FedSIG: Privacy-Preserving **Fed**erated Recommendation via **S**ynthetic **I**nteraction **G**eneration

Thirasara Ariyarathna
University of New South Wales
Sydney, Australia
thirasaradevanmini@gmail.com

Salil Kanhere
University of New South Wales
Sydney, Australia
salil.kanhere@unsw.edu.au

Meisam Mohammady
Iowa State University
Iowa, USA
meisam@iastate.edu

Hye-Young (Helen) Paik
University of New South Wales
Sydney, Australia
h.paik@unsw.edu.au

***Abstract*—Recommendation Systems (RS) play an important role in our everyday life in this data-driven digital era by providing users with the convenience of navigating the plethora of available choices. An RS collects user behavioural data to provide them with valuable suggestions. The growing privacy concerns regarding private data collection have led to the use of Federated Learning (FL) to implement RS. However, many research works have exposed the privacy leakages in FL gradient sharing. The embedding gradients shared by FL users during the RS model training can be used to infer the items that users have interacted with. Existing defences, such as random noise injection or pseudo-interaction sampling to obfuscate the privacy-sensitive information reflected by the shared gradients. However, these techniques provide limited protection and often result in substantial degradation of recommendation performance, leading to an unfavourable privacy–utility trade-off.**

In this paper, we propose FedSIG (Federated Synthetic Interaction Generation), a defence mechanism that mitigates user-item interaction inference in federated recommendation systems by generating synthetic interaction data using generative models. The generated items are selectively used to replace or augment real user interactions, thereby obfuscating sensitive data while preserving user preference signals. To further enhance utility, we design an item selection module based on an attention mechanism to identify less contributive interactions for replacement. Extensive experiments conducted on five real-world datasets and two state-of-the-art recommendation models demonstrate that FedSIG achieves a significantly improved privacy–utility balance compared to existing approaches, effectively reducing inference success rates while maintaining competitive recommendation accuracy.

***Index Terms*—Federated Learning, Recommendation, Privacy-Protection**

I. Introduction

Recommendation systems (RS) are crucial in today's digital age, significantly enhancing the user experience across various platforms. They are a powerful tool for navigating the digital landscape, helping users discover relevant content, products, or services tailored to their preferences and behaviors. This is particularly important in sectors like e-commerce, entertainment, news, and healthcare, where the sheer volume of information can be overwhelming. For example, many service applications, such as Amazon, Google, YouTube, etc., use RS to help users navigate the plethora of available choices. Several recent research works have proposed improvements in recommendation systems [1], [2]. By accurately predicting user preferences, recommendation systems not only streamline the decision-making process for users but also increase engagement, customer satisfaction, and, ultimately, business revenue [3]–[7]. Furthermore, they can foster a sense of personalization and connection, making users feel understood and valued. However, it is crucial to balance the benefits of personalization with respect for user privacy, ensuring that data collection and usage are transparent and ethical.

While RS offer numerous benefits, it also raises significant privacy concerns. RS are mainly trained on users' past interaction information over time, which requires collecting vast amounts of user behavioral data by those service applications [1], [2], [8] to make accurate recommendations. This data can include user attributes, behaviors, social relations, and context information. The collection and use of such sensitive information can lead to potential privacy breaches such as identity inference, behavior tracking, location tracking, etc. Therefore, concern for privacy in centralized recommendation has led to training recommendation models in a decentralized fashion. Federated Recommender Systems (FedRecs) have been increasingly explored in recent years as they use intermediate parameters instead of real user data to train the RS. Furthermore, FedRecs enables the system to learn from data across multiple devices while minimizing the risk of sensitive information exposure. By reducing the need for data transmission, FedRecs also lower the risk of data breaches during transit. Figure 1 shows an overview of a federated recommendation system on a service level where service providers initialize the recommendation model, which is then trained on client behavior data locally for several rounds. In each round, users share the local gradient updates with the service provider, and the service provider will aggregate the local updates to obtain the global model update for that round, which is then shared again with users for the next round. This process repeats until a converged FedRec model is obtained.

Previous studies have shown that gradient updates in decentralized training still expose sensitive information, particularly user–item interactions, through inference attacks [9]–[12].

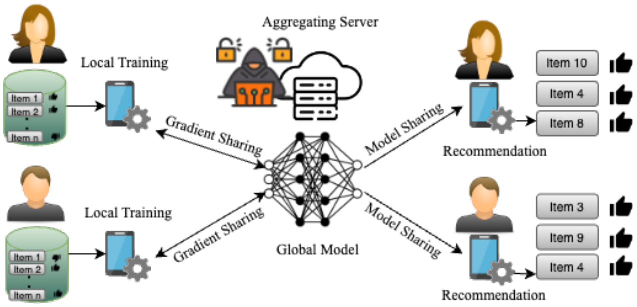

Fig. 1: **Workflow of Federated Recommendation System (FedRS)**. A central server aggregates user updates each round, but gradients may still leak sensitive interaction information.

Such vulnerabilities risk exposing assets, including individual preferences, browsing patterns, and consumer histories, which can in turn lead to downstream harms such as targeted profiling, exposure of sensitive attributes, or manipulation. [9], [10].

FedRecs are vulnerable to such inference attacks as most RS learn user and item representations (embeddings) by exploiting users' historical interaction data. There are two main privacy inferences commonly seen in FedRecs [13]. The first is user attribute inference [14], and the second is user-item interactions inference [9]. In this work, we focus on protecting user-item interactions. In FedRecs, the associated item set of a particular user can be easily inferred by analyzing the change in embedding updates. Many FedRecs use randomly sampled pseudo-interacted items [15], [16] to minimize the user-item interaction inference. Here, the inference attack success rate is directly proportional to the pseudo-interaction sampling rate. Therefore, to effectively reduce the inference success rate, at least up to a random guess, the pseudo-interacted item set should equal an actual interacted item set. However, randomly sampling an equal number of pseudo-interactions leads to a drastic decrease in recommendation accuracy. Furthermore, adding randomly sampled noise to the gradients is also used in several works [15] to achieve a privacy guarantee for the underlying user data. Differential Privacy (DP) techniques are typically used in FL to implement privacy protection in these systems. However, this method of adding noise also results in a drastic decrease in recommendation performance, as we later discuss in IV section.

This work proposes FedSIG: Synthetic Interaction Generation for FedRecs to generate synthetic user-item interactions using Generative Adversarial Networks (GAN) to address the issues as mentioned above. We employ a generative model to generate items that closely represent the users' preference information and use these generated items to replace or augment the actual user-item interactions. This approach can protect privacy-sensitive user-item interactions while capturing user preferences, which helps preserve recommendation performance. The successful creation of synthetic data that preserves privacy has been evidenced in several instances. For instance, several works have focused on generating synthetic location data derived from actual locations to safeguard the existence and true location of each individual in the original dataset [17]–[19]. Similarly, synthetic health data are generated to protect the private data of patients [20]. Introducing such privacy-preserving synthetic data in RS eliminates the risk of privacy-sensitive user interactions being identified by potential adversaries, thereby reducing the likelihood of behavioral and identity inferences.

The proposed synthetic interaction generation model can provide a privacy guarantee for the original data at the user-item interaction level. Specifically, for each user participating in training FedRec, we design a selection module to select the items that are replaced/augmented by the generated synthetic items. To maximize the utility of synthetic data inspired by [21], we employ an attention mechanism to estimate the contribution of each item (e.g., movies, locations, songs, etc.) to the user's preference. We then compare different settings to obfuscate the original user-item interactions, considering negative sampling [9]. The user-item interaction inference on FedRecs without negative sampling is straightforward compared to traditional membership inference in other ML tasks (e.g. classification), as we can observe the item-embedding change before and after the local update. However, if negative sampling is used when training the FedRec model, then it adds inherent noise to item embeddings. Therefore, more sophisticated attacks such as Interaction Membership Inference Attack (IMIA) [9] is used to infer the user-item interactions. Therefore, in this paper, we use IMIA to evaluate the robustness of FedSIG.

We use the self-attention mechanism to select items from the positive interacted item set and replace or augment the positive item interactions with synthetic items to increase the utility of the recommendation. Then, we use the proposed synthetic item generator to generate the synthetic item using the selected items while considering the user's preferences. In the first method, we replace the least contributing items with generated items, and in the second method, we augment items by increasing the proportion of contributing items. Our findings show that replacing negative samples gives a better privacy-utility balance. Extensive experiments on five real-world datasets have been carried out to demonstrate the effectiveness of our method. In summary, the main contributions of this work are summarized as follows:

- We highlight the limitations of existing privacy-preserving methods in FedRecs and propose a novel generative model to generate synthetic interactions for users to obfuscate the original user-item interactions.
- We experiment with different approaches to generate the item interaction datasets (item replacement and item augmentation) to show the privacy protection obtained by each method and identify the best strategy.
- Extensive experiments conducted on five real-world datasets show that the proposed method of item replacement with synthetic items outperforms the utility achievement of SOTA privacy-preserving FedRecs.

The rest of the paper is organized as follows. Section II provides the notions, notations, definitions and key ideas used in the paper. In section III, baseline PRR models, threat model and the proposed FedSIG model are presented. Section IV presents the experiments and results analysis. We discuss and analyze the privacy protection capability of FedSIG in section V Related works are presented in section VI, and concluding remarks are given in section VII.

II. Preliminaries

This section introduces the federated recommendation setting, the notations used throughout the paper, and the membership attack model that motivates our defense. We conclude with the threat model.

A. Federated Recommendation

A traditional recommendation system consists of two entities: *users* and *items*. The system recommends items to users based on historical interactions (implicit feedback) or explicit ratings. Formally, let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of users (clients) and items, respectively. Each user $u \in \mathcal{U}$ holds a local dataset $\mathcal{D}_u$, consisting of interaction records $b = (u_j, v_j, r_j)$. In explicit settings, $r_j \in \mathcal{R}$ denotes a rating value. In implicit settings, $r_j = 1$ if user u_j interacted with item v_j and $r_j = 0$ otherwise. In this work, we adopt the implicit feedback formulation.

We consider the standard Federated Learning (FL) setting, where a central server coordinates training using the FedAvg algorithm [22]. Each client performs multiple steps of local Stochastic Gradient Descent (SGD) and shares model updates ω_i with the server. The global model is updated as the average of local gradients:

$$\omega' = \frac{1}{N}\sum_{i=1}^{N} \omega_i,$$

where ω' is the global model update and N is the number of participating users.

B. Notations

Table I summarizes key notations.

TABLE I: Notations used in this paper

Symbol	Definition
$\mathcal{D}_u$	Local dataset of user u
$\mathcal{D}_{aux}$	Auxiliary item–interaction dataset
$\mathcal{U}$	Set of all users
$\mathcal{V}$	Set of all items
$\Phi(i)$	Embedding vector of item i
$\Phi(u)$	Embedding vector of user u
$\mathcal{M}_f$	Global federated recommendation model
$\mathcal{M}_u$	Local model of user u
$\mathcal{M}'_u$	Altered model of user u after FedSIG

C. Membership Inference Attacks

Following [23], we formalize membership inference in the FL setting.

Definition 1 (Membership experiment $Exp^M(\mathcal{A}, A, \mathcal{D})$): Let $\mathcal{A}$ be an adversary, A be a learning algorithm, $\mathcal{D}_u$ be a target user's dataset, and $\mathcal{D}_{aux}$ be the adversary's auxiliary dataset. For each $z \in \mathcal{D}_{aux}$, define $b = 0$ if $z \in \mathcal{D}_u$ and $b = 1$ otherwise. Then

$$Exp^M(\mathcal{A}, A, z) = 1 \quad \text{if} \quad \mathcal{A}(z, A(\mathcal{D}_u), \mathcal{D}_{aux}) = b.$$

This experiment quantifies $\mathcal{A}$'s ability to decide whether z belongs to $\mathcal{D}_u$.

D. Threat Model

We assume the standard federated recommendation architecture (Fig. 1), where clients train locally on $\mathcal{D}_u$ and send model updates $\mathcal{M}_u$ (including item embeddings $\Phi(i)$) to an aggregator.

Our assumptions are as follows:

i **Adversary.** The aggregator is *honest-but-curious*: it follows the FL protocol correctly but attempts to infer sensitive user information from observed updates.

ii **Adversarial goal.** The server seeks to determine whether a target item i' is part of a user's dataset, i.e., $i' \in \mathcal{D}_u$, thereby revealing user–item interactions.

iii **Auxiliary data.** The adversary may hold an auxiliary dataset $\mathcal{D}_{aux}$ containing user–item interactions drawn from the same domain as $\mathcal{V}$. This enables training of shadow models for inference attacks.

iv **Channel assumption.** Communication between clients and server is over a secure channel. We do not consider integrity or availability attacks (e.g., parameter tampering or denial-of-service).

This model captures realistic privacy risks: although raw data never leaves the client, updates expose embeddings that leak membership information, enabling inference of sensitive user preferences and behaviors.

III. Methodology

This section will first describe the baseline FedRecs and threat model used in this paper and then present details of the proposed FedSIG defence mechanism. We consider the typical FL setting and the server as the adversary trying to infer user-item interactions.

A. Base Recommendation Systems

Most ML tasks can be transformed to work in an FL setting. Based on the same argument, most ML-based recommendation systems can be implemented in the FL setting. In this paper, we implemented two SOTA centralized recommendation systems in an FL setting, namely Neural Matrix Factorization (NeuMF) [1] and Light Graph Convolutional Network (LightGCN) [2]. The former is implemented using feedforward neural networks, and the latter using Graph neural networks (GNN). These recommendation systems train an ML model

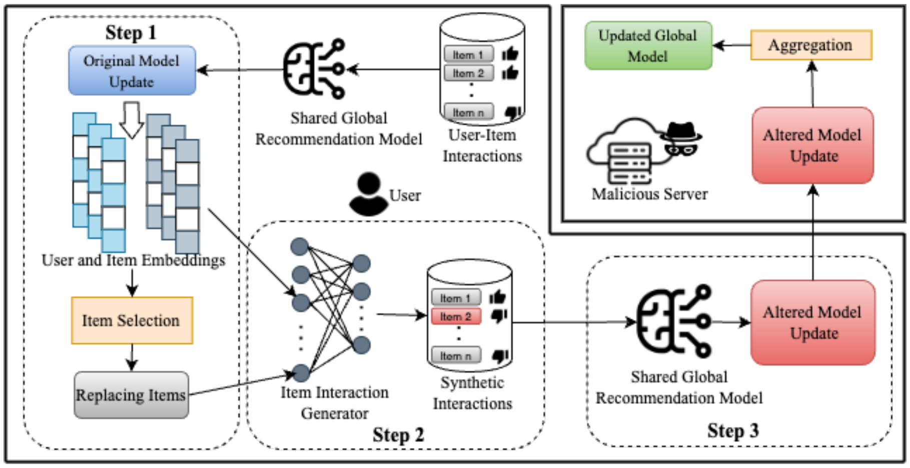

Fig. 2: Overview of the proposed defence

that learns to model user preferences using their historical item interactions.

1) NeuMF: This model uses a multi-layer perceptron (MLP) to learn user-item interactions and then uses a combination of MF and MLP to do the collaborative filtering (CF). Let $\Phi(i)$ and $\Phi(u)$ denote the latent vectors (embeddings) of user u and item i, then NeuMF estimates an interaction r_{ui} as follows:

$$\hat{r}_{ui} = f\{\Phi(\mathcal{U})^T\Phi(i), \Phi(\mathcal{V})^T\Phi(u) \;|\Phi(\mathcal{U}), \Phi(\mathcal{V}), \Theta_f\} \quad (1)$$

Here, $\Phi(\mathcal{U}) \in \mathbb{R}^{M\times K}$ and $\Phi(\mathcal{V}) \in \mathbb{R}^{N\times K}$ denotes the latent factor matrix for users and items respectively where M is the total number of users and N is the total number of items. Θ_f denotes the interaction function f. Therefore, equation 1 shows the NeuMF learning function.

NeuMF will minimize the loss $\mathcal{L}$ as in equation 2 to learn the interaction prediction function.

$$\mathcal{L} = -\sum_{(u,i)\in(\mathcal{U},\mathcal{V})} r_{ui}\log\hat{r}_{ui} + (1-r_{ui})\log(1-\hat{r}_{ui}) \quad (2)$$

2) LightGCN: In recommenders implemented using GNNs, users and items are represented as a bipartite graph. LightGCN performs graph convolution iteratively, aggregating features of neighbours as the new representation of a target node. Therefore, in this model, user and item embeddings are learnt by propagating their neighbour nodes' embeddings according to equations 3 and 4.

$$\Phi(u)^l = \sum_{k\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}\Phi(i)^{l-1} \quad (3)$$

$$\Phi(i)^l = \sum_{j\in\mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}}\Phi(u)^{l-1} \quad (4)$$

Here, $\mathcal{N}_u$ and $\mathcal{N}_i$ denote the sets of users u's and item i's neighbours, while l denotes the propagation layer. We aggregate all layers' embeddings together after propagation through all L-layers as the final user and item embeddings as follows:

$$\Phi(u) = \sum_{l=0}^{L}\Phi(u)^l, \Phi(i) = \sum_{l=0}^{L}\Phi(i)^l \quad (5)$$

Then, according to equation 1, the user interactions $\hat{r}_{ui}$ are learned by minimizing the loss presented in equation 2.

B. Inference Attacker

This section presents an overview of the implementation of the Interaction-level Membership Inference Attack (IMIA) [9], which aims to infer the positively interacted items of a given user. When a user u_i shares the local model update $M^t_{u_i}$ with the server after communication round t, the server can observe which item embeddings $\Phi(v_j)$ have been updated. This allows the server to infer which items v_j were involved in the user's local training. However, such inference is inconclusive, as an updated item v_j may have been sampled as either a positive or negative instance. Therefore, the server seeks to determine whether an updated item satisfies $r_{ij} = 1$ in the user's local dataset $\mathcal{D}_{u_i}$.

Let V^t_u denote the set of items whose embeddings were updated by user u_i after round t. The objective of IMIA is to infer the value of r_{ij} for each item $v_j \in V^t_u$. To achieve this, the server constructs a fake dataset $\mathcal{D}^{\text{fake}}_{u_i}$ by randomly assigning binary ratings to the items in V^t_u, guided by a predefined negative sampling ratio η. For instance, if $\eta = 1:4$, the server randomly selects 20% of the updated items as positive and the remaining 80% of updated items as negative.

Using $\mathcal{D}^{\text{fake}}_{u_i}$, the server trains a shadow model $M^{\text{fake}}_{u_i}$ to emulate the training behavior of the user. Upon completion of training, the server calculates the distance between the item embeddings produced by $M^{\text{fake}}_{u_i}$ and those in the received model update $M^t_{u_i}$. The top $\gamma \cdot |V^t_u|$ items with the smallest distances are then selected as correctly identified user interactions, under the assumption that smaller embedding shifts indicate a higher likelihood of being true positives. These inferred ratings are fixed in subsequent iterations.

This procedure is repeated iteratively until the server completes the reconstruction of the user's positive interaction set.

C. Federated Synthetic Interaction Generation

In this subsection, we will present the details of the proposed FedSIG defence against user-item interaction inference in FedRecs. The learning process begins with the server sharing the global recommendation model and the initial generator, trained on public data. As shown in Fig. 2, the user u's local training procedure involves three steps. The first step is to obtain the original gradients from the FL user by getting the local update $\mathcal{M}_u$ of the shared global recommendation model $\mathcal{M}_f$ and selecting the items that contribute less to the user preference, depending on the embeddings from the local update. The second step is to generate a set of synthetic user-item interactions based on the selected items. In the third step we obtain the local update on generated synthetic interactions and upload it to the server for aggregation. Each of these steps is described in detail in the following subsections.

1) Step1: Item Selection: In this step, we select interacted items I_u in the original dataset of the user u to be replaced by the generated interactions S_u. We first obtain the local update M_u of the shared global model M_f. We can extract $\Phi(\mathcal{V})$ from M_u. Then, we determine the number of items to be replaced using the replacement ratio $R = |I_u \cap S_u|/|I_u|$. The more items that are replaced, the less the data leakage risk. If we choose the set of item interactions that contribute to the user's preference least, then we can maximize the utility of the recommendation task. For this, we employ the representation of items i that u interacted with, and then the user's preferences can be presented as in equation 6:

$$\Phi(u) = \frac{1}{|I_u|} \sum_{i \in I_u} a_{ui} \Phi(i) \tag{6}$$

Here, $a_{ui} \in A_u$ is a trainable parameter denoting the attention weight of item i for u's preference and $\Phi(i) \in \Phi(\mathcal{V})$. A_u is the weight set of the items interacted with by user u. We calculate p_u by getting the attention a_{ui} from the attention mechanism.

Then we select the replacing item set I_u^r of user u as follows:

$$I_u^r = i | I \in I_u, a_{ui} \in A_u^R \tag{7}$$

Here, A_u^R represents the R percentage of items whose attention weights are smaller than those of others.

2) Step2: Synthetic Data Generation: In this step, we generate a set of synthetic interactions based on the item set we selected in Section III-C1. Here, we fine-tune a global generator trained on a public auxiliary dataset $\mathcal{D}_{aux}$ to accommodate user preferences. Training the generator to incorporate user preferences will generate items likely to be chosen by the user. The generator uses the concatenation of the user embedding vector $\Phi(u)$ and the selected item embedding vector $\Phi(i)$ where $i \in \mathcal{I}_u^r$ as the input to get a latent feature for the output, which is given by the equation 8.

$$\mathcal{Q}_{u,i} = W(\Phi(u), \Phi(i)) + b \tag{8}$$

Here, W and b are weight and bias vectors. Then, we calculate the similarity $s_{u,i}$ between the latent feature $\mathcal{Q}_{u,i}$ and all item embeddings $\Phi(\mathcal{V})$ according to equation 9, which will be learned in generator training.

$$s_{u,i} = \Phi(\mathcal{V})^T \mathcal{Q}_{u,i} \tag{9}$$

Finally, we estimate the probability distribution over all candidate items as $r_{u,i} = softmax(s_{u,i})$ and then the top n candidate items are selected according to the replacement ratio R.

Inspired by the CF recommendation from [], we assume that user u will give a higher score to item i if they prefer i. To maximize the recommendation utility, we aim to generate user-preferred synthetic items. Therefore, we have to maximize the similarity $s_{u,i}$ by minimizing the generated loss $\mathcal{L}_g$ according to equation 10.

$$\mathcal{L}_g = \sum_{(u,v)} = -ln\sigma(\Phi(u)^T \Phi(i)) \tag{10}$$

After the synthetic interaction generation, we again obtain the altered local model update M_u' from the shared global recommendation model M_f and upload it to the server for aggregation.

3) Step3: Federated Aggregation: The final step involves federated aggregation of client local updates. For aggregation, we use the FedAvg algorithm proposed in [22] to update the global parameters according to the equation 11.

$$\Theta_{t+1} = \sum_{i=0}^{M} \frac{\Theta_t}{M} \tag{11}$$

As defined in the threat model in section II-D, the curious server would do the federated averaging process honestly according to equation 11. However, the server would receive altered local updates $\mathcal{M}_u'$ from users to infer the user interactions. We use the inference model proposed in [9] to infer the positively interacted item set.

IV. Experiments and Results

In this section, we evaluate the effectiveness of FedSIG with respect to two primary objectives: (i) preserving end-user privacy and (ii) maintaining the utility of the recommendation model. To this end, we address the following research questions:

- **RQ1:** To what extent do the proposed FedSIG defense strategies—specifically, item replacement and item augmentation—improve recommendation utility compared to state-of-the-art baseline?
- **RQ2:** How effective is FedSIG in reducing the attack success rate of the Interaction-level Membership Inference Attack (IMIA)?

Privacy-Setting	Dataset	NeuMF		LightGCN	
		Precision	Recall	Precision	Recall
Original	MovieLens	0.0926	0.2474	0.1327	0.2823
	TYO-PoI	0.0083	0.1289	0.0622	0.1633
	NYC-PoI	0.0091	0.0664	0.0146	0.0700
	Electronics	0.0372	0.1350	0.0742	0.1646
	Clothes	0.0084	0.0713	0.0120	0.1095
Replacement	MovieLens	0.0715	0.2052	0.1183	0.2469
	TYO-PoI	0.0052	0.0862	0.0440	0.1024
	NYC-PoI	0.0085	0.0511	0.0120	0.0580
	Electronics	0.0274	0.0961	0.0694	0.1257
	Clothes	0.0079	0.0752	0.0106	0.0813
Augmented	MovieLens	0.0774	0.2441	0.1497	0.2904
	TYO-PoI	0.0135	0.1352	0.0473	0.1692
	NYC-PoI	0.0096	0.0557	0.0186	0.0736
	Electronics	0.0294	0.1387	0.0718	0.1855
	Clothes	0.00841	0.0868	0.0169	0.1247

TABLE II: **Recommendation performance comparison original dataset and generated datasets. We compared original dataset with replacement and augmentation settings with 0.2 replacement and augmentation ratio.**

- **RQ3:** In what ways does similarity-based synthetic item interaction generation contribute to achieving a more favorable privacy-utility trade-off?

A. Experimental Setup

1) Datasets: - We conducted evaluations using five real-world datasets, MovieLens (ML) and Foursquare, in two cities, New York (NY) and Tokyo (TYO) and Amazon, using two item paradigms, namely, Electronics and Fashion. The first dataset is the MovieLens dataset [24], which is widely used to evaluate RS. This dataset contains one million ratings, and each user has at least 20 ratings. The ML dataset is an explicit feedback dataset where movie ratings range from a score of 1 to 5. We intentionally chose to investigate the performance of FedSIG from the implicit feedback (whether a user has interacted with an item, i.e., whether the user has watched a movie and liked it) [25] of the explicit feedback. We transform explicit feedback into implicit data, where each record is marked as 1 or 0 depending on whether the user rated the item (movie). The second and third datasets are user check-in data obtained from the Foursquare dataset [26] by selecting the Points of Interest (PoIs) from New York City (NYC) and Tokyo (TYO). The fourth and fifth datasets are Amazon product review datasets. Similar to the ML dataset, we transformed user reviews into implicit feedback from explicit feedback. For this study, we do not focus on solving the cold start problem, in which we study how to recommend items to users with fewer or no previous interactions, so we conducted a data preprocessing step to make the data denser. We preprocessed the data according to previous studies [14], [27], [28] by eliminating users with less than 10 interactions and combined users to represent one client in the FL system, such that one client has at least 20 item interactions. Table III shows the statistics of the datasets after preprocessing.

Dataset	#users	#items	#int	sparsity
MovieLens-1M	6040	3900	1000209	95.76%
NYC PoI	29858	48981	1027370	99.91%
TYO PoI	18737	32510	1278274	98.94%
Amazon Clothes	18209	17317	150889	99.95%
Amazon Electronics	13174	5970	103593	99.87%

TABLE III: **Statistics of five datasets after preprocessing.**

2) Evaluation Metrics: To measure the *recommendation accuracy*, we adopt an evaluation metric widely used in previous works [29]–[31] called *leave-one-out*. We reserve the most recent interaction of each user for the test set, while the rest of the data is employed for training. Given the extensive time required to rank all items for each user, we adopted a widely used approach [25], [32] that randomly selects 100 items not previously interacted with by the user and ranks the test item among these 100 items. The effectiveness of a ranked list is assessed by the Hit Ratio (HR) [33]. We limit the ranked list to 20 for all metrics. HR essentially determines whether the test item appears in the top-20 list.

We measured the *attack success* rate of user-item interaction inference with adversarial advantage [34]. Adversarial advantage calculates the gain from a membership attack by taking the difference between true and false positive rates and neutralizing a random guess by offering an advantage of 0.

3) Baselines: We compared FedSIG with Random pseudo-interaction addition (RI) for three different replacement ratios (R). RI is used in several recent SOTA research works [15] to ensure the privacy of item interactions of users by hiding actual interactions among randomly chosen items with which the user has not interacted. Furthermore, we compare the recommendation performance.

B. Recommendation Performance After Defence

In this section, we address **RQ1**. Table II presents the Precision@20 and Recall@20 metrics for the LightGCN and NeuMF recommendation algorithms across five datasets. As shown in Table II, the absolute precision and recall values are relatively low, often below 10%. This is expected due to the extreme sparsity of the Foursquare and Amazon datasets (over 99% sparse, see Table III) and the leave-one-out evaluation protocol with 100 negative samples, which creates a strict ranking task. These results are consistent with prior FL-based recommendation work, where absolute metrics are lower than centralized settings but relative improvements remain meaningful.

For all experiments, we set the negative sampling ratio to 1:4, where four negative samples are selected for each positive interaction. Uniform negative sampling [1] is employed as the sampling strategy.

We evaluate two privacy-preserving settings—replacement (FedSIG-R) and augmentation (FedSIG-A)—to generate synthetic datasets and compare their recommendation performance with the baseline. We begin by examining the performance of the recommendation models on the original dataset. Overall, LightGCN outperforms NeuMF, yielding higher precision and recall values. This may be attributed to LightGCN's more effective embedding representation learning, which in turn facilitates more accurate generation of synthetic interactions.

Next, we assess the performance of the models after applying the FedSIG defense under both replacement and augmentation settings. In general, the replacement setting yields lower performance compared to augmentation, particularly at a 0.2 replacement/augmentation ratio. This difference may be due to all positive training samples retained in the augmentation setting, which better reflect user preferences compare to replacement setting. As expected, the recommendation performance on the replacement-based dataset is lower than on the original dataset as original interactions are being replaced by similar item interactions. Nevertheless, the synthetic dataset generated via replacement still captures user preferences reasonably well, resulting in only a marginal decrease in utility.

Finally, we analyze the impact of the replacement ratio R, which determines the proportion of items replaced or augmented during synthetic dataset generation. Figure 3 illustrates the recommendation performance across different datasets using the LightGCN model, which is selected due to its superior performance relative to NeuMF. The utility of the replacement-based datasets tends to decline as the replacement ratio increases. This may be because replacing a larger portion of the original dataset makes it more difficult to accurately model user preferences. In contrast, the utility of augmented datasets fluctuates slightly with the augmentation ratio and varies across datasets. Notably, in some cases, augmentation leads to improved performance over the original dataset.

C. Attack Resilience

This section will answer the question **RQ2**. To measure the attack success, we used the IMIA [9] (details in section III-B) on the LightGCN recommendation model trained on three datasets with different replacement ratios. We used LightGCN model to evaluate attack resilience because LightGCN performs well compared to NeuMF in recommendation. Figure 4 shows the attack success rate of FedSIG and RI for different replacement/augment ratios RR. Although replacing and augmenting items introduce noise to positively interacted items, the attack resilience is higher for datasets with replacement when compared to augmentation. This may be because the IMIA attack leverages the similarity of item embedding vectors to infer the user-item interactions. Because the augmented datasets still contain all the positive interactions among synthetic interactions IMIA adversary can infer the interacted items with higher precision. The attack success of RI with augmentation is greater than the attack success rate of FedSIG with augmentation. The reason for this might be that augmented items in FedSIG help it to learn more meaningful embeddings than RI, which aids in inferring similar embeddings. Therefore, we can conclude that IMIA attacks can be effectively mitigated using item replacement compared to pseudo-interaction addition.

D. Privacy-Utility Trade-Off

First, to get an idea about the privacy-utility trade-off after applying the most common SOTA privacy-preserving mechanism, we conducted the RS training with LDP for different noise strengths λ. Figure 5 shows the recommendation accuracy (Hit@20) in the right y-axis and attack success rate in the left y-axis for different noise strengths λ. We can see that to lower the adversarial advantage by 21%, we have to compromise the recommendation accuracy by 72% for the TYO dataset. When the λ increases, all the gradients are perturbed equally proportionate to λ. Therefore, the overall impact on recommendation accuracy is much more significant than on adversarial advantage, which only utilizes item embeddings for the inference. However, when comparing the performance of FedSIG, even with replacement, we can achieve better performance by lowering the adversarial advantage approximately by 20% on the Movie Lens and TYO datasets. When comparing the adversarial advantage of FedSIG-R and FedSIG-A we can see that for all the RRs the item interaction inference is below 0.5 for FedSIG-R. Therefore, by choosing 0.2 as the replacement ratio in FedSIG-R we can achieve better privacy-utility balance. This because for all datasets, FedSIG-R with 0.2 item replacement, we can achieve comparably higher recommendation accuracy.

V. Privacy Analysis

This section evaluates the privacy characteristics of three model design choices in FedSIG: negative sampling, synthetic interaction generation (augmentation vs. replacement), and

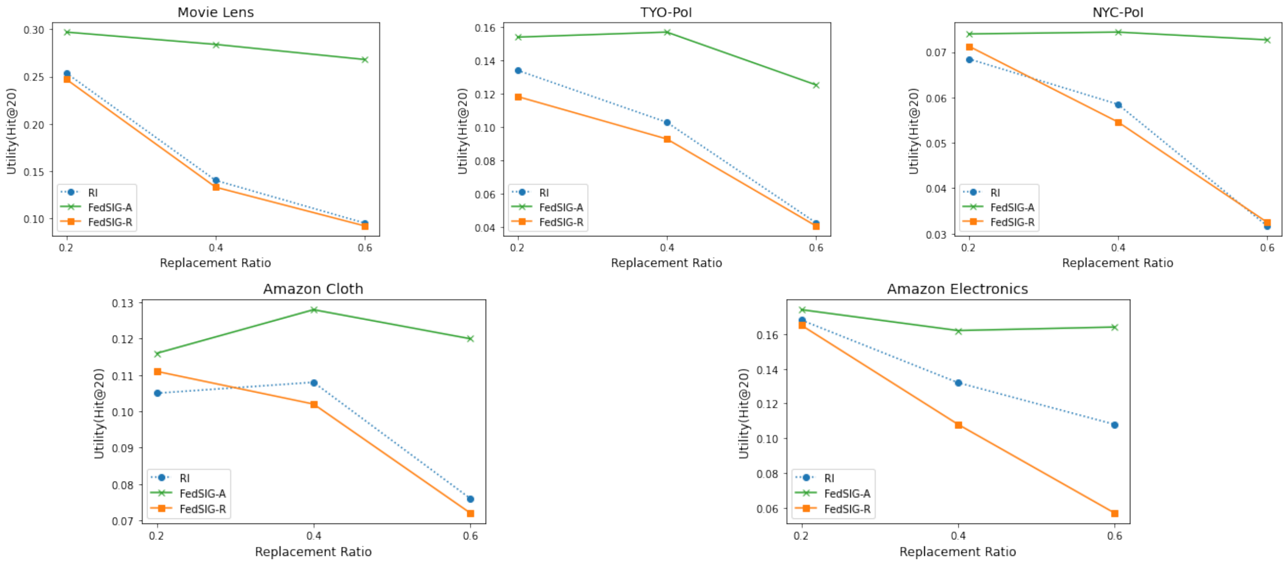

Fig. 3: **Recommendation accuracy after FedSIG for five datasets. Different lines show the different replacement settings: Positive sample replacement, Negative sample replacement and Random replacement.**

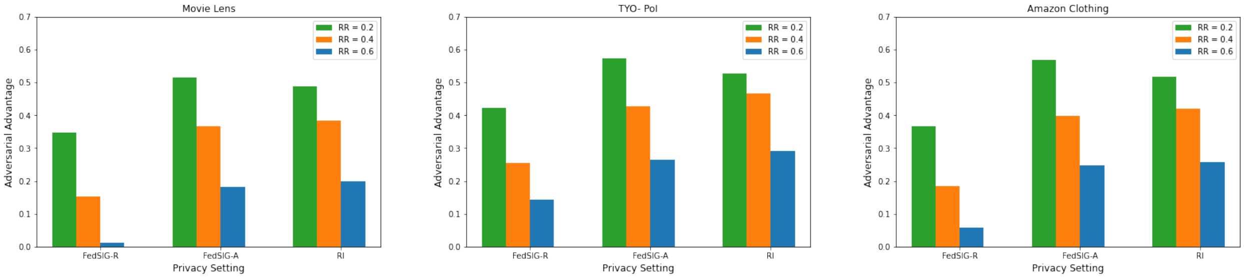

Fig. 4: **Attack success rate measured by Adversarial Advantage for 3 different privacy settings. FedSIG-R is synthetic interaction addition with replacement, FedSIG-A is a synthetic interaction addition with augmentation, and RI is a pseudo interaction addition with augmentation.**

noise-injection mechanism (FedSIG vs. DP-SGD). The analysis focuses on vulnerability to inference attack, particularly Interaction-level Membership Inference Attack (IMIA).

A. *Effect of Negative Sampling*

Training recommendation models exclusively with positive samples introduces a distinct privacy vulnerability. In federated settings, only the embeddings of items with which a user has interacted are updated during local training. As a result, an adversary monitoring the trajectory of item embedding updates over communication rounds can infer user-item interactions with high precision.

Introducing negative samples mitigates this risk. When both positive and negative items are involved in training, a broader subset of item embeddings undergo updates. Adding random pseudo-interactions (baseline RI) also extends the similar idea and expands the subset of updated item embeddings. This obfuscates the identity of true interactions by introducing ambiguity into the gradient update. The adversary cannot easily distinguish which updates correspond to actual user preferences, thereby reducing the effectiveness of inference attacks. Thus, from a privacy standpoint, negative sampling serves as a stochastic obfuscation mechanism that enhances protection against user interaction leakage.

B. *Effect of Item Replacement vs Item Augmentation*

Both item augmentation and item replacement aim to reduce the identifiability of original user interactions by modifying the set of positive samples before training. However, the privacy guarantees differ significantly.

Item replacement entirely removes original interactions, replacing them with semantically similar generated items. This eliminates direct signals from the original data, making it infeasible for the adversary to infer specific user-item interactions, even under gradient-based or semantic analysis

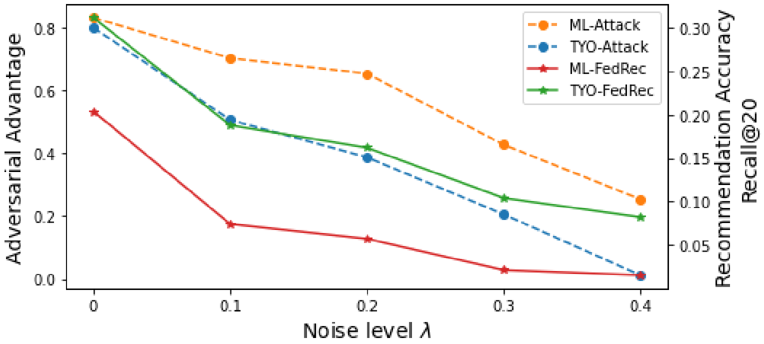

Fig. 5: Recommendation performance and IMIA attack performance of Movie Lens dataset and TYO dataset under difference noise levels

in IMIA setting. Therefore, item replacement achieves strong privacy by design.

Item augmentation, by contrast, retains original interactions and introduces additional, similar items. While this increases uncertainty in the IMIA attacker, original interactions still contribute to the gradient updates. Consequently, adversaries with more sophisticated semantic inference capabilities may recover the underlying interactions. Hence, item augmentation offers weaker privacy protection than item replacement, particularly under stronger adversarial models.

C. *Effect of FedSIG vs. Differential Privacy*

Differential privacy (DP) provides a formal guarantee that the output of a mechanism is nearly indistinguishable on neighboring datasets. Formally, a randomized mechanism M is (ϵ, δ)-DP if for any neighboring datasets D and D' differing in a single record, and for all events $S \subseteq \text{Range}(M)$,

$$\Pr[M(D) \in S] \leq e^{\epsilon} \cdot \Pr[M(D') \in S] + \delta.$$

In practice, DP-SGD achieves such guarantees by injecting calibrated Gaussian noise into the gradients during training. This ensures that the presence or absence of any single interaction has a limited effect on the model updates. However, because all gradients are perturbed uniformly, achieving strong privacy (i.e., small ϵ) typically comes at the cost of substantial degradation in model utility.

FedSIG adopts a different approach. Instead of adding unstructured random noise, FedSIG-R introduces *structured randomization* through interaction replacement: a fraction R of true user interactions are replaced by synthetic interactions generated from a global auxiliary model. This mechanism can be viewed as analogous to randomized response, where each true interaction is hidden with probability R. Intuitively, this reduces the distinguishability of positive interactions, thereby lowering the adversary's inference power. While this randomization is DP-inspired, it does not constitute a formal (ϵ, δ)-DP guarantee. In contrast, FedSIG-A augments user datasets with synthetic items but retains all original interactions, offering weaker protection since gradient signals for true interactions are preserved.

Empirical comparison. Our results demonstrate that FedSIG achieves DP-like privacy–utility trade-offs in practice. As shown in Figure 5 on the MovieLens dataset, DP-SGD reduces the adversarial advantage under IMIA from 0.814 to below 0.5, but at the cost of Recall@20 dropping sharply from 0.365 to 0.117. By contrast, FedSIG-R with a replacement ratio of $R = 0.2$ achieves a similar reduction in adversarial advantage (below 0.5) while maintaining Recall@20 at 0.148. This demonstrates that selective replacement can achieve comparable empirical privacy protection while preserving significantly more utility.

While DP-SGD remains the gold standard for formal privacy guarantees, FedSIG offers a pragmatic alternative when strict (ϵ, δ)-DP is not required, delivering similar empirical privacy improvements at a fraction of the accuracy loss. In applications where user experience is strongly tied to recommendation quality, FedSIG provides a viable balance between privacy and utility.

D. *Differential-Privacy View of Replacement (FedSIG-R)*

We show that the replacement step in FedSIG-R can be cast as a (local) DP randomizer under mild conditions, yielding central DP via amplification by shuffling. We first recall the definition: a randomized mechanism M is (ϵ, δ)-DP if for any neighboring datasets D, D' differing in one record and any event S, $\Pr[M(D) \in S] \leq e^{\epsilon} \Pr[M(D') \in S] + \delta$.

a) Per-interaction randomizer.: Fix the item domain $\mathcal{V}$. For a user u and an interaction item $i \in \mathcal{V}$, define the *replacement randomizer* $R_{R,P}$: with probability $1 - R$ output i (keep), and with probability R sample $Y \sim P$ and output Y (replace). Here $R \in (0, 1)$ is the replacement ratio and P is a *public* distribution over $\mathcal{V}$ (independent of the specific input record). We assume full support: $P(v) \geq p_{\min} > 0$ for all $v \in \mathcal{V}$.

Proposition 1 (Interaction-level LDP for replacement): Under the mechanism $R_{R,P}$ defined above:

1) If replacement may resample the original item (Y can equal i), then $R_{R,P}$ is ϵ-LDP with

$$\epsilon \;\leq\; \ln\!\Big(1 + \tfrac{1-R}{R\,p_{\min}}\Big).$$

2) If replacement explicitly excludes the original item (sample Y from $\mathcal{V} \setminus \{i\}$ with renormalized P), then $R_{R,P}$ is ϵ-LDP with

$$\epsilon \; \le \; \ln\Big(\frac{1-R}{R\,p_{\min}}\Big).$$

Proof sketch. For any inputs $i, j \in \mathcal{V}$ and any output o, bound the likelihood ratio $\Pr[\text{out} = o \,|\, i] / \Pr[\text{out} = o \,|\, j]$. The worst case occurs for $o = i$, yielding ratio $\frac{(1-R)+RP(i)}{RP(i)} = 1 + \frac{1-R}{RP(i)}$ in case (1), and $\frac{1-R}{RP(i)}$ in case (2). Taking $P(i) \ge p_{\min}$ gives the stated bounds. □

b) From local to central DP via shuffling.: If each user applies $R_{R,P}$ independently to their interactions and the server only observes the *shuffled* multiset of (noised) records, standard amplification-by-shuffling results imply an (ϵ, δ) central-DP guarantee with

$$\epsilon_{\text{shuf}} \; = \; \tilde{O}\Big((e^{\epsilon_{\text{loc}}} - 1)\sqrt{\frac{\log(1/\delta)}{m}}\Big),$$

where ϵ_{loc} is from Prop. 1 and m is the total number of randomized records. (Post-processing preserves DP, so subsequent gradient computation/aggregation does not weaken the guarantee.)

c) Practical instantiation and caveat.: To invoke Prop. 1 *formally*, the replacement distribution P must not depend on the specific input record; a public P (or user-level P_u independent of the particular item being randomized) suffices. Our implemented FedSIG-R uses a generator conditioned on user embeddings to improve utility; this can be made DP-compliant by enforcing that, *conditional on replacing*, the output is drawn from a pre-committed P (or P_u) with $P_{\min} > 0$, independent of the particular item being replaced. In this DP-compliant variant, the replacement ratio R acts as a privacy knob via ϵ_{loc}, and shuffling yields central DP as above.

d) Empirical alignment.: With $R = 0.2$, we observe adversarial advantage < 0.5 across datasets, matching the qualitative effect predicted by Prop. 1: as R increases, the per-record indistinguishability improves (ϵ_{loc} decreases), while utility degrades mildly compared to DP-SGD. This supports replacement as a DP-inspired mechanism with a favorable privacy–utility profile.

E. Discussion and Limitations

An important consideration raised in peer feedback concerns the assumptions and risks in FedSIG's design. In particular, two aspects warrant closer discussion: the reliance on auxiliary data for generator initialization and the limits of the augmentation strategy.

Auxiliary data quality. FedSIG employs an auxiliary dataset $\mathcal{D}_{aux}$ to bootstrap its generator. Although the generator is fine-tuned using user embeddings to reduce mismatch with private data, the quality and distribution of $\mathcal{D}_{aux}$ remain critical. Poorly aligned auxiliary data may lower recommendation accuracy by producing interactions that are semantically plausible but behaviorally irrelevant to users with sparse or complex preferences. This could weaken both privacy and utility guarantees. As such, careful selection of auxiliary data or the development of domain-adaptive generation techniques represents an important direction for future work.

Augmentation vs. replacement. While FedSIG's augmentation strategy (*FedSIG-A*) enriches datasets with synthetic interactions, it retains the original gradients tied to true interactions. This leaves residual privacy risk, since sophisticated adversaries could still distinguish real items using semantic correlation or temporal patterns. By contrast, the replacement strategy (*FedSIG-R*) directly masks gradients associated with sensitive items, yielding stronger privacy protection, albeit with a modest utility trade-off. This distinction highlights the need to select the defense variant based on the privacy sensitivity of the application domain.

VI. Related Work

Collaborative filtering (CF) [35]–[37] is a well-established approach for building recommendation systems, where users and items are represented as embeddings in a shared latent space, and interactions predict preferences. Matrix Factorization (MF) [25], [37] minimizes reconstruction errors of the user-item interaction matrix using the dot product and has inspired many variants. The advent of deep learning has further enhanced CF by enabling better embeddings and modeling complex user-item interactions, as seen in models like NeuMF [1] and A3NCF [38]. More recently, Graph Convolutional Networks (GCNs) [2] have gained traction for their ability to model high-order relationships, with methods like NGCF and IMP-GCN leveraging graph structures to propagate embeddings. However, these approaches depend heavily on historical interaction data, making them less effective when data is sparse or users hesitate to share data due to privacy concerns. Privacy-preserving synthetic data offers a potential solution to these limitations.

Open data sharing and unrestricted data exchange can significantly advance research and development; however, such practices are often infeasible when dealing with sensitive data involving privacy concerns, such as health, location, and behavioral information [11]. To address these challenges, the literature broadly categorizes existing privacy-preserving approaches into two main groups.

The first category comprises data anonymization techniques [8], [39], [40], which apply various sanitization strategies to prevent the re-identification of individuals in a dataset. Although these methods have demonstrated applicability in specific domains, they often lack formal privacy guarantees and are susceptible to re-identification attacks under adversarial conditions.

The second category focuses on synthetic data generation methods, which aim to produce realistic yet privacy-preserving synthetic datasets. Many of these approaches are grounded in rigorous differential privacy frameworks [17]. For synthetic data to be useful, its distribution should approximate that of the original data as closely as possible, while avoiding the generation of synthetic samples that are overly similar to real data instances—since such similarity may lead to privacy leakage.

For example, Acs et al. [41] proposed a method that first partitions the original dataset into k clusters using differentially private kernel k-means, followed by training generative neural networks to produce synthetic data within each cluster. In contrast, Bindschaedler et al. [42] introduced the concept of *plausible deniability*, where a privacy threshold is enforced such that an adversary cannot confidently determine whether a specific real instance corresponds to any released synthetic data record. Furthermore, Wang et al. [17] developed a practical technique for generating synthetic location data that preserve individual privacy by obfuscating both the presence and true location of individuals in the original dataset.

Although these approaches have been shown to work in some cases, generating privacy-preserving synthetic data in recommendation scenarios for federated learning is still challenging. This work presents a privacy-preserving synthetic data generation model for a federated recommendation setting.

VII. Conclusion

In this paper, we proposed a defence against inference attacks in federated recommendation systems to protect user interactions while maintaining the recommendation performance of the system. The main observation made in analysing privacy attacks on FedRecs was that we could not mitigate the attack success rate by adding SOTA privacy-preserving mechanisms unless we compromised the recommendation accuracy by a large margin, which results in invalidating the use of the system itself. Therefore, the proposed methodology generates synthetic interactions representing user preferences to replace a fraction of the original user interactions or augment the dataset using a proportion of similar items. The former aids user to remove direct interactions by replacing them with similar interactions and the latter allows users to hide the original interactions among similar interactions. Both these methods preserve the accuracy of the recommendation task. The experiments on five real-world datasets show that we can boost the accuracy of the recommendation model up to 65% compared to adding random synthetic interactions, which achieved the same privacy protection. In conclusion, our method can successfully mitigate the user item interaction inference attack while maintaining the performance of the federated recommendation model, achieving a better privacy-utility balance.

References

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

[2] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. 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*, pp. 639–648, 2020.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pp. 158–167, 2000.

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pp. 158–167, 2000.

[5] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "Drn: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 world wide web conference*, pp. 167–176, 2018.

[6] D. Delling and D. Wagner, "Pareto paths with sharc," in *Experimental Algorithms* (J. Vahrenhold, ed.), (Berlin, Heidelberg), pp. 125–136, Springer Berlin Heidelberg, 2009.

[7] W. Yue, Z. Wang, J. Zhang, and X. Liu, "An overview of recommendation techniques and their applications in healthcare," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 701–717, 2021.

[8] J. Wang, N. Wu, and X. Zhao, "Personalized route recommendation with neural network enhanced a* search algorithm," *IEEE Transactions on Knowledge & Data Engineering*, pp. 1–1, mar 5555.

[9] W. Yuan, C. Yang, Q. V. H. Nguyen, L. Cui, T. He, and H. Yin, "Interaction-level membership inference attack against federated recommender systems," *arXiv preprint arXiv:2301.10964*, 2023.

[10] S. Zhang, W. Yuan, and H. Yin, "Comprehensive privacy analysis on federated recommender system against attribute inference attacks," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[11] T. Ariyarathna, M. Mohammady, H.-Y. Paik, and S. S. Kanhere, "Vlia: Navigating shadows with proximity for highly accurate visited location inference attack against federated recommendation models," in *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pp. 1261–1271, 2024.

[12] T. Ariyarathna, M. Mohommady, H.-y. Paik, and S. S. Kanhere, "Deepsneak: User gps trajectory reconstruction from federated route recommendation models," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 1, pp. 1–22, 2024.

[13] D. Javeed, M. S. Saeed, P. Kumar, A. Jolfaei, S. Islam, and A. N. Islam, "Federated learning-based personalized recommendation systems: An overview on security and privacy challenges," *IEEE Transactions on Consumer Electronics*, 2023.

[14] K. Zhao, Y. Zhang, H. Yin, J. Wang, K. Zheng, X. Zhou, and C. Xing, "Discovering subsequence patterns for next poi recommendation.," in *IJCAI*, vol. 2020, pp. 3216–3222, 2020.

[15] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie, "Fedgnn: Federated graph neural network for privacy-preserving recommendation," *arXiv preprint arXiv:2102.04925*, 2021.

[16] V. Perifanis and P. S. Efraimidis, "Federated neural collaborative filtering," *Knowledge-Based Systems*, vol. 242, p. 108441, 2022.

[17] X. Wang, X. Liu, Z. Lu, and H. Yang, "Large scale gps trajectory generation using map based on two stage gan," *Journal of Data Science*, vol. 19, no. 1, pp. 126–141, 2021.

[18] J. W. Kim and B. Jang, "Deep learning-based privacy-preserving framework for synthetic trajectory generation," *Journal of Network and Computer Applications*, vol. 206, p. 103459, 2022.

[19] K. Liu, X. Jin, S. Cheng, S. Gao, L. Yin, and F. Lu, "Act2loc: a synthetic trajectory generation method by combining machine learning and mechanistic models," *International Journal of Geographical Information Science*, vol. 38, no. 3, pp. 407–431, 2024.

[20] H. Murtaza, M. Ahmed, N. F. Khan, G. Murtaza, S. Zafar, and A. Bano, "Synthetic data generation: State of the art in health care domain," *Computer Science Review*, vol. 48, p. 100546, 2023.

[21] F. Liu, Z. Cheng, H. Chen, Y. Wei, L. Nie, and M. Kankanhalli, "Privacy-preserving synthetic data generation for recommendation systems," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1379–1389, 2022.

[22] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2017.

[23] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282, IEEE, 2018.

[24] GroupLens, "Movielens 1m dataset," 2024.

[25] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, 2008.

[26] Foursquare, "Foursquare datasets," 2024.

[27] L. Qi, Y. Liu, Y. Zhang, X. Xu, M. Bilal, and H. Song, "Privacy-aware point-of-interest category recommendation in internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21398–21408, 2022.

[28] Y. Liu, H. Wu, K. Rezaee, M. R. Khosravi, O. I. Khalaf, A. A. Khan, D. Ramesh, and L. Qi, "Interaction-enhanced and time-aware graph convolutional network for successive point-of-interest recommendation in traveling enterprises," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 635–643, 2022.

[29] I. Bayer, X. He, B. Kanagal, and S. Rendle, "A generic coordinate descent framework for learning from implicit feedback," in *Proceedings of the 26th international conference on world wide web*, pp. 1341–1350, 2017.

[30] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 549–558, 2016.

[31] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[32] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the 24th international conference on world wide web*, pp. 278–288, 2015.

[33] X. He, T. Chen, M.-Y. Kan, and X. Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 1661–1670, 2015.

[34] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282, IEEE, 2018.

[35] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: Rating prediction with ratings and reviews," in *Proceedings of the 2018 world wide web conference*, pp. 639–648, 2018.

[36] D. Neumann, A. Lutz, K. Müller, and W. Samek, "A privacy preserving system for movie recommendations using federated learning," *arXiv preprint arXiv:2303.04689*, 2023.

[37] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 Eighth IEEE international conference on data mining*, pp. 263–272, Ieee, 2008.

[38] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. S. Kankanhalli, "Aˆ 3ncf: An adaptive aspect attention model for rating prediction.," in *IJCAI*, pp. 3748–3754, 2018.

[39] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[40] J. Wang, N. Wu, and X. Zhao, "Personalized route recommendation with neural network enhanced a* search algorithm," *IEEE Transactions on Knowledge & Data Engineering*, pp. 1–1, mar 5555.

[41] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[42] B. Liu, Y. Guo, and X. Chen, "Pfa: Privacy-preserving federated adaptation for effective model personalization," in *Proceedings of the Web Conference 2021*, WWW '21, (New York, NY, USA), p. 923–934, Association for Computing Machinery, 2021.