Robust Recommendation against Shilling Attacks via List Consistency and Counterfactual Neighbor Analysis

Fan Mo †§ Chongxian Chen ‡ Xin Fan ‡ and Hayato Yamana †

† Faculty of Science and Engineering, Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan
‡ Dept. of CSCE, Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan
§Zozo research, Chiba, Japan
E-mail: † bakubonn@toki.waseda.jp, yamana@yama.info.waseda.ac.jp
‡ fan_xin@fuji.waseda.jp, chenc@toki.waseda.jp

Abstract Shilling attacks pose a critical threat to recommendation systems, where malicious users inject crafted interactions or comments to promote target items. Existing defenses can be categorized into explicit detection of fake users or interactions, and anti-shilling recommendation methods. Recent studies increasingly focus on anti-shilling recommendation, which aims to preserve the consistency of recommendation lists after and before attacks, rather than explicitly detecting malicious users. However, existing anti-shilling methods assume the presence of attacks and focus on malicious suppression, which may lead to additional computational overhead while potentially degrading recommendation performance when no shilling attacks are present in the system. Besides, they lack generality, as the anti-shilling methods are designed for specific recommendation models and training procedures, which limits their applicability and practicality in real-world systems where recommenders are frequently updated and continuously evolving. In this work, we propose a counterfactual-based reranking framework for anti-shilling recommendation. This proposed method improves accuracy by refining noisy rankings via list-based consistency when no shilling attacks are present, and naturally suppresses maliciously injected recommendations when attacks occur. Specifically, we construct a recommendation list similarity graph from top-K outputs of a base recommender, leveraging the collaborative filtering assumption that users with similar preferences receive similar recommended items. This list-level consistency is used to suppress anomalous recommendations weakly supported by similar users. Besides, we introduce a counterfactual neighbor–based analysis that measures the stability of user representations by randomly masking neighbors during training. Users exhibiting large embedding variations are regarded as suspicious, as genuine users typically exhibit stable representations; meanwhile, in the absence of shilling attacks, stable representations contribute to consistent preference learning, therefore improve accuracy.

Keyword graph neural network, collaborative filtering, shilling attack, anti-shilling, recommendation system

1. Introduction

Recommendation systems play a central role in modern online platforms by filtering vast amounts of content and guiding user decision-making. In recent years, a variety of learning-based recommendation have been developed, including neural collaborative filtering (NCF) [1] approaches that model nonlinear user–item interactions, graph-based methods [2] [3] [9] [10] [32] that leverage user–item graphs to propagate collaborative signals and model high-order connectivity.

However, the training of recommendation systems relaying on user-generated interactions makes them particularly vulnerable to shilling attacks, a class of adversarial behaviors in which malicious users inject crafted interactions or reviews to artificially promote target items. Such attacks can significantly distort recommendation results, degrade user experience, and undermine the credibility of the platform.

To solve the problem of shilling attack, existing studies can be broadly categorized into two groups: detection-based methods [4] and anti-shilling-based methods [5] [6]. Detection-based approaches suffer from two inherent limitations. 1) Training an effective detector typically requires a large number of labeled fake users to ensure reliable performance; however, in real-world systems, the proportion of ground-truth fake users is usually extremely low [7], leading to severe class imbalance in the training data; 2) Constructing representative features and collecting sufficient training examples is labor-intensive and may further raise concerns regarding privacy compliance in user profiling [8].

Therefore, in recent years, increasing attention has been directed toward anti-shilling recommendation methods [5] [6]. Instead of explicitly detecting fake users or malicious

interactions, anti-shilling approaches aim to mitigate the impact of shilling attacks on recommendation outcomes, ensuring that the recommendation lists remain stable and reliable even in the presence of malicious behaviors. You et al. [5] pioneered the study of anti-shilling recommendation. They proposed a GCN-based anti-shilling recommendation framework that consists of two stages. The model first predicts the probability of a user being fake, and then integrates the predicted scores into the recommendation process to prevent the propagation of negative impacts caused by shilling attacks. Based on You et al., Mu et al. [6] propose Trust-GRS, which estimates the probabilities of users and items being fake by exploiting training dynamics and interaction frequency anomalies. Specifically, the method identifies suspicious users in early training stages and employs a PageRank-based algorithm, termed Shilling-Rank, to propagate fake probabilities over the user–item graph.

However, their methods suffer from several limitations. 1). These methods lack generality, as existing anti-shilling methods are often tightly coupled with specific recommendation models and training procedures, making them difficult to transfer or deploy across different recommendation models. Specifically, applying these methods requires detailed knowledge of the internal structure of the underlying recommender and corresponding modifications to the recommendation model, which limits their applicability in practical scenarios where the base recommender is fixed, proprietary, or costly to retrain, and therefore cannot be easily extended as a black-box component. 2). Existing anti-shilling methods are designed under the explicit assumption that shilling attacks are present, and thus primarily focus on suppressing malicious behaviors. Therefore, when no shilling attacks are present, existing anti-shilling methods are not explicitly designed to optimize recommendation performance under clean settings, which may introduce extra computational overhead and may potentially affect recommendation results.

In this paper, we propose ConsisRec (consistency-aware recommendation), a post-processing approach that operates on the output of a base recommender system. Given the top-K recommendation outputs of a base recommender, we first construct a candidate list-based similarity graph, inspired by collaborative filtering, where users with similar preferences tend to receive similar recommended items. This list-level consistency provides a robust collaborative signal that enables us to suppress anomalous recommendations that are weakly supported by similar users. Furthermore, we introduce a counterfactual neighbor–based analysis to assess the stability of user representations. By masking neighbors during training, we measure how user embeddings vary under neighborhood perturbations. Users exhibiting large embedding variations are regarded as suspicious, as genuine users typically maintain stable representations. This stability signal is used in a soft and model-independent manner, without explicitly detecting or removing users.

Our framework is scalable and non-intrusive, because it operates purely as a post-processing mechanism on top of any base recommender. When no shilling attacks are present, the proposed reranking mechanism improves recommendation accuracy by refining noisy ranking results through list-based collaborative consistency. When shilling attacks occur, the same mechanism naturally suppresses maliciously injected recommendations, achieving effective anti-shilling robustness without sacrificing performance in benign settings. Our contributions are listed as follows.

- We propose ConsisRec, a scalable and model-agnostic post-processing framework for anti-shilling recommendation, applicable to various recommender models.
- We exploit list-based collaborative consistency via a candidate list similarity graph propagation to refine ranking results.
- We introduce a counterfactual stability signal to softly suppress the influence of suspicious users.
- We will further investigate the dual role of ConsisRec in both clean and adversarial environments. Our preliminary results reveal that even under shilling attacks, ConsisRec achieves higher recommendation accuracy than the base recommender in attack-free settings, indicating that the proposed method not only mitigates the negative impact of malicious interactions but also fundamentally improves representation quality beyond the original model's capability.

The remainder of the paper is organized as follows: Section 2 reviews related work on shilling attacks and anti-shilling recommendation. Section 3 introduces the preliminaries. Section 4 presents the proposed ConsisRec framework. Section 5 describes the experimental setup and reports the experimental results. Finally, Section 6 concludes the paper.

2. Related Work

2.1. Shilling attack

Shilling attacks manipulate recommendation systems by strategically injecting artificial user profiles whose

interaction patterns are deliberately designed to promote target items. Shilling attacks can be categorized into three types according to how the attack profiles are generated: heuristic attacks, neural-network-based attacks, and gradient-based attacks. Heuristic attacks generate fake user profiles by following predefined item selection rules [11] [12]. Popularity attacks construct fake user profiles by interacting with target items and a set of popular items, aiming to maximize overlap with genuine users. Random attacks generate fake profiles by combining interactions on target items with randomly selected filler items, to mimic normal user behavior. Recently, shilling attacks based on neural networks and gradient optimization have gained increasing attention. Neural network–based attacks leverage deep learning models to automatically learn realistic interaction patterns for constructing fake user profiles. For example, PRec [13] formulates shilling attack generation as a reinforcement learning problem, while GOAT [14] adopt generative adversarial networks to synthesize attack profiles. Gradient-based attacks cast shilling attack generation as a bi-level optimization problem, where approximate gradients are exploited to iteratively modify the original data and generate the final attack profiles. Neural network–based and gradient-based attacks make defending against shilling attacks increasingly challenging, as such attacks can adaptively optimize injected interactions to closely mimic genuine user behaviors and exploit model-specific vulnerabilities.

2.2. Shilling Attack Defense

Existing defenses against shilling attacks can be divided into two categories: explicit detection of shilling users or items, and anti-shilling recommendation methods that mitigate the impact of malicious manipulation on recommendation results. Explicit detection has been regarded as one of the most straightforward defenses against shilling attacks [15]. Early studies by Burke et al. trained classifiers using carefully designed features extracted from the rating matrix to identify malicious users [16]. Bhaumik et al., proposed unsupervised detection methods based on clustering and data mining techniques to identify fake profiles by exploiting statistical discrepancies between genuine and malicious data [17]. Wu et al. [18], proposed a probabilistic method to train a Naïve Bayes classifier on labeled data and infer posterior probabilities for unlabeled users. In recent years, graph-based detection methods have attracted increasing attention from the researchers. Li et al.[20] proposed SpDetector, which constructs user and item hypergraphs to extract spectral features capturing high-order interactions, and integrates them with rating prediction errors to accurately distinguish fake users from genuine ones. Zhang et al. [19] proposes a user similarity–based graph convolutional network (USGSAD) for the detection, which jointly model user rating correlation and deviation to identify malicious users without manual feature engineering. Hao et al. [21] modifies the graph structure by reweighting edges. They extracted popularity- and rating-based user features, constructed a weighted user graph, and employed a two-stage scheme with partial labeling and regularized GCN to detect hybrid model-generative shilling attacks. Despite their effectiveness, the above explicit detection-based methods have two limitations: 1) They rely on large amounts of labeled fake users, which are scarce in practice and lead to severe class imbalance. 2) Feature engineering and data collection are labor-intensive and raise privacy concerns.

To solve the problems, You et al. [5] pioneered the concept of anti-shilling recommendation by proposing a GCN-based framework that estimates fake-user probabilities and integrates the predicted scores into the recommendation process to mitigate shilling attacks. Building on this idea, Mu et al. [6] proposed Trust-GRS, which identified suspicious users in early training stages and employs a PageRank-based algorithm, termed Shilling-Rank, to propagate fake probabilities over the user–item graph. However, existing anti-shilling methods are model-specific and assume the presence of shilling attacks, which limits their generality and may introduce unnecessary overhead or performance degradation in clean settings.

3. Preliminary

This section introduces the preliminary knowledge about shilling attacks.

3.1. Recommendation task

$U = \{u_1, u_2, u_3, \dots, u_X\}$ and $I = \{i_1, i_2, i_3, \dots, i_Y\}$ denote the sets of users and items, $N = \{N_{u_x} | 1 \le x \le X\}$ denotes the set of N_{u_x}, and $N_{u_x} = \{i_1, i_2, \dots, i_y\}$ consists of the items checked by user u_x. The goal of the recommendation task is to predict a user's preference over unseen items and recommend those that the user is likely to click in the future. Table 1 provides a summary of the notations used in this paper.

3.2. Attacker's goal

We consider the most common shilling attack setting as You et al.[5] and Mu et al. [6], where the attacker aims to boost the ranking of a set of target items I^T. Specifically, the goal is to make the target items appear in the

Table 1: Notations

Notation	Definition
$e_{u_x}^{k}$	The output embedding of user u_x at k^{th} GCN layer
$e_{i_y}^{k}$	The output embedding of item $\boldsymbol{i_y}$ at k^{th} GCN layer
$\boldsymbol{e_{u_x}}$	The final output embedding produced by the GCN for user $\boldsymbol{u_x}$
$\boldsymbol{e_{i_y}}$	The final output embedding produced by the GCN for item $\boldsymbol{i_y}$
C_{u_x}	candidate item set of user u_x
C_{i_y}	candidate user set of item i_y
γ_{u_x}	risk assessment coefficient of user u_x, ranging from 0 to 1, controls the amount of information aggregated from user u_x during GCN propagation
γ_{i_y}	risk assessment coefficient of item $\boldsymbol{i_y}$, ranging from 0 to 1, controls the amount of information aggregated from item $\boldsymbol{i_y}$ during GCN propagation
r_{u_x}	The risk score of the user u_x, represents the estimated likelihood that the user is fake
r_{i_y}	The risk score of the item i_y, represents the estimated likelihood that the item is a target item

recommendation lists of as many users as possible.

3.3. Attacker's capability

To avoid easy detection, the number of malicious user profiles is limited; by default, same as You et al.[5], the injection rate is set to 1%.

3.4. Defender's knowledge

We assume that the defender only has access to check-in data, without any additional side information. Besides, the defender has no prior knowledge of the specific attack strategies.

4. Proposed method

This section introduces the details of the ConsisRec. As illustrated in Fig. 1, the architecture of the proposed method consists of three parts. Step1: The model initializes user and item embeddings using a Gaussian distribution. The shilling risk r_{u_x}/r_{i_y} of user and item is initialized to zero. Step2: Through risk-aware graph convolution propagation, the model produces the final embedding representations for users and items. At this stage, we first construct a graph structure based on the candidate item sets (Section 4.1). We then perform risk-aware neighbor aggregation, where information from suspicious neighbors is adaptively down-weighted during message passing (Section 4.2). We further apply a counterfactual masking strategy that selectively removes a subset of neighboring nodes to evaluate the stability of user representations (Section 4.3). Users whose embeddings exhibit pronounced sensitivity to such perturbations are regarded as suspicious, whereas genuine users are expected to maintain relatively stable representations under counterfactual graph structures. Step3: Based on the user and item embeddings obtained in the second stage, the model computes preference scores via inner products and generates the final recommendation list.

4.1. Candidate-Induced Graph Construction

For each user u_x, we first produce a candidate item set C_{u_x} using the base model by returning the top-L items, where top-L >>top-K. top-K denotes the number of items in the final recommendation list. This ensures that post-processing module training from a sufficiently rich information to mitigate shilling attacks. Based on these candidate sets, we further construct C_{i_y} for each item i_y, which consists of users whose candidate lists include item i_y.

After that, we construct a user-item graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that connects users and items based on the candidates. Specifically, $\mathcal{V}$ denotes the set of user and item nodes while $\mathcal{E}$ is a edge set, is defined as Eq. 1.

$$\mathcal{E} = \{(u_x, i_y) | u_x \in U, i_y \in C_{u_x}\} \tag{1}$$

4.2. Risk-aware GCN propagation

After constructing the graph structure, at each GCN layer, we design user and item representations by aggregating information from their neighbors, as formulated in Eq. 2.

$$e_{u_x}^{k} = \sum_{i_y \in C_{u_x}} \frac{\gamma_{i_y}}{\sqrt{|C_{u_x}| * |C_{i_y}|}} e_{i_y}^{k-1}$$
$$e_{i_y}^{k} = \sum_{u_x \in C_{i_y}} \frac{\gamma_{u_x}}{\sqrt{|C_{u_x}| * |C_{i_y}|}} e_{u_x}^{k-1} \tag{2}$$

, where $k \in \{1,2,3\}$, $e_{u_x}^{k}$ and $e_{i_y}^{k}$ represent the user and item embeddings produced at the k^{th} GCN layer while $e_{u_x}^{0}$ and $e_{i_y}^{0}$ represent the initial user and item embeddings, respectively. The term $1/\sqrt{|C_{u_x}| * |C_{i_y}|}$ serves as a degree-based normalization factor to stabilize message propagation and mitigate over-smoothing, where $|.|$ denotes the size of the candidate-based neighbor set. γ_{u_x} and γ_{i_y} are risk assessment coefficients ranging from 0 to 1, where 1 indicates a fully trusted neighbor. The corresponding computation is defined as Eq. 3.

$$\gamma_{u_x} = max\,(0, 1 - \sigma(\beta * r_{u_x}))$$
$$\gamma_{i_y} = max\,(0, 1 - \sigma(\beta * r_{i_y})) \tag{3}$$

, where β is a hyperparameter that controls the strength of risk r_{u_x}/r_{i_y} of a user/item node. The risk scores r_{u_x} and r_{i_y} represent the estimated likelihood that a user or an item is involved in shilling attack. σ is sigmoid function, used to smoothly map the estimated risk into the range 0 to 1. As

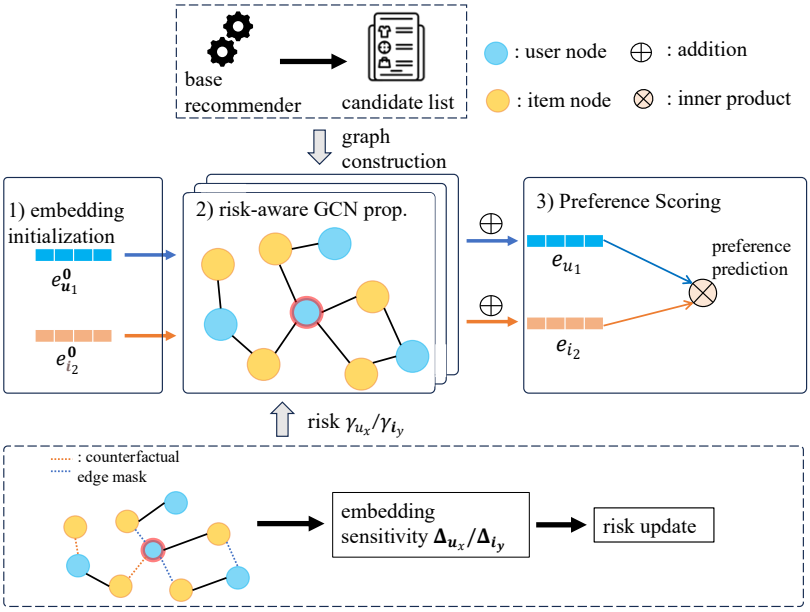

Figure 1: architecture of the proposed method

the estimated risk increases, less information is aggregated from the corresponding node by inversely weighting its contribution during neighbor aggregation. The update details of the risks r_{u_x} and r_{i_y} are described in Section 4.3. After the model outputs the high-order user and item representations, we aggregate all layer's outputs by mean pooling, as Eq. 4.

$$\boldsymbol{e}_{u_x} = \sum_{k=0}^{K} \alpha_k \boldsymbol{e}_{u_x}^{k} \qquad (4)$$
$$\boldsymbol{e}_{i_y} = \sum_{k=0}^{K} \alpha_k \boldsymbol{e}_{i_y}^{k}$$

, where α_k is a weighting coefficient fixed at $1/(K+1)$, with $K=3$. $\boldsymbol{e}_{u_x}$ and $\boldsymbol{e}_{i_y}$ represent the final embedding representations of user $\boldsymbol{u_x}$ and item $\boldsymbol{i_y}$.

4.3. Counterfactual-based risk estimation

This section describes how we quantify and update the risk associated with each user and item based on the counterfactual stability analysis. Specifically, we leverage the sensitivity of user representations to counterfactual neighbor perturbations as a risk signal, where unstable embedding behaviors indicate potential anomalous or unreliable interactions. On the user–item candidate graph $\mathcal{G}$, we first perform S times independent random edge masking operations. For each mask step s, we randomly mask 30% of the edges, treating the corresponding neighbors as absent, and generate the counterfactual graph $\mathcal{G}_s$. On the counterfactual graph $\mathcal{G}_s$, we apply Eqs. 2 to 4 to calculate the user and item embeddings under the counterfactual setting, denoted as $\boldsymbol{e}_{u_x,s}$ and $\boldsymbol{e}_{i_y,s}$.

We then compute the embedding sensitivity magnitude $\boldsymbol{\Delta}_{u_x}/\boldsymbol{\Delta}_{i_y}$ by using l_2 distance as defined in Eq. 5. Alternative distance measures, such as KL divergence, is left for future work.

$$\boldsymbol{\Delta}_{u_x} = \frac{1}{S}\sum_{1}^{S} \left\| \boldsymbol{e}_{u_x} - \boldsymbol{e}_{u_x,s} \right\|_2 \qquad (5)$$
$$\boldsymbol{\Delta}_{i_y} = \frac{1}{S}\sum_{1}^{S} \left\| \boldsymbol{e}_{i_y} - \boldsymbol{e}_{i_y,s} \right\|_2$$

We maintain dynamically updated risk scores r_{u_x} and r_{i_y}. After computing embedding sensitivity magnitude, we update risks using exponential moving average, as Eq. 6.

$$r_{u_x} \leftarrow \boldsymbol{\alpha} r_{u_x} + (1-\boldsymbol{\alpha})\boldsymbol{\Delta}_{u_x} \qquad (6)$$
$$r_{i_y} \leftarrow \boldsymbol{\alpha} r_{i_y} + (1-\boldsymbol{\alpha})\boldsymbol{\Delta}_{i_y}$$

, where $\boldsymbol{\alpha}$ ranges from 0 to 1, controlling the smoothing strength.

4.4. Preference Scoring

Given a user $\boldsymbol{u_x}$ and an item $\boldsymbol{i_y}$ along with their embeddings $\boldsymbol{e}_{u_x}$ and $\boldsymbol{e}_{i_y}$, we compute the preference score using the inner product, which is widely adopted in GCN-based recommendation systems[30] [2], as Eq. 7.

$$\widehat{r_{u_x,i_y}} = \boldsymbol{e}_{u_x}{}^{T} \boldsymbol{e}_{i_y} \qquad (7)$$

We return the top-K items with the highest preference scores from the candidate set as the recommendation list for the user.

4.5. Model training

Pairwise ranking objectives are commonly employed in implicit-feedback recommendation scenarios. Among them, the Bayesian Personalized Ranking (BPR) loss has been extensively used due to its effectiveness in learning personalized ranking signals from implicit interactions [9] [23] [22] [33]. The BPR loss adopts a pairwise optimization strategy by sampling negative items for each observed

user–item interaction and encourages the model to assign higher preference scores to observed (positive) items than to unobserved (negative) ones. Instead of drawing negative items solely from the set of unobserved interactions, we further refine the negative sampling process by excluding items that appear in the candidate set C_{u_x}, same as Mo et al. [10]. The resulting training objective is defined as Eq. 8.

$$L_{BPR} = -\sum_{(u_x, i_y, i_{y'})\epsilon D} ln\sigma\left(\widehat{r_{u_x,i_y}} - \widehat{r_{u_x,i_{y'}}}\right) + \mu\|\boldsymbol{\omega}\|^2 \quad (8)$$

, where $D = \{(u_x, i_y, i_{y'})|i_y \epsilon N_{u_x}, i_{y'} \epsilon N - N_{u_x} - C_{u_x}\}$. The parameter set $\boldsymbol{\omega}$ includes all learnable variables of the model, while μ controls the magnitude of the l_2 regularization term used to alleviate overfitting. Other loss terms, such as regularizing the embedding sensitivity magnitude $\boldsymbol{\Delta}_{u_x}/\boldsymbol{\Delta}_{i_y}$, are left for future work.

5. Preliminary Experiment

5.1. Dataset

We conduct experiments on the ML-1M dataset[24] [31], which contains 6,040 users, 3,952 items and 1,000,209 ratings. Data preprocessing is performed by ARLib [24]. The dataset is randomly split into 70% for training, 10% for tuning, and 20% for testing. We directly use the preprocessed datasets provided by ARLib. The dataset statistics are summarized in Table 2.

5.2. Base Recommender System

We choose LightGCN[2], a representative and widely adopted recommender, as the backbone model.

5.3. Attack and Defense Method

With the help of ARLib, we conducted preliminary experiments to validate the effectiveness of the proposed method under random attacks. We used the recommendation results of the base LightGCN model as the baseline for comparison.

5.4. Hyperparameter setting

Our method involves several important hyperparameters, including the size of the candidate set generated by the base recommender (top-L), the exponent β that controls the strength of risk-aware aggregation, the number of masking operations S, and the risk update exponent α. We set the candidate list size top-L to be twice the final recommendation length, i.e., top-L =40. β is set to 0.7. The number of counterfactual masking operations S is set to 3. The risk update smoothing exponent α is set to 0.9.

5.5. Metric

Following You et al [5], we recommend 20 items for each user. To evaluate recommendation accuracy, we adopt commonly used metrics, including Recall@20, and NDCG@20. Comparing recommendation accuracy with the base recommender validates the effectiveness of our method. Besides, to directly evaluate anti-shilling effectiveness, we measure the Target Item Exposure (TIE), defined as the number of times the target item appears in users' recommendation lists, where a lower value indicates better defense performance.

In this preliminary study, we compare our method with the base recommender in terms of recommendation accuracy. The evaluation with additional metrics TIE, more datasets, additional recommender models (e.g., NCF [1], NFCG[25], NCL [26], SimGCL[27]), stronger attack strategies (e.g., DLAttack[28], GOAT[14], Pipattack[29]), comparisons with state-of-the-art anti-shilling defenses(e.g., Anti-fakeu [5], Trust-GRS[6]), and directly applying ConsisRec to the base recommender model are left for future work. Before launching the attack, we randomly select five unpopular items as target items and generate fake users accounting for 1% of the number of normal users.

5.6. Experimental Results

In this section, we conduct a comparative evaluation between the proposed method and baseline methods. Table 3 presents the experimental results.

5.7. Comparison between ConsisRec with baselines

Compare LightGCN+RandomAttack with LightGCN, under random attack settings, the performance of the base recommender is affected by maliciously injected interactions. By contrast, even in the presence of shilling attacks, ConsisRec achieves notable improvements over LightGCN, yielding relative gains of 2.11% in Recall and 2.35% in NDCG. These results demonstrate that ConsisRec can naturally suppress anomalous recommendations introduced by shilling attacks, while preserving and even enhancing recommendation accuracy. Notably, ConsisRec treats the base recommender as a black box and relies solely on the candidate sets it generates, without modifying the internal structure or parameters of the base model, which ensures the proposed framework with high generality and makes it readily applicable to various recommender systems.

Table 2: Dataset Statistics

Dataset	#user	#item	#interaction	sparsity
MovieLens 1M	6,040	3,952	1,000,209	95.81%

Table 3: Experimental Results on MovieLens

Model	Recall	NDCG
LightGCN	0.2563	0.1996
LightGCN +RandomAttack	0.2545	0.1992
LightGCN +RandomAttack +ConsisRec	0.2617 +2.11%	0.2043 +2.35%

5.8. Comparison between ConsisRec with Anti-fakeu

You et al. [5] reported the Recall and NDCG of the base recommender and Anti-fakeu under shilling attack scenarios. The results indicate that Anti-fakeu degrades recommendation performance, leading to lower accuracy compared with state-of-the-art methods. In contrast, our method achieves even higher recommendation performance than the recommender trained without shilling attacks, highlighting the ability of our noise-filtering–based approach to improve recommendation accuracy.

6. Conclusion and Future Work

In this paper, we propose ConsisRec, a scalable and model-agnostic post-processing framework for anti-shilling recommendation. Unlike existing anti-shilling methods that are tightly coupled with specific recommendation models or training procedures, ConsisRec operates purely on the output recommendation lists of base recommenders, enabling flexible deployment without modifying model architectures or retraining processes. Preliminary experiments show that ConsisRec improves recommendation accuracy under random shilling attacks, achieving performance gains over the base LightGCN model. There are several promising directions for future work. First, we plan to evaluate ConsisRec on more datasets and additional backbone recommender models, including directly applying ConsisRec to the base recommender model, to further verify its generality. Second, we will extend the evaluation to stronger and more sophisticated shilling attack strategies and conduct systematic comparisons with state-of-the-art anti-shilling defense methods. Third, we will incorporate direct anti-shilling metrics, such as Target Item Exposure (TIE), into comprehensive evaluations to better quantify defensive effectiveness.

7. Acknowledgments

This work is supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant Number 25K21304.

References

[1] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182).

[2] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020, July). 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* (pp. 639-648).

[3] Liu, F., Cheng, Z., Zhu, L., Gao, Z., & Nie, L. (2021, April). Interest-aware message-passing GCN for recommendation. In *Proceedings of the web conference 2021* (pp. 1296-1305).

[4] Zhang, S., Yin, H., Chen, T., Hung, Q. V. N., Huang, Z., & Cui, L. (2020, July). 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* (pp. 689-698).

[5] You, X., Li, C., Ding, D., Zhang, M., Feng, F., Pan, X., & Yang, M. (2023, April). Anti-fakeu: Defending shilling attacks on graph neural network based recommender model. In *Proceedings of the ACM web conference 2023* (pp. 938-948).

[6] Mu, L., Liu, Z., Zhu, Z., & Lin, Z. (2025, April). Trust-GRS: A Trustworthy Training Framework for Graph Neural Network Based Recommender Systems Against Shilling Attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 39, No. 12, pp. 12408-12416).

[7] Tang, J., Wen, H., & Wang, K. (2020, September). Revisiting adversarially learned injection attacks against recommender systems. In Proceedings of the 14th ACM Conference on Recommender Systems (pp. 318-327).

[8] Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., ... & Papernot, N. (2021, May). Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)* (pp. 141-159). IEEE.

[9] Mo, F., & Yamana, H. (2023). EPT-GCN: Edge propagation-based time-aware graph convolution network for POI recommendation. *Neurocomputing*, *543*, 126272.

[10] Mo, F., Fan, X., Chen, C., & Yamana, H. (2025). Synergistic fusion framework: Integrating training and non-training processes for accelerated graph convolution network-based recommendation. *Pattern Recognition*, 111829.

[11] Huang, H., Mu, J., Gong, N. Z., Li, Q., Liu, B., & Xu, M. (2021). Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644*.

[12] Lam, S. K., & Riedl, J. (2004, May). Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web* (pp. 393-402).

[13] Song, J., Li, Z., Hu, Z., Wu, Y., Li, Z., Li, J., & Gao, J. (2020, April). Poisonrec: an adaptive data poisoning framework for attacking black-box recommender systems. In *2020 IEEE 36th international conference on data engineering*

(ICDE) (pp. 157-168). IEEE.

[14] Wu, F., Gao, M., Yu, J., Wang, Z., Liu, K., & Wang, X. (2021). Ready for emerging threats to recommender systems? A graph convolution-based generative shilling attack. *Information Sciences*, *578*, 683-701.

[15] Zhang, S., Yin, H., Chen, T., Hung, Q. V. N., Huang, Z., & Cui, L. (2020, July). 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* (pp. 689-698).

[16] Burke, R., Mobasher, B., Williams, C., & Bhaumik, R. (2006, August). Classification features for attack detection in collaborative recommender systems. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 542-547).

[17] Bhaumik, R., Mobasher, B., & Burke, R. (2011). A clustering approach to unsupervised attack detection in collaborative recommender systems. In *Proceedings of the International Conference on Data Science (ICDATA)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[18] Wu, Z., Wu, J., Cao, J., & Tao, D. (2012, August). 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* (pp. 985-993).

[19] Zhang, Y., Hao, Q., Zheng, W., & Xiao, Y. (2025). User similarity-based graph convolutional neural network for shilling attack detection. *Applied Intelligence*, *55*(5), 340.

[20] Li, H., Gao, M., Zhou, F., Wang, Y., Fan, Q., & Yang, L. (2021). Fusing hypergraph spectral features for shilling attack detection. *Journal of information security and applications*, *63*, 103051.

[21] Hao, Y., Meng, G., Wang, J., & Zong, C. (2023). A detection method for hybrid attacks in recommender systems. *Information Systems*, *114*, 102154.

[22] Mo, F., Fan, X., Chen, C., Bai, C., & Yamana, H. (2024). Sampling-based epoch differentiation calibrated graph convolution network for point-of-interest recommendation. *Neurocomputing*, *571*, 127140.

[23] Zhou, X., Lin, D., Liu, Y., and Miao, C. 2023. Layer-refined graph convolutional networks for recommendation. In Proceedings of the 2023 IEEE 39th International Conference on Data Engineering, pp. 1247-1259.

[24] Wang, Z., Yu, J., Gao, M., Yuan, W., Ye, G., Sadiq, S. W., & Yin, H. (2024). Poisoning Attacks and Defenses in Recommender Systems: A Survey. *CoRR*.

[25] Wang, X., He, X., Wang, M., Feng, F., & Chua, T. S. (2019, July). Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 165-174).

[26] Lin, Z., Tian, C., Hou, Y., & Zhao, W. X. (2022, April). Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022* (pp. 2320-2329).

[27] Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., & Nguyen, Q. V. H. (2022, July). 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* (pp. 1294-1303).

[28] Huang, H., Mu, J., Gong, N. Z., Li, Q., Liu, B., & Xu, M. (2021). Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644*.

[29] Zhang, S., Yin, H., Chen, T., Huang, Z., Nguyen, Q. V. H., & Cui, L. (2022, February). Pipattack: Poisoning federated recommender systems for manipulating item promotion. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (pp. 1415-1423).

[30] Mo, F., & Yamana, H. (2022, November). GN-GCN: Combining geographical neighbor concept with graph convolution network for POI recommendation. In *International conference on information integration and web* (pp. 153-165). Cham: Springer Nature Switzerland.

[31] Fang, M., Yang, G., Gong, N. Z., & Liu, J. (2018, December). Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference* (pp. 381-392).

[32] Chen, C., Mo, F., Fan, X., Bai, C., & Yamana, H. (2023, March). Mobarec-gcnfp: Champion recommendation for multi-player online battle arena games using graph convolution network with fewer parameters. In *2023 IEEE 8th International Conference on Big Data Analytics (ICBDA)* (pp. 147-153). IEEE.

[33] Fan, X., Mo, F., Chen, C., Bai, C., & Yamana, H. (2024, March). Connectivity-Aware Experience Replay for Graph Convolution Network-Based Collaborative Filtering in Incremental Setting. In *2024 9th International Conference on Big Data Analytics (ICBDA)* (pp. 233-242). IEEE.