RecPS: Privacy Risk Scoring for Recommender Systems

Jiajie He[1*], Yuechun Gu[1†], Keke Chen[1‡]

[1]University of Maryland, Baltimore County

Abstract

Recommender systems (RecSys) have become an essential component of many web applications. The core of the system is a recommendation model trained on highly sensitive user-item interaction data. While privacy-enhancing techniques are actively studied in the research community, the real-world model development still depends on minimal privacy protection, e.g., via controlled access. Users of such systems should have the right to choose *not* to share highly sensitive interactions. However, there is no method allowing the user to know which interactions are more sensitive than others. Thus, quantifying the privacy risk of RecSys training data is a critical step to enabling privacy-aware RecSys model development and deployment. We propose a membership-inference attack (MIA)- based privacy scoring method, RecPS, to measure privacy risks at both the interaction and user levels. The RecPS interaction-level score definition is motivated and derived from differential privacy, which is then extended to the user-level scoring method. A critical component is the interaction-level MIA method RecLiRA, which gives high-quality membership estimation. We have conducted extensive experiments on well-known benchmark datasets and RecSys models to show the unique features and benefits of RecPS scoring in risk assessment and RecSys model unlearning.

1 Introduction

Recommender systems (RecSys) utilize machine learning algorithms to analyze user-item interactions (i.e., user implicit preferences on items, such as book reviews on Amazon and movie ratings on Netflix) and recommend items to users who might like them [1]. They have been deployed in numerous applications, including e-commerce, social media, and entertainment. The success of recommendation systems relies on large-scale user personal data, which often contains private information about user preferences, actions, and social contexts [2], thereby raising significant privacy concerns.

Several studies have been made to ensure the privacy of data contributors in RecSys modeling. However, successful deployments of privacy-enhancing techniques have been limited. Earlier efforts in RecSys privacy protection focused on data anonymization techniques [2]. Recently, Federated RecSys [3] applied the federated learning framework to RecSys modeling, allowing data contributors to keep their data locally and only upload intermediate computational results, thereby avoiding the exposure of raw private data and the application of flawed anonymization techniques. However, federated RecSys does not protect the learned model. Recent studies [4–7] show that membership inference attacks (MIAs) can reveal private information in training data by only accessing the RecSys model API. Although differentially private machine learning is a provable method for protecting privacy leakage in RecSys models [8], it leads to significant reductions in model quality due to noise addition and gradient clipping [8], which is not well accepted by practitioners.

In practice, recommender system (RecSys) practitioners rely primarily on minimal privacy protections, such as controlled access, which fully preserves data utility. In controlled access settings, data curators and authorized users are trusted to protect data privacy. However, numerous risks such as insider threats, system vulnerabilities, and emerging model-API-based attacks [2, 9, 10] demonstrate that controlled access alone is insufficient to ensure robust privacy protection.

Consequently, it is essential that data contributors, who are also users of models, can proactively assess their privacy risks and make informed decisions regarding their participation in RecSys projects. Upon receiving estimated privacy risks, data contributors should have the right to request the removal of their data from training datasets [11, 12] or require the model to unlearn their records [13, 14]. Recent pri-

[*] jiajih1@umbc.edu

[†] ygu2@umbc.edu

[‡] kekechen@umbc.edu

vacy regulations, such as GDPR [15] and CCPA [16], emphasize the responsibility of data curators to transparently communicate potential privacy risks and enable contributors to opt out of high-risk activities. Despite this regulatory framework, a lack of formal, quantitative tools remains for systematically and transparently evaluating privacy risks.

Scope of Research. Inspired by recent developments in hypothesis-based membership inference [17], e.g., the likelihood ratio test (LiRA) method, we design an effective RecSys Privacy Scoring Tool (RecPS) to assess the potential risk of a user participating in a RecSys modeling task. This privacy scoring tool has many potential uses. One critical application is for users to determine whether they want to withdraw samples from a RecSys modeling task or "unlearn" [11] selected samples from an existing RecSys model.

Our scoring approach consists of two critical components: the theory and the implementation. It is backed by the theory of differential privacy, i.e., how difficult an attacker can distinguish whether a user or their specific record is in the training data of a RecSys model. This is the most fundamental privacy threat – if the attacker cannot confidently determine the membership, the information collected from any other type of attacks, e.g., data reconstruction and property inference [9], cannot be linked to the specific user. We define the formal scoring method for a single user-item interaction and then extend it to the collection of user-item interactions for a specific user. The core is to estimate the optimal ratio between the True Positive Rate (TPR) and False Positive Rate (FPR) of MIA, TPR/FPR. A previous study [18] has used MIA's TPR to represent the privacy score. However, we find that the ratio TPR/FPR directly links to the definition of differential privacy, and thus it's more theoretically justifiable.

However, there are challenges in applying RecSys MIA to the proposed scoring method. (1) Our scoring method requires a powerful MIA at the interaction level. However, we found that most existing RecSys MIA methods only apply to the user level [6, 7, 19] and the interaction-level MIAs [4, 5] have limited attacking abilities and do not meet the requirement of our scoring framework. (2) A promising record-level MIA method, likelihood-ratio attack (LiRA) [17], has shown state-of-the-art performance on classification models [20–22]. However, it remains unclear whether and how LiRA can be applied to RecSys models for our scoring purposes.

We designed the RecLiRA MIA for our scoring framework based on LiRA. RecLiRA shows significantly higher TPR in the low-FPR region for interaction-level MIA compared to the best existing interaction-level MIA, MINER [5]. It works for models that predict the likelihood of interaction, such as neural collaborative filtering (NCF) [23], and light graph convolution network (LightGCN) [24].

We have evaluated the proposed method with three well-known benchmark recommendation datasets: Amazon Digital Music (ADM) [25], the Amazon Beauty [25], and Movielens-1m (ml-1m) [26]. RecLiRA demonstrates significantly better MIA performance than MINER, thereby ensuring the quality of the privacy scores. We also simulate the scenario where users request to remove their records from the RecSys model and show how user-level removals can significantly impact the model performance. We then demonstrate how interaction-level privacy scores can be utilized to selectively remove sensitive interactions, enabling finer-grained privacy-utility tradeoffs.

Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to propose and study privacy scoring in RecSys, which can serve as a critical tool for both data contributors and RecSys owners to assess the privacy risks of participating in RecSys modeling.
- Our RecPS method can generate privacy scores at both the user level and the user-item interaction level using a high-quality MIA method, RecLiRA, which provides multiple granularities for better privacy-utility tradeoffs.
- We have conducted extensive experiments to demonstrate the quality of RecLiRA and the unique features of the RecPS scoring method.

The remaining sections are organized as follows. Section 2 briefly describes the background knowledge; Section 4 shows the principles and implementations of our designed RecPS in detail; Section 5 presents our experimental results; Section 3 summarizes the related work; and finally Section 6 concludes our work.

2 Preliminaries

In this section, we introduce the basic definitions and notations in Section 2.1 and the basic LiRA method that will be adapted to RecSys for our scoring method in Section 2.2.

2.1 Definitions and Notations

A recommender system analyzes the interactions between users, denoted by the user set U, and items, denoted by the item set I, and recommends items that a user may be interested in. A user-item interaction with $u \in U$ and $i \in I$ is denoted as a 3-tuple (u, i, r), where r is a rating that user u gives to item i, or simply (u, i), representing u interacted (e.g., clicked) i. For simplicity, we used the interaction definition of (u, i) in this paper. A recommender model takes (u, i) as the input and outputs a recommendation score: $s_{(u,i)} = f((u, i))$. We can recommend i if $s_{u,i}$ is in the top-k recommendation scores for all $i \in I$. The output might also be normalized or interpreted as a probability, $s_{(u,i)} \in [0, 1]$. A threshold τ is given to convert the output to 1, i.e., u "hits" i, if $s_{(u,i)} > \tau$; 0 otherwise.

Existing models employ either the scoring function approach, such as ALS [27], or the probability output approach, as seen in NCF [23] and LGCN [24]. We have adopted the probability output in our approach to conveniently convert the recommender model to a classification model, i.e., the RecSys will predict if it will recommend i to u or not.

2.2 Likelihood Ratio Attack (LiRA)

LiRA is a membership inference attack (MIA), which infers the likelihood of a victim sample being a member of the training data for a target model. It has been applied to classification models, where a labeled dataset $D = \{(x_i, y_i), i = 1..N\}$ is used for training the model, where x_i is a sample and y_i is the task-specific label. The membership inference attack can be formally defined as a hypothesis test about a specific sample (x, y) [17].

- H_0: the sample (x,y) was not included in the training dataset
- H_1: the sample (x,y) was included in the training dataset.

To decide between these two hypotheses, LiRA leverages per-sample likelihood ratio test to determine the membership.

$$\Lambda(f) = \frac{P(f \mid L_D)}{P(f \mid L_{\neg D})}$$

where f is the model, L_D represents the distribution of logit transformation of the model's highest confidence score for members, and $L_{\neg D}$ represents the distribution of the highest confidence score for nonmembers. $P(f \mid L_D)$ represents the likelihood of f using the sample in training, after observing the logit transformation of the model's highest confidence score of the target sample. LiRA can be done using online or offline testing methods. The offline method is more cost-effective but of slightly lower quality. We have used the offline method in our approach.

3 Related work

MIA on RecSys. The earlier RecSys MIA studies are focused on the user level. Zhang et al. [6] propose the Item-Diff method for inferring membership in a target RecSys by analyzing the similarity between a user's historical interactions and recommended items. The core idea is that, for users in the training set, their historical interactions are likely to be more closely aligned with the items recommended by the system. Wang et al. [7] propose the DL-MIA framework to improve Item-Diff with a VAE-based encoder and weight estimator to address issues with Item-Diff. Note that these user-level MIA methods cannot be modified to perform interaction-level attacks, and thus cannot be adopted by our scoring method. More recently, Wei et al. [4] propose an interaction-level membership inference on federated RecSys. However, since users in federated RecSys do not expose their records in the first place, it's impossible to calculate the MIA's TPR and FPR for our scoring purpose. Zhong et al. [5] propose another interaction-level membership inference on KG-based RecSys, utilizing the similarity matrix between the interacted items and the recommended items. However, the version modified for LGCN and NCF models still performs significantly worse than our RecLiRA.

Unlearning on RecSys models. Users can request to remove sensitive interactions from the recommender system. A closely related issue is machine unlearning [12]. Current recommender-model unlearning technologies [11] operate at the user level, assuming the RecSys owner must completely remove all data associated with the user. Such an approach can significantly degrade the overall system performance and exacerbate cold-start issues. We show that interaction-level privacy risk analysis enables finer-grained removal to ensure both utility and privacy guarantees.

Privacy Score. Training data privacy has been a top concern in AI modeling. While methods like differentiated private learning [8] allow data contributors to quantify acceptable privacy loss, model utility is often significantly damaged. In practice, controlled data access remains a mainstream method for protecting data privacy in many industrial and research environments. In controlled data access, authorized model builders work in a restricted environment to access sensitive data, which can fully preserve data utility with reduced risk of data leak. However, unlike differential privacy, there is no quantitative measure for individual data contributors to tell their privacy risk before participating in a machine learning task. Gu et al. [28] first proposed personalized privacy scoring in Fine tuning LLM but did not conduct in-depth research. At the same time, it is still unknown how to design personalized privacy score for other downstream tasks such as recommendation systems.

4 RecPS: Privacy Scoring for RecSys

In this section, we first introduce the threat model for RecPS in Section 4.1. Then, we define the privacy score in Section 4.2 and describe the score estimation method and implement it in Section 4.3.

4.1 Threat Model

A typical RecSys consists of two primary components: the offline model training and the online recommendation service that processes user requests. In the scoring process, the model owner simulates a relatively powerful adversary to conduct MIA attacks, who can apply the RecSys model to determine the probability of a user-item interaction. However, the adversary cannot access the offline model training component.

Adversary's Goal: The adversary aims to determine whether a particular user's interaction records were included in training the target recommendation model. Successfully inferring the user's presence in the training dataset could reveal sensitive details about the user's historical interactions, directly compromising the user's privacy and increasing the risk of legal and business consequences for the data collector. The adversary can also conduct other attacks, e.g., data reconstruction, to infer information about a subset or the entire training data. However, without MIA to determine the membership of a user or a record, the adversary cannot confidently link the inferred information to a specific user.

Adversary's Knowledge: The adversary knows the list of candidate items and the model output, i.e., the probability associated with each recommended item. However, the adversary does not have access to the offline model training process and thus lacks knowledge about the exact users or user-item interactions involved in training.

Under this threat model, RecSys owners can proactively quantify the privacy risk associated with each data contributor. This transparency enables data contributors to understand their privacy risks clearly and make informed decisions about their participation. In particular, when contributors exercise their "right to be forgotten," we show that the scores can help determine a more fine-grained sample removal strategy that preserves more utility with a comparable privacy guarantee.

4.2 Defining Privacy Score in RecSys

Determining the likelihood that a user's records were used in modeling is a critical step in defining the privacy risk. Without this information, the adversary cannot link the result of any other type of attack to the specific user. We derive the theoretically justifiable privacy score based on the definition of differential privacy and its link with MIA. Let's start with the basic definition of differential privacy, e.g., the widely adopted relaxed (ϵ, δ) - differential privacy.

Background: Differential Privacy (DP). An algorithm M satisfies (ϵ, δ)-differential privacy if, for any two adjacent datasets D_0 and D_1 differing by exactly one record, and for all measurable outputs $\mathcal{O}$, the following condition is met:

$$Pr(M(D_0) \in \mathcal{O}) \leq e^{\epsilon} Pr(M(D_1) \in \mathcal{O}) + \delta,$$

where ϵ denotes the privacy budget and δ represents a small probability, commonly set as $\delta = 1/N$ for a dataset containing N records. The symmetry between D_0 and D_1 encapsulates the concept of "indistinguishability" governed by ϵ.

The hypothesis testing interpretation of DP [29] links the probabilities $Pr(M(D_0) \in \mathcal{O})$ and $Pr(M(D_1) \in \mathcal{O})$ to MIA's True Positive Rate (TPR) and False Positive Rate (FPR). Thus, MIAs can serve as auditing tools to verify whether a claimed (ϵ, δ)-differentially private model meets the condition: $\ln\left(\frac{TPR}{FPR}\right) \leq \epsilon$ for *every sample.* If a sample causes the violation of this inequality, there might be some DP implementation errors that lead to the false claim of the privacy bound [21, 22, 30].

Interaction-level privacy risk scoring. Interestingly, since modeling is a statistical process, we have observed that even without a DP noise injection mechanism, a *non-differentially private modeling method* M_0, for two datasets D_0 and D_1 differing by exactly one *known record* r, there exists a sample-specific bound ϵ_r,

$$\ln\left(\frac{Pr(M_0(D_0) \in \mathcal{O})}{Pr(M_0(D_1) \in \mathcal{O})}\right) \leq \epsilon_r, \tag{1}$$

where ϵ_r represents the "true" but unknown privacy risk associated with record r. Ideally, a perfectly powerful MIA could precisely estimate this risk with $\ln\left(\frac{TPR}{FPR}\right)$. Motivated by this intuition, we formally define the interaction-level privacy risk as follows.

In RecSys modeling, since a single record corresponds to an interaction, e.g., (u, i), we call the *interaction-level* privacy risk bound the *privacy score*:

$$\epsilon_{(u,i)} = \sup \ln \frac{\Pr\big(M_0(D_{0,(u,i)}) \in \mathcal{O}\big)}{\Pr\big(M_0(D_{1,(u,i)}) \in \mathcal{O}\big)}. \qquad (2)$$

where $D_{0,(u,i)}$ and $D_{1,(u,i)}$ differ by (u, i). Without loss of generalization, let $D_{0,(u,i)} = D_{1,(u,i)} \cup (u, i)$. Since we don't know whether an MIA is ideal, we try to find the most powerful MIAs to approach the ideal ϵ_r.

User-level privacy risk scoring. Users may also have concerns about their overall privacy risk rather than specific interactions. Existing user-level attacks rely on empirical approaches rather than principled methodologies, such as LiRA [17], which complicates theoretical privacy analyses. To address this challenge, we propose to derive user-level privacy risk scoring directly from interaction-level privacy risk assessments. Let's denote a user u's interaction set, I_u, and $D_{0,u}$ and $D_{1,u}$ represent datasets differing by I_u so we can get $D_{0,u} = D_{1,u} \cup I_u$. Then, $Pr(M_0(D_{0,u}) \in \mathcal{O}) = \prod_{i \in I_u} Pr(M_0(D_{0,(u,i)}) \in \mathcal{O})$ and $Pr(M_0(D_{1,u}) \in \mathcal{O}) = \prod_{i \in I_u} Pr(M_0(D_{1,(u,i)}) \in \mathcal{O})$. With Eq. 2, we can derive the user's privacy risk, ϵ_u, which is bounded by $\sum_{i \in I_u} \epsilon_{u,i}$.

To avoid overestimating the risk of users who have a large number of interactions, we have used the average of the interaction scores to represent the user-level score.

$$\epsilon_u = 1/|I_u| \sum_{(u,i) \in I_u} \epsilon_{(u,i)}. \qquad (3)$$

4.3 RecLiRA: Estimating Privacy Score with MIA

We have formally defined the RecPS scoring methods for both interaction-level and user-level privacy risk bounds. As discussed, we can use an MIA's sample-level TPR and FPR to estimate the interaction-level score, but the MIA's quality is essential to the score. However, very few RecSys-specific MIAs can be applied to the interaction level so far [4, 5], whose performance is not strong enough for our scoring method. Thus, we developed a new interaction-level MIA – RecLiRA, based on LiRA originally developed for classification models. In experiments, we have shown that RecLiRA outperforms the best interaction-level MIA reported so far [5].

According to the previous discussion, if an interaction-level RecSys MIA, $\mathcal{A}$, exists, we can use its TPR and FPR for a specific interaction (u, i), i.e., $\ln(\text{TPR}_{\mathcal{A},(u,i)}/\text{FPR}_{\mathcal{A},(u,i)})$, to approximate the defined privacy score, i.e.,

$$\hat{\epsilon}_{(u,i)} = \ln(\text{TPR}_{\mathcal{A},(u,i)}/\text{FPR}_{\mathcal{A},(u,i)}),$$

where $\hat{\epsilon}_{(u,i)} \leq \epsilon_{(u,i)}$. With the quality of MIA improving, $\hat{\epsilon}_{(u,i)} \to \epsilon_{(u,i)}$. Correspondingly, we have

$$\hat{\epsilon}_u = 1/|I_u| \sum_{(u,i) \in I_u} \hat{\epsilon}_{(u,i)}.$$

Algorithm 1: The preparation stage

```
Input: Training dataset $D$, number of shadow
       models $m$
Output: Shadow datasets $\{\mathcal{S}_j, j = 1..m\}$,
        shadow models $\{M_j, j = 1..m\}$, and
        distribution $\mathcal{N}_{\text{out}}$
$\phi_{\text{out}} \leftarrow \{\}$;
for $j \in \{1, ..., m\}$ do
    $\mathcal{S}_j \leftarrow$ Random sampling from the training
      dataset with each sample selected with
      0.5 probability;
    $M_j \leftarrow \text{Train}(\mathcal{S}_j)$  ▷ Train a shadow model;
    for $(u, i) \in \mathcal{D} \setminus \mathcal{S}_j$ do
        $\phi_{\text{out}} \leftarrow \phi_{\text{out}} \cup \{\phi(M_j((u, i)))\}$;
                ▷ Estimate the OUT distribution's
  parameters: $\mu$ and $\sigma$ with $k$ OUT samples' $\phi$
  values $\mu_{\text{out}} \leftarrow \text{mean}(\phi_{\text{out}})$;
$\sigma_{\text{out}} \leftarrow \text{var}(\phi_{\text{out}})$;
return $\{\mathcal{S}_j\}$, $\{M_j\}$, $\mu_{\text{out}}$, $\sigma_{\text{out}}$;
```

Our next objective is to design a high-quality *interaction-level* MIA, RecLiRA, for RecSys.

We consider the most popular RecSys models, such as NCF [23] and LightGCN [24], which output the probability of a user interacting with an item, i.e., $p = M((u, i)), p \in [0, 1]$. It can be conveniently converted into a binary classifier, with a confidence vector $(p, 1 - p)$ for probabilities of ("interaction", "no interaction"), respectively. With the classifier representation, we are ready to apply LiRA[1]. We have adopted the offline version of LiRA for our scoring approach. Our experiments show that this adaptation is quite successful for RecSys models.

The basic idea of the offline LiRA is that the output of an IN-training sample r_{IN}, $M(r_{IN})$, is highly distinguishable from the OUT-training sample's output distribution, $M(r_{OUT})$. Carlini et al. [17] have

[1] For some scoring models, such as matrix factorization [31] and ALS [27], which generate scores for ranking and cannot be converted to a classification problem, RecLiRA does not apply. We will develop effective MIAs for such approaches in our future work.

```
Algorithm 2: ScoreQuery($u$)
Input: Training dataset $D$, shadow datasets $\{\mathcal{S}_j\}$, shadow models $\{M_j\}$ for
       $j = 1 \dots m$, OUT distribution $\mathcal{N}_{out}$,
       and interaction set $I_u$ of user $u$
Output: Scores: $\{\hat{\epsilon}_{(u,i)}\}_{(u,i)\in I_u}$ and $\hat{\epsilon}_u$
for $(u,i) \in I_u$ do // For each interaction
    for $j \in \{1 \dots m\}$ do
        $\Lambda_{(u,i),M_j} \leftarrow 1 - \Pr\big(Z > \phi(q)\big), \quad Z \sim \mathcal{N}_{out}$;
        $L_j \leftarrow \begin{cases} 1 & \text{if } (u,i) \in \mathcal{S}_j, \\ 0 & \text{otherwise} \end{cases}$;
         // Ground-truth IN/OUT
    // Gather candidate thresholds from OUT models
    $T \leftarrow \{\}$;
    for $j \in \{1 \dots m\}$ do
        if $(u,i) \notin \mathcal{S}_j$ then
            $T \leftarrow T \cup \{\Lambda_{(u,i),M_j}\}$;
    // Find maximum TPR/FPR
    $\hat{\epsilon}_{(u,i)} \leftarrow 0$;
    for $t \in T$ do
        $P \leftarrow \{\}$ ; // Predicted labels with threshold $t$
        for $j \in \{1 \dots m\}$ do
            $P_j \leftarrow \begin{cases} 1 & \text{if } \Lambda_{(u,i),M_j} > t, \\ 0 & \text{otherwise} \end{cases}$;
        (TPR, FPR) $\leftarrow$ Compute TPR and FPR using $\{L\}$ and $\{P\}$;
        if $FPR \neq 0$ and $\hat{\epsilon}_{(u,i)} < \ln\big(\frac{TPR}{FPR}\big)$ then
            $\hat{\epsilon}_{(u,i)} \leftarrow \ln\Big(\frac{\text{TPR}}{\text{FPR}}\Big)$;
$\hat{\epsilon}_u \leftarrow \frac{1}{|I_u|} \sum_{(u,i)\in I_u} \hat{\epsilon}_{(u,i)}$;
return $\{\hat{\epsilon}_{(u,i)}\}_{(u,i)\in I_u}$ and $\hat{\epsilon}_u$;
```

identified that the logit transformation of the classification model's output confidence, denoted q, can be a useful feature for this task. Intuitively, the IN-training sample is much easier to predict for the classifier than an OUT-training sample, with which the IN samples' highest confidence scores will be much higher. It's shown [17] that the logit transformation, $\phi(q) = \log(q/(1-q))$ for both IN and OUT samples has Gaussian-like distributions, which can be conveniently used by hypothesis testing. We have used the absolute difference between the predictions: "interaction" and "no interaction" to define q for the recommender models to gain satisfactory performance.

The offline LiRA assumes that OUT samples' outputs for different models have a similar $\phi(q)$ distribution and thus can be shared across different models, which significantly saves the cost of MIA. The MIA task can be conducted with a one-side hypothesis testing to determine whether the tested sample's $\phi(q)$ is in the OUT distribution.

RecLiRA Scoring Framework. We name the LiRA attack for RecSys models: RecLiRA. The framework consists of two stages: the offline preparation stage and the online scoring stage. The offline stage prepares the shadow models and estimates the OUT $\phi(q)$ distribution and the scoring stage calculates the privacy score on demand for interactions and users.

As we treat the selected RecSys models as binary classification models, the output confidence vector for a record (u,i) is denoted as $(p, 1-p) = M_0((u,i))$.Intuitively, if one sample is an IN-training sample, the difference between p and $1-p$ should be large, while an OUT-training sample should have p close to $1-p$. Thus, we compute the difference between these two as:

$$q = |p - (1-p)| = |2p - 1|$$

where $q \in [0,1]$. With a logit transformation:$\phi(q) = \log\left(\frac{q}{1-q}\right)$, $\phi(q)$'s distribution is approximately Gaussian.

In the preparation stage, we first train m shadow RecSys models,
$\{M_1, ..., M_m\}$ with the corresponding m sample datasets $\{\mathcal{S}_1, ..., \mathcal{S}_m\}$ that are generated as follows. For each sample set, each sample in the original dataset D is selected with 0.5 probability. Thus, each interaction $(u,i) \in D$ shows up in about $M/2$ sample sets. For a sample set, $\mathcal{S}_j$, we name any $(u,i) \in \mathcal{S}_j$ an IN record for shadow model M_j; otherwise, an OUT record for M_j. RecPS then collects the $\phi(q)$ values of a few OUT-training samples for estimating the Gaussian distribution $\mathcal{N}_{out}$'s parameters. Algorithm 1 describes the detailed steps in the offline preparation stage.

In the online scoring stage, for each interaction (u,i), we calculate its $\phi(q_j)$ for each shadow model M_j, where $q_j = M_j((u,i))$, and compute the probability Λ as:

$$\Lambda_j = 1 - Pr(Z > \phi(q_j)), \quad Z \sim \mathcal{N}_{out} \qquad (4)$$

A higher Λ_j indicates the target sample is more likely to be an IN-training sample. We can set up a threshold T for Λ as the cutoff separating IN and OUT samples: if $\Lambda_j > T$, we predict the interaction as an IN-training sample. In practice, a global threshold, e.g., $T = 0.5$, has led to good results in classification [17], but may not be optimal. To maximize the ratio TPR/FPR for scoring, we have probed the interaction-specific threshold $T_{(u,i)}$ for each interaction (u, i) as Algorithm 2 shows.

In practice, the number of shadow models, m, should be greater than 500 to generate a statistically significant estimation for TPRs and FPRs [21, 22, 30]. The OUT samples are used to estimate the parameters of $\mathcal{N}_{out}$, with a minimum of 30 samples to ensure reliable parameter estimation [32].

Cost analysis. The dominating cost of the proposed RecLiRA is shadow model training. Let's use the cost of training one shadow model as the basic cost unit, T. In contrast, model application incurs a much lower cost, denoted as t. The whole process involves training m models, applying k OUT samples to infer the OUT distribution, and for each sample testing each of the m models. Thus, the offline cost is $O(mT + kt)$, and the online per-sample cost is $O(mt)$. In experiments, we have used m around 500 and k around 30.

5 Experiments

We design experiments to answer the following research questions.

- (RQ1) The quality of MIA determines the effectiveness of our privacy scoring. We want to know how effective RecLiRA is on benchmark datasets and RecSys models.
- (RQ2) As mentioned in Section 1, a critical task to comply with privacy laws is removing specific training samples and unlearning them from a trained model. RecPS privacy scores can guide the removal/unlearning process to preserve more utility. We want to understand how the scores may enable finer-grained privacy-utility tradeoffs.
- (RQ3) The dynamic aspect of RecPS privacy scores: we want to verify whether removing interactions (or users) may affect remaining interactions' and users' scores.

Dataset	#Users	#Items	#Interactions
MovieLens-1M	6,040	3,706	1,000,209
Amazon Digital Music	840,372	456,992	1,584,082
Amazon Beauty	1,210,271	249,274	2,023,070

Table 1: Statistics of datasets.

5.1 Experiment Setup

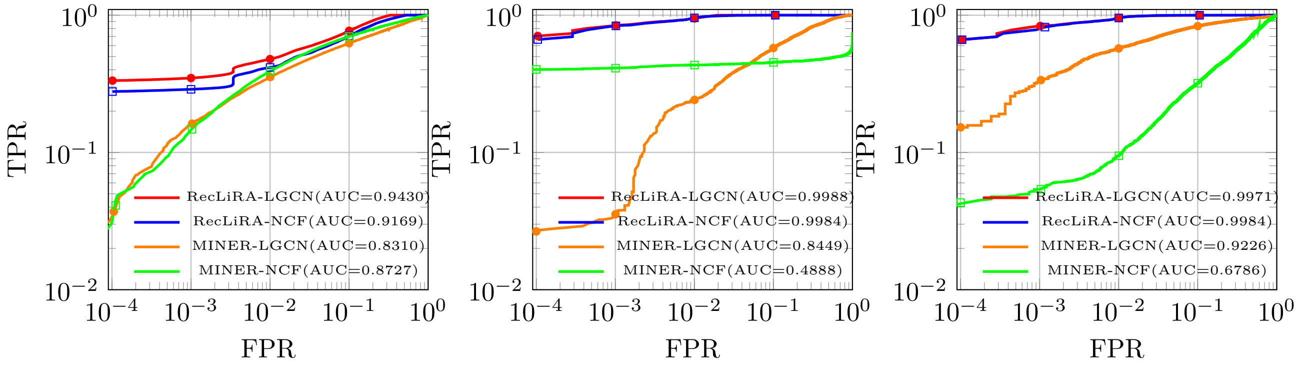
Datasets. We utilize three real-world datasets in our experiments, including Movielens-1M(ML-1M) [33], Amazon Digital Music (ADM), and Amazon Beauty(AB) [34], to evaluate our attack strategies. All these datasets are commonly used benchmark datasets for evaluating recommendation systems [2]. We only keep the users with more than 20 interactions to ensure the performance of the recommender systems [6]. Note that only ratings in these datasets are used for our evaluation in the experiments. Scores range from 1 to 5, which indicates how much users like movies (ML-1M), music (ADM), and beauty or personal care (Amazon Beauty). The statistics of these datasets are shown in Table 1. 1.

Recommender Models. According to the definition of privacy scores and our implementation of RecLiRA, this method can be applied to any RecSys. In our experiments, we consider two widely used RecSys methods with publicly available implementations for reproducibility: Neural Collaborative Filtering (NCF) [23] and LightGCN (LGCN) [24]. For NCF, we utilize the original implementation provided by [23]. For LightGCN, we configure the model with an embedding dimension of 64 and 3 graph convolution layers. To construct the training and evaluation datasets, we first sort each user's interactions in chronological order according to their timestamps. For every user, we hold out the two most recent interactions: the last interaction is used as the test instance, and the second-to-last interaction is used for validation. All remaining interactions constitute the training set with a negative sampling ratio of 1:4. Model training is performed using stochastic gradient descent (SGD) with a learning rate of 0.001, a batch size of 256, and a maximum of 30 epochs. We apply early stopping if the model's performance does not improve over 5 consecutive epochs.

The target and shadow models share identical architectures and training strategies, aligning with our threat model. To evaluate MIA performance, we randomly split the entire training dataset into an 8:2 ratio without overlap at the user level, using 80% of the users to train the target model and 20% for shadow models. All models are trained on 8 NVIDIA RTX 2080 Ti GPUs.

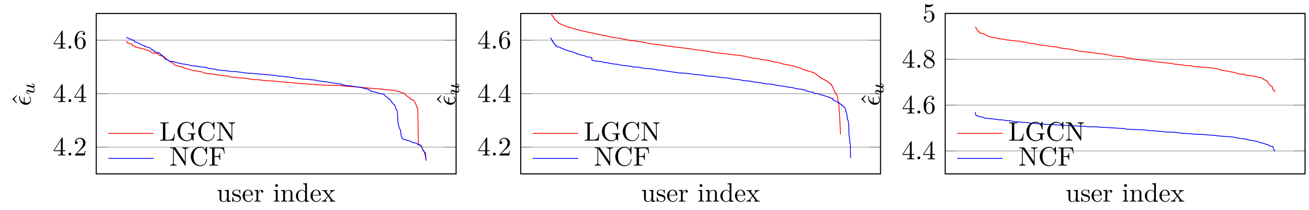
Evaluation measures. To evaluate the performance of RecSys, we adopt "the hit ratio at top-k recommended items" (HR@k) as the metric to evaluate the recommendation performance, where k = 100

[2] all these datasets are downloaded from https://cseweb.ucsd.edu/~jmcauley/datasets.html and https://grouplens.org/datasets/movielens/



(a)MovieLens-1M (b)Amazon Digital Music (c)Amazon Beauty

Figure 1: Interaction level: TPR vs FPR on RecLiRA vs MINER.



(a)MovieLens-1M (b)Amazon Digital Music (c)Amazon Beauty

Figure 2: Combined Analysis: user index vs privacy risk score(ln(TPR/FPR))

is used by previous studies [6, 7]. Specifically, we sort the recommender model's outputs for a set of (u, i) and take the top-k results. These k items intersect with the ground-truth items (i.e., the items the user actually clicked) to calculate HR@k.

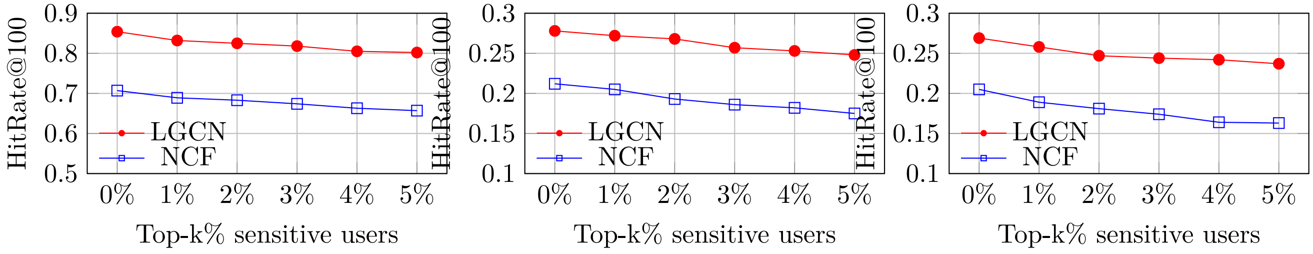
To evaluate the MIA performance, we use AUC (Area under the ROC curve) and TPR at low FPR, which have been commonly used by MIA studies [17, 35–37].

5.2 Results Analysis

RecLiRA Performance. We compare RecLiRA with the best performing interaction-level RecSys MIA, MINER[5]. Considering that MINER was originally designed for knowledge graph (KG)-based recommender systems with a focus on long-tailed distribution and is not directly applicable to NCF and LGCN, we modified certain components of MINER to ensure compatibility with these models. We carefully verified the performance of the modified MINER, specifically focusing on true positive rate (TPR) under low false positive rate (FPR) conditions. Notably, the original MINER paper reported TPR about 0.15 at an FPR of 5%, while the modified version achieves comparable or superior performance at low FPR levels. Figure 1 compares RecLiRA and MINER on the interaction-level MIA. The results show that RecLiRA outperforms on all datasets and models. Its overall AUC values are above 0.9 for all the dataset/model combinations. For Music and Beauty datasets, AUC values are almost perfect, around 0.99. More importantly, we have seen high TPRs at the low FPR range (< 0.1), where high TPR/FPR ratios are observed. In contrast, MINER has a significantly worse AUC and yields lower TPRs in the low FPR range.

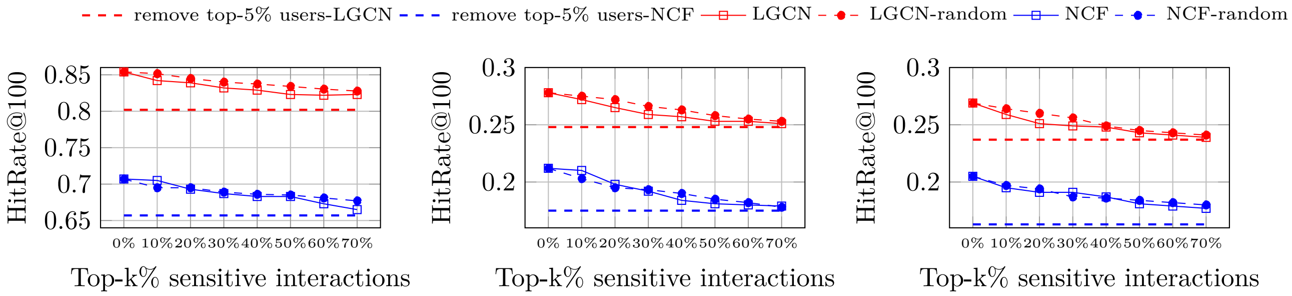
User-Level Privacy Score. To observe how the RecPS scores look like, we also generate user-level scores for 60% randomly selected users. Figure 2 shows the sorted user-level ln(TPR/FPR) values. They are all located on the range $[4.06, 4.97]$. The values are approximately divided into three bands. The top 10-20% has much higher scores, the bottom 10-20% has much lower, and the middle band shares similar values. Different RecSys models also give different score ranges. LGCN gives higher scores than NCF, seemingly consistent with its better model performance – LCGN's HR@100 is 10-20% better than NCF on these datasets.

How scores affect downstream tasks. Users have the right to remove their data from a RecSys model, and according to the laws [15, 16], the model owner



(a)MovieLens-1M (b)Amazon Digital Music (c)Amazon Beauty

Figure 3: Model utility is significantly reduced if sensitive users' interactions are all removed.



(a)MovieLens-1M (b)Amazon Digital Music (c)Amazon Beauty

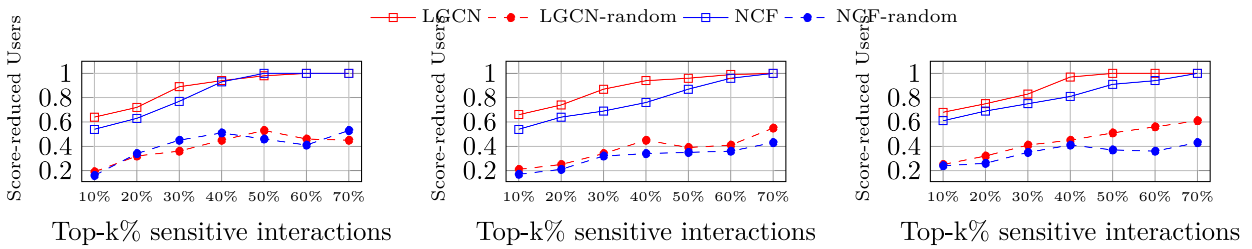
Figure 4: Partially removing sensitive users' interactions will preserve RecSys performance better. x-axis: percentage of the top sensitive interactions removed; dash lines are the worst case where all interactions of the target users are removed (the 5%-users case in Figure 3).

must ensure that this right is implemented. Consequently, a model retraining or unlearning process [11] must be applied. We study how model utility is affected in such scenarios and how scores can help both users and model owners make more informed decisions, thereby preserving more model utility. The existing RecSys unlearning strategies [11] assume that we will remove the entire set of a requested user's interactions, which, however, may lead to a significant decline in the performance of RecSys and cause the cold-start issue for this user. Figure 3 illustrates that with the top-%1 to top-5% sensitive users removed based on their privacy scores, HR@100 consistently decreases significantly across all experiments. For instance, on Amazon Digital Music, removing the top 5% of sensitive users results in a 10.79% performance drop for LGCN and an even more substantial 37.05% drop for NCF.

One may wonder: since some users may have privacy concerns about specific historical interactions but not all, can we remove only the top sensitive interactions to preserve a significantly better model utility, while still protecting privacy satisfactorily? While simulating the actual data removal request stream is impractical in this experiment, we consider an oversimplified case: removing a certain percentage of top-sensitive interactions for the top-sensitive users. Let's define a less aggressive goal. Let the minimum score of the top 5% users be the cutoff value θ. We consider these users' privacy protection goals to be achieved if their privacy scores (re-evaluated in the new dataset and model) are reduced to below θ after removing some of their sensitive interactions.

In Figure 4, we show how removing top-sensitive interactions of the top-5% users affects HR@100. The dashed lines are the model performance if all the top-5% users' interactions are removed. As expected, we see that better model utility is preserved.

Figure 5 shows how many of the top-5% users get their scores reduced below the cutoff θ with a percentage of their top sensitive interactions removed. We observed less model performance reduction. For example, on Amazon Digital Music dataset, with the top 70% sensitive interactions removed, 100% of the top 5% users successfully get their scores demoted below the cutoff score with only 15.56% reduction on NCF model performance (in comparison, 37.05% if removing all interactions); with 70% of top sensitive interactions removed, 100% of the top 5% users successfully get their scores demoted below the cut-



(a)MovieLens-1M

(b)Amazon Digital Music

(c)Amazon Beauty

Figure 5: Removing top-k% sensitive interactions of the top-5% sensitive users. Y-axis: the percentage of users whose scores are reduced below the cutoff value θ. The score-guided interaction removals preserve privacy better.

off score with only 9.71% reduction on LGCN model performance (in comparison, 10.79% if removing all). Furthermore, the score-guided interaction removal works much better than random interaction removal. Figure 5 also shows that random removal does not efficiently reduce users' privacy risk. This experiment indicates that the scores can serve as a means for us to fine-tune the data-removal strategies to effectively preserve RecSys utility.

Privacy onion effect. Since the score-generation process utilizes the training dataset, the change of dataset may affect scores. We have also conducted experiments to investigate the dynamic aspect of RecPS scores. When the top 5% of sensitive users are removed from the training data, the remaining users' privacy scores change – the so-called privacy onion effect [20] does exist. Figure 6 shows the score-change distributions, where x-axis is the score_difference = new_score - old_score. In all experimental settings, we observe the changes are relatively small, within the range $[-0.03, 0.015]$. A small portion of users (around 10-20%) get scores increased, while most users have scores reduced or unchanged. We observed that interaction-level removal can reduce the privacy onion effect. If we remove only the top 70% sensitive interactions of the top 5% sensitive users, the privacy onion effect becomes weaker. The score-increased users are reduced by around 1-7%. The privacy onion effort in the RecPS score evaluation indicates the unique complexity of privacy protection and the interaction-level scoring helps handle this complexity.

6 Conclusion

Recommender models are built using sensitive user-item interaction data, which may be vulnerable to various model-based attacks. However, current privacy-enhancing techniques are not mature enough to be deployed in recommender systems. Thus, users and model owners must assess the potential privacy risks associated with sharing specific interactions for recommender modeling. We propose the RecPS privacy scoring framework derived from the theory of differential privacy. It can estimate privacy risks at the interaction and user levels with the developed RecLiRA method. Our experimental results demonstrate that the scoring method can generate high-quality scores that effectively guide downstream privacy-enhancing tasks, such as record removal or model unlearning.

Acknowledgment. This research was partially supported by the National Science Foundation under Grant No. 2517121.

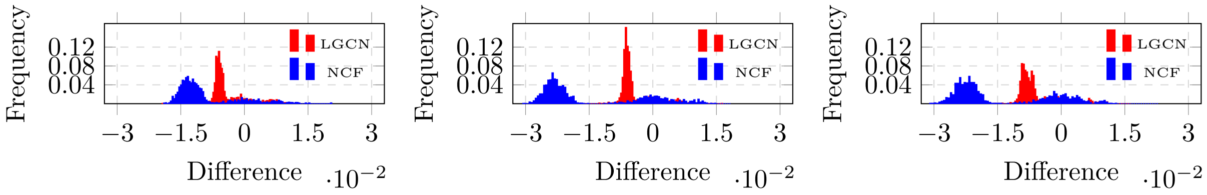
References

[1] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Biao Gong, Jun Wang, and Linxun Chen. Selective and collaborative influence function for efficient recommendation unlearning. *Expert Systems with Applications*, 234: 121025, 2023.

[2] Yingqiang Ge, Shuchang Liu, Zuohui Fu, Juntao Tan, Zelong Li, Shuyuan Xu, Yunqi Li, Yikun Xian, and Yongfeng Zhang. A survey on trustworthy recommender systems. *ACM Transactions on Recommender Systems*, 3(2):1–68, 2024.

[3] Qinyong Wang, Hongzhi Yin, Tong Chen, Junliang Yu, Alexander Zhou, and Xiangliang Zhang. Fast-adapting and privacy-preserving federated recommender system. *The VLDB Journal*, 31(5):877–896, 2022.

[4] Wei Yuan, Chaoqun Yang, Quoc Viet Hung Nguyen, Lizhen Cui, Tieke He, and Hongzhi Yin. Interaction-level membership inference at-



(a)MovieLens-1M

(b)Amazon Digital Music

(c)Amazon Beauty

Figure 6: Histogram of users' privacy score difference after removing entire top-5% sensitive users on each dataset.

tack against federated recommender systems. In *Proceedings of the ACM Web Conference 2023*, pages 1053–1062, 2023.

[5] Da Zhong, Xiuling Wang, Zhichao Xu, Jun Xu, and Wendy Hui Wang. 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*, pages 3433–3442, 2024.

[6] 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 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 864–879, 2021.

[7] 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 the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1959–1968, 2022.

[8] Peter Müllner, Elisabeth Lex, Markus Schedl, and Dominik Kowald. Differential privacy in collaborative filtering recommender systems: a review. *Frontiers in big Data*, 6:1249997, 2023.

[9] Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra. A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.

[10] Fatemeh Rezaimehr and Chitra Dadkhah. A survey of attack detection approaches in collaborative filtering recommender systems. *Artificial Intelligence Review*, 54:2011–2066, 2021.

[11] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. Recommendation unlearning. In *Proceedings of the ACM Web Conference 2022*, pages 2768–2777, 2022.

[12] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022.

[13] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[14] Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. A review on machine unlearning. *SN Computer Science*, 4(4): 337, 2023.

[15] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[16] California consumer privacy act of 2018 (ccpa). Legislation enacted by the State of California, 2018. Available at https://oag.ca.gov/privacy/ccpa.

[17] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[18] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632, 2021.

[19] Zhihao Zhu, Chenwang Wu, Rui Fan, Defu Lian, and Enhong Chen. Membership inference attacks against sequential recommender systems. In *Proceedings of the ACM Web Conference 2023*, pages 1208–1219, 2023.

[20] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. In S. Koyejo,

S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 13263–13276. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/564b5f8289ba846ebc498417e834c253-Paper-Conference.pdf.

[21] Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy auditing with one (1) training run. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 1631–1648, 2023.

[23] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[24] 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 the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[25] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.

[26] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[27] Gábor Takács and Domonkos Tikk. Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 83–90, 2012.

[28] Yuechun Gu, Jiajie He, and Keke Chen. Demo: Ft-privacyscore: Personalized privacy scoring service for machine learning participation. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 5075–5077, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706363. doi: 10.1145/3658644.3691366. URL https://doi.org/10.1145/3658644.3691366.

[29] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 61(6):3469–3481, 2015.

[30] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33: 22205–22216, 2020.

[31] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.

[32] Zachary R Smith and Craig S Wells. Central limit theorem and sample size. In *annual meeting of the Northeastern Educational Research Association, Kerhonkson, New York*, 2006.

[33] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

[34] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 507–517, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee. ISBN 9781450341431. doi: 10.1145/2872427.2883037. URL https://doi.org/10.1145/2872427.2883037.

[35] Grant Ho, Aashish Sharma, Mobin Javed, Vern Paxson, and David Wagner. Detecting credential spearphishing in enterprise settings. In *26th USENIX security symposium (USENIX security 17)*, pages 469–485, 2017.

[36] Alex Kantchelian, Michael Carl Tschantz, Sadia Afroz, Brad Miller, Vaishaal Shankar, Rekha Bachwani, Anthony D Joseph, and J Doug Tygar. Better malware ground truth: Techniques for weighting anti-virus vendor labels. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, pages 45–56, 2015.

[37] J Zico Kolter and Marcus A Maloof. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research*, 7 (12), 2006.