A Generic Data Synthesis Framework for Privacy-Preserving Point-of-Interest Recommender Systems

Longyin Cui
University of Kentucky
Lexington, KY, USA
lcu225@uky.edu

Xiwei Wang
Northeastern Illinois University
Chicago, IL, USA
xwang9@neiu.edu

Ting Gu
College of the Holy Cross
Worcester, MA, USA
tgu@holycross.edu

ABSTRACT

Personalization services offered by Point-of-Interest (POI) recommender systems are becoming increasingly popular, especially in the context of mobile devices. However, data privacy regulations and user concerns regarding privacy often prevent the transfer and storage of user data, which poses a challenge for these systems. To address this issue, privacy-preserving recommender systems have gained importance. This paper proposes a generic framework for generating synthetic user data for POI recommendations based on differential privacy, random response, and user grouping. The proposed framework can accommodate various data feedback without compromising privacy and is compatible with non-private recommender systems, allowing for future improvements and flexibility. Our experiments on real-world datasets demonstrate that the framework strikes a balance between privacy protection and accurate recommendations.

CCS CONCEPTS

• **Security and privacy** → **Privacy-preserving protocols**; • **Information systems** → **Clustering**; *Collaborative filtering*; *Location based services.*

KEYWORDS

privacy preserving, recommender system, POI recommendation, differential privacy, data clustering, virtual users

ACM Reference Format:
Longyin Cui, Xiwei Wang, and Ting Gu. 2023. A Generic Data Synthesis Framework for Privacy-Preserving Point-of-Interest Recommender Systems. In *International Conference on Research in Adaptive and Convergent Systems (RACS '23), August 6–10, 2023, Gdansk, Poland*. ACM, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/3599957.3606241

1 INTRODUCTION

The Recommender system (RS) predicts user interactions with products and is widely used to curate personalized content lists from vast online options [25]. These systems [31] analyze attributes, feedback, and contextual details. As these systems evolve, the integration of more information types is increasingly common for improved prediction accuracy [8, 20].

RACS '23, August 6–10, 2023, Gdansk, Poland

ACM ISBN 979-8-4007-0228-0/23/08.
https://doi.org/10.1145/3599957.3606241

However, the rise of privacy concerns due to data breaches and regulations creates a challenging environment for information collection, especially for POI recommender systems. For example, national security concerns are leading to competition for exclusive data access rights, and regulations have also led to an unfriendly environment for recommender systems to collect useful information. They all call for an increased emphasis on privacy preservation in these systems.

As a result, privacy concerns are driving increased regulations in the field of recommender systems [6, 15, 26]. These regulations aim to prevent irresponsible or malicious data mining and analytics. POI RS, which relies on large quantities of data and various data sources, is particularly affected by these regulations. Besides, the growing use of mobile devices has led to the proliferation of location-based services (LBS).

Various privacy-preserving recommender systems have emerged with approaches including decentralization, anonymization, obfuscation, and traditional cryptography. Decentralization allows for local RS operation, eliminating the need to store sensitive data on the cloud. Anonymization and obfuscation-based systems perturb or remove identifiable user details to protect sensitive information. Traditional cryptographic methodologies are routinely customized to augment the capacity for privacy preservation as well. However, these methods have limitations, such as increased communication needs and limited compatibility with newer models.

This paper introduces a flexible framework using local differential privacy (LDP) and location-based clustering to generate synthetic data for POI recommendations. Our proposed framework enables local recommender systems to offer predictions without directly sharing sensitive personal information. Our main contributions are:

- A model-independent user data collection mechanism for privacy-preserving POI recommender systems.
- Secure data communication using estimated user location and LDP to generate personalized recommendations while preserving user privacy.
- Empirical validation of the framework's effectiveness using real-world datasets.

The paper is structured as follows: "Related Work" discusses related topics and background information. "Framework Description" depicts the framework's structure and the challenges we aim to solve. "Preliminaries" introduces fundamentals, notation, proof, and problem formulation. "Experiment and Discussion" presents datasets, optimization, and results. "Conclusion and Future Work" concludes the paper.

2 RELATED WORK

2.1 Existing Privacy-Preserving Recommender System

Privacy concerns have led to various privacy-preserving recommender system models, largely divided into decentralized and centralized methods. Decentralized strategies, such as distributed and federated models, exchange gradients or model weights without centralized data collection [7, 21]. Centralized frameworks employ encryption protocols, k-anonymity, and perturbations, often facing the challenge of balancing privacy, data transfer, and user volume demands[11, 18].

Differential Privacy (DP), a privacy standard garnering increased attention, has been adapted to protect sensitive user data in recommender systems. For instance, the application of DP to matrix factorization-based recommender systems has been explored, providing optimal privacy-preserving techniques [5]. Local DP (LDP), a more localized DP variant, has been used to protect user feedback from central servers [28]. Other research, such as [4], has improved upon existing challenges by providing deniability not just to rating values but also to rating behaviors.

2.2 Data Clustering

Clustering is widely used for two purposes in POI RS. First, it is used to improve performance and address data scarcity problems, such as predicting how groups of people choose online streaming services or nearby restaurants [2, 9, 22]. Second, clustering has been used to protect user identities and mask sensitive user information in privacy-preserving schemes for RS and general data mining, helping to protect user feedback and certain contextual information [14, 27, 30].

3 FRAMEWORK DESCRIPTION

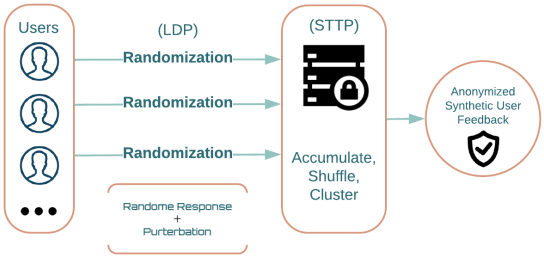

Figure 1: STTP's User Data Collecting Scheme from Mobile Devices

The Generic User Synthesizer (GUS) framework is designed to maintain privacy while generating personalized recommendations using aggregated user data. The framework employs a Semi-Trusted Third Party (STTP) for data collection. As Figure 1 shows, this STTP applies LDP-based clustering algorithms to protect user privacy and groups users based on implicit location. The centroids of these clusters represent synthetic users.

The central server calculates POI similarities and prepares user and item embeddings and a similarity matrix. This data is made available for download by local RS on user devices, which then generate recommendations by integrating the user's personal rating record.

Due to computational constraints, an adaptive training strategy is used for the simulation of numerous mobile devices. The local RS uses similar synthetic users to derive a weighted average for the final prediction. Further sections will provide an in-depth explanation of this approach.

4 PRELIMINARIES

4.1 Local Differential Privacy

4.1.1 Definitions and Theorems. ***Definition 1*** **(*ϵ*-Local Differential Privacy)**. Suppose there is a randomized Algorithm A and a single dataset D, for any pair of different user feedback $r, r' \in D$ and for any outcome $S \subseteq Range(A)$, we have

$$Pr[A(r) \in S] \leq exp(\epsilon) \cdot Pr[A(r') \in S] \tag{1}$$

where the randomization is applied to each user's feedback independently. The value of ϵ is aligned with data integrity but the opposite of privacy protection. Our LDP mechanism is based on Generalized Randomized Response (GRR) [16, 29], whose definition is given below:

Definition 2 **(Generalized Randomized Response (GRR))**. Given i being the true visited POI, let $\bar{i}$ be the random response instead. If $\bar{K}$ is the option space from which the random response is selected, i.e., $i \in \bar{K}$, then we have the following formulation:

$$Pr(\bar{i} = i) = \begin{cases} \frac{e^\epsilon}{e^\epsilon + |\bar{K}| - 1}, \text{if } (i = \bar{i}) \\ \frac{1}{e^\epsilon + |\bar{K}| - 1}, \text{if } (i \neq \bar{i}) \end{cases} \tag{2}$$

As the above formula shows, when $|\bar{K}| = 2$, which means the user can only choose one of two items/POIs, then it becomes the conventional true or false scenario where the traditional Randomized Response (RR) is implemented. Next, the feedback value is also guarded by the perturbation technique using Duchi's solution [10].

Duchi's Solution. Given a tuple $r \in$ [-1, 1], a perturbed tuple r' that equals either $\frac{e^\epsilon+1}{e^\epsilon-1}$ or $-\frac{e^\epsilon+1}{e^\epsilon-1}$ is returned according to the following probability t:

$$Pr(r'_{ui} = t | r_{ui}) = \begin{cases} \frac{e^\epsilon-1}{2e^\epsilon+2} \cdot r_{ui} + \frac{1}{2}, \text{if } (t = \frac{e^\epsilon+1}{e^\epsilon-1}) \\ \frac{1-e^\epsilon}{2e^\epsilon+2} \cdot r_{ui} + \frac{1}{2}, \text{if } (t = -\frac{e^\epsilon+1}{e^\epsilon}) \end{cases} \tag{3}$$

where u is the user given ratings, and i is the target POI. The perturbed ratings r'_{ui} are unbiased estimators of the original ratings, according to Duechi et al.

Since LDP mechanisms in our framework include GRR and Duchi's Solution to protect user rating behavior, i.e., whether a user visited a POI, and POI preference, i.e., whether the user likes their visited place, respectfully, we introduce the following properties.

Theorem 1 **(Sequential Composition)**. If a mechanism G contains a series of n independent randomized functions $G = \{g_1, g_2, ..., g_n\}$, and each function offer ϵ_i-*LDP* guarantee where $i \in n$, then the mechanism G provides $(\sum_{i=1}^{n} e^\epsilon)$-*LDP*.

Meanwhile, the LDP mechanisms are introduced at the very beginning, so the post-processing property is involved to guarantee

the safety of the data manipulation and training process following the initial step.

***Theorem 2* (Post-processing)**. Given a dataset D, and a function f that guarantees ϵ-*LDP* where $f : D \rightarrow \mathbb{R}$, for any randomized function $f' : \mathbb{R} \rightarrow \mathbb{R}'$ we have $f \circ f'$ also being ϵ-*LDP*.

These theorems guarantee the privacy-preserving qualities for all the following computing and data processing procedures.

4.1.2 Rating Value Protection. Considering that LDP-dependent algorithm is to calculate centroids' ratings, which comes from averaging all user ratings in each group, we choose Duchi's solution. The algorithm is shown in Algorithm 1.

It is worth noting that the input of the following algorithms is a list of tuples that are in the format of $\{userID, itemID, rating\}$. Each tuple can be denoted as r_{ui} representing that user u possesses a rating r toward item i. Specifically, the collection of ratings thereby forms the rating list R_{list}. In Algorithm 1, each rating after normalization is perturbed according to the Equation (3).

Algorithm 1: Rating Perturbation Using Duchi et al.'s Solution

Input: list of rating tuples R_{list}, privacy parameter ϵ_1
Output: list of perturbed rating tuples R'_{list}

for *each* r_{ui} *in* R_{list} **do**
  Normalize r_{ui} such that $r_{ui} \in [-1, 1]$
  $r_{ui} = \frac{1}{2} \cdot (r_{ui} - 1) - 1$
end

for *each* r_{ui} *in* R_{list} **do**
  Sample a Bernoulli variable t where:
  $Pr(t = 1) = \frac{e^{\epsilon_1}-1}{2e^{\epsilon_1}+2} \cdot r_{ui} + \frac{1}{2}$
  if $t = 1$ **then**
    $r'_{ui} = \frac{e^{\epsilon_1}+1}{e^{\epsilon_1}-1}$
  else
    $r'_{ui} = \frac{e^{\epsilon_1}+1}{1-e^{\epsilon_1}}$
  end
end
for *each* r'_{ui} *in* R'_{list} **do**
  De-normalize r'_{ui} such that $r'_{ui} \in [1, 5]$
end

return R'_{list}

4.1.3 Rating Behavior Protection. Previous research [13] has shown that safeguarding rating behaviors in recommender systems is challenging due to the large column space of potential POIs and the sparsity of the rating matrix. We introduce a dynamically precalculated similarity matrix on the central server, which allows each *locationID* in the rating tuple $\{userID, locationID, rating\}$ to be perturbed.

Algorithm 2 shows how the POIs are randomized. The input, the list of rating tuples R'_{list}, is the exact output from Algorithm 1. Variable k is the number of options. For example, when $k = 2$, the *locationID* in $\{userID, locationID, rating\}$ can only be replaced by the most similar POI's *locationID*. When $k = 3$, the *locationID* in $\{userID, locationID, rating\}$ is replaced by either one of the top two most similar POIs' *locationIDs*. The larger k is, the higher the privacy budget. Eventually, the output R''_{list} is sent to the aggregating and clustering server.

Algorithm 2: Protecting Rating Presence

Input: list of rating tuples R'_{list} from Algorithm 1, privacy parameter ϵ_2, similarity matrix S_I, number of options $k \geq 2$
Output: list of perturbed rating tuples R''_{list}

for *each* r'_{ui} *in* R'_{list} **do**
  Sample a Bernoulli variable t where:
  $Pr(t = 1) = \frac{e^{\epsilon_2}}{e^{\epsilon_2}+k-1}$
  if $t = 1$ **then**
    $r''_{ui} = r'_{ui}$
  else
    $r''_{ui} = r'_{uS_I[i]}$
  end
end
return R''_{list}

4.2 Privacy Analysis

Theorem 1. *Algorithm 1 satisfies ϵ_1-LDP with respect to users' rating values.*

Proof. According to the definition of ϵ-LDP, we want to prove that it is equally likely to generate the same output $r'_{list} = [r'_{ui}]_{i=1}^{n}$ for any two input $r^1_{list} = [r_{ui}]_{i=1}^{n}$ and $r^2_{list} = [r_{ui}]_{i=1}^{n}$ in Algorithm 1. Let x, X^1, and X^2 be any values in $[-1, 1]$, we have

$$\frac{Pr[r'_{ui} = x | r_{ui} = X^1]}{Pr[r'_{ui} = x | r_{ui} = X^2]} \leq \frac{\max_{X^1}(Pr[r'_{ui} = x | r_{ui} = X^1])}{\min_{X^2}(Pr[r'_{ui} = x | r_{ui} = X^2])}$$

$$= \frac{\max_{X^1}(\frac{e^{\epsilon_1}-1}{2e^{\epsilon_1}+2}X^1 + \frac{1}{2})}{\min_{X^2}(\frac{e^{\epsilon_1}-1}{2e^{\epsilon_1}+2}X^2 + \frac{1}{2})} = \frac{\frac{e^{\epsilon_1}-1}{2e^{\epsilon_1}+2}(1) + \frac{1}{2}}{\frac{e^{\epsilon_1}-1}{2e^{\epsilon_1}+2}(-1) + \frac{1}{2}} = e^{\epsilon_1}.$$

Thus, the perturbation of R_{list} in Algorithm 1 satisfies ϵ_1-LDP and Algorithm 1 satisfies ϵ_1-LDP with respect to users' rating values. □

Theorem 2. *Algorithm 2 satisfies $(\epsilon_1 + \epsilon_2)$-LDP for both users' rating values and rating behaviors.*

Proof. We start by proving that it is equally likely to generate the same output $r''_{list} = [r''_{ui}]_{i=1}^{n}$ for any two inputs $r^1_{list} = [r'_{ui}]_{i=1}^{n}$ and $r^2_{list} = [r'_{ui}]_{i=1}^{n}$ in Algorithm 2. Let y, Y^1, and Y^2 be any values in $[-1, 1]$. According to Algorithm 2, we have

$$\frac{Pr[r''_{ui} = y | r'_{ui} = Y^1]}{Pr[r''_{ui} = y | r'_{ui} = Y^2]} \leq \frac{\max_{Y^1}(Pr[r''_{ui} = y | r'_{ui} = Y^1])}{\min_{Y^2}(Pr[r''_{ui} = y | r'_{ui} = Y^2])}$$

$$= \frac{\frac{e^{\epsilon_2}}{e^{\epsilon_2}+k-1}}{\frac{k-1}{e^{\epsilon_2}+k-1}} = \frac{e^{\epsilon_2}}{k-1} \leq e^{\epsilon_2}.$$

Thus, the perturbation of R'_{list} in Algorithm 2 satisfies ϵ_2-LDP and Algorithm 2 satisfies ϵ_2-LDP with respect to users' rating behavior. Since Algorithm 2 protects user's rating value by using Algorithm 1 and Algorithm 1 satisfies ϵ_1-LDP, according to the sequential composition property, we can conclude that Algorithm 2 satisfies $(\epsilon_1 + \epsilon_2)$-LDP.

□

4.3 Notations and Methods for Clustering and POI Similarity Calculation

It is common to use user-item interaction records to perform clustering. However, this approach is not effective for POI recommender systems, as users in datasets often visit only an extremely small portion of a city and do not have a significant number of interactions. In comparison, we choose to use estimated user locations (center of visited locations)

Various clustering methods have been investigated in our experiment. Initially, our approach started with DBSCAN [12]. However, as Figure 2 shows (Upper-left), this method leads to grouping users who are far from each other into one cluster. The red dots represent the users that are grouped into the first cluster. On the other hand, when testing OPTICS clustering [3], we shrank the maximum distance to avoid overly large groups. As a result, a large portion of the users are considered noise (black dots).

Moreover, we also considered hierarchical clustering methods such as the Partition Around Medoids (PAM) [17] clustering algorithm (upper-right). Unfortunately, it creates many empty clusters despite our active tuning, and the result is stretched easily by outliers. As a result, the k-mean yields comparatively better results. Spectral clustering [23] has also been tested as well by constructing the affinity matrix first. However, it is too computationally costly.

4.4 Personalization

The central RS in our approach performs initial training to reduce the computational burden on mobile devices and imputes missing data in sparse POI datasets. Synthetic data is used to generate pre-trained models for download, but adaptive training on local RS is still required to impute missing probabilities. Simulating this scenario during the experiment is time-consuming, so we take an alternative approach where real users rank artificial users based on rating similarities to get personalized recommendations through the weighted average of the top t similar centroids.

4.5 Evaluation and Metrics

Because of the non-linear feature of our system, it is challenging to optimize it as a whole. Therefore, we have decided to optimize each of the three parts (perturbation, clustering, and final results) separately, as this approach allows us to tackle the heavy workload in a more efficient and effective way.

4.5.1 Perturbation. To evaluate the performance of the perturbation part, we use Mean Absolute Error (MAE) to measure the relationship between the LDP budget (ϵ_1, ϵ_2) and the resulting errors between real users and virtual users. By comparing the MAE values at different levels of the LDP budget, we can assess the impact of the budget on the accuracy of following the clustering procedure. The formula is shown below:

$$MAE = \frac{1}{|R|} \sum_{i \in R} |r_i - \bar{r}_i| \tag{4}$$

where R is the set of all ratings. The variables r_i and $\bar{r}_i$ are the corresponding ratings from real users and virtual users, respectively.

4.5.2 Clustering. Here we state the metric for finding the optimal group number k: the within-cluster sum of squares (WSS). The formula is listed below:

$$\text{WSS} = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - c_i\|^2 \tag{5}$$

where k is the number of clusters, C_i is the i-th cluster, x_j is the j-th data point in the i-th cluster, c_i is the centroid of the i-th cluster, and $||x_j - c_i||^2$ represents the squared Euclidean distance in between.

4.5.3 Final Result. Regarding the assessment of the final result, we use precision at k and recall at k to study the trade-off from the framework output. If we let I_r and I_v denote the recommended POI set and visited POI set, then the following equations show the details of the definition:

$$Recall@k = \frac{|I_r \cap I_v|}{|I_v|} \tag{6}$$

$$Precision@k = \frac{|I_r \cap I_v|}{k} \tag{7}$$

where $Recall@k$ measures the portion of visited POIs in all visited POIs, and $Precision@k$ measures the portion of recommended POIs that are actually visited in the top-k POIs.

5 EXPERIMENTS AND DISCUSSION

5.1 The Datasets

We experimented with user feedback from two areas: the Champaign–Urbana (CU) metropolitan area and the Phoenix (PH) city area. Data were filtered from Yelp [1] and Google Local [19] datasets, respectively. As shown in Figure 2, the locations of the users outline the shape of the city.

The user feedback in our datasets has the following characteristics:

- The feedback is explicitly rated in $\{1, 2, 3, 4, 5\}$.
- User ratings are unique, which means that each user can only give a rating to the same place once.
- The sparsity of the datasets is comparatively much lower than that of conventional recommendation datasets, even with filtering. A clear comparison of the statistics of the four areas can be found in Table 1.

5.2 Parameters and Tuning

The optimization is performed on each of the three parts:

- Users' mobile devices perform two tasks: selecting privacy budget (ϵ_1 and ϵ_2) and determining the number of synthetic users (n_v) for optimal performance.
- k-means clustering is used on the STTP, with the number of clusters chosen beforehand using the elbow method.

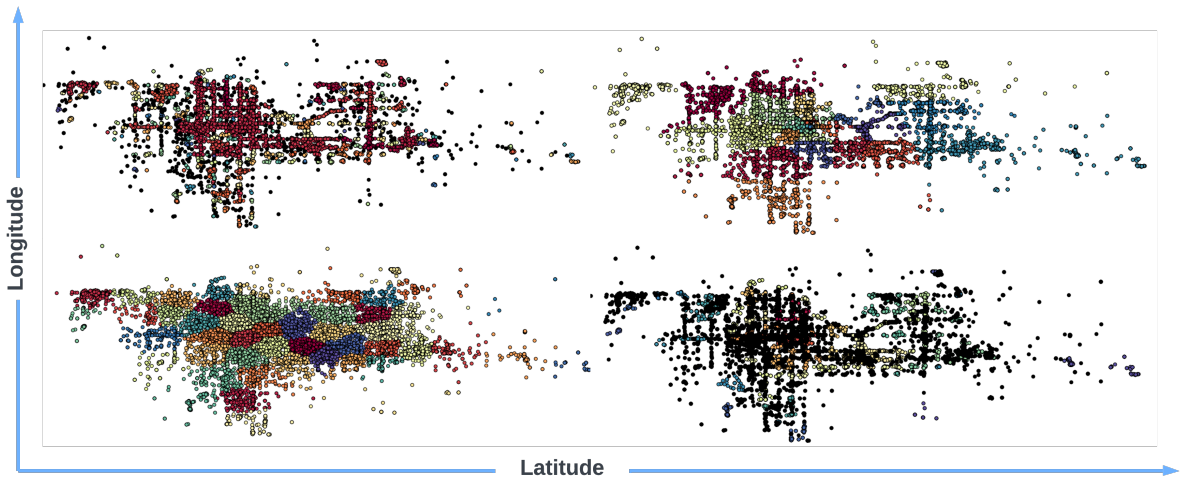

Figure 2: The visualization of user location clustering (Phoenix). The x and y axes represent latitude and longitude, respectively. The clustering algorithms shown in the picture include DBSCAN (upper left), PAM (upper right), k-mean (lower left), and OPTICS (lower right).

Table 1: Datasets Statistics

Area	#Users	#POIs	#Feedback	Density
CU (Yelp)	11953	1579	33990	0.1802%
PH (Yelp)	204887	17213	576700	0.0163%
CU (Google)	1910	876	3310	0.1978%
PH (Google)	24899	7801	37245	0.0192%

- On the central server, the hyperparameters of the recommendation algorithm, the neural collaborative filtering (NCF) model, are tuned.

To streamline the hyperparameter tuning process for GUS, we have developed a strategy that decomposes the tuning process into smaller, manageable sub-tasks that focus on optimizing specific sub-components.

In the optimization process, we calculate the MAE between feedback generated from centroids with and without LDP to identify the optimal trade-off point. Based on the analysis in Figure 3, we set ϵ_1 and ϵ_2 to 0.6 to achieve high accuracy and robust privacy protection within the tolerable increase in MAE.

In the second stage, by considering the elbow method and computation cost, the number of clusters is set to 45.

In the last part, we employed an automatic approach to tuning our model due to a large number of parameters, a wide range of possible values, and a lack of consensus on optimal settings. The approach [24] utilizes a Bayesian optimization framework that treats the overall performance of the model as a sample from a Gaussian process. Our findings suggest that a low number of latent factors ($k = 2$) yields the best results. We set the NCF's multi-layer perceptron structure to be [128, 64, 32] and used 20 initial centroids to calculate the final prediction. The number of centroids could be reduced to 9 without affecting the final result.

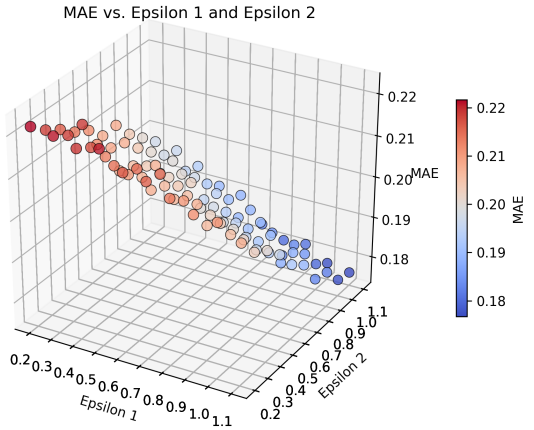

Figure 3: Comparison of Perturbation Results for Varying Epsilon Values: Scatter plot of MAE values for different combinations of Epsilon 1 (ϵ_1) and Epsilon 2 (ϵ_2) values.

5.3 Results and Comparison

We compared four models in our study: the Neural Collaborative Filtering (NCF) baseline model, our proposed framework (NCF+GUS), the Decentralized Matrix Factorization (DMF) model [7], and the federated recommender system (MetaMF) [21]. We selected these models for the following reasons:

- Given its prominence as a widely studied neural network-based collaborative filtering model, the NCF provides an ideal baseline for comparison in our research.
- As a decentralized approach that places a strong emphasis on user privacy, DMF provides a valuable reference point for our research. By relying on gradient exchange and other security

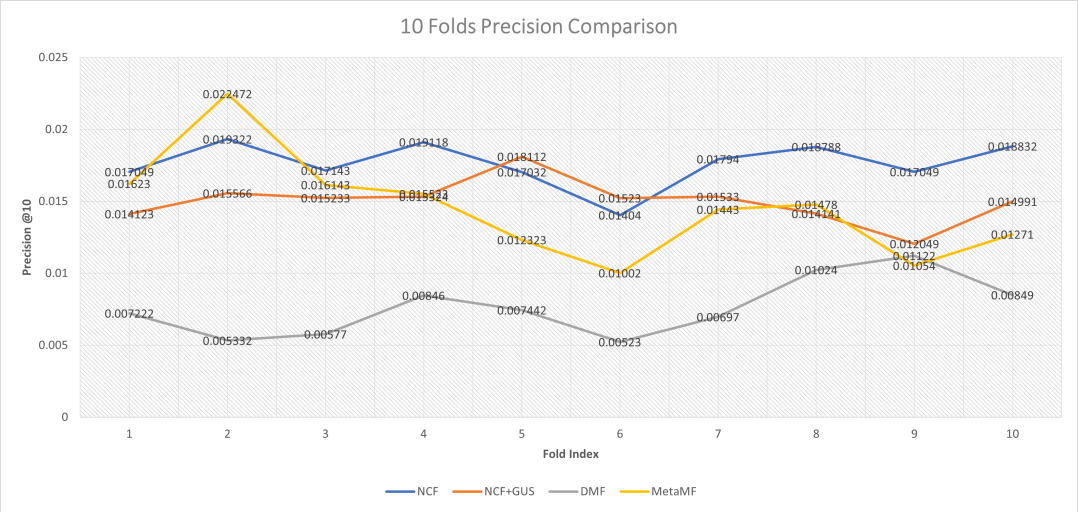

Figure 4: The Precision comparisons among the four models for each training-testing dataset pair in the City of Phoenix. The dataset is partitioned chronologically.

measures, this model is able to protect against malicious attacks and untrusted users.

- MetaMF is a privacy-preserving RS model with a federated structure that strikes an excellent balance between privacy and prediction accuracy. Its semi-distributed structure separates central and local training processes.

The overall average performance of each model is shown in Tables 2 and 3. Each of the entries in Tables 2 and 3 is the average performance of its corresponding model on all training-testing folds. If we expand each entry, we get Figure 4. For example, Figure 4 displays precision comparisons for each training-testing dataset pair in the City of Phoenix, comparing the performance of the four models. The datasets are partitioned chronologically, with each point on the graph representing a distinct training-testing pair.

Table 2: Average Result Comparison @5

CU				
Metrics	P@5		R@5	
Dataset	Yelp	Google	Yelp	Google
NCF	0.0147	**0.0812**	**0.0512**	**0.0045**
NCF+GUS	**0.0152**	0.0670	0.0447	0.0033
DMF	0.0067	0.0042	0.0081	0.0038
MetaMF	0.0130	0.0708	0.0438	0.0041
PH				
Metrics	P@5		R@5	
Dataset	Yelp	Google	Yelp	Google
NCF	0.0152	**0.0907**	**0.0667**	**0.0066**
NCF+GUS	**0.0155**	0.0771	0.0538	0.0043
DMF	0.0122	0.0485	0.0423	0.0022
MetaMF	0.0149	0.0767	0.0609	0.0062

Table 3: Average Result Comparison @10

CU				
Metrics	P@10		R@10	
Dataset	Yelp	Google	Yelp	Google
NCF	**0.0143**	0.0086	**0.0708**	0.0046
NCF+GUS	0.0135	**0.0102**	0.0446	0.0051
DMF	0.0086	0.0041	0.0068	0.0017
MetaMF	0.0137	0.0114	0.0714	**0.0062**
PH				
Metrics	P@10		R@10	
Dataset	Yelp	Google	Yelp	Google
NCF	0.0176	**0.0104**	**0.0708**	**0.0074**
NCF+GUS	0.0150	0.0082	0.0446	0.0041
DMF	**0.0776**	0.0043	0.0077	0.0023
MetaMF	0.0147	0.0071	0.0428	0.0033

The selected datasets yielded lower precision and recall values compared to other popular datasets due to the sparsity of location-based services. While we could modify all comparable models to better combat the sparsity problem in POI recommendations, doing so would harm the integrity of the challenged models. Also, universal modification is not employable in this situation. Nevertheless, while NCF had the highest precision, our proposed framework with the GUS had the best performance among privacy models, balancing accuracy and privacy concerns.

Our testing procedure simulates real-world scenarios where the recommendation system receives data sequentially and over time. During each round of cross-validation, the testing set becomes the training set, and the next fold is used as the testing set. The inflection points of the curves in Figure 4 represent the results of the validation process. To ensure that each fold has sufficient users, we adjust the temporal window size of each fold so that each user

has at least two ratings in the training set. This process is applied to all models being tested.

The lightweight and efficient integration of the Generic User Synthesizer (GUS) in our proposed framework strikes a balance between precision and recall on both datasets in a real-world setting. Unlike DMF, which underperforms due to low data density, GUS generates local recommendations without burdening users with additional computational tasks. The model's training process is made efficient and fast by using LDP standards and clustering, which preserves user privacy. Additionally, the modularized and detachable GUS framework enables easy integration into existing recommendation systems, unlike other privacy-preserving models, such as DMF and MetaMF, which require a complete switch and have limitations on future data usage. Furthermore, the data-mineable synthetic data, which cannot be traced back to real users, can be safely transferred without compromising privacy.

Lastly, an important benefit of our framework is generating synthetic user-item interaction data that can be used in future research and shared without concern for privacy liability.

6 CONCLUSIONS AND FUTURE WORK

This privacy-preserving point-of-interest framework estimates users' location while protecting privacy. The framework uses a general user data collection approach based on LDP and data clustering to achieve scalability and geolocation awareness. The third-party server collects direct user feedback through LDP and obfuscates private data through clustering before uploading data to the central server, ensuring privacy. We optimize the trade-off between prediction accuracy and privacy protection by employing judicious processing techniques, robust testing conditions, and the application of real-world datasets. The optimization process uses a Bayesian framework to reduce the number of steps required to find optimal parameters.

Future work includes upgrading the data collecting mechanism with advanced contextual information processing tools and integrating more user data synthesis techniques, such as generative adversarial networks and variational autoencoders.

REFERENCES

[1] [n. d.]. Yelp dataset. ([n. d.]). https://www.yelp.com/dataset

[2] Rishabh Ahuja, Arun Solanki, and Anand Nayyar. 2019. Movie recommender system using k-means clustering and k-nearest neighbor. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 263–268.

[3] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.

[4] Ting Bao, Lei Xu, Liehuang Zhu, Lihong Wang, Ruiguang Li, and Tielei Li. 2021. Privacy-preserving collaborative filtering algorithm based on local differential privacy. *China Communications* 18, 11 (2021), 42–60.

[5] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. 2015. Applying differential privacy to matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 107–114.

[6] Igor Calzada. 2022. Citizens' data privacy in China: The state of the art of the Personal Information Protection Law (PIPL). *Smart Cities* 5, 3 (2022), 1129–1150.

[7] Chaochao Chen, Ziqi Liu, Peilin Zhao, Jun Zhou, and Xiaolong Li. 2018. Privacy preserving point-of-interest recommendation using decentralized matrix factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[8] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[9] Zhihua Cui, Xianghua Xu, XUE Fei, Xingjuan Cai, Yang Cao, Wensheng Zhang, and Jinjun Chen. 2020. Personalized recommendation system based on collaborative filtering for IoT scenarios. *IEEE Transactions on Services Computing* 13, 4 (2020), 685–695.

[10] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2018. Minimax optimal procedures for locally private estimation. *J. Amer. Statist. Assoc.* 113, 521 (2018), 182–201.

[11] Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L Lagendijk. 2012. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE transactions on information forensics and security* 7, 3 (2012), 1053–1066.

[12] Ian Fiske and Richard Chandler. 2011. Unmarked: an R package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of statistical software* 43 (2011), 1–23.

[13] Arik Friedman, Shlomo Berkovsky, and Mohamed Ali Kaafar. 2016. A differential privacy framework for matrix factorization recommender systems. *User Modeling and User-Adapted Interaction* 26 (2016), 425–458.

[14] Md Zahidul Islam and Ljiljana Brankovic. 2011. Privacy preserving data mining: A noise addition framework using a novel clustering technique. *Knowledge-Based Systems* 24, 8 (2011), 1214–1223.

[15] Garrett A Johnson, Scott K Shriver, and Samuel G Goldberg. 2023. Privacy and market concentration: intended and unintended consequences of the GDPR. *Management Science* (2023).

[16] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. 2016. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*. PMLR, 2436–2444.

[17] Leonard Kaufman and Peter J Rousseeuw. 1990. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis* 344 (1990), 68–125.

[18] Sungwook Kim, Jinsu Kim, Dongyoung Koo, Yuna Kim, Hyunsoo Yoon, and Junbum Shin. 2016. Efficient privacy-preserving matrix factorization via fully homomorphic encryption. In *Proceedings of the 11th ACM on Asia conference on computer and communications security*. 617–628.

[19] Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. *arXiv preprint arXiv:2202.13469* (2022).

[20] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1754–1763.

[21] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Dongxiao Yu, Jun Ma, Maarten de Rijke, and Xiuzhen Cheng. 2020. Meta matrix factorization for federated rating predictions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 981–990.

[22] R Logesh and V Subramaniyaswamy. 2019. Exploring hybrid recommender systems for personalized travel applications. In *Cognitive Informatics and Soft Computing: Proceeding of CISC 2017*. Springer, 535–544.

[23] Andrew Ng, Michael Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems* 14 (2001).

[24] Fernando Nogueira et al. 2014. Bayesian Optimization: Open source constrained global optimization tool for Python. *URL https://github.com/fmfn/BayesianOptimization* (2014).

[25] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. 2012. A literature review and classification of recommender systems research. *Expert systems with applications* 39, 11 (2012), 10059–10072.

[26] Dhanya Pramod. 2022. Privacy-preserving techniques in recommender systems: state-of-the-art review and future research agenda. *Data Technologies and Applications* n.d., ahead-of-print (2022).

[27] Sina Shaham, Ming Ding, Bo Liu, Shuping Dang, Zihuai Lin, and Jun Li. 2020. Privacy preserving location data publishing: A machine learning approach. *IEEE Transactions on Knowledge and Data Engineering* 33, 9 (2020), 3270–3283.

[28] Hyejin Shin, Sungwook Kim, Junbum Shin, and Xiaokui Xiao. 2018. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1770–1782.

[29] Shaowei Wang, Liusheng Huang, Pengzhan Wang, Hou Deng, Hongli Xu, and Wei Yang. 2016. Private weighted histogram aggregation in crowdsourcing. In *Wireless Algorithms, Systems, and Applications: 11th International Conference, WASA 2016, Bozeman, MT, USA, August 8-10, 2016. Proceedings 11*. Springer, 250–261.

[30] Xiwei Wang, Minh Nguyen, Jonathan Carr, Longyin Cui, and Kiho Lim. 2020. A group preference-based privacy-preserving POI recommender system. *ICT Express* 6, 3 (2020), 204–208.

[31] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems* 41, 3 (2023), 1–43.