Trust-GRS: A Trustworthy Training Framework for Graph Neural Network Based Recommender Systems Against Shilling Attacks

Lingyu Mu[1,2], **Zhengxiao Liu**[1,2*], **Zhitong Zhu**[1,2], **Zheng Lin**[1,2*]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{mulingyu, liuzhengxiao, zhuzhitong, linzheng}@iie.ac.cn

Abstract

Graph neural network (GNN) based recommender systems have been widely used in diverse service platforms as they can more effectively capture users' interests. Nevertheless, recent investigations have revealed that the neighborhood aggregation and contrastive learning mechanisms render GNN-based recommender systems more vulnerable to fake profile injection attacks, i.e., *shilling attacks*. Despite numerous defenses against shilling attacks having emerged, these approaches still face certain challenges, such as the demand for prior knowledge and the difficulty in defending against multiple attacks. Therefore, this paper proposes a two-stage trustworthy GNN-based recommender systems training framework (Trust-GRS), which models the probability of data being fake in a zero-knowledge scenario and establishes a trustworthy neighborhood aggregation and contrastive learning mechanisms. Through extensive experiments on multiple benchmark datasets against 12 state-of-the-art shilling attacks, we demonstrate that Trust-GRS substantially mitigates the influence of fake data in all attacks, up to 100%, while preserving the original recommendation performance. Benefiting from the absence of a requirement for prior knowledge, Trust-GRS holds significant application value for real-world recommendation platforms.

Code — https://github.com/IIE-MLY/Trsut-GRS

Introduction

Recommender systems (RSs) as one of the solutions to alleviate the problem of information overload are widely applied in various service platforms (Wang et al. 2021), such as e-commerce, social media, and search engines. In recent years, GNN-based RSs (He et al. 2020; Wu et al. 2021b) represent data as user-item graphs, effectively capturing users' interests and item characteristics by explicitly encoding higher-order neighbor information from interaction behavior. Consequently, GNN-based RSs have emerged as one of the mainstream recommendation architectures.

However, GNN-based RSs are vulnerable to attacks from malicious profiles or fake data injection, i.e., *shilling attacks,* which is accomplished by injecting a set of fake profiles to the training set of the RSs to maliciously alter the ranking of target items. Recent studies (You et al. 2023; Wang et al. 2023; Zhang et al. 2023) have shown that neighborhood aggregation (NA) (Kipf and Welling 2016) and contrastive learning (CL) (Hadsell, Chopra, and LeCun 2006) exacerbate the impact of fake users, making GNN-based RSs more susceptible to shilling attacks compared to traditional RSs. Existing defenses (Yu et al. 2021) against shilling attacks overlook this aspect and the majority of these methods are only applicable in white-box scenarios. Therefore, we consider developing a black-box universal training framework for GNN-based RSs. To achieve this goal, we aim to address two research problems: **(i)** In the black-box scenarios, how to precisely identify fake data? **(ii)** When not all malicious samples can be detected, how to mitigate the impact of fake data while preserving benefits of GNN?

To address the aforementioned problems, we investigated the inherent differences between real and fake data in the black-box scenarios based on model training dynamics approaches (Saxe, McClelland, and Ganguli 2013; Li et al. 2021) and found that the average training loss of fake data rapidly converges to 0 than real data. Based on this observation, the users with the lowest loss during the early stages of training can form a subset, which consists mainly of fake users. Note not all the losses of fake users are lower than real users, this subset cannot accurately include all fake users. However, since the number of fake items is small and fake users often share fixed common targets (Si and Li 2020), it is possible to identify target items via this subset and subsequently recognize all potential malicious user files in the training set. Building on the above findings, we propose a two-stage training framework for the GNN-based RSs, namely Trust-GRS, which consists of two stages: (i) fake data identification stage and (ii) trustworthy GNN-based RSs training stage. In the first stage, we take a certain percentage of users with the lowest loss as a subset in the early stages of training. In this subset, fake users in the majority frequently interact with target items due to their structural similarity, resulting in interaction frequencies of fake items far exceeding those of real items. Since PageRank (Page et al. 1999) can compute rankings based on node interaction counts, we devise an algorithm named Shilling-Rank. Shilling-Rank can map the probability of items being the target of shilling attacks and further calculate the probability of users being fake in the entire training set. In the sec-

*Corresponding authors: Zhengxiao Liu, Zheng Lin.

ond stage, we design trustworthy NA and CL founded on the probabilities calculated by Shilling-Rank of users and items being fake. Specifically, in the trustworthy NA, each node acquires the influence of its neighboring nodes weighted based on the probabilities. For the trustworthy CL, we design a new distance formula that modifies the spacing between data points in the representation space based on the probabilities. Through trustworthy NA and CL, Trust-GRS can eliminate the impact of fake data while preserving the advantages of both mechanisms.

We conduct extensive assessments of Trust-GRS, evaluating its defense performance against 12 state-of-the-art (SOTA) attack approaches on three datasets, encompassing neural network-based attacks such as AUSH (Lin et al. 2020), CLeaR (Wang et al. 2023), and gradient-based attacks such as A_ra (Rong, He, and Chen 2022), FedRec (Rong et al. 2022). Trust-GRS could minimize the recommendation metrics of the malicious targeted items to 0 against all attacks in the black-box scenarios, which is significantly superior to the results of defense baselines such as PCA (Mehta and Nejdl 2009), FAP (Zhang et al. 2015) in the white-box scenarios.

Our contributions can be summarized as follows:

- This paper proposes a novel method named Shilling-Rank, which uses model training dynamics to identify poisoned data in shilling attacks. This approach does not require any prior knowledge and distinguishes between real and fake data merely by exploiting inherent attribute differences during training.
- This paper introduces a robust GNN-based RSs learning process. By leveraging fake data probability distribution, we refine the NA and CL mechanisms. The new trust module mitigates the impact of fake data while maintaining the original RSs performance.
- Our framework can be integrated into any type of GNN-based RSs. We conducted extensive experiments, demonstrating that this training framework completely eliminates the impact of fake data against 12 SOTA attacks.

Related Work

Shilling attacks. Existing shilling attacks can be classified into three categories based on their generation methods: heuristic attacks, neural-network-based attacks, and gradient-based attacks. Since early heuristic attack profiles (Lam and Riedl 2004) were simple and easy to detect, recent research has focused on the latter two attack methods. *(i) Neural-Network-based attacks* utilize neural networks to automatically learn and design patterns for fake users. P_Rec (Song et al. 2020) uses reinforcement learning to design fake data. GOAT (Wu et al. 2021a) and AUSH use generative adversarial networks to train a set of attack files. *(ii) Gradient-based attacks* formulate the problem as a bi-level optimization process and uses approximate gradients to update the original data (Christakopoulou and Banerjee 2019).

As one of the most widely used architectures, GNN-based RSs (He et al. 2020) are more vulnerable to shilling attacks (You et al. 2023) due to the neighborhood aggregation which propagates target items to multi-hop neighbors. Furthermore, CL for GNN-based RSs (Wu et al. 2021b; Yu et al. 2022) increases the separation of data, making it easier for fake data to infiltrate users' Top-K lists (Wang et al. 2023).

Therefore, we target GNN-based RSs with CL for defense, against 12 SOTA attacks from two categories.

Defenses against shilling attacks. Existing defenses against shilling attacks can be classified into three categories based on whether there is supervision or not: supervised, unsupervised, and semi-supervised. *(i) Supervised methods* rely on the labeled poisoned dataset to train a classifier for identifying fake data(Zhang et al. 2020). AntiFU (You et al. 2023) used the Laplacian matrix of the users to train a probability generator. Burke et al. (Burke et al. 2006) trained the classifier using specific attributes of the rating matrix. *(ii) Unsupervised methods* such as clustering (Bhaumik, Mobasher, and Burke 2011) and data mining detect fake data by exploiting differences in the statistical properties. PCA-Based (Mehta and Nejdl 2009) used principal component analysis to cluster malicious users. *(iii) Semi-supervised methods* utilize both supervised and unsupervised poisoned datasets for hybrid detection, which requires the highest level of knowledge. For example, HySAD (Wu et al. 2012) trained a Naïve Bayes classifier using labeled data and predicts the posterior probabilities for unlabeled data.

The aforementioned defense approaches all possess certain limitations. For practical applications, our method conducts defense in a completely black-box scenario.

Methodology

Preliminaries

Recommendation problem. Let $\mathcal{U}$ and $\mathcal{I}$ respectively denote the sets of users and items, and $\mathcal{D}$ represent the set of all data. $R \in \mathbb{R}^{M \times N}$ represents interaction data, where M (N) denotes the number of users (items) respectively. $e_u, e_v \in \mathbb{R}^d$ denote d-dimensional embeddings of users and items learned by RSs, used to compute user preferences for items. This work uses Bayesian Personalized Ranking (BPR) (Rendle et al. 2012) as the recommendation objective function, which is defined as follows:

$$L_{BPR} = -\sum\nolimits_{(u,i,j)\in R} ln\sigma(e_u^T e_i - e_u^T e_j) + \lambda\|\Theta\|_2,$$

where σ is the sigmoid function, λ is regularization parameter and Θ denotes the model parameters. i and j respectively denote the positive sample that user u has interacted with and the negative sample that u has not interacted with.

Neighborhood aggregation. In each NA iteration, the node aggregates features from all adjacent nodes to update its embedding. The update of users is formulated as follows:

$$e_u^{(k+1)} = \sum\nolimits_{v\in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_v|}} e_v^{(k)}, \tag{1}$$

where N_u and N_v represent the neighbors of user and item nodes. $e_u^{(k)}$ denotes embedding at the k-th layer. The learned embedding is a weighted representation of multiple aggregation layers.

Contrastive learning. CL enhances embeddings by learning from multiple perspectives of graph, typically using InfoNCE (Oord, Li, and Vinyals 2018) as the loss function. The loss for users is defined as follows:

$$L_{cl}^{user} = \sum\nolimits_{u \in \mathcal{U}} -log \frac{exp(s(e'_u, e''_u)/\tau)}{\sum_{n \in (\mathcal{U} \cup \mathcal{I})} exp(s(e'_u, e''_n)/\tau)}, \quad (2)$$

where e'_u and e''_u represent the embeddings of user u learned from two different views. The function $s(\cdot)$ measures the similarity between two representations. τ is a temperature coefficient that controls the penalty strength for negative samples. Let γ be the coefficient that controls the magnitude of the CL loss. The combined loss function is:

$$\mathcal{L}_{rec} = L_{BPR} + \gamma L_{cl}. \quad (3)$$

Attacker's goal. In this work, we focus on the most common attack scenarios, namely the promoting attack, whose aim is to enhance the ranking of target items I^T.

Attacker's capability. The number of malicious user profiles $\mathcal{U}_M$ is limited, as too many can be easily detected. Unless otherwise specified, we assume an injection rate of 5%.

Defender's knowledge. For practical applications, we assume that defenders rely solely on interaction data without obtaining any additional knowledge.

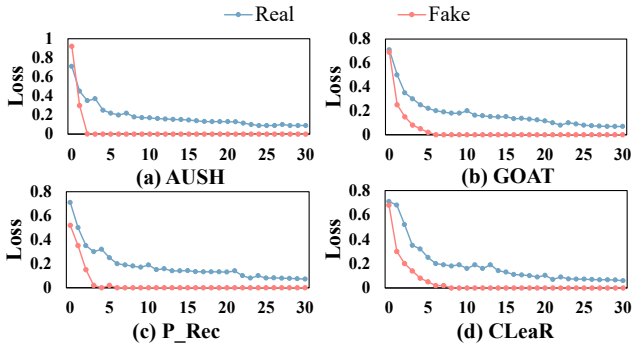

Figure 1: The average training loss between real and fake data under AUSH, GOAT, P_Rec, and CLeaR.

Pilot Experiments

Data behaviors in training dynamics. To accomplish defense in the black-box scenarios, we analyzed training behavior differences between real and fake data, including average loss, gradient, and parameter changes during training. We selected 9 attacks for testing, encompassing the neural network-based AUSH, GOAT, LegUP (Lin et al. 2022), GSP (Nguyen Thanh et al. 2023), P_Rec, RL (Zhang et al. 2021), and the gradient-based PGA (Li et al. 2016a), A_ra, and FedRecAttack and then trained LightGCN on the poisoned DouBan (Zhao, Qian, and Xie 2016) dataset, repeating each experiment 10 times and taking the average result. The results show that the loss of fake data decreases more rapidly compared to real data in the early stages of training for all 9 attacks. The average training loss under partial attack results was plotted in Figure 1, with the complete results

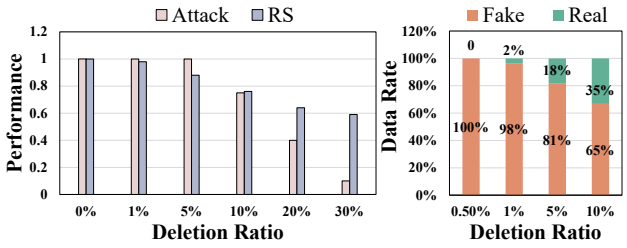

Figure 2: (a) Attack and recommendation performance after removing a certain proportion of data with the lowest loss (left). (b) The proportion of fake and real data under different deletion ratios (right).

presented in Appendix A.1. Moreover, simpler attacks, like AUSH, tend to exhibit a faster decrease. Additionally, with an increasing proportion of fake data, its average loss decreases more rapidly, as detailed in Appendix A.2. We guess that the small number of fake users and their structural similarity (Si and Li 2020) contribute to this difference.

Heuristic data deletion. Inspired by this difference, we designed a heuristic-based data deletion method and tested its effect on the DouBan, GOAT, and LightGCN. Specifically, after training on the poisoned dataset for 5 epochs, we remove data with the lowest loss as fake data, and then reinitialize the model to train on the new dataset. The results are shown in Figure 2 (left). As the deletion ratio increases, although attack performance gradually decreases, target items were still recommended, and the RSs performance reached a maximum reduction of 41% . To explain the results, figure 2 (right) shows the proportion of fake and real data among the removed samples. Since not all fake data have a lower loss than real, a small amount of fake data remains in the dataset after deletion. If training epochs are sufficiently long, the model can still learn the preference on target items. Moreover, heuristic deletion will cause the removal of some real data with lower loss, thereby affecting the RSs performance.

Trust-GRS

Building on the insights gained from pilot experiments, we propose Trust-GRS which is a two-stage joint training framework designed for GNN-based RSs. It divides the model's training into two stages: fake data identification stage and trustworthy RSs training stage. As shown in Figure 3, the first stage incorporates the data label probability modeling module, and the second stage encompasses two modules: trustworthy NA and trustworthy CL.

Data Label probability modeling. Without directly obtaining data labels, we propose an algorithm named Shilling-Rank to model the label probability distribution. Due to the similar interaction structure of fake data, fake users often frequently interact with the same target items. Consequently, in a dataset dominated by fake users, fake items will have significantly higher interaction counts than genuine items, indicating that interaction counts can reflect the probability of an item's label. At an early training epoch denoted as T_A,

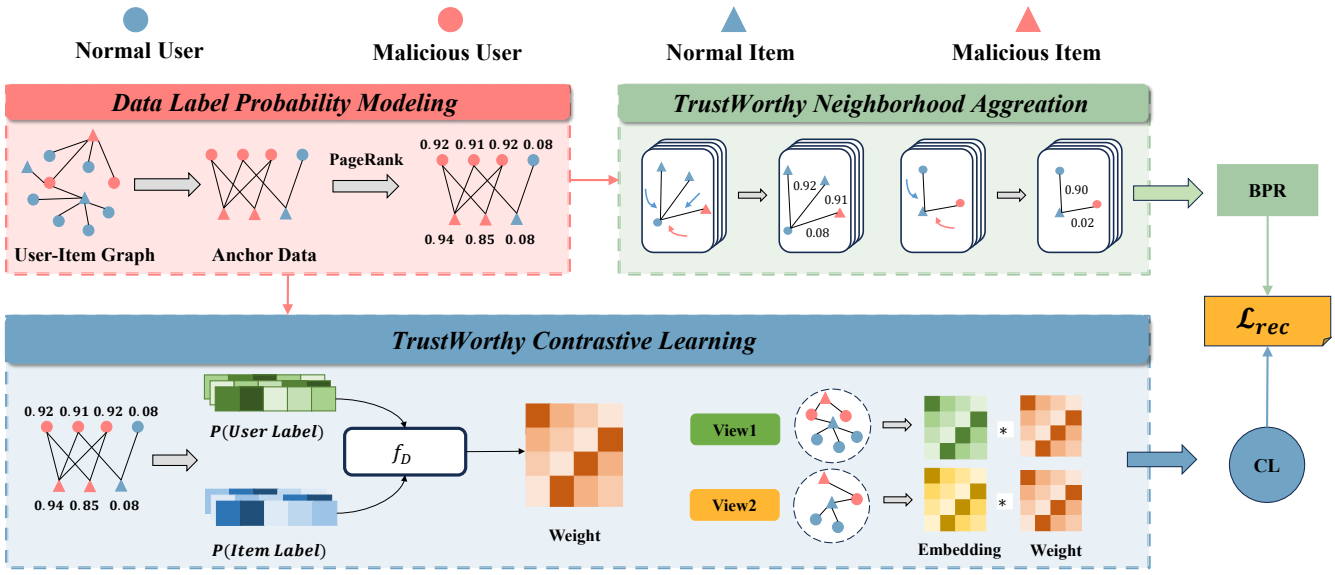

Figure 3: The overview of Trust-GRS: A Trustworthy zero-knowledge two-stage GNN-based RSs training framework.

we select a small subset of users with the lowest losses as **anchor data** D_A, which is a collection consisting almost entirely of fake users and then construct D_A as a user-item bipartite graph G_A. Given the shared goals of fake users, items linked by a larger number of users in G_A tend to be fake. Therefore, we employ a PageRank-based algorithm to calculate the probability that users and items are fake in G_A. Specifically, we freeze the user nodes and only calculate the PageRank values PR_v for item nodes. After iterations, items linked to more users have a higher PR_v and are more likely to be fake. We apply Min-Max normalization to the PR_v sequence, mapping it to the range [0, 1] as the probability $P_{v_i}^{Fake}$ that the item node is fake, as follows:

$$P_{v_i}^{Fake} = \frac{PR_{v_i} - min(PR_v)}{max(PR_v) - min(PR_v)}. \tag{4}$$

Since D_A contains a sufficient number of structurally similar fake users, D_A nearly encompasses all fake items. So we set the $P_{v_i}^{Fake}$ of items not included in D_A to 0. Given that fake items are typically less popular, users interacting with more fake items are prone to be fake. Therefore, the probability of a user being fake is computed as $P_{u_i}^{Fake} = \frac{1}{|N_v|}\sum_{u \in N_v} P_{v_i}^{Fake}$. When calculating user probabilities, fake users outside D_A will be correctly mapped based on their interactions with fake items.

Trustworthy neighborhood aggregation. After obtaining data label probabilities, we first modify the NA. Intuitively, if a node's neighbor is fake, setting its aggregation value to 0 will limit the propagation of this neighbor. Therefore, we employ the real probability $P_{v_i}^{Real} = 1 - P_{v_i}^{Fake}$ as the weight for aggregation. For example, the trustworthy NA formula for an item node is:

$$e_v^{(k+1)} = \sum\nolimits_{u \in N_v} \frac{P_{v_i}^{Real}}{\sqrt{|N_v|}\sqrt{|N_u|}} e_u^{(k)}, \tag{5}$$

where a higher probability of a neighboring node being fake results in a smaller weight. To model process using matrices, let $P_U \in \mathbb{R}^{M \times N}$ ($P_I \in \mathbb{R}^{M \times N}$) be the probability matrix of items (users) interacted with by users (items) as:

$$P_U = \begin{bmatrix} P_{v_{11}} & \cdots & P_{v_{1N}} \\ \vdots & \ddots & \vdots \\ P_{v_{M1}} & \cdots & P_{v_{MN}} \end{bmatrix}, P_I = \begin{bmatrix} P_{u_{11}} & \cdots & P_{u_{1M}} \\ \vdots & \ddots & \vdots \\ P_{u_{N1}} & \cdots & P_{u_{NM}} \end{bmatrix},$$

Let P_D be the adjacency probability matrix as $P_D = \begin{pmatrix} 0 & P_U \\ P_I & 0 \end{pmatrix}$. The trustworthy NA can be expressed as:

$$E^{(k+1)} = (D^{-\frac{1}{2}} A D^{-\frac{1}{2}} P_D) E^{(k)}. \tag{6}$$

The probabilities P^{Fake} generated in the first stage directly impact the defense performance in the second stage, as the essence of P_D is to reduce the influence of suspected fake data based on their probability. When D_A contains too many real users, the difference in interaction counts between fake and real items decreases, leading to a reduction in the P^{Fake} of fake items. That increases the weight of fake data during the aggregation, thereby reducing the defense performance in the second stage. To correctly classify fake items, we introduce the **confidence threshold** α. When $P_{v_i} > \alpha$, set $P_{v_i}^{Fake} = 1$, i.e., v_i is fake, otherwise to 0. Adjusting α will change the number of data classified as fake in D_A, achieving a trade-off between defense and RSs performance.

Trustworthy contrastive learning. To counteract the vulnerability introduced by original CL mechanism, we develop a new contrastive loss based on the label probabilities. Directly canceling the margin adjustment of CL will cause the model to lose its ability to address the cold-start problem. Thus, we adjust the margin and data augmentation based on the label probabilities. We design a distance function f_D:

$$f_D = P_1^{Real} * P_2^{Real} + \beta |P_1^{Real} - P_2^{Real}|, \tag{7}$$

Datasets	Defense	Attacks											
		AUSH	RL	P_Rec	DL	FedRec	GSP	LegUP	GOAT	PGA	A_ra	Bi_B	CLeaR
ML-1M	Attack	0.0020	0.0020	0.0021	0.0020	0.0004	0.0019	0.0020	0.0021	0.0018	0.0024	0.0023	0.0029
	PCA	0.0020	0.0020	0.0019	0.0020	0.0004	0.0020	0.0020	0.0021	0.0020	0.0024	0.0022	0.0028
	FAP	0.0021	0.0020	0.0020	0.0020	0.0004	0.0020	0.0020	0.0021	0.0020	0.0022	0.0022	0.0026
	Semi-SAD	0.0020	0.0020	0.0022	0.0020	0.0004	0.0022	0.0019	0.0019	0.0020	0.0024	0.0019	0.0028
	Degree-SAD	0.0	0.0017	0.0009	0.0010	0.0	0.0012	0.0012	0.0020	0.0020	0.0016	0.0020	0.0016
	AntiFU	0.0	0.0	0.0016	0.0	0.0	0.0022	0.0006	0.0008	0.0	3e-5	0.0011	0.0013
	Trust-GRS	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
DouBan	Attack	0.0023	0.0022	0.0022	0.0021	0.0022	0.0022	0.0022	0.0022	0.0023	0.0024	0.0024	0.0024
	PCA	0.0020	0.0021	0.0022	0.0022	0.0021	0.0020	0.0022	0.0016	0.0020	0.0024	0.0019	0.0026
	FAP	0.0022	0.0020	0.0021	0.0018	0.0021	0.0020	0.0016	0.0022	0.0020	0.0024	0.0024	0.0022
	Semi-SAD	0.0020	0.0020	0.0018	0.0020	0.0021	0.0022	0.0021	0.0022	0.0019	0.0024	0.0024	0.0026
	Degree-SAD	0.0012	0.0011	0.0012	0.0008	0.0009	0.0010	0.0012	0.0014	0.0014	0.0011	0.0012	0.0008
	AntiFU	0.0020	0.0	0.0022	0.0	0.0	0.0	0.0006	0.0013	0.0020	0.0018	0.0	0.0023
	Trust-GRS	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Epinions	Attack	0.0022	0.0021	0.0021	0.0021	0.0118	0.0019	0.0021	0.0034	0.0020	0.0123	0.0108	0.0130
	PCA	0.0022	0.0022	0.0019	0.0020	0.0110	0.0020	0.0020	0.0031	0.0020	0.0124	0.0104	0.0110
	FAP	0.0022	0.0021	0.0021	0.0019	0.0118	0.0020	0.0019	0.0032	0.0022	0.0107	0.0102	0.0130
	Semi-SAD	0.0020	0.0018	0.0022	0.0020	0.0126	0.0017	0.0022	0.0032	0.0022	0.0102	0.0090	0.0132
	Degree-SAD	2e-5	0.0007	0.0014	0.0010	0.0090	0.0012	0.0006	0.0012	0.0011	0.0120	0.0070	0.0110
	AntiFU	0.0022	0.0019	0.0	0.0	0.0	0.0022	0.0020	0.0	0.0	0.0	0.0	0.0113
	Trust-GRS	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

Table 1: Performance comparison of different defense methods on GNN-based RSs under multiple attacks. The best result is indicated in bold, and the runner-up is indicated with an underline. The numbers represent HR@50 of the target items.

where β is a scaling factor used to control the amplification of the distance between real and fake data. Since P^{Real} is close to 1 for real data and close to 0 for fake data, the interval between real and fake data is amplified by β, while the distance between real data remains largely unchanged and the distance of the fake data approaches 0. After obtaining f_D, we enhance the data using P^{Real} and conduct a trustworthy CL loss. The loss for users is defined as follows:

$$L^{user}_{trust_{cl}} = \sum\nolimits_{u \in \mathcal{U}} -log \frac{exp(P^{Real}_{u_i} * s(e'_u, e''_u)/\tau)}{\sum_{n \in (\mathcal{U} \cup \mathcal{I})} exp(f_D * s(e'_u, e''_n)/\tau)}. \tag{8}$$

We present the Trust-GRS algorithm in Appendix B.

Experiments

Experiment Settings

Datasets. We use three different scales public datasets for experiments: DouBan, ML-1M (Fang et al. 2018), and Epinions (Pan, He, and Yu 2020). Each dataset is split into training, validation, and test sets in a 70/10/20% ratio. The statistical attributes of datasets are presented in Appendix C.1.

Recommender systems. We select 5 recent GNN-based RSs: LightGCN, SSL4Rec (Yao et al. 2021), SGL, SimGCL and XSimGCL (Yu et al. 2023) as backbones of study. For each model, we adhered to the experimental settings and hyperparameters provided in the original papers. The training epochs are set to 30 and the embedding size is set to 64.

Attack & defense methods. With the help of ARLib (Wang et al. 2024), we evaluate the defense performance of Trust-GRS against 12 SOTA baseline attacks: A_ra, AUSH, Bi_B (Chen and Li 2019), CLeaR, DL (Huang et al. 2021), FedRec, GOAT, GSP, LegUP, PGA, P_Rec and RL. We compare our approach with five defense baselines: PCA, FAP, Degree-SAD (Li et al. 2016b), Semi-SAD (Cao et al. 2013), and AntiFU. The introduction and implementation of each baseline can be found in Appendix C.2. Before the attack, we randomly select 5 unpopular items as the targets.

Evaluation protocol. We use three of the most common metrics for measuring recommendation performance (Wang et al. 2023) : Hit-Ratio@50, Recall@50, and NDCG@50.

Defense Performance Comparison Settings

Defense on GNN-based RSs. We select LightGCN as the victim model and apply different defense methods for comparison. Some baseline defense strategies involve classifying real and fake data, and our strategy for them is to remove data classified as fake before training. The hyperparameters α and β of Trust-GRS are set to 0.6 and 10 respectively. Based on the pilot experiments, taking 0.5% of the data with the lowest loss as anchor data and set T_A to 5. Note all metrics are 0 in the absence of attacks. The HR@50 which calculates the frequency of target items is shown in Table 1 and the complete measurement results can be found in Appendix C.3. The original attack performance in each dataset is initially presented in the table, followed by the results of each defense. By examining Table 1, we can find:

- The proposed Trust-GRS is the best defense method, reducing metrics to 0 under all attacks.

Datasets	Model	Degree-SAD		AntiFU		Trust-GRS	
		A_ra	CLeaR	A_ra	CLeaR	A_ra	CLeaR
ML-1M	SSL4Rec	0.018 / 0.011	0.045 / 0.032	0.018 / 0.0003	0.045 / 0.013	**0.018 / 0.0**	**0.045 / 0.0**
	SGL	0.034 / 0.026	0.037 / 0.032	0.034 / 0.023	0.037 / 0.012	**0.034 / 0.0**	**0.037 / 0.0**
	SimGCL	0.061 / 0.058	0.095 / 0.072	0.061 / 0.019	0.095 / 7e-5	**0.061 / 0.0**	**0.095 / 0.0**
	XSimGCL	0.087 / 0.076	0.090 / 0.045	0.087 / 0.079	0.090 / 0.091	**0.087 / 0.0**	**0.090 / 0.0**
DouBan	SSL4Rec	0.023 / 0.012	0.046 / 0.034	0.023 / 0.017	0.046 / 0.042	**0.023 / 0.0**	**0.046 / 0.0**
	SGL	0.042 / 0.033	0.037 / 0.032	0.042 / 0.026	0.037 / 0.019	**0.042 / 0.0**	**0.037 / 0.0**
	SimGCL	0.047 / 0.022	0.081 / 0.036	0.047 / 0.033	0.081 / 0.079	**0.047 / 0.0**	**0.081 / 0.0**
	XSimGCL	0.079 / 0.067	0.094 / 0.071	0.079 / 0.021	0.094 / 0.072	**0.079 / 0.0**	**0.094 / 0.0**
Epinions	SSL4Rec	0.084 / 0.082	0.093 / 0.091	0.084 / 0.004	0.093 / 0.082	**0.084 / 0.0**	**0.093 / 0.0**
	SGL	0.072 / 0.067	0.094 / 0.046	0.072 / 0.036	0.094 / 0.093	**0.072 / 0.0**	**0.094 / 0.0**
	SimGCL	0.092 / 0.089	0.095 / 0.076	0.092 / 0.036	0.095 / 0.092	**0.092 / 0.0**	**0.095 / 0.0**
	XSimGCL	0.093 / 0.079	0.091 / 0.092	0.093 / 0.026	0.091 / 0.088	**0.093 / 0.0**	**0.091 / 0.0**

Table 2: Performance comparison of different defense methods on CL-based recommender systems under multiple attacks. Each column shows the HR@50 of target items before and after defense, separated by a slash.

- PCA, FAP and Semi-SAD exhibit poor performance, with negligible reduction in metrics, indicating that they struggle to defend SOTA attacks. Degree-SAD shows relative improvement over them, but a small amount of fake data remains in the dataset still accomplish attack. AntiFU is suboptimal in most cases as it undertakes specialized processing on NA. However, it sometimes provides almost no defense against attacks, as its performance is constrained by the predefined attacks. Trust-GRS leverages the inherent characteristics of fake data, enabling it to defend against all types of attacks.
- The strongest attacks are P_Rec, A_ra, Bi_B, and CLeaR. Especially for CLeaR, which utilizes CL for attacks, making it difficult to defend for baselines.

Defense on GNN-based RSs with CL. We select the two most powerful attacks: A_ra and CLeaR to compare the defense performance on the SGL, SimGCL, and XSimGCL. As CL can further promote attacks, we chose Degree-SAD and AntiFU which performed relatively well in previous experiments as baselines. The HR@50 results are shown in Table 2. More attacks and metrics results are provided in Appendix C.4. Specially, we used t-SNE (Van der Maaten and Hinton 2008) to visualize the representations learned by Trust-GRS in Appendix C.9. Through Table 2, we can find:

- Trust-GRS exhibits the best performance across all models and attacks. Previously well-performing AntiFU shows a significant drop, as it only addresses the impact of NA and fails to defend effectively in CL scenarios.
- Trust-GRS is a universal framework. Trust-GRS specifically addresses the negative impact caused by modifications to representation space distance and can defend against various CL-based RSs.

Real and fake data classification. We also compared the data classification accuracy of Trust-GRS, with the results presented in Appendix C.5. The results show that our method achieved the highest accuracy across all attacks.

No malicious data. To the best of our knowledge, existing defenses cannot directly detect whether a dataset contains poisoned data. We investigate the impact of each defense on RSs performance in clean datasets. Trust-GRS has minimal impact on RSs performance, as shown in Appendix C.6.

The average training loss of Trust-GRS. Finally, we speculate that Trust-GRS essentially penalizes the loss of fake data, preventing it from converging. As shown in Figure 4, we plotted loss variation of real and fake data during the Trust-GRS training process under the AUSH attack, which confirms our speculation.

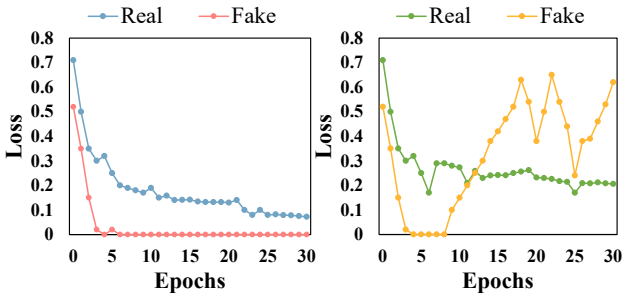

Figure 4: AUSH (left) and Trust-GRS (right) data loss variation during training.

Study on Hyperparameter Impact

Impact of α on defense performance. We test the defense performance of the trustworthy NA module without α and with different α values across 3 datasets and 12 attacks. Partial results are presented in Table 3, with the full results available in Appendix C.7. Through Table 3, we observe that in most attacks, Trust-GRS can reduce the metrics to 0 without confidence threshold α. For A_ra, Bi_B, and CLeaR, a lower α is needed to also reduces the attack metrics to 0. This is because fake data generated by stronger attacks is more similar to real data, causing a lower P^{Fake} for

fake data generated in the first stage, which further reduces the defense performance in the second stage. Based on the above observations, we can find that confidence threshold α can mitigate the effects of errors in the first stage on the second stage. Furthermore, the dataset can also influence the defense performance. In DouBan, all metrics can be reduced to 0 without setting α. This is because DouBan is denser and attacking it is simpler (You et al. 2023).

Datasets	α	Attacks					
		AUSH	RL	DL	A_ra	Bi_B	CLeaR
ML-1M	w/o	0.0	0.0	0.0	0.0021	0.0012	0.0022
	0.8	0.0	0.0	0.0	0.0013	0.0016	0.0009
	0.6	0.0	0.0	0.0	0.0	0.0	0.0
	0	0.0	0.0	0.0	0.0	0.0	0.0
DouBan	w/o	0.0	0.0	0.0	0.0	0.0	0.0
	0.8	0.0	0.0	0.0	0.0	0.0	0.0
	0.6	0.0	0.0	0.0	0.0	0.0	0.0
	0	0.0	0.0	0.0	0.0	0.0	0.0

Table 3: Attack performance with respect to α. "w/o α" represents not applying confidence threshold adjustment to the probabilities generated in the first stage of Trust-GRS.

Impact of α on recommendation performance. Lowering α may misclassify the real data and lead to a decline of the RSs performance. According to Table 3, we test A_ra, Bi_B, and CLeaR with AUSH as a comparison. The HR@50 results are shown in Figure 5. As α decreases, the RSs performance on A_ra, Bi_B, and CLeaR begins to decline significantly. We find that the performance decline of AUSH is the lowest, while that of CLeaR can reach up to 31%. This difference is due to the varying amounts of real data included in D_A. Therefore, under complex attacks and datasets, α needs to be carefully set to balance RSs performance and defense effectiveness.

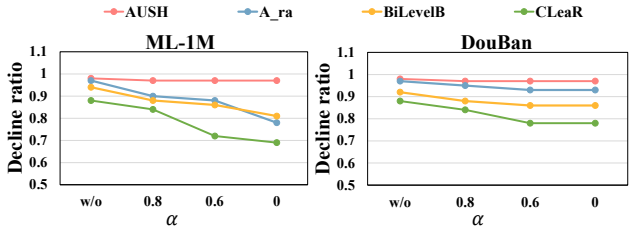

Figure 5: The decline ratio of RSs performance under different values of α.

Impact of β. β mainly affects the separation between real and fake in f_D. We conducted experiments with Trust-CL on SGL and ML-1M datasets, with the results shown in Table 4. As β increases, the attack performance gradually declines, indicating that fake data begins to move away from real data. However, when β is too large, such as 100, the CL loss becomes NaN and model cannot converge. Therefore, If β is too small, the defense performance will be poor; if it is too large, the loss will not converge. We recommend setting β between 2 and 10 to minimize the negative impact of CL.

Datasets	β	Attacks		
		A_ra	Bi_B	CLeaR
ML-1M	w/o	0.034	0.037	0.072
	2	0.032	0.036	0.071
	6	0.009	0.012	0.023
	10	3e-5	0.0022	0.0026
	100	NaN	NaN	NaN

Table 4: Attack performance with respect to β.

Ablation Study

In this section, we explore the roles of the trustworthy NA (Trust-GRS$_{NA}$) and trustworthy CL (Trust-GRS$_{CL}$) modules. We conducted experiments on XSimGCL under A_ra and CLeaR. The HR@50 results are shown in Table 5, and other models and metrics results are available in Appendix C.8. From Table 5, we find that CL contributes more to fake data and Trust-GRS is the only defense that considers both mechanisms. The Trust-GRS$_{NA}$ and Trust-GRS$_{CL}$ as independent modules significantly reduce the metric compared to the attack scenarios. Finally, with the synergy of both modules, Trust-GRS can completely reduce the metric to 0.

Datasets	Cases	Attacks	
		A_ra	CLeaR
ML-1M	Attack$_{NA}$	0.0023	0.0029
	Attack$_{NA+CL}$	0.087	0.090
	Trust-GRS$_{NA}$	0.008	0.017
	Trust-GRS$_{CL}$	0.003	0.0009
	Trust-GRS	0.0	0.0

Table 5: The effect of the two modules of Trust-GRS.

Conclusion

In this work, we propose a zero-knowledge two-stage trustworthy GNN-based recommender systems training framework called Trust-GRS, aimed at eliminating the facilitating effects of the GNN architecture on shilling attacks. We utilize model training dynamics to identify fake data in the black-box scenarios and progressively mitigate the negative impact of fake data through weighted trustworthy neighborhood aggregation and trustworthy contrastive learning. Our framework can be widely adapted to GNN-based RSs, and since it doesn't demand any additional knowledge, Trust-GRS can assist the real-world recommendation platforms in defending against potential malicious data within unknown datasets. Inspired by the inherent attribute differences, we consider using adversarial training and perturbation injection to smooth the intrinsic characteristics between real and fake data, thereby enhancing the stealth and aggressiveness of the attacks in the future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62472419, No.62472420).

References

Bhaumik, R.; Mobasher, B.; and Burke, R. 2011. A clustering approach to unsupervised attack detection in collaborative recommender systems. In *Proceedings of the International Conference on Data Science (ICDATA)*, 1. Citeseer.

Burke, R.; Mobasher, B.; Williams, C.; and Bhaumik, R. 2006. Classification features for attack detection in collaborative recommender systems. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 542–547.

Cao, J.; Wu, Z.; Mao, B.; and Zhang, Y. 2013. Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. *World Wide Web*, 16: 729–748.

Chen, H.; and Li, J. 2019. Data poisoning attacks on cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2177–2180.

Christakopoulou, K.; and Banerjee, A. 2019. Adversarial attacks on an oblivious recommender. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 322–330.

Fang, M.; Yang, G.; Gong, N. Z.; and Liu, J. 2018. Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference*, 381–392.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, 1735–1742. IEEE.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 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.

Huang, H.; Mu, J.; Gong, N. Z.; Li, Q.; Liu, B.; and Xu, M. 2021. Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Lam, S. K.; and Riedl, J. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, 393–402.

Li, B.; Wang, Y.; Singh, A.; and Vorobeychik, Y. 2016a. Data poisoning attacks on factorization-based collaborative filtering. *Advances in neural information processing systems*, 29.

Li, W.; Gao, M.; Li, H.; Zeng, J.; Xiong, Q.; and Hirokawa, S. 2016b. Shilling attack detection in recommender systems via selecting patterns analysis. *IEICE TRANSACTIONS on Information and Systems*, 99(10): 2600–2611.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021. Anti-Backdoor Learning: Training Clean Models on Poisoned Data. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 14900–14912. Curran Associates, Inc.

Lin, C.; Chen, S.; Li, H.; Xiao, Y.; Li, L.; and Yang, Q. 2020. Attacking recommender systems with augmented user profiles. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 855–864.

Lin, C.; Chen, S.; Zeng, M.; Zhang, S.; Gao, M.; and Li, H. 2022. Shilling black-box recommender systems by learning to generate fake user profiles. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1): 1305–1319.

Mehta, B.; and Nejdl, W. 2009. Unsupervised strategies for shilling detection and robust collaborative filtering. *User Modeling and User-Adapted Interaction*, 19: 65–97.

Nguyen Thanh, T.; Quach, N. D. K.; Nguyen, T. T.; Huynh, T. T.; Vu, V. H.; Nguyen, P. L.; Jo, J.; and Nguyen, Q. V. H. 2023. Poisoning GNN-based recommender systems with generative surrogate-based attacks. *ACM Transactions on Information Systems*, 41(3): 1–24.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford infolab.

Pan, Y.; He, F.; and Yu, H. 2020. Learning social representations with deep autoencoder for recommender system. *World Wide Web*, 23(4): 2259–2279.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Rong, D.; He, Q.; and Chen, J. 2022. Poisoning deep learning based recommender model in federated learning scenarios. *arXiv preprint arXiv:2204.13594*.

Rong, D.; Ye, S.; Zhao, R.; Yuen, H. N.; Chen, J.; and He, Q. 2022. Fedrecattack: Model poisoning attack to federated recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2643–2655. IEEE.

Saxe, A. M.; McClelland, J. L.; and Ganguli, S. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Si, M.; and Li, Q. 2020. Shilling attacks against collaborative recommender systems: a review. *Artificial Intelligence Review*, 53: 291–319.

Song, J.; Li, Z.; Hu, Z.; Wu, Y.; Li, Z.; Li, J.; and Gao, J. 2020. Poisonrec: an adaptive data poisoning framework for attacking black-box recommender systems. In *2020 IEEE 36th international conference on data engineering (ICDE)*, 157–168. IEEE.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, S.; Cao, L.; Wang, Y.; Sheng, Q. Z.; Orgun, M. A.; and Lian, D. 2021. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)*, 54(7): 1–38.

Wang, Z.; Yu, J.; Gao, M.; Ye, G.; Sadiq, S.; and Yin, H. 2024. Poisoning Attacks and Defenses in Recommender Systems: A Survey. *arXiv preprint arXiv:2406.01022*.

Wang, Z.; Yu, J.; Gao, M.; Yin, H.; Cui, B.; and Sadiq, S. 2023. Poisoning Attacks Against Contrastive Recommender Systems. *arXiv preprint arXiv:2311.18244*.

Wu, F.; Gao, M.; Yu, J.; Wang, Z.; Liu, K.; and Wang, X. 2021a. Ready for emerging threats to recommender systems? A graph convolution-based generative shilling attack. *Information Sciences*, 578: 683–701.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021b. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 726–735.

Wu, Z.; Wu, J.; Cao, J.; and Tao, D. 2012. HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 985–993.

Yao, T.; Yi, X.; Cheng, D. Z.; Yu, F.; Chen, T.; Menon, A.; Hong, L.; Chi, E. H.; Tjoa, S.; Kang, J.; et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 4321–4330.

You, X.; Li, C.; Ding, D.; Zhang, M.; Feng, F.; Pan, X.; and Yang, M. 2023. Anti-fakeu: Defending shilling attacks on graph neural network based recommender model. In *Proceedings of the ACM Web Conference 2023*, 938–948.

Yu, H.; Zheng, H.; Xu, Y.; Ma, R.; Gao, D.; and Zhang, F. 2021. Detecting group shilling attacks in recommender systems based on maximum dense subtensor mining. In *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 644–648. IEEE.

Yu, J.; Xia, X.; Chen, T.; Cui, L.; Hung, N. Q. V.; and Yin, H. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2): 913–926.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 1294–1303.

Zhang, K.; Cao, Q.; Sun, F.; Wu, Y.; Tao, S.; Shen, H.; and Cheng, X. 2023. Robust Recommender System: A Survey and Future Directions. arXiv:2309.02057.

Zhang, S.; Yin, H.; Chen, T.; Hung, Q. V. N.; Huang, Z.; and Cui, L. 2020. GCN-Based User Representation Learning for Unifying Robust Recommendation and Fraudster Detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, 689–698. New York, NY, USA: Association for Computing Machinery. ISBN 9781450380164.

Zhang, Y.; Tan, Y.; Zhang, M.; Liu, Y.; Chua, T.-S.; and Ma, S. 2015. Catch the black sheep: unified framework for shilling attack detection based on fraudulent action propagation. In *Twenty-fourth international joint conference on artificial intelligence*.

Zhang, Y.; Yuan, X.; Li, J.; Lou, J.; Chen, L.; and Tzeng, N.-F. 2021. Reverse attack: Black-box attacks on collaborative recommendation. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 51–68.

Zhao, G.; Qian, X.; and Xie, X. 2016. User-service rating prediction by exploring social users' rating behaviors. *IEEE transactions on multimedia*, 18(3): 496–506.