Interaction-level Membership Inference Attack against Recommender Systems with Long-tailed Distribution

Da Zhong
Stevens Institute of Technology
Hoboken, New Jersey, USA
dzhong2@stevens.edu

Xiuling Wang
Stevens Institute of Technology
Hoboken, New Jersey, USA
xwang193@stevens.edu

Zhichao Xu
University of Utah
Salt Lake City, Utah, USA
zhichao.xu@utah.edu

Jun Xu
University of Utah
Salt Lake City, Utah, USA
junxzm@cs.utah.edu

Wendy Hui Wang
Stevens Institute of Technology
Hoboken, New Jersey, USA
hwang4@stevens.edu

Abstract

Recommender systems (RSs) are susceptible to *Interaction-level Membership Inference Attacks* (IMIAs), which aim to determine whether specific user-item interactions are present in the training data of the target RS. However, existing IMIAs struggle with inferring the membership of tail interactions, i.e., the interactions involving tail items, due to the limited information available about these items. This paper introduces MINER, a new IMIA designed to enhance attack performance against RSs with long-tailed item distribution. To address the scarcity issue of tail items, first, MINER leverages the *Knowledge Graphs* (KGs) to obtain the auxiliary knowledge of tail items. Second, MINER leverages a *Bilateral-Branch Network* (BBN) to initially learn from the head interactions and gradually shift attention to tail interactions. The BBN trains two branches independently, with one branch trained on interaction samples with the original long-tailed item distribution and the other on interaction samples with a more balanced item distribution. The outputs of the two branches are aggregated using a cumulative learning component. Our experimental results demonstrate that MINER significantly enhances the attack accuracy of IMIA, especially for tail interactions. Beyond attack design, we design a defense mechanism named RGL to defend against MINER. Empirical evaluations demonstrate that RGL effectively mitigates the privacy risks posed by MINER while preserving recommendation accuracy. Our code is available at https://github.com/dzhong2/MINER.

CCS Concepts

• **Security and privacy**;

Keywords

Membership inference attack; Recommender system; Long-tailed distribution; Privacy of machine learning

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679804

ACM Reference Format:
Da Zhong, Xiuling Wang, Zhichao Xu, Jun Xu, and Wendy Hui Wang. 2024. Interaction-level Membership Inference Attack against Recommender Systems with Long-tailed Distribution. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679804

1 Introduction

Recommender systems (RSs) suggest new items to users by learning user preferences from historical interactions [36]. Recently, RSs have been widely used across various domains such as finance, healthcare, and education.

Despite their effectiveness, RSs are vulnerable to the *Membership Inference Attacks* (MIAs) [37]. The existing MIAs on RSs fall into two categories: (i) *User-level MIA* (**UMIA**) infers whether the interactions of a specific user were used by the target RS for training [41, 46]; (ii) *Interaction-level MIA* (**IMIA**) [44] infers if a specific user-item interaction was present in the training data.

This paper primarily focuses on IMIA. Specifically, we follow [46] and consider an adversary who has black-box access to the target RS, i.e., he can access the top-k items recommended to the users. However, the adversary does not know the respective preference scores. The adversary's goal is to infer if the interaction between a specific user u and a particular item i exists in the training data of the target RS system, based on u's top-k recommendations. The user-item pair (u, i) is called a *member* if it exists in the training data and a *non-member* otherwise.

Although the existing IMIA [44] has demonstrated its effectiveness, it suffers from two significant drawbacks: (i) It relies on the impractical assumption that the adversary possesses white-box access to the target RS (i.e., the adversary can access the model parameters); (ii) As will be shown in our empirical study (Sec. 5), it exhibits poor performance on the inference of *tail interactions*, i.e., the interactions involving *tail items* (e.g., unpopular or new products). However, the tail interaction can unveil more sensitive information than the head ones. For example, consider a healthcare recommender system. Determining whether a patient has been treated ("interacted") with HIV (a rare disease) carries greater sensitivity than discerning whether the user has been treated with flu, a common disease. Given the prevalence of long-tailed distributions

Table 1: Comparison between the existing MIA attacks against recommender systems and our work.

Work	Adversary access	KG	Long-tailed distribution	MIA
[41, 46]	Black-box	✗	✗	User-level
[44]	White-box	✗	✗	Interaction-level
MINER (Ours)	Black-box	✓	✓	

in recommender systems [34, 43], it is crucial to understand the privacy vulnerabilities of tail interactions against IMIAs.

Attack design. We introduce MINER, a new Interaction-level Membership INferencE Attack tailored for Recommender Systems with long-tail item distribution. Table 1 summarizes the key difference between MINER and the prior works. Intuitively, it is challenging to infer the membership of tail interactions due to their scarcity in the data. To tackle the scarcity issue, first, MINER leverages *Knowledge Graphs* (KGs), which can be publicly accessible,[1] as external sources to harness auxiliary information of items, especially the tail ones. Second, MINER includes an attack model that utilizes a *Bilateral-Branch Network* (BBN) [50] comprising two branches: the "main branch" and the "regularizer branch", for membership inference. The two branches are trained with the samples whose attack features are derived from the head and tail interactions, respectively. Specifically, the samples for the main branch were obtained by employing a uniform sampler over the original long-tailed distribution, while the samples for the regularizer branch were obtained by utilizing a re-balanced sampler from a more balanced item distribution. The bilateral branches are trained independently. Their outputs are aggregated using a cumulative learning model with an adaptive parameter that controls the attack model to initially learn from the head interactions and gradually shift attention to tail interactions.

Attack evaluation. We conduct an extensive set of experiments to evaluate the performance of MINER on three mainstream KG-based RSs and three real-world RS datasets. The empirical evaluation demonstrates the effectiveness of MINER, achieving a consistent attack accuracy of around 0.8. Notably, the attack accuracy of head and tail interactions can be as high as 0.852 and 0.779, respectively, demonstrating a remarkable increase of 35.5% in overall attack accuracy and a 44.4% improvement in the inference accuracy of tail interactions compared with the existing attacks [44, 46].

Defense. We design RGL to mitigate the privacy risk of MINER. RGL incorporates a regularizer term, which penalizes the target model's loss with the attack model's ability to distinguish between member and non-member interactions, with the loss function of the target model. As RGL does not have access to the attack model, it trains a *surrogate attack model* to mimic MINER's attack behaviors. The empirical evaluation demonstrates that RGL is effective in protecting the recommender systems against MINER while preserving the recommendation accuracy, even when the surrogate attack model has a different architecture from that of MINER.

[1] Many large knowledge graphs, e.g., DBpedia, Wikidata, WordNet, and Geonames, are openly available.

2 Preliminaries

Consider $\mathcal{U}$ and $\mathcal{I}$ as the sets of users and items, respectively. The user-item interactions are represented as a binary matrix $I \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, where $I_{u,i} = 1$ indicates user u being interacted with item i, and $I_{u,i} = 0$ otherwise. Recommender systems (RSs) aim to recommend items to each user based on their interactions. In the following discussions, we use $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$ to denote the user u's recommendations.

Interaction graph. The historical user-item interactions take the form of an *interaction graph* (denoted as $\mathcal{IG}$), in which each node corresponds to either a user or an item, and each edge corresponds to a historical user-item interaction.

Knowledge graph. Knowledge graphs (KGs) represent entities as nodes and relations between entities as edges [5, 16, 42, 45]. A KG with an entity set $\mathcal{E}$ and a relation set $\mathcal{R}$ can be formally defined as $\mathcal{KG} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where each triple (h, r, t) denotes a fact of the relation r between the entities h and t.

Head and tail interactions. In many real-world recommender systems, user-item interactions typically follow a long-tailed distribution, wherein a small fraction of items (*head items*) are popular and attract the majority of user interactions. Meanwhile, the remaining items (*tail items*) are seldom interacted with by users [34, 43, 48]. The itemset $\mathcal{I}$ can be partitioned into the head **H** and the tail **T** by selecting a cutting point ζ [34]. In this paper, we adhere to the convention in the literature [34, 43] and choose $\zeta = 20\%$, meaning that 20% of items with the highest frequency constitute head items, while the remaining 80% are considered tail items. An interaction that involves a head item is referred to as a *head interaction*; otherwise, it is a *tail interaction*.

3 Problem Formulation

Adversary knowledge. Following the prior work [46], we assume that the adversary possesses *restricted black-box access* to the target model $\mathcal{M}$ by which the adversary can access the top-k items $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$ recommended to the target user u by $\mathcal{M}$. We consider two scenarios regarding $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$: (1) *with ranking*, where the adversary is aware of the ranking of the items in $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$; and (2) *without ranking*, wherein the adversary only possesses a list of recommended items but lacks knowledge of their ranking. In both scenarios, the adversary cannot access the preference scores of any recommended items, which aligns with real-world RSs such as Amazon and Netflix [46].

Besides access to the top-k recommendations, we assume that the adversary possesses knowledge of a *shadow knowledge graph* (denoted as $\mathcal{KG}^S$), from which the adversary can extract side information (such as features) of items. The adversary can obtain $\mathcal{KG}^S$ through the public repositories of knowledge graphs[2] or by crawling on open platforms which are provided by many real-world RSs (e.g., Amazon and Netflix). $\mathcal{KG}^S$ may belong to different domains and exhibit different distributions compared to the knowledge graph $\mathcal{KG}$ used by the target model.

In addition to the shadow knowledge graph, the adversary may have access to a *shadow interaction graph* (denoted as $\mathcal{IG}^S$), which comprises a collection of user-item interactions[3]. $\mathcal{IG}^S$ and $\mathcal{IG}$

[2] As mentioned in Note 1.

[3] Many data repositories for recommender systems are available online (e.g., https://github.com/caserec/Datasets-for-Recommender-Systems).

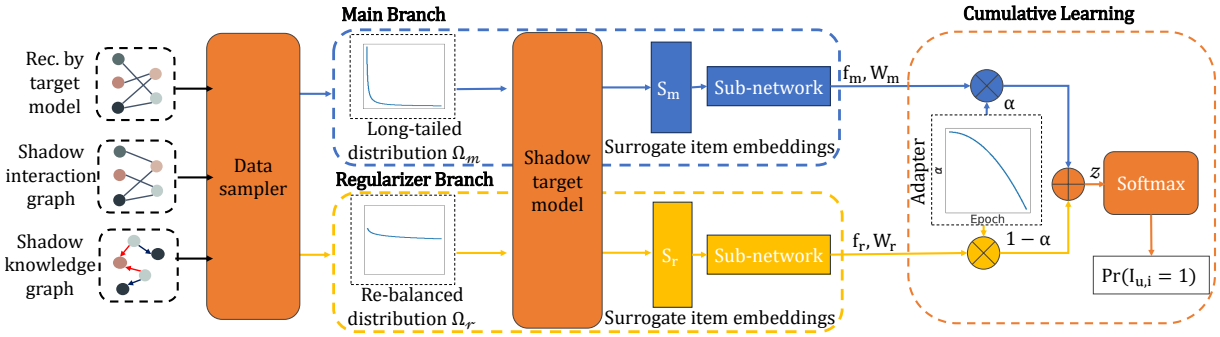


Figure 1: The framework of MINER.

can originate from distinct domains and distributions and do not necessarily share nodes and edges.

Problem definition. Given a knowledge graph $\mathcal{KG}$, an interaction graph $\mathcal{IG}$ and its corresponding interactin matrix I, and a target RS $\mathcal{M}$ (referred to as the target model) trained on both $\mathcal{KG}$ and $\mathcal{IG}$, the adversary seeks to infer whether there is an interaction between a user u and an item i in $\mathcal{IG}$ based on u's recommendations $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$ and the adversary knowledge which consists of the shadow knowledge graph $\mathcal{KG}^S$ and the shadow interaction graph $\mathcal{IG}^S$. The problem can be formulated as the design of a binary classifier $\mathcal{A}$ that predicts the membership label y,

$$\mathcal{A} : u, i, \hat{\boldsymbol{r}}_{\boldsymbol{u}}, \mathcal{IG}^S, \mathcal{KG}^S \rightarrow y,$$

where $y = 1$ if $\mathcal{A}$ predicts $I_{u,i} = 1$, and $y = 0$ otherwise.

4 MINER: Our Attack

In this section, we present the details of MINER. Figure 1 illustrates the framework of MINER. It consists of four components: (1) training of *knowledge-enhanced shadow target models*; (2) derivation of attack features; (3) training of the attack model; and (4) attack inference. Specifically, our attack model consists of two branches: the "main branch" and the main branch "regularizer branch". The main branch is trained on the samples with the original long-tailed item distributions. In contrast, the regularizer branch is trained on samples derived from a relatively balanced item distribution. Each branch undergoes independent classifier training, enabling each to learn the unique knowledge of inferring head and tail interactions, respectively. Additionally, MINER incorporates a "cumulative learning" component to aggregate the predicted outputs of the bilateral branches using an "α-adaptor" to control the attack classifier to emphasize learning from the head interactions initially and gradually shift focus to the tail interactions. Next, we provide the details of these components.

4.1 Training Knowledge-enhanced Shadow Target Model

Inspired by the existing MIAs against RSs [44, 46], MINER relies on the similarity between the embeddings of interacted items and recommended ones for inference. As these item embeddings are not available to the adversary who only has black-box access to the target RS, we adopt the *shadow model* approach used in prior MIA research [37, 46] to learn item embeddings. As the embeddings of tail items generated by the shadow model may be of poor quality due to the information scarcity of these items, we leverage knowledge graphs as external sources to harness auxiliary information and enhance the quality of item embeddings, particularly for tail items.

Specifically, the adversary first constructs a shadow training graph G_{shadow}, which includes three components: (1) the shadow knowledge graph $\mathcal{KG}^S$, (2) the shadow interaction graph $\mathcal{IG}^S$, and (3) the recommendations $\hat{R}$ made by the target RS $\mathcal{M}$ obtained through black-box access to $\mathcal{M}$. Including the shadow knowledge graph $\mathcal{KG}^S$ in G_{shadow} enhances the generated item embeddings by aggregating items and their features in $\mathcal{KG}^S$, while including $\hat{R}$ in G_{shadow} ensures that $\mathcal{M}^S$ emulates the behaviors of $\mathcal{M}$. Next, the adversary trains a shadow model $\mathcal{M}^S$ on G_{shadow}. The architecture of $\mathcal{M}^S$ can differ from that of $\mathcal{M}$. Finally, the adversary uses the item embeddings generated by $\mathcal{M}^S$, referred to as *surrogate item embeddings*, to derive the attack features of MINER.

4.2 Deriving Attack Features

In general, recommender systems recommend items similar to those that users have interacted with in the past. MINER exploits this fact to derive the attack features from the similarity between the interacted items and the recommended ones. Specifically, let $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$ be the top-k recommendations of a given user u. We consider two scenarios of $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$:

- **Case 1. Recommendations with ranking**: All items in $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$ are ranked by their preference scores.
- **Case 2. Recommendations without ranking**: The items in $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$ are not ranked.

For Case 1, we compute the *discounted similarity* (denoted as $DS(i, i')$) between a given item i and any item $i' \in \hat{\boldsymbol{r}}_{\boldsymbol{u}}$:

$$DS(i, i') = \frac{d(e_i, e_{i'})}{log_2(r_{i'} + 1)}, \tag{1}$$

where $d()$ is a distance function, e_i and $e_{i'}$ are the embeddings of i and i', respectively, and $r_{i'}$ is the ranking of i' in $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$. Here the embeddings of i and i' are the surrogate embeddings generated by the shadow target model $\mathcal{M}^S$. The rationale behind using $\frac{1}{log_2(r_{i'}+1)}$ as a weight which is logarithmically inversely proportional to its ranking, is rooted in the intuition that higher-ranked items hold

more significance than lower-ranked ones. Therefore, assigning greater weight to the similarity with items of higher ranking underscores their increased importance in the inference. The DS metric can be readily adapted to Case 2 where the ranking of items in $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$ is not available by assigning all items with the same ranking as 1, leading to $\frac{1}{log_2(r_{i'}+1)} = 1$ for all items.

To capture different aspects of item similarity and provide a comprehensive evaluation of item similarity for inference, we consider multiple distance functions $d_1(), \ldots, d_t()$ for Eqn. (1). Specifically, we consider four distance functions ($t = 4$): the L1 distance, L2 distance, Cosine distance, and Bray-Curtis distance [7]. This will lead to t similarity values for each item pair (i, i'). We concatenate these t values and form the *similarity vector* $\boldsymbol{s}_{\boldsymbol{i},\boldsymbol{i'}}$ of (i, i') as follows:

$$\boldsymbol{s}_{\boldsymbol{i},\boldsymbol{i'}} =< DS_1(i, i')|| \ldots ||DS_t(i, i') > \tag{2}$$

where $DS_j(i, i')$ is the discounted similarity of i and i' (Eqn. (1)) measured by the j-th similarity function.

Based on the similarity vector, given a user-item pair (u, i), its attack feature (denoted as $\mathbf{x}$) of (u, i) is derived as follows:

$$\mathbf{x} = \forall_{i' \in \hat{\boldsymbol{r}}_{\boldsymbol{u}}} || \boldsymbol{s}_{\boldsymbol{i},\boldsymbol{i'}} \tag{3}$$

The length of the feature $\mathbf{x}$ is $t \times k$, where t is the number of similarity functions, and k is the number of recommendations in $\hat{\boldsymbol{r}}_{\boldsymbol{u}}$.

4.3 Training Bilateral Branches

To address the scarcity issue of tail items at the sample level, we design MINER as a bilateral branch network that consists of two branches: A *main branch* trained on the long-tailed distribution, and a *regularizer branch* trained on a relatively balanced distribution. Next, we explain the details of the bilateral-branch structure.

Sub-network structure. Each branch trains a sub-network independently. In this paper, we use Multi-Layer Perceptron (MLP) for both sub-networks. Both sub-networks share the same MLP architecture, including the number of layers and the number of neurons per layer. However, they do not share the parameters such as weights and biases. This design choice allows the two branches to learn distinct knowledge for head and tail interactions respectively.

Data samplers. Both branches perform their training on different samples respectively. Specifically, the main branch utilizes a uniform sampler that retains the long-tailed item distribution (denoted as Ω_m), while the regularizer branch employs a re-balanced sampler to generate a set of interactions with a relatively balanced item distribution (denoted as Ω_r). Ω_m emphasizes the head interactions by following the original long-tailed distribution. Ω_r, on the other hand, allocates more attention to the tail interactions by down-sampling the head interactions.

Concretely, the uniform sampler draws a sample with an equal probability $P = 1/N$, where N is the number of interactions in the training set. For the re-balanced sampler, we categorize the interactions into two classes ($K = 2$): the *head interaction class* and the *tail-interaction class*. The re-balanced draws a sample from the class j with a probability P_j computed as follows:

$$P_j = \frac{w_j}{\sum_{j=1}^{K} w_j}, \quad w_j = (\frac{N_{max}}{N_j})^{\frac{1}{\beta}}. \tag{4}$$

where β adjusts the reversed distribution of tail interactions, N_j is the sample size of class j, and N_{max} is the maximum sample size for all the classes. As the data follows the long-tailed distribution, $N_{max} = N_H$ (i.e., the number of head interactions), and thus $w_H = 1$. As β increases, the re-balancing effect increases. We choose β to be sufficiently large so that $w_T \approx w_H$.

Training data for each branch. First, we sample two sets of interactions, denoted as I_m and I_r, from the shadow interaction graph $\mathcal{IG}^S$ by employing the uniform sampler and the re-balanced sampler, respectively. Then we feed I_m and I_r to the shadow RS $\mathcal{M}^S$ and obtain the surrogate item embeddings (denoted as $\boldsymbol{S}_m$ and $\boldsymbol{S}_r$), respectively. Next, from I_m, we sample a set of user-item pairs $\{u, i\}$, with 50% of them being interacted (members) in I_m, and the remaining 50% not interacted (non-members). For each sampled user-item pair, we generate a corresponding sample (x, y) where x is derived from $\boldsymbol{S}_m$ by following Eqn. (3), and y is the membership label (0/1) of (u, i) in I_m. We use these samples to train the sub-network of the main branch. Similarly, we generate the training dataset from I_r and use it to train the sub-network of the regularizer branch. The two sub-networks thus are trained simultaneously on two different datasets.

4.4 Cumulative Learning

To bridge the gap between head and tail interactions, we fuse the decoupled information from both branches through a cumulative learning component that utilizes a α-adapter. By adjusting the hyperparameter α in the adapter, we can shift the training attention from head interactions to tail interactions in a soft and flexible way.

Specifically, let $\boldsymbol{f}_m$ and $\boldsymbol{f}_r$ be the feature vectors of the two sub-networks, and $\boldsymbol{W}_m$ and $\boldsymbol{W}_r$ be the weights of the sub-networks respectively. We integrate the two branches by controlling the weights for $\boldsymbol{f}_m$ and $\boldsymbol{f}_r$ with an adaptive trade-off parameter α. The predicted logits are formulated as follows:

$$z = \alpha \boldsymbol{W}_m^{\mathsf{T}} \boldsymbol{f}_m + (1 - \alpha) \boldsymbol{W}_r^{\mathsf{T}} \boldsymbol{f}_r, \tag{5}$$

where α is a γ-adaptor, which is a function of the training epoch t:

$$\alpha = 1 - (\frac{t}{\gamma \times T})^2 \tag{6}$$

Here $\gamma > 1$ is the regularizer rate, and T is the total number of training epochs. Thus α will gradually decrease as the training progresses. With α decreasing, MINER turns the emphasis from the main branch to the regularizer branch, and thus shifts the attention from head interactions to tail interactions.

With the logits z calculated by Eqn. (5), we calculate the probability that the user-item pair (u, i) is a member/non-member through a softmax function:

$$\hat{p}(y = y_j) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad for\ j = 1, 2 \tag{7}$$

where y_1=1 (member) and y_2=2 (non-member).

Finally, we formalize the loss function as the aggregation over the loss of the two branches:

$$\mathcal{L}_{total} = \alpha \mathcal{L}(\hat{\mathbf{p}}, y_m) + (1 - \alpha) \mathcal{L}(\hat{\mathbf{p}}, y_r), \tag{8}$$

where $\mathcal{L}$ denotes the cross-entropy loss, $\hat{\mathbf{p}}$ is the output probability distribution (Eqn. (7)), and y_m and y_r denote the ground-truth membership labels of the samples in I_m and I_r, respectively.

Table 2: Description of the three datasets.

Statistics		Book	LastFM	Yelp
Interaction Graph	# of users	70,679	23,566	45,919
	# of items	24,915	48,123	45,538
	# of interactions	742,730	1,474,722	1,059,575
Knowledge Graph	# of entities	88,572	58,266	90,961
	# of relations	39	9	42
	# of triplets	2,557,746	464,567	1,853,704

4.5 Inference

During the inference phase, MINER feeds the target user-item pair to the trained attack classifier for inference. In particular, the target user-item pair is fed into both branches of the attack classifier. As both branches are equally important during testing, we simply fix α to 0.5 in the test phase. Then, the features are fed to their corresponding branches to obtain the prediction logits. Finally, the logits are aggregated using equal weights to obtain the membership inference result of the target user-item pair.

5 Evaluation

This section presents the results of our empirical evaluation addressing three research questions:

- **RQ$_1$**: How effective is MINER against the representative RSs?
- **RQ$_2$**: How do various factors influence the performance of MINER.
  RQ$_3$: Do the different components of MINER necessarily boost the effectiveness of MINER?

5.1 Experimental Setup

All the experiments are performed on a server with eight NVIDIA A100 GPUs. The algorithms are implemented in Python. Each experiment is repeated five times and the average results are reported.

Target recommender systems and datasets. We employ three representative KG-based recommender systems, namely KGAT [40], CKE [45], and ECFKG [5]. We set the embedding length as 64. We use a 3-layer KGAT model, with 64, 32, and 16 nodes at Layer 1, 2, and 3 respectively. We follow the same parameter settings of CKE and ECFKG as in [45] and [5], respectively. We conducted experiments on three real-world datasets: Amazon Book dataset (**Book**) [1], LastFM dataset (**LastFM**) [2], and Yelp18 dataset (**Yelp**) [3]. Table 2 summarizes the details of the three datasets.

Attack training and testing data. The attack training dataset consists of all the interactions in the shadow training graph as members. We also randomly sample a number of non-interacted user-item pairs from the shadow training graph, and include them in the attack training dataset as non-members. Regarding the attack testing data, we randomly sample 10% of interactions from the target graph as the members and a set of non-interacted user-item pairs in the target graph as the non-members. Both training and testing datasets are balanced between the members and non-members as well as between head and tail interactions. The head and tail interactions are evenly split between members and non-members.

Evaluation metrics. We consider two types of metrics to evaluate attack performance: (1) *Attack accuracy* (**ACC**) is measured as the fraction of samples in the attack testing data that are correctly inferred (either as members or non-members): (2) *True Positive Rate at Low False Positive Rate* (TPR@FPR) [8]: we measure TPR@5%FPR in this paper. Regarding target model accuracy, we measure the hit ratio of the top-k recommendations[14, 24, 42] (**HR@K**), i.e., the percentage of the ground-truth items that are included in the top-k recommendations.

Table 3: Attack accuracy of MINER and baselines under the non-transfer setting (Book dataset, γ=2). The highest attack accuracy is marked with green.

Model	Attack	ACC			TPR@5%FPR		
		All	Head	Tail	All	Head	Tail
CKE	MINER	0.82	0.85	0.78	0.15	0.18	0.14
	MIARS	0.70	0.86	0.54	0.10	0.18	0.01
	WBI	0.61	0.64	0.57	0.07	0.07	0.05
KGAT	MINER	0.82	0.87	0.76	0.16	0.18	0.15
	MIARS	0.70	0.88	0.52	0.10	0.19	0.02
	WBI	0.65	0.66	0.63	0.05	0.07	0.03
ECFKG	MINER	0.81	0.85	0.76	0.17	0.19	0.14
	MIARS	0.75	0.88	0.61	0.09	0.20	0.02
	WBI	0.63	0.63	0.62	0.06	0.09	0.04

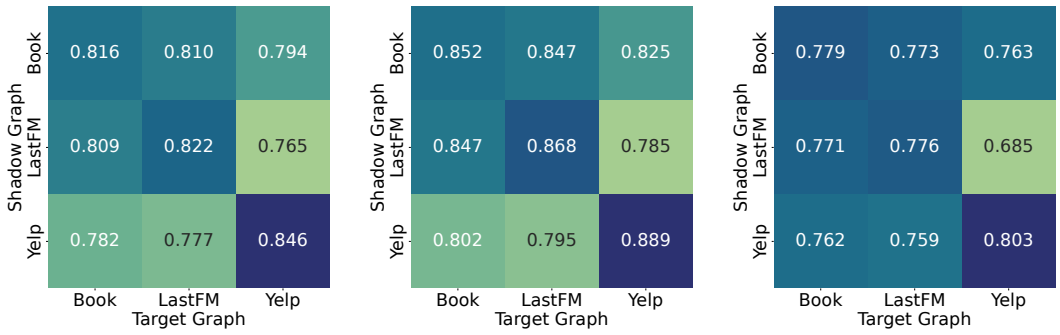
Baselines. We compare MINER with two existing attacks:

- **MIARS** [46]: As MIARS is a user-level MIA, we modified its attack features for interaction-level inference. Specifically, the attack features of MIARS were derived from the similarity between the embeddings of the top-k recommended items (instead of all the recommended items by MIARS) and the target item. We use the same MLP architecture outlined in [46] for the attack model.
- **WBI** [44]: WBI is a white-box IMIA that leverages the item embeddings generated by the target model for attack inference. Unlike MINER, WBI utilizes white-box access to the target RS. Furthermore, it utilizes neither a shadow knowledge graph nor a shadow interaction graph. Instead, it initially hypothesizes an interaction graph (a shadow training dataset) by randomly linking items with the target user. Subsequently, it leverages the similarity between the item embeddings generated from the original graph and those from the shadow training dataset to assess the correctness of the inferred interactions.

5.2 Performance Evaluation (RQ$_1$)

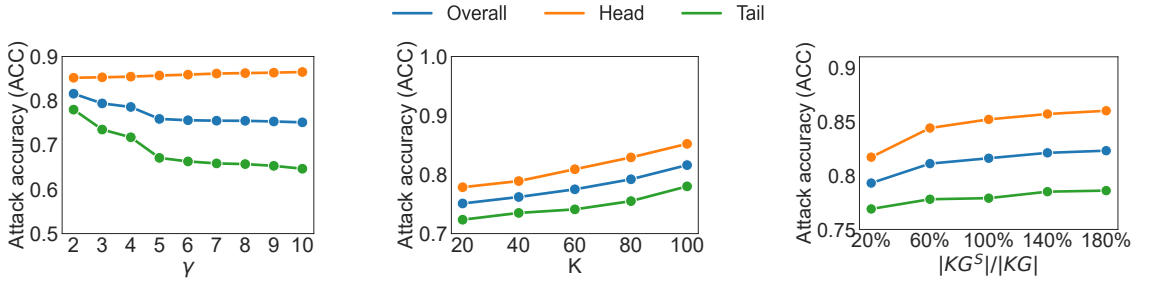
In this section, we present the performance of MINER under two settings: (1) **Non-transfer setting** where the shadow graph and the target graph are sampled from the same datasets; (2) **transfer setting** where the shadow graph and the target graph are sampled from different domains as well as different distributions.

Non-transfer setting. Table 3 reports the attack accuracy (ACC) and TPR@5%FPR under the non-transfer setting. Overall, MINER exhibits high effectiveness in all the settings, with the overall attack ACC no less than 0.81, and the ACC for head interactions and tail interactions as high as 0.87 and 0.78, respectively. Furthermore, MINER consistently outperforms both baselines in terms of both the overall ACC and the attack ACC of tail interactions. For instance, when CKE is the target model, MINER surpasses MIARS by 17.1% and 44.4% in the overall attack ACC and for tail interactions,



(a) Overall accuracy (b) Head interactions (c) Tail interactions

Figure 2: Attack accuracy (ACC) of MINER under transfer setting



(a) Parameter γ of cumulative learning (b) k: # of recommended items (c) Size of shadow knowledge graph ($\mathcal{KG}^S$)

Figure 3: Impact of various factors on the performance of MINER (CKE Model, Book dataset)

respectively. Moreover, MINER outperforms WBI by 34.4% and 36.8% in terms of the overall attack ACC and for tail interactions, respectively, even though WBI is a white-box attack. We believe this is attributed to the fact that the embeddings of the tail items generated by the target model (and used by WBI) do not encode sufficient information for the attack due to their scarcity in the training data. We have similar observations for TPR@5%FPR results and thus omitted the discussion due to limited space. These observations demonstrate MINER's effectiveness in improving attack accuracy on tail interactions while maintaining competitive performance on head interactions.

Transfer setting. Figure 2 presents the attack ACC of MINER under the transfer setting, using the CKE model for both the target and shadow models. Overall, as presented in Figure 2 (a), MINER maintains high attack accuracy even when the shadow and target graphs originate from different datasets. For instance, when the shadow graph is sampled from the LastFM dataset and the target graph sampled from the Book dataset, MINER can still achieve an attack accuracy as high as 0.809. Second, as presented in Figure 2 (b) & (c), MINER maintains effectiveness for both head and tail interactions under the transfer setting, with accuracy no lower than 0.785 and 0.685 respectively. These results demonstrate MINER's ability to transfer the knowledge, i.e., the fact that the interacted items (members) are more similar to the recommended items than the non-members, across different datasets.

5.3 Factor Analysis (RQ$_2$)

In this section, we study the impact of three factors: (1) the parameter γ (Eqn. (6)), (2) the number k of recommended items available to the adversary, and (3) the size of the shadow knowledge graph, on the attack performance of MINER. Due to the space limit, we only present the results of the setting where the CKE model is the target model and the Book dataset is the training graph.

Parameter γ. We vary γ from 2 to 10 and report the attack accuracy results of MINER in Figure 3 (a). We observe that increasing γ results in an improvement in the attack accuracy for head interactions. However, a contrasting pattern emerged in the overall accuracy and accuracy of tail interactions. This behavior can be explained by considering the relationship between γ and α (Eqn. (6)). A higher γ corresponds to an increased α, directing MINER to predominantly learn from the main branch while diminishing the contribution of the regularizer branch. As a higher γ reduces the regularizer branch's learning capacity, the attack accuracy for tail interactions experiences a decline, while an increase is observed for head interactions. This phenomenon emphasizes the importance of carefully selecting the γ value to strike a balance in achieving optimal attack accuracy across head and tail interactions.

Number of top-k recommendations. We vary k from 20 to 100 and report the attack accuracy in Figure 3 (b). MINER demonstrates effective inference — its accuracy is consistently high (at least 0.75) even when the number of recommended items is as small as 20. Furthermore, while MINER shows an increase in attack accuracy

Table 4: Ablation study (CKE model, Book dataset)

Method	Overall	Head	Tail
Base (MINER)	0.82	0.85	0.78
w/o ranking-based weight	0.79	0.84	0.74
w/o main branch	0.67	0.69	0.64
w/o regularizer branch	0.70	0.86	0.54

for both head and tail interactions as k increases, it demonstrates a slightly higher improvement for head interactions (9%) compared to tail interactions (7%).

Size of shadow knowledge graph. Figure 3 (c) reports the attack accuracy of MINER for the shadow knowledge graph of various sizes. Notably, MINER maintains high effectiveness, with an overall attack accuracy of no less than 0.8 in all settings even when the shadow knowledge graph is as small as 20% of the target knowledge graph $\mathcal{KG}$. Furthermore, the attack accuracy of MINER grows with the increase in the size of the shadow knowledge graph. In particular, MINER's overall attack accuracy rises from 0.79 to 0.82, while the attack accuracy of head interactions increases from 0.82 to 0.86, and the attack accuracy of tail interactions improves from 0.77 to 0.79, respectively, when the size of the shadow knowledge graph grows from 20% to 180% of $\mathcal{KG}$.

5.4 Ablation Study (RQ_3)

In this section, we conduct an ablation study by systematically removing individual components of MINER to assess their impact on the attack performance. The evaluated components include: (i) the ranking-based weights in the similarity function (Eqn. (1)) and (ii) each branch of the Bilateral-Branch Network (BBN).

Table 4 presents the results of the ablation study for the attack accuracy (ACC) of MINER using the CKE model as the target RS and the Book dataset as the training graph. The findings reveal that the removal of any of these components leads to a reduction in attack accuracy, demonstrating the positive impact of the three components of MINER, especially the main branch and the regularizer branch, in maintaining the effectiveness of MINER. Specifically, first, eliminating the ranking-based weights from the attack features results in a 3% decrease in overall accuracy, accompanied by a 1% drop in the accuracy of head interactions and a more substantial 5% drop for tail interactions. This implies that the ranking-based weights have a minor impact on improving the performance of MINER. Second, when the main branch is removed, a more significant accuracy drop (around 18%) is observed in overall attack accuracy as well as the attack accuracy of head tail interactions. Conversely, removing the regularizer branch results in a milder accuracy drop (14%) in overall accuracy compared to removing the main branch. This implies that the main branch is more important to MINER than the regularizer branch. We believe this is because the main branch is impacted less by the shift in item distribution than the regularizer branch.

6 Defense

In this section, we present our defense mechanisms against MINER. We consider the party who is responsible for training the target RS as the defender. Thus, the defender possesses full access to both the target RS and its training data. We also assume the defender possesses some partial information about the attack model. Specifically, the defender is aware that the attack model is designed as a binary classifier. However, he has no knowledge of the model architecture of the binary classifier. Therefore, he trains a binary classifier as the *surrogate attack model*. The architecture of the surrogate attack model can differ fundamentally from that of MINER.

6.1 Details of the Defense Mechanism

We adapt the regularization-based defense technique [27] to our setting. In general, we introduce a regularization term to the training loss function. The regularization term quantifies the (estimated) attack's power to distinguish between members and non-members. In this paper, we estimate the attack's power of distinguishing between members and non-members as the distance between the probability vector distributions of members and non-members determined by the attack. Formally, let $X = x_1, \ldots, x_n$ and $Y = y_1, \ldots, y_n$ denote the sets of random variables drawn from distributions $\mathcal{P}$ and $\mathcal{Q}$, respectively. The distance between X and Y is measured by the KL-divergence between these two distributions:

$$Dis(X, Y) = KL(X||Y) = \sum_{j=1}^{n} P(x_j) \log(\frac{x_j}{y_j})) \tag{9}$$

In our context, each $x_j(y_j)$ is the softmax output of the member (non-member) classes by the attack. Intuitively, the smaller the distance value is, the closer the distribution of label confidence for members and non-members is.

Based on the KL-divergence, RGL adds a penalty term to the loss function of the target RS as follows:

$$\mathcal{L}_{total} = \mathcal{L}^{rec} + \lambda Dis(P^+, P^-), \tag{10}$$

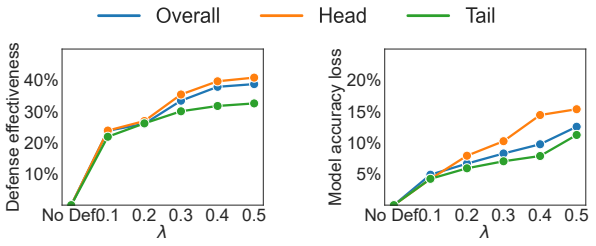
where $\mathcal{L}^{rec}$ is the loss of the target RS, P^+ and P^- are the attack's inference probability distribution of member and non-members, and λ controls the amount of regularization applied to the model. Intuitively, RGL penalizes the settings where the members and non-members have significantly different distributions.

As the defender lacks access to the attack model, he will employ the surrogate attack model to derive P^+ and P^-. Concretely, the defender generates the training data RGL^{Train} in which member/non-member samples are randomly drawn from the training and testing data of the target RS. Next, he trains a binary classifier on RGL^{Train}. The architecture of the classifier and the features in RGL^{Train} can differ from those of MINER. For instance, the surrogate attack model might adopt a linear regression model and utilize the attack features derived from item similarity measured by a single similarity function (as opposed to multiple similarity functions as by MINER). We will demonstrate that RGL remains effective across various types of surrogate attack models (Sec. 6.2).

6.2 Evaluation

6.2.1 Setup. We use the same datasets and RSs as those for the attack evaluation (Sec. 5).

Evaluation metrics. In terms of the effectiveness of the defense mechanisms, we measure the downgrade in attack accuracy after the defense. Formally, **defense effectiveness** is evaluated as:



(a) Defense effectiveness (b) Model accuracy loss

Figure 4: Defense effectiveness and model accuracy loss (CKE model, Book dataset). "No def." denotes no defense.

Defense effectiveness = $\frac{Acc_O - Acc_D}{Acc_O}$, where Acc_O and Acc_D denote the attack accuracy before and after defense mechanisms. A higher value indicates greater defense effectiveness. Concerning the impact of defenses on the target model accuracy, we measure the **model accuracy loss** because of defenses. Formally, it is calculated as: *Model accuracy loss* = $\frac{HR_O - HR_D}{HR_O}$. where HR_O and HR_D denote the hit ratio of the target model before and after defenses. A lower value indicates less model accuracy loss.

Baselines. We consider three methods for comparison with RGL:

- **Noisy embedding (NE) [49]**: it introduces noise to the item embeddings. The noise follows the Laplace distribution with probability density function $Pr(Lap(\beta) = x) = \frac{1}{2\beta} e^{-\frac{|x-\mu|}{\beta}}$, where μ denotes the mean of the distribution, and β is the scale parameter. A higher β results in larger amounts of noise, enhancing the defense. We experiment with $\beta = 1, 2, 3, 4, 5$.
- **Embedding-based regularizer (ER) [44]**: it introduces a regularizer term, which measures the difference between the item embeddings in consecutive epochs, to the loss function of the target RS: $\mathcal{L} = \mathcal{L}^{rec} + \lambda||\vec{V}_{t-1} - \vec{V}_t||$ (where $\vec{V}_{t_1}$ and $\vec{V}_t$ represent the item embeddings in the previous and current epochs, respectively). We use the same λ values for both ER and RGL.
- **Differential privacy (DP-SGD)**: We equip the target model with differential privacy [11], a de facto privacy standard, by adapting the DP-SGD method [4] to the target model. DP-SGD adds Gaussian noise to the stochastic gradient descent (SGD) during training of the target model. We use the privacy parameter $\epsilon = 0.1, 0.5, 1.0, 3.0, 5.0$ in the experiments. Note that, although CKE is not a deep learning model, it still employs an SGD algorithm to update the model parameters.

6.2.2 Performance of Defense. In this section, we evaluate the performance of RGL in terms of its defense effectiveness, model accuracy loss, and the trade-off between defense effectiveness and model accuracy loss.

Defense effectiveness. Figure 4 (a) showcases the effectiveness of RGL when the CKE model trained on the Book dataset is the target model. We vary the parameter $\lambda = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ (Eqn. (10)) in our experiments. Notably, RGL proves effective in protecting the target RS against MINER. Its defense effectiveness grows with the increase of λ, evidenced by the increase in defense effectiveness to be as high as 37% when $\lambda = 0.5$. Moreover, RGL provides effective defense for both head and tail interactions across all the settings, with the defense effectiveness on the head interactions consistently higher than that on the tail interactions. Additionally, while both head and tail interactions witness increases in defense effectiveness when λ increases, such an increase is more significant for head interactions than tail interactions. This demonstrates that RGL is more effective on head interactions than tail ones.

Table 5: Performance of RGL with various surrogate attack models (CKE model, Book dataset, $\lambda = 0.4$).

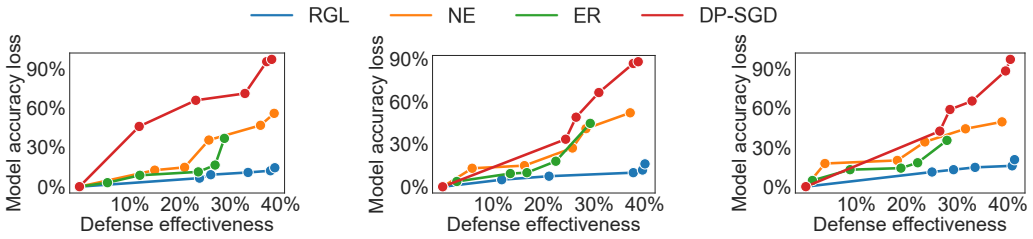
Surrogate attack model	Attack features	Defense eff.	Accuracy loss
MLP	Same as MINER	35.6%	12.5%
LR		31.4%	11.5%
SVM		32.7%	11.9%
MLP	A subset of MINER	31.7%	11.5%
LR		29.4%	10.9%
SVM		30.4%	11.6%
MINER		37.9%	12.7%

Model accuracy loss. Figure 4 (b) exhibits the results of model accuracy loss by RGL. Overall, RGL does not result in significant accuracy loss (at most a 13% decrease). Furthermore, the head interactions experience a higher accuracy loss (15%) compared to that of the tail ones (11%). Additionally, the model accuracy loss consistently increases when the value of λ grows. Together with the results of defense effectiveness (Figure 4 (a)), these demonstrate that stronger defense requires more sacrifice on model accuracy.

Defense-accuracy trade-off. Since RGL and the baselines utilize different privacy parameters, direct comparisons of their defense capabilities would be unfair. Therefore, we compare these methods in terms of the trade-off between their defense effectiveness and target model accuracy. To visualize this trade-off, we construct a defense-accuracy curve that illustrates pairs of values representing defense effectiveness and model accuracy loss. This curve is generated by varying the privacy parameters of RGL and the two baselines while measuring the corresponding defense effectiveness and model accuracy loss for each parameter value.

Figure 5 visualizes the defense-utility curve of RGL and the three baselines when CKE model is the target model. RGL demonstrates a superior trade-off between defense effectiveness and model accuracy loss compared to the baselines. In particular, RGL achieves higher defense effectiveness than the baselines under equivalent model accuracy loss. Meanwhile, it suffers from substantially fewer model accuracy loss than the baselines when they present similar defense effectiveness. For instance, when the Book dataset is used as the training data (Figure 5 (a)), RGL experiences 14% accuracy loss when the defense effectiveness reaches 40%, while NE and DP-SGD incur 58% and 92% of accuracy loss, respectively, under the same defense effectiveness. This signifies the effectiveness of RGL in balancing defense strength and maintaining utility.

Varying surrogate attack models. The surrogate attack model of RGL may not have the same attack ability as MINER, especially when it has a different architecture and/or features from that of MINER. Thus we assess the defense effectiveness of RGL whose



(a) Book dataset (b) LastFM dataset (c) Yelp dataset

Figure 5: Trade-off between defense and target model performance (CKE model) of RGL and three baselines. The defense with higher defense effectiveness and lower model accuracy loss has a better trade-off.

surrogate attack models have different architectures from MINER. We consider three types of surrogate attack models: (1) a Multi-layer Perception (MLP) neural network comprising 2 layers and 128 neurons per layer; (2) a linear regression (LR) model; and (3) a Support Vector Machine (SVM). These three models use either the same features as MINER (Eqn. (3)) or a subset of MINER's features. Specifically, among the four similarity functions that MINER employ, we choose Cosine similarity and L2 similarity, and concatenate these two types of similarity values (Eqn. (2)) to derive the attack features of the surrogate attack model.

Table 5 presents the results of defense effectiveness and model accuracy loss for RGL using seven surrogate attack models. Overall, the defense effectiveness of RGL is only slightly lower (by 6% to 22%) compared to using MINER as the attack model, even when employing a simple linear regression model as the surrogate attack model and utilizing only a subset of attack features. Furthermore, the model accuracy loss by RGL is (marginally) better than that of using MINER as the attack model, showing the trade-off between defense power and target model accuracy.

7 Related Work

Membership inference attacks (MIAs). Homer et al. [21] first introduced the concept of MIAs for genomics data analytics. Shokri et al. [37] proposed the first MIA framework in the context of machine learning. Recent years have witnessed active research efforts in the MIA encompassing a variety of machine learning models including federated learning [30], generative models [15], language models [38], recommender systems [41, 44, 46], and graph neural networks [19, 32]. We refer the readers to [22, 23] for some excellent surveys on MIA and its defenses.

Privacy inference attacks against RSs. Recently, various types of inference attacks, including *attribute inference attacks* [13, 47] and *membership inference attacks* [41, 44, 46], have been developed to attack RSs. Zhang et al. [46] introduced the first user-level MIA (UMIA) against RSs to infer the inclusion of a user's interactions in the training data of the given RS. Wang et al. [41] also considered UMIA and proposed a defense mechanism based on disentangled representations. Yuan et al. [44] developed the first interaction-level MIA (IMIA). They focused on Federated recommender systems. In their approach, the adversary performs the attack by leveraging white-box access to the item embeddings generated by the target RS. In contrast to [44], we consider the black-box access to the target RS, and utilize the shadow model to generate the surrogate item embeddings for the attack inference.

Recommendations over long-tail data. Long-tail data has been identified as a major challenge for recommender systems [26]. The existing techniques on long-tail recommendations can be broadly categorized into three classes: *pre-processing methods*, *in-processing methods*, and *post-processing methods*. The pre-processing techniques modify the input graph through re-sampling [10], injecting noise to the input graph [6], and injecting additional item-to-item edges for tail items [29]. In-processing methods modify the model by either decoupling the learning process of memorization and generalization [48] or incorporating preference mechanisms for long-tail items [28]. The post-processing methods modify the recommendations by re-ranking [33, 35], amplifying the exposure of long-tail items [39], and substituting popular items with less popular ones [25].

8 Conclusion

In this paper, we introduce MINER, a new IMIA tailored for recommender systems with long-tail item distribution. The key component of MINER is a Bilateral-Branch Network (BBN) that aggregates the knowledge learned from both head and tail interactions. Our empirical evaluation demonstrates the effectiveness of MINER. Furthermore, we design a novel defense mechanism named RGL to mitigate the privacy risks posed by MINER, and demonstrate that RGL can effectively defend against MINER while maintaining recommendation accuracy.

We outline potential avenues for future research. One avenue involves exploring whether the privacy risks associated with membership inference attacks extend to other forms of recommender systems that do not leverage knowledge graphs [17, 18]. Another direction is to investigate the susceptibility of recommender systems to different types of privacy inference attacks, such as attribute inference attacks [9, 31] and model inversion attacks [12, 20].

Acknowledgements

We thank the anonymous reviewers for their feedback. This work was supported by the National Science Foundation (CNS-2029038; CNS-2135988; OAC-2319880). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agency.

References

[1] Amazon product review dataset. https://jmcauley.ucsd.edu/data/amazon/.
[2] Lastfm dataset. http://www.lastfm.com.
[3] Yelp dataset. https://www.yelp.com/dataset.
[4] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2016.
[5] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 2018.
[6] Bing Bai, Yushun Fan, Wei Tan, and Jia Zhang. Dltsr: A deep learning framework for recommendations of long-tail web services. *IEEE Transactions on Services Computing*, 2017.
[7] J Roger Bray and John T Curtis. An ordination of the upland forest communities of southern wisconsin. *Ecological monographs*, 1957.
[8] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *Proceedings of the IEEE Conference on Symposium on Security and Privacy*, 2022.
[9] Abdelberi Chaabane, Gergely Acs, Mohamed Ali Kaafar, et al. You are what you like! information leakage through users' interests. In *Proceedings of annual network & distributed system security symposium*, 2012.
[10] Jin Chen, Defu Lian, Binbin Jin, Kai Zheng, and Enhong Chen. Learning recommenders for implicit feedback with importance resampling. In *Proceedings of ACM World Wide Web Conference*, 2022.
[11] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on Theory and Applications of Models of Computation*, 2008.
[12] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2015.
[13] Christian Ganhör, David Penz, Navid Rekabsaz, Oleg Lesota, and Markus Schedl. Unlearning protected user attributes in recommendations with adversarial training. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022.
[14] Elaheh Malekzadeh Hamedani and Marjan Kaedi. Recommending the long tail items through personalized diversification. *Knowledge-Based Systems*, 2019.
[15] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Procedings of Privacy Enhancing Technologies Symposium*, 2019.
[16] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.
[17] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.
[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of ACM World Wide Web Conference*, 2017.
[19] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing links from graph neural networks. In *Procedings of USENIX Security Symposium*, 2021.
[20] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of Annual Computer Security Applications Conference*, 2019.
[21] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 2008.
[22] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 2022.
[23] Li Hu, Anli Yan, Hongyang Yan, Jin Li, Teng Huang, Yingying Zhang, Changyu Dong, and Chunsheng Yang. Defenses to membership inference attacks: A survey. *ACM Computing Surveys*, 2023.
[24] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. 2010.
[25] Iordanis Koutsopoulos and Maria Halkidi. Efficient and fair item coverage in recommender systems. In *Proceedings of the Dependable, Autonomic and Secure Computing, Pervasive Intelligence and Computing, Big Data Intelligence and Computing and Cyber Science and Technology Congress*, 2018.
[26] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of International Conference on Ubiquitous Information Management and Communication*, 2008.
[27] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2021.
[28] Siyi Liu and Yujia Zheng. Long-tail session-based recommendation. In *Proceedings of the ACM Conference on Recommender Systems*, 2020.
[29] Sichun Luo, Chen Ma, Yuanzhang Xiao, and Linqi Song. Improving long-tail item recommendation with graph augmentation. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2023.
[30] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of the IEEE Conference on Symposium on Security and Privacy*, 2019.
[31] Gong Neil, Zhenqiang and Liu Bin. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *Procedings of USENIX Security Symposium*, 2016.
[32] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. In *Proceedings of the IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*, 2021.
[33] Enrico Palumbo, Diego Monti, Giuseppe Rizzo, Raphaël Troncy, and Elena Baralis. entity2rec: Property-specific knowledge graph embeddings for item recommendation. *Expert Systems with Applications*, 2020.
[34] Yoon-Joo Park and Alexander Tuzhilin. The long tail of recommender systems and how to leverage it. In *Proceedings of the ACM Conference on Recommender Systems*, 2008.
[35] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of ACM World Wide Web Conference*, 2020.
[36] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 1997.
[37] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the IEEE Conference on Symposium on Security and Privacy*, 2017.
[38] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2020.
[39] Rama Syamala Sreepada and Bidyut Kr Patra. Mitigating long tail effect in recommendations using few shot learning technique. *Expert Systems with Applications*, 2020.
[40] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2019.
[41] Zihan Wang, Na Huang, Fei Sun, Pengjie Ren, Zhumin Chen, Hengliang Luo, Maarten de Rijke, and Zhaochun Ren. Debiasing learning for membership inference attacks against recommender systems. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.
[42] Zhichao Xu, Hansi Zeng, Juntao Tan, Zuohui Fu, Yongfeng Zhang, and Qingyao Ai. A reusable model-agnostic framework for faithfully explainable recommendation and system scrutability. *ACM Transactions on Information Systems*, 2023.
[43] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, 2012.
[44] Wei Yuan, Chaoqun Yang, Quoc Viet Hung Nguyen, Lizhen Cui, Tieke He, and Hongzhi Yin. Interaction-level membership inference attack against federated recommender systems. In *Proceedings of ACM World Wide Web Conference*, 2023.
[45] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2016.
[46] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhunmin Chen, Pengfei Hu, and Yang Zhang. Membership inference attacks against recommender systems. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2021.
[47] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Lizhen Cui, and Xiangliang Zhang. Graph embedding for recommendation against attribute inference attacks. In *Proceedings of the World Wide Web Conference*, 2021.
[48] Yin Zhang, Ruoxi Wang, Derek Zhiyuan Cheng, Tiansheng Yao, Xinyang Yi, Lichan Hong, James Caverlee, and Ed H Chi. Empowering long-tail item recommendation through cross decoupling network. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.
[49] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference attacks against graph neural networks. In *Proceedings of USENIX Security Symposium*, 2022.
[50] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.