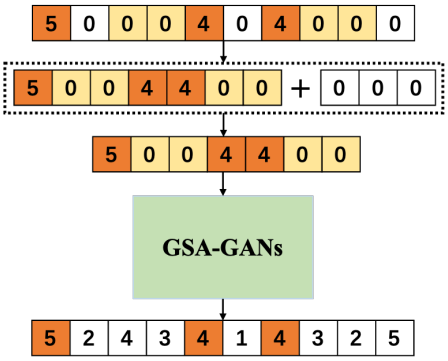

Fig. 4. The example of sampling unrated items and showing results of training. Orange ones stand for rated items. Yellow ones stand for unrated items that we sample. White ones stand for the left unrated items.

$$\min \mathcal{L}_G = \min(\sum_u (1 - D(\hat{r}_u)) + \mathcal{U}_{attack}), \tag{5}$$

where $\mathcal{U}_{attack}$ denotes the utility of attack w.r.t. the generated fake profiles. Following [24], we consider a factor when designing the utility function $\mathcal{U}_{attack}$. Attackers boost the prediction ratings of the target items, so $\mathcal{U}_{attack}$ should reflect the changes of the predicted ratings for target items before and after an attack – a.k.a. target item attack utility. We design two types of $\mathcal{U}_{attack}$ for rating prediction and ranking prediction, respectively, then get the following Equation 6 and Equation 7:

$$\min \mathcal{L}_G = \min(\sum_u (1 - D(\hat{r}_u)) + \sum_{u \in U, i \in target\ item} (Max\ rating - \tilde{P}_{ui})), \tag{6}$$

$$\min \mathcal{L}_G = \min(\sum_u (1 - D(\hat{r}_u)) - \sum_{u \in U, i \in target\ item, i_* \notin target\ item} (\tilde{P}_{ui} - \tilde{P}_{ui_*})), \tag{7}$$

where $\tilde{P}_{ui}$ denotes the predicted rating of user u for target item i after attack, and $\tilde{P}_{ui_*}$ denotes the predicted rating of user u for a non-target item i_* after attack.

Following previous work [2], we design Equation 6 for attacking rating prediction algorithms to make the predicted ratings of target items close to the maximum ratings, boosting the ratings of target items. The design of $\mathcal{U}_{attack}$ in Equation 6 could achieve the goal to affect rating prediction algorithms that focus on the variety of ratings, but it is not suitable for attacking ranking prediction algorithms because it hardly guarantees the target item in the recommendation list of normal users. Non-target items could also be close to the maximum ratings, so that target items may fail to appear in the recommendation list of normal users. Motivated by [10], we define Equation 7 to ensure that the predicted rating score of the target item is greater than those of items in the top-N recommendation list of every normal user.

However, $\tilde{P}_{ui}$ and $\tilde{P}_{ui_*}$ in Equation 6 and Equation 7 are often unknown since GSA-GANs is a gray-box attack such that attackers lack knowledge of the underlying algorithms deployed in the recommender systems. Inspired by RAGAN[5], we propose a recommendation simulator (see Fig. 5) to predict ratings for target items. The recommendation simulator has a generator G and a discriminator D. The generator generates rating vectors, and the discriminator D distinguishes whether the rating vectors are similar to real rating vectors according to the

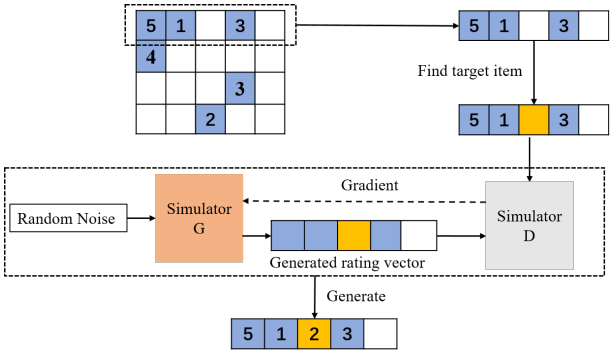

Fig. 5. The process of rating prediction for target items.

observed ratings (in the blue cells). G will optimize itself according to D's gradient, and then D optimizes itself based on the generated rating vectors from G. The G and D compete continually until arriving an equilibrium. Finally, G can generate rating vectors where the ratings in blue cells can mimic patterns of observed ratings, and those in the orange cells are for the target items.

Algorithm 1 shows the GSA-GANs for generating fake user profiles.

4 EXPERIMENTS AND ANALYSIS

In this section, we first introduce our experimental setup, including datasets, evaluation metrics, attack models, and recommendation and detection algorithms. We then evaluate the performance of the proposed GSA-GANs. Specifically, we aim to answer the following research questions.

- **RQ1:** How effective is the proposed GSA-GANs compared against other shilling attack models and how is the transferability of GSA-GANs?
- **RQ2:** How effective is GSA-GANs in escaping from shilling attack detection?
- **RQ3:** What is the performance of GSA-GANs with varied size of injected attack profiles?
- **RQ4:** What is the performance of GSA-GANs for users of different activity levels?

4.1 Experimental Setups

4.1.1 Datasets. We perform experiments on three public available datasets: MovieLens[1], Epinions[2], and FilmTrust[3]. MovieLens consists of 100,004 ratings of 681 users for 9,125 items; FilmTrust contains 1,508 users, 2,071 movies, and 35,494 ratings; and Epinions contains 664,824 ratings of 49,289 users over 139,738 movies. The detailed statistics are shown in Table 2. The ratings in these datasets are integers of in the range of [1, 5]. We randomly choose 80% of the data for training and the remaining for testing. All the experiments are repeated 20 times, and the average performance is reported.

4.1.2 Evaluation Metrics. We use the following metrics, shown in Table 3, to measure the performance of shilling attack, recommendation, and shilling attack detection, which are commonly used in the related domains [43].

Attack evaluation metrics: we use different metrics to measure attack utilities on different recommendation algorithms, including rating prediction and ranking prediction. The first metric *Prediction Shift* stands for the

[1]https://grouplens.org/datasets/movielens/
[2]https://github.com/CQU-CSE/DatasetCollection
[3]https://github.com/CQU-CSE/DatasetCollection

Algorithm 1 The proposed GSA-GANs model (all experiments performed in this paper use the default values η= 0.0001, m=100, t=10)

Input: R – real ratings. η – the learning rate. m – the batch size. t – the number of iterations of the discrimination per generator iteration.

Initial discriminator parameters θ_G, initial discriminator parameters θ_D, initial generator distribution Z.
#Get GAN items:
while not converged **do**
#D-step:
  for $i = 0, ..., t$ **do**
    Sample$\{R_u\}_{u=1,...,m} \sim R$, a batch from the real ratings.
    Sample$\{Z_u\}_{u=1,...,m} \sim Z$, a batch of prior sample.
    $J_D = \sum_u D(R_u) + \sum_u (1 - D(\theta_G(Z_u)))$
    $\theta_D \leftarrow \theta_D + \eta \nabla_w J_D$
  end for
#G-step:
  Sample$\{Z_u\}_{u=1,...,m} \sim Z$, a batch of prior samples.
  Calculate $\mathcal{U}_{attack}$
  $J_G = \sum_u (1 - D(\theta_G(Z_u))) + \mathcal{U}_{attack}$
  $\theta_G \leftarrow \theta_G - \eta \nabla_w J_G$
end while
#Set target items max rating:
Sample$\{Z_u\}_{u=1,...,m} \sim Z$, a batch of prior samples.
Get $\{X_u\}_{u=1,...,m}$, where $X_u = \theta_G(Z_u)$
for *each* $u \in U$ **do**
  $X_{u,I_{target}}$ = max rating
end for

Output: Sample set$\{X_u\}_{u=1,...,m}$

Table 2. The detailed statistics of the used datasets.

Information/dataset	MovieLens	FilmTrust	Epinions
#User	681	1,508	49,289
#Item	9,125	2,071	139,738
#Rating	100,004	35,494	664,824

difference of predicted ratings before and after attacks. The definition of *Prediction Shift* is formulated as follows:

$$\text{Prediction Shift} = \frac{\sum_{u,i} \left(\tilde{P}_{u,i} - P_{u,i}\right)}{N_{predict}}, \tag{8}$$

where $\tilde{P}_{u,i}$ indicates the predicted ratings from user u to target item i after attacks, $P_{u,i}$ indicates the predicted ratings for user u to target item i before attacks, and $N_{predict}$ indicates the number of ratings to predict (missing ratings). The second metric *Hit Ratio* stands for the probability that the target items appear in the top-K recommendation list. We set K = 10 unless otherwise specified, because this experimental setting is often used in the

Table 3. Evaluation metrics for shilling attack, recommendation, and shilling detection algorithms.

Type	Metric			
Evaluation metrics for attacks	Prediction Shift	Hit Ratio	PST	HRT
Evaluation metrics for recommendation	MAE	Precision	-	-
Evaluation metrics for shilling attack detection	Precision	Recall	F1	-

top-K recommendation task [42]. The definition of *Hit Ratio* is as follows:

$$\text{Hit Ratio} = \frac{N_{\text{target item}}}{N_{\text{topK}}}, \tag{9}$$

where $N_{\text{target item}}$ indicates the number of target items in the Top-K recommendation list, and N_{topK} indicates the total number of items in the top-K recommendation list.

Attack transferability metrics: The metrics of *Prediction Shift Transferability* (PST) and *Hit Ratio Transferability* (HRT) are used to measure whether the developed attack strategies can be transfer to different recommendation algorithms, and these two metrics are respectively defined as follows:

$$\text{PST} = \sqrt{\frac{\sum_i (PS_i)^2}{\left(N_{\text{algorithm}} - 1\right)}} / (PS_{Average}), \tag{10}$$

$$\text{HRT} = \sqrt{\frac{\sum_i (HR_i)^2}{\left(N_{\text{algorithm}} - 1\right)}} / (HR_{Average}), \tag{11}$$

where PS_i is the *Prediction Shift* of the i-th recommendation algorithm, $PS_{Average}$ is the *average Prediction Shift* of all recommendation algorithms, HR_i is the *Hit Ratio* of the i-th recommendation algorithm, $HR_{Average}$ is the *average Hit Ratio* of all recommendation algorithms, and $N_{\text{algorithm}}$ is the number of all recommendation algorithms.

Recommendation performance evaluation metrics: We use *Mean Absolute Error* (MAE) for rating prediction and *precision* for ranking evaluation to show the performance of recommendation algorithms, and they are defined as below:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} \left| \widehat{P}_{u,i} - P_{u,i} \right|, \tag{12}$$

$$\text{precision (recommendation)} = \frac{N_{topK}}{N_{all}}, \tag{13}$$

where $\widehat{P}_{u,i}$ is the predicted rating, $P_{u,i}$ is the real rating, N_{all} is the number of all predicted items, and N_{topK} is the number of predicted top-K items.

Shilling attack detection evaluation metrics: We use three metrics *precision*, *recall*, and *F1* to measure the results of shilling attack detection, which can help us determine whether the attack models can escape the detection algorithms. They are defined as below:

$$\text{precision (detection)} = \frac{\text{\# true positive}}{\text{\# true positive + \# false positive}}, \tag{14}$$

Table 4. The baseline shilling attack models, recommendation algorithms, and shilling attack detection algorithms.

Model type	Further classification	Algorithm name for short	Full name/Description
Attack models	-	RA	Random Attack
		AA	Average Attack
		BA [36]	Bandwagon Attack
		UMA [33]	Unorganized Malicious Attacks
Recommendation algorithms	Rating prediction algorithms	BasicMF	The Original Matrix Factorization Model
		SVD [21]	Singular Value Decomposition
		PMF [30]	Probabilistic Matrix Factorization
		EE [20]	Euclidean Embedding
	Ranking prediction algorithms	BPR [35]	Bayesian Personalized Ranking
		WRMF [18]	Weighted Regularized Matrix Factorization
		APR [16]	Adversarial Personalized Ranking
		NeuMF [17]	Neural Matrix Factorization
Shilling attack detection algorithms	Supervised	DegreeSAD [25]	Detection Based on Item Popularity
		BayesDetector [41]	Detection Based on Bayes Estimation
	Unsupervised	PCA [29]	Principal Component Analysis
		FAP [45]	Fraudulent Action Propagation

$$\text{recall} = \frac{\#\text{ true positive}}{\#\text{ true positive} + \#\text{ false negative}}, \tag{15}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{16}$$

where # true positive denotes the number of correctly detected attack profiles, # false positive denotes the number of misclassified real profiles, and # false negative denotes the number of attack profiles that are misclassified.

4.1.3 Baselines. In the experiments, we adopt 4 shilling attack models as baselines, employ 8 recommendation algorithms to evaluate the attack effectiveness, and take advantage of 4 shilling attack detection algorithms to evaluate assess whether the attack models can conceal the detection algorithms. The detailed information of baseline shilling attack models, recommendation and shilling attack detection algorithms are shown in Table 4.
Baseline Attack Models: We compare the proposed attack model GSA-GANs with four other shilling attack models [2]: (1) *Random Attack* (RA): Filler items are chosen randomly and rated based on a normal distribution. (2) *Average Attack* (AA): Filler items are chosen randomly and those items are assigned with average ratings. (3) *Bandwagon Attack* (BA): Filler items are chosen randomly. Except for the filler items, attackers select and rate the popular items as selected items. (4) *Unorganized Malicious Attacks* (UMA): Combine some attack models to achieve UMA. Specifically, in our experiments, we use RA, AA, and BA to generate fake user profiles respectively and inject into system together. Unless otherwise specified, we follow the previous work [22, 43]to set a fixed attack

size (the total number of fake profiles to the total number of user profiles) as 3%. The filler size of GSA-GANs is not manually specified, but traditional attack model is. In the experiments, we set filler sizes of traditional shilling models to 1.3% in FilmTrust, 1.8% in MovieLens, and 1.1% in Epinions, which are average filler sizes of GSA-GANs in those datasets. More details and analysis of filler size are shown in Section 4.8.

Recommendation Algorithms: We select several representative recommendation algorithms for rating prediction and ranking prediction to demonstrate the effectiveness and transferability of the proposed GSA-GANs. In particular, we choose BasicMF, SVD [21], PMF [30], and EE [20] for rating prediction; BPR [35], WRMF [18], APR [16], and NeuMF [17] for ranking prediction.

Shilling Attack Detection Algorithms: We adopt four shilling attack detection algorithms to show how GSA-GANs can camouflage the fake profiles from advanced detection algorithms. Here, we choose two popular supervised shilling detection algorithms (DegreeSAD [25] and BayesDetector [41]), and two widely used unsupervised methods (PCA [29] and FAP [45]).

4.1.4 Experimental Settings. To compare GSA-GANs with baseline attack models, we inject fake profiles generated by baselines and GSA-GANs into the original data. It should be noted that, to make it consistent with other research works on shilling attack [43], all the original users in the data are assumed as the real users. The experimental processes are summarized as follows:

- We choose target items, filler items, and selected items for the baselines AA, RA, BA, and UMA. There is no need to choose filler items and selected items for GSA-GANs because they are generated by adversarial training.
- We generate fake profiles and they are injected into the original data.
- We then conduct the recommendation and shilling detection algorithms on the original data and new injected data respectively, and perform evaluation with all the introduced metrics in Section 4.1.2.

Following [22, 36], the target items set consists of 10 different items, which are randomly chosen without any human intervention. The average ratings of these target items are lower than 3, since it is not practical to push prediction results of an item that already has high rating scores.

4.2 Effectiveness and Transferability of Shilling Attacks (RQ1)

In this subsection, we attempt to answer the first research question. We first assess the effectiveness of the developed attack model GSA-GANs by comparing it with several baseline attack models. The comparison results in Tables 5, 6, and 7 demonstrate (1) the effectiveness of the proposed shilling attack models; (2) the good transferability of the proposed method on different recommendation algorithms. For instance, on MovieLens dataset, GSA-GANs improves the Prediction Shift upon the best attack model over 0.185 for SVD, and improves Hit Ratio upon the best baseline method around 170% for WRMF. In terms of transferability, the injected fake profiles by GSA-GANs can well jeopardize various recommendation algorithms. The compared attack models show good performance of transferability to some extent, but our model is better.

4.3 Camouflage of Generated Fake Profiles (RQ2)

In this subsection, we aim to answer the second research question by investigating whether the injected fake profiles by GSA-GANs can escape the detection of advanced shilling attack detection models. As can be seen from Table 8, shilling detection results demonstrate that the fake profiles injected by our method is much harder to be detected than others in most cases. As a summary, GSA-GANs has powerful camouflage against both unsupervised and supervised shilling attack detection methods.

To show the effectiveness of attacks under detection, we use four detection methods mentioned in Section 4.1.3 to detect fake users and perform experiments on the fake-user removed, new datasets. We conduct experiments on MovieLens and choose BasicMF (a rating prediction algorithm) and APR (a ranking prediction algorithm)

Table 5. Comparison of shilling attack models on FilmTrust.

	Rating Prediction Algorithm (Prediction Shift)				PST	Ranking Prediction Algorithm (Hit Ratio)				HRT
	BasicMF	SVD	PMF	EE		BPR	WRMF	APR	NeuMF	
AA	0.049	0.112	0.064	0.073	0.109	1.20%	1.05%	1.14%	1.15%	1.52%
RA	0.152	0.077	0.079	0.020	0.145	1.19%	1.01%	0.92%	1.05%	1.40%
BA	0.135	0.125	0.076	0.088	0.149	1.30%	0.84%	0.97%	1.02%	1.41%
UMA	0.115	0.132	0.076	0.107	0.149	1.20%	1.11%	1.17%	1.12%	1.53%
GSA-GANs	**0.224**	**0.192**	**0.154**	**0.190**	**0.257**	**1.82%**	**1.46%**	**1.63%**	**1.25%**	**2.09%**

Table 6. Comparison of shilling attack models on MovieLens.

	Rating Prediction Algorithm (Prediction Shift)				PST	Ranking Prediction Algorithm (Hit Ratio)				HRT
	BasicMF	SVD	PMF	EE		BPR	WRMF	APR	NeuMF	
AA	**0.150**	0.285	0.243	0.112	0.296	3.47%	2.58%	2.84%	3.40%	4.16%
RA	0.127	0.292	0.156	0.138	0.270	3.05%	3.29%	**3.86%**	**4.25%**	4.90%
BA	0.091	0.292	0.148	0.107	0.265	2.41%	2.70%	3.19%	2.02%	3.17%
UMA	0.103	0.296	0.231	0.120	0.295	2.94%	2.42%	3.40%	3.01%	3.89%
GSA-GANs	0.138	**0.481**	**0.297**	**0.237**	**0.457**	**3.95%**	**5.71%**	3.65%	3.80%	**5.44%**

Table 7. Comparison of shilling attack models on Epinions.

	Rating Prediction Algorithm (Prediction Shift)				PST	Ranking Prediction Algorithm (Hit Ratio)				HRT
	BasicMF	SVD	PMF	EE		BPR	WRMF	APR	NeuMF	
AA	0.024	0.125	0.192	0.055	0.189	0.32%	**0.40%**	0.19%	0.13%	0.40%
RA	0.151	0.036	0.153	0.146	0.189	0.47%	0.18%	0.25%	0.24%	0.43%
BA	0.043	0.161	0.236	0.098	0.230	0.36%	0.08%	0.15%	0.27%	0.36%
UMA	0.123	0.246	0.140	0.084	0.230	0.25%	0.17%	0.20%	0.15%	0.26%
GSA-GANs	**0.175**	**0.277**	**0.345**	**0.151**	**0.350**	**0.59%**	0.25%	**0.35%**	**0.34%**	**0.57%**

because of their robustness shown in Table 6. The Prediction Shift and Hit Ratio on the new datasets are shown in Fig. 6. GSA-GANs has the smallest drop compared with other attack models, especially in rating prediction scenario, where attack effectiveness drops by around 10% on average, while those of other attack models drop by at least 36%. These results show that our attack model has strong attack effectiveness against recommendation and camouflage ability against detection.

4.4 The Influence of Attack Size (RQ3)

To answer the third research question, we change the attack size (the ratio of fake profiles to total number of profiles) from 3% to 18%. The experimental results are shown in Fig. 7. First, our proposed method is quite effective in promoting target items. For example, the Prediction Shift values for SVD are larger than 0.48 all the time. Second, the attack model becomes more effective as attack size increases, and the trend slows down while the attack size is large enough in most cases. For instance, the Prediction Shift for PMF shows steep curve when the attack size is from 3% to 12%, while it turns quite smooth after the attack size is larger than 12%.

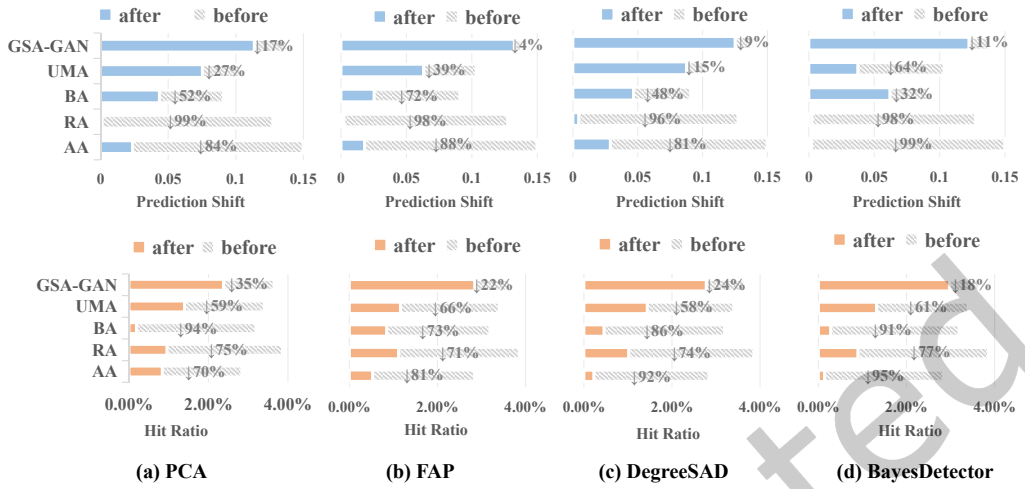

Fig. 6. The Prediction Shift and Hit Ratio of GSA-GANs on MovieLens after detection.

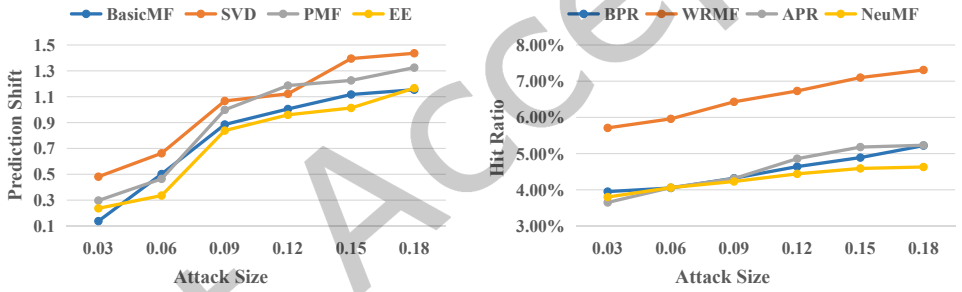

Fig. 7. The Prediction Shift and Hit Ratio of GSA-GANs on MovieLens with varied attack sizes.

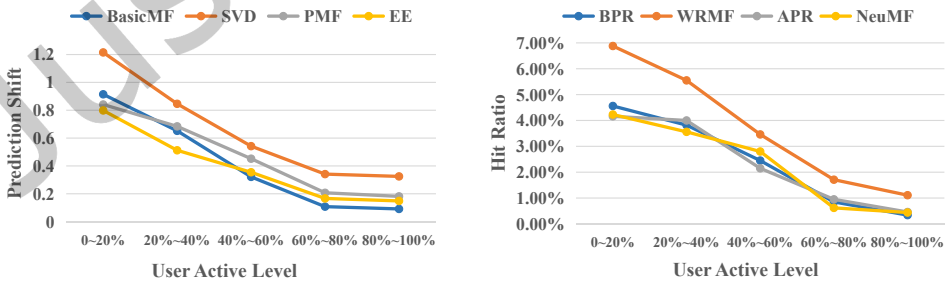

Fig. 8. Users analysis under GSA-GANs. (0 ~ 20%) stands for cold-start user group and (80% ~ 100%) stands for active user group.

Table 8. Shilling attack detection results on MovieLens w.r.t. different attack models.

		Unsupervised		Supervised	
		PCA	FAP	Degree -SAD	Bayes- Detector
AA	Precision	0.9427	0.9742	0.9429	0.9570
	Recall	0.8725	0.5095	0.9710	0.9783
	F1	0.9062	0.6619	0.9567	0.9675
RA	Precision	0.9694	0.9756	0.9427	0.9787
	Recall	0.8972	0.5110	0.9638	0.9783
	F1	0.9319	0.6612	0.9531	0.9719
BA	Precision	0.9568	0.9743	0.9572	0.9645
	Recall	0.9710	0.5125	0.9677	**0.9493**
	F1	0.9639	0.6610	0.9625	0.9559
UMA	Precision	**0.8959**	0.9727	0.9439	0.9429
	Recall	0.8942	0.5095	0.9716	0.9710
	F1	0.8950	0.6604	0.9576	0.9567
GSA-GANs	Precision	0.9036	**0.9402**	**0.9328**	**0.9167**
	Recall	**0.8525**	**0.5094**	**0.9467**	0.9574
	F1	**0.8773**	**0.6532**	**0.9400**	**0.9366**

4.5 The Influences of GSA-GANs for Different Users (RQ4)

The last question is to explore the influence of GSA-GANs for different types of users. We separate users into 5 different groups based on their active levels (from low to high), and analyze how different types of users are affected by GSA-GANs. For example, (0 ~ 20%) refers to cold-start user groups and (80% ~ 100%) denotes active user groups. We can find that cold-start users are vulnerable to GSA-GANs but active users are not, as can be seen from Fig. 8. We hold the opinion that cold-start users that lacks historical ratings are affected by attacks most under different recommendation algorithms. Naturally, we should pay more attention to these cold start users when developing defending strategies.

4.6 The Rating Distribution of Generated Users Profiles and Genuine Data

GSA-GANs tries to make a trade-off between mimicking the genuine data and achieving effective attacks. We conduct experiments to demonstrate that the rating distribution of generated user profiles do not change much even considering the effectiveness of attacks. As shown in Fig. 9, the rating distributions of three datasets do not change a lot from the genuine data to the generated data. Most of the ratings are 4 and 5, and rating differences are quite small. It demonstrates that the generated data well mimics the properties of genuine data.

4.7 The Effectiveness of Attack When Partial Ratings is Available

The above experimental results are based on the assumption that attackers know all genuine ratings of users, and many research efforts [24, 32] are based on this assumption. However, it may be challenging to get all genuine ratings in real systems. To further evaluate GSA-GANs, we keep partial ratings (20%, 40%, 60%, and 80% of all ratings for one user) to conduct experiments on MovieLens. As shown in Fig.10, we can find that the attack effectiveness increases as the ratio of known ratings increases. To sum up, the ratio of known ratings is highly correlated with the attack effectiveness of GSA-GANs.

4.8 The Attack Effectiveness of Different Filler Size

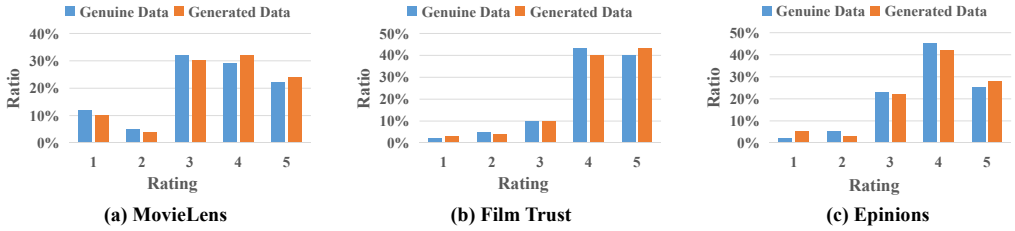

Fig. 9. The rating distribution of generated user profiles and genuine data.

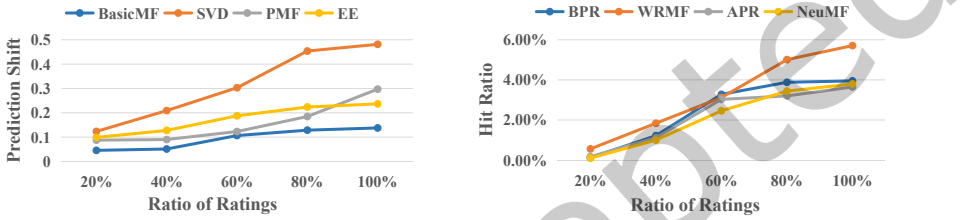

Fig. 10. The Prediction Shift and Hit Ratio on MovieLens w.r.t. different ratio of known ratings.

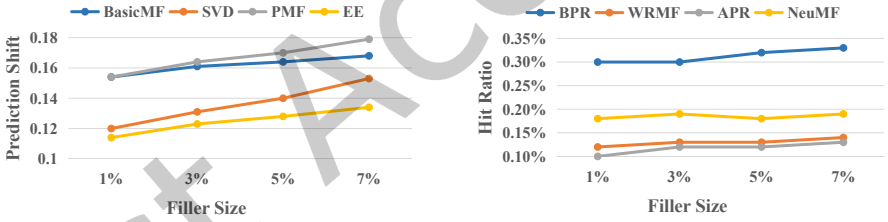

Fig. 11. The Prediction Shift and Hit Ratio using 3% attack size on MovieLens w.r.t. different filler sizes.

As mentioned above, we follow studies [22, 43] to specify the attack size as 3% to ensure a fair comparison. To further evaluate GSA-GANs, we conduct more experiments on 3% attack size with several filler sizes, i. e. 1%, 3%, 5%, and 7%, respectively. As shown in Fig. 11, the attack effectiveness still keeps stable. Besides, we find that for almost all the algorithms, the attack effectiveness does not change much w.r.t. different filler sizes. Based on these findings, we loosen the filler size of GSA-GANs, rather than fixing it, and set the filler size of traditional shilling models as 1.3% in FilmTrust, 1.8% in MovieLens, and 1.1% in Epinions, which are the average filler sizes of GSA-GANs in such datasets.

4.9 GSA-GANs Defensive Strategies

Based on our experimental results and empirical findings, here we provide some insights in developing defensive strategies against the proposed GSA-GANs. Although the proposed GSA-GANs is an attack model for attacker, the

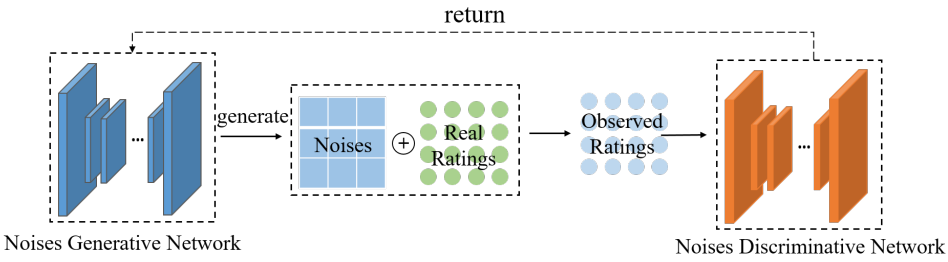

Fig. 12. A generative framework of defensive noises.

analysis of the attack model can help us better understand the vulnerability of the recommendation algorithms, and the analysis of the defensive strategies can further enhance the defense of recommender systems against the attacks.

Add noises to the ratings. GSA-GANs heavily relies on real data to develop the attack strategy. In practice, attackers usually crawl ratings from an open recommender system platform. In this case, defenders could add some noises into ratings to fool attackers. But on the other hand, it is also easy for attackers to remove a fixed noise distribution from the original data distribution of ratings. Thus, we suggest that researchers can leverage adversarial learning to train defensive noises, and a possible framework is shown in Fig. 12. In short, the generative network adds noises into the ratings to form observed samples, and then such samples are fed into a discriminative network, which returns results to train the generative network. Finally, we repeat the process until the generated noises can successfully conceal attackers.

Leverage the power of auxiliary information. Auxiliary information, such as user comments and user relations, not only can be used to improve the recommendation algorithms but also can enhance the robustness of systems, because the requirements of more auxiliary information will raise cost of attacks when users have behaviors, and we can detect fake user profiles based on auxiliary information.

Focus on cold-start users. Cold-start users are susceptible to GSA-GANs, as demonstrated in Section 4.6. A possible method to mitigate this problem is to use adversarial learning based data augmentation that can improve the robustness of recommendation under attacks because it can generate reliable ratings to mitigate the data sparsity issue.

5 CONCLUSION AND FUTURE WORK

In this paper, we employ adversarial learning to propose a gray-box shilling attack model GSA-GANs, which contains two GAN modules. The first GAN aims to keep the distribution of fake profiles close to that of real profiles; while the second one is to predict missing ratings to better evaluate the utilities of attack models. The two GAN modules cooperatively work together to build a powerful shilling attack model that can better affect recommendation results and escape the detection of sophisticated shilling attack detection models. We also perform extensive empirical evaluations on real-world datasets, and the results demonstrate that GSA-GANs is very effective in terms of attack utilities, attack transferability, and camouflage. Additionally, we also provide some feasible defensive strategies against GSA-GANs. For future work, we will explore more powerful attack models based on GANs by considering more auxiliary information, e.g., social relationships or sentiment of users, because lots of researchers have been trying to take the auxiliary information into consideration in recommendation algorithms. Besides, we will study detection models based on adversarial learning to detect fake user profiles. Moreover, we plan to develop noise generation models that can learn how to add useful noises to the genuine

data, in order to prevent the attackers from learning genuine data distribution while improve the robustness of the recommendation algorithms.

ACKNOWLEDGMENTS

Min Gao is supported by the National Natural Science Foundation of China (62176028) and the Overseas Returnees Innovation and Entrepreneurship Support Program of Chongqing (cx2020097). Zongwei Wang is supported by the Natural Science Foundation of Chongqing (cstc2020jcyj-msxmX0690).

REFERENCES

[1] Runa Bhaumik, Chad Williams, Bamshad Mobasher, and Robin Burke. 2006. Securing collaborative filtering against malicious attacks through anomaly detection. In *Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization (ITWP'06), Boston*, Vol. 6. 10.

[2] Wilander Bhebe and Okuthe P Kogeda. 2015. Shilling Attack Detection in Collaborative Recommender Systems Using a Meta Learning Strategy. In *2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*. 56–61.

[3] HongYun Cai and Fuzhi Zhang. 2019. Detecting Shilling Attacks in Recommender Systems based on Analysis of User Rating Behavior. *Knowledge-Based Systems* (2019), 22–43.

[4] Xiaoyan Cai, Junwei Han, and Libin Yang. 2018. Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 5747–5754.

[5] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jaeho Choi. 2019. Rating Augmentation with Generative Adversarial Networks towards Accurate Collaborative Filtering. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. 2616–2622.

[6] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. CFGAN: A Generic Collaborative Filtering Framework based on Generative Adversarial Networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. 137–146.

[7] Konstantina Christakopoulou and Arindam Banerjee. 2019. Adversarial attacks on an oblivious recommender. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*. 322–330.

[8] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2019. Visual Interfaces for Recommendation Systems: Finding Similar and Dissimilar Peers. *ACM TIST* (2019), 9:1–9:23.

[9] Wenqi Fan, Tyler Derr, Yao Ma, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep Adversarial Social Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. 1351–1357.

[10] Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. 2020. Influence Function based Data Poisoning Attacks to Top-N Recommender Systems. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 3019–3025.

[11] 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, ACSAC 2018, San Juan, PR, USA, December 03-07, 2018*. 381–392.

[12] Min Gao, Zhongfu Wu, and Feng Jiang. 2011. Userrank for Item-based Collaborative Filtering Recommendation. *Inf. Process. Lett.* (2011), 440–446.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2672–2680.

[14] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 5767–5777.

[15] Ihsan Gunes, Cihan Kaleli, Alper Bilge, and Huseyin Polat. 2014. Shilling Attacks against Recommender Systems: A Comprehensive Survey. *Artif. Intell. Rev.* (2014), 767–799.

[16] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 355–364.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*. 173–182.

[18] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. 263–272.
[19] Neil Hurley, Zunping Cheng, and Mi Zhang. 2009. Statistical Attack Detection. In *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*. 149–156.
[20] Mohammad Khoshneshin and W. Nick Street. 2010. Collaborative Filtering via Euclidean Embedding. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*. 87–94.
[21] Yehuda Koren. 2009. Collaborative Filtering with Temporal Dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*. 447–456.
[22] 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, WWW 2004, New York, NY, USA, May 17-20, 2004*. 393–402.
[23] Xian Yeow Lee, Aaron J. Havens, Girish Chowdhary, and Soumik Sarkar. 2019. Learning to Cope with Adversarial Attacks. *CoRR* (2019).
[24] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. 2016. Data Poisoning Attacks on Factorization-Based Collaborative Filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 1885–1893.
[25] Wentao Li, Min Gao, Hua Li, Jun Zeng, Qingyu Xiong, and Sachio Hirokawa. 2016. Shilling Attack Detection in Recommender Systems via Selecting Patterns Analysis. *IEICE Transactions* (2016), 2600–2611.
[26] Zilong Lin, Yong Shi, and Zhi Xue. 2018. IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection. *CoRR* (2018).
[27] Yifan Liu, Zengchang Qin, Tao Wan, and Zhenbo Luo. 2018. Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Wasserstein Generative Adversarial Networks. *Neurocomputing* (2018), 78–87.
[28] Bhaskar Mehta, Thomas Hofmann, and Peter Fankhauser. 2007. Lies and Propaganda: Detecting Spam Users in Collaborative Filtering. In *Proceedings of the 12th International Conference on Intelligent User Interfaces, IUI 2007, Honolulu, Hawaii, USA, January 28-31, 2007*. 14–21.
[29] Bhaskar Mehta and Wolfgang Nejdl. 2009. Unsupervised Strategies for Shilling Detection and Robust Collaborative Filtering. *User Model. User-Adapt. Interact.* (2009), 65–97.
[30] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic Matrix Factorization. In *Advances in neural information processing systems*. 1257–1264.
[31] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Jeff J Sandvig. 2007. Attacks and remedies in collaborative recommendation. *IEEE Intelligent Systems* 22, 3 (2007), 56–63.
[32] 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.
[33] Ming Pang, Wei Gao, Min Tao, and Zhi-Hua Zhou. 2018. Unorganized Malicious Attacks Detection. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. 6976–6985.
[34] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*. 452–461.
[36] Ahmet Murat Turk and Alper Bilge. 2019. Robustness Analysis of Multi-criteria Collaborative Filtering Algorithms against Shilling Attacks. *Expert Syst. Appl.* (2019), 386–402.
[37] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. 515–524.
[38] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 515–524.
[39] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. 2018. Exploring Hierarchical Structures for Recommender Systems. *IEEE Trans. Knowl. Data Eng.* (2018), 1022–1035.
[40] Zongwei Wang, Min Gao, Xinyi Wang, Junliang Yu, Junhao Wen, and Qingyu Xiong. 2019. A Minimax Game for Generative and Discriminative Sample Models for Recommendation. In *Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part II*. 420–431.
[41] Fan Yang, Min Gao, Junliang Yu, Yuqi Song, and Xinyi Wang. 2018. Detection of Shilling Attack Based on Bayesian Model and User Embedding. In *IEEE 30th International Conference on Tools with Artificial Intelligence, ICTAI 2018, 5-7 November 2018, Volos, Greece*.

639–646.

[42] Junliang Yu, Min Gao, Jundong Li, Hongzhi Yin, and Huan Liu. 2018. Adaptive Implicit Friends Identification over Heterogeneous Network for Social Recommendation. In *Proceedings of the 27th ACM International Conference on Informationand Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. 357–366.

[43] Junliang Yu, Min Gao, Wenge Rong, Wentao Li, Qingyu Xiong, and Junhao Wen. 2017. Hybrid Attacks on Model-based Social Recommender Systems. *Physica A: Statistical Mechanics and its Applications* (2017), 171–181.

[44] Weina Zhang, Xingming Zhang, Haoxiang Wang, and Dongpei Chen. 2019. A Deep Variational Matrix Factorization Method for Recommendation on Large Scale Sparse Dataset. *Neurocomputing* (2019), 206–218.

[45] Yongfeng Zhang, Yunzhi Tan, Min Zhang, Yiqun Liu, Tat-Seng Chua, and Shaoping Ma. 2015. Catch the Black Sheep: Unified Framework for Shilling Attack Detection Based on Fraudulent Action Propagation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. 2408–2414.