electronics

MDPI

Article

A Cascade Framework for Privacy-Preserving Point-of-Interest Recommender System

Longyin Cui [1,*] and Xiwei Wang [2]

[1] Department of Computer Science, College of Engineering, University of Kentucky, Lexington, KY 40508, USA
[2] Department of Computer Science, Northeastern Illinois University, Chicago, IL 60625, USA; xwang9@neiu.edu
* Correspondence: lcu225@uky.edu

Abstract: Point-of-interest (POI) recommender systems (RSes) have gained significant popularity in recent years due to the prosperity of location-based social networks (LBSN). However, in the interest of personalization services, various sensitive contextual information is collected, causing potential privacy concerns. This paper proposes a cascaded privacy-preserving POI recommendation (CRS) framework that protects contextual information such as user comments and locations. We demonstrate a minimized trade-off between the privacy-preserving feature and prediction accuracy by applying a semi-decentralized model to real-world datasets.

Keywords: recommender system; privacy preserving; POI recommendation; collaborative filtering; clustering

check for updates

Citation: Cui, L.; Wang, X. A Cascade Framework for Privacy-Preserving Point-of-Interest Recommender System. *Electronics* **2022**, *11*, 1153. https://doi.org/10.3390/electronics11071153

Academic Editor: George Angelos Papadopoulos

Received: 26 February 2022
Accepted: 1 April 2022
Published: 6 April 2022

1. Introduction

The development of recommender systems accelerated during the past decade in different fields such as online shopping, streaming services, and point-of-interest (POI) recommendations. People have cultivated a habit of surrendering various life details in exchange for the powerful tool, colloquially known as the recommender system, in coping with information overload. However, as the awareness of privacy issues continues to grow, more and more general data protection regulations and consumer privacy acts are being implemented worldwide [1–3].

Traditional recommendation methods, e.g., matrix factorization [4], infer users' preferences on items using only the implicit or explicit feedback data. Over the years, newer models have integrated more contextual information to generate better and more accurate predictions. From timeSVD++ [5], factorization machine [6], to the neural network with attention-based novel RS models [7], recommendation methods keep including more forms of sensitive user data.

Furthermore, privacy concerns have started to arise from the framework of traditional recommender systems. In the area of POI recommendations, users employ their end devices to for restaurant and gas station recommendations, as well as for many other places that might interest them. While the users enjoying the convenience brought by a recommender system, it collects sensitive data such as private user profiles or interaction records from personal devices and stores them in a centralized server. Should there be any data breach or mismanagement, all private data may be at risk which could lead to severe consequences.

Over the years, researchers have experimented with many different privacy-preserving methods to lower the risks of privacy breach issues. For example, decentralized RS aims to push off the computation entirely to the users' end [8], removing the need for saving any sensitive data. In contrast, the k-anonymity based-RS, which keeps the central server, hides user identities and broadens certain sensitive information [9]. Other methods such as obfuscation focus more on perturbations and noise injection [10]. However, each technique still faces unique challenges today. For example, K-anonymity-like algorithms are especially vulnerable to reverse engineering when platforms purposely collect any available contextual user information; decentralization requires more communication among users and more computing powers from each device; Obfuscation, although easy to implement, may cause the original data to be no longer available.

In 2020, we proposed a group-based RS where ratings are spread out among user devices to accumulate sufficient information for forming groups in order to enhance identity obfuscation [11]. The performance showed reasonable trade-off compared to its centralized counterpart demonstrating the effectiveness of user groups in POI recommendation.

In this paper, we present the cascade recommender system (CRS), a novel framework with a cascade structure that integrates and processes sensitive data differently at various levels while maintaining high prediction accuracy. Generally speaking, the data in our proposed model are processed in three different levels. Firstly, we make use of clustering methods to replace users with centroids based on users' feedback on their visited POIs and their estimated locations. At the same time, we extract user reviews from local POIs to calculate POI similarities using the Doc2Vec method [12]. Secondly, we conduct our recommendation algorithm using the centroids' information to generate several data fragments for users to download in order to reconstruct an imputed centroids preference matrix. Finally, after the reconstruction, the users' mobile devices will impute the users' missing preferences to compare against their own, thus obtaining the final recommendations. Based on two large-scaled location-based social network (LBSN) datasets, the test results show a satisfying accuracy of POI recommendations under a secure framework.

Our main contributions are summarized as follows:

- We propose a cascade recommendation framework for POI recommendation, whereby sensitive user information is processed at different levels to accommodate the gradually unsecured environment. The multiple aggregator servers added to the model ease the extra computational privacy cost on the central server and user devices.
- We propose to sever the connection between contextual information such as text comments and user identity while keeping it serviceable. Our model collects and uses such information, but when the processed information is fed to the central server, the highly processed information cannot retrace the customers.
- We conduct experiments on real-world publicly available datasets, and the results demonstrate the effectiveness and a reasonable trade-off.

The remainder of the paper is organized as follows. Section "Related Work" discusses topics pertinent to the proposed research. "Proposed Model and Methodology" depicts the problem and our approach. In section "Datasets and Experiment", the datasets and results are presented. Finally, the conclusions and future work are given in Section "Conclusion and Future Work".

2. Related Work

In this section, we review related background context and techniques which were utilized to develop our model, including conventional recommender systems, privacy-preserving recommender systems, user comments embedding, and clustering methods.

2.1. Conventional Recommender Systems

There are a variety of models and approaches on the traditional point-of-interest (POI) recommender system (RS) and they can be differently categorized by various standards. For example, a broad categorization of the RSes can be content-based RSes [13], collaborative filtering (CF)-based RSes [14], and hybrid RSes that are a combination of multiple kinds. Collaborative filtering, as one of the most well-known recommendation techniques, focuses on finding user preferences from similar user–item interactions. After reviewing enough feedback, these RS models carry out a set of predictions by filtering down the entire item set for a new user using the knowledge obtained from similar users. The models assume that users with similar behaviors tend to interact with similar items or locations. Within CF-based RS models, there are a large number of variants including the matrix factorization (MF) models [4], regression-based latent factor models [15], Bayesian personalized ranking models [16] and deep MF models [17].

Due to the nature of POI recommendations, POI RS is heavily associated with LBSN. Figure 1 shows the diagram of data flow for a POI recommender system based on LBSN. Aside from collecting direct user feedback, more contextual information including sensitive personal data are also being stored.

In recent years, researchers increasingly focused on analyzing auxiliary information from users and POIs. Although RS models mainly use user feedback directly to predict future user actions, integrating auxiliary information vastly increases the models' performance. Thus, in another dimension, traditional recommender systems can also be categorized by the auxiliary information it focuses on to enhance the results. The earlier auxiliary information on which most researchers focused was the temporal effect, such as users' check-in time and visiting frequency [5]. This kind of study first focused on the temporal effect independently (time-aware RS) and later successively (sequence-aware RS) [18]. Meanwhile, the geographical information is an especially important trait for POI recommendations [19] and many interesting traits uniquely appear in this field. For example, users are usually active in only one city, and their visiting behaviors are often bound by where they live [8]. Moreover, the users' text comments also play an increasingly critical role in POI RSes due to the advancement from the field of natural language processing. Not only can researchers more accurately predict user behavior, but they are also able to make the machine decisions explainable [7]. Lastly, even though integrating user social relations is not as popular as previously mentioned auxiliary information, some researchers improved their results with the help of social network information [20]. Indirect methods such as group-based POI recommendations have also found ways to use the relationships among individuals and the group in which they are to boost prediction accuracy [21]. Recently, correlations have also been found between long-distance visits and social relations [22].

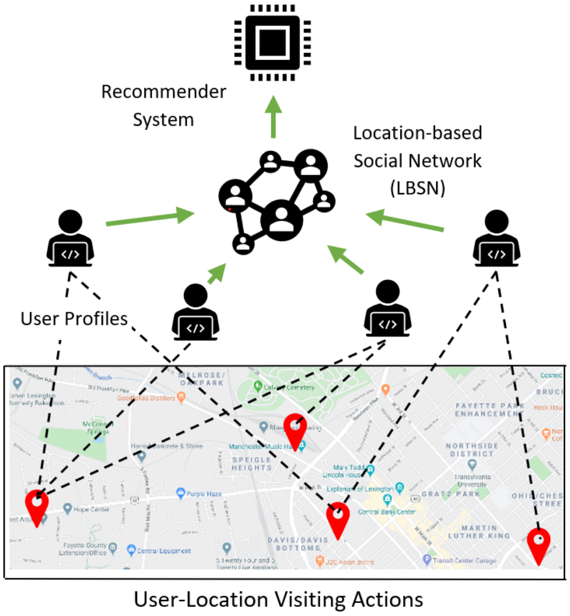
Recommender System
Location-based Social Network (LBSN)
User Profiles
User-Location Visiting Actions

Figure 1. The framework of a traditional centralized POI recommender system above an LBSN. The users' private information is collected by location-based social networks and aggregated in the central server.

2.2. Privacy-Preserving Recommender Systems

Because of the ever-increasing privacy risks that recommender systems face, researchers have proposed different methods and models over the years. The first approach is through distribution. Two typical examples were completely distributed recommender systems [8] and the federated recommender systems [23]. The former pushes the computation burden entirely to the user-end, meaning that each user has their separate private data and recommendation model. In contrast, the latter only requires the private data to be stored on each user's device while a central server learns the neural network through gradient aggregation. The second most popular method is obfuscation, such as randomized perturbation. The idea is to protect users' original private data by adding extra noises. As a traditional privacy-preservation technique, it has been frequently used in RS models [10,24–26]. At last, the third type is by adding cryptographical protocols to secure users' identities and data transmission [27–30]. The advantage is that there is no or little sacrifice of the performance of the RS model, but the downside is it requires extra computation power or time.

As mentioned before, in our previous work [11], a two-layered system was proposed, employing clustering and grouping as a means for obfuscation. Instead of focusing on improving the prediction accuracy of individuals' preferences as in [21], the grouping mechanism was implemented as a way for further concealing users' identities. While the experiment showed a promising trade-off between the privacy-preservation feature and prediction accuracy, both of them have much room for improvement.

Despite their differences, privacy-preserving recommender systems (PPRS) share the same categorizations, compositions, and challenges due to the fact that they are derived from traditional recommender systems. However, from the perspective of privacy preservation, the privacy of the utilization of auxiliary information has not been studied as thoroughly as direct user feedback such as user ratings or approvals. In this paper, we share ways for protecting and safely collecting different contextual information in a POI recommender system.

2.3. Clustering Method

Data clustering is a popular data mining technique as well as a way to mask or obfuscate private user information. Since clustering is an integral part of our experiment, we briefly mention the differences among the grouping algorithms, including k-means, spectral [31], DBSCAN [32], and fuzzy c-means [33].

k-means, probably the most well-known clustering algorithm, groups n observations into k clusters with the nearest mean. On the other hand, spectral clustering treats the grouping task as a graph partition task. Both k-means and spectral clustering require a predefined number of groups. In addition, DBSCAN clustering is based on the number of observations in a certain radius, i.e., density. Compared to the former methods, it is better at eliminating outliers, and most importantly, does not require a predefined k. Lastly, the fuzzy c-means (FCM) algorithm is a soft clustering algorithm because it assigns observations to different clusters by percentage instead of assigning an observation entirely to one group.

It is worth noting that relying on customers to be physically close to each other while there is always Wi-Fi Direct can be challenging in real-world scenarios. Therefore, comparing with the work in [11], we add a layer to use aggregator servers to actively aggregate user information. Consequently, to reflect this change, we need users' GPS information to establish the hypothetical locations of aggregator servers. Since this information is not readily available, especially in our dataset, we use the user-visited POIs' public GPS information to estimate their locations. However, the original k-means can no longer satisfy the current system due to how user affinities are constructed. To still perform the clustering, we tested extensively on clustering methods, especially DBSCAN, which automatically decides the number of clusters but finally chose to use spectral clustering, which is more potent for locational classification.

The finished framework does not involve all of the clustering methods mentioned above, but the choice of picking is based on the comparison of each one's final results.

2.4. User Comments and Word Embedding

In most e-commerce and LBSN websites, users can write free-text reviews along with an explicit or implicit rating. The text comment feedback can be very informative in the sense that it contains the evaluation and subjective opinions toward a product or a place from multiple perspectives. The review information weighs separately in potential customers' minds when choosing the next place to visit. This is especially true in POI recommendations since there are often limited ratings for unpopular places.

Some studies, such as [34,35], focus on preprocessing user comments, and these models aim to separate normal comments from "bad" comments (irrelevant comments, advertisements, or scams). Others, such as [36–38], integrate user comments to affect the latent factors for users and items while training. However, better results have been found for integrating speech recognition and natural language processing (NLP) [39]. In [7], Chen et al. proposed a neural attentional regression model with review-level explanations (NARRE) to evaluate user reviews while generating accurate predictions.

In our model, word embedding is used to find the association among the POIs with similar comment sections. The term "word embedding" refers to the problems in NLP where we want similar vectors to represent similar texts. Specifically, we use "Doc2Vec" [12], an unsupervised algorithm to generate vectors for a document, to convert user comments to vectors. This algorithm is an extension of the algorithm "Word2Vec" [40] that generates vectors for a single word.

The motivation for integrating such a feature is because privacy-preserving POI recommender systems such as [8,11,23] tend to ignore such commonly existing LBSN resources. Additionally, the clustering processes in our framework narrow down available user records, rendering other contextual information more invaluable. However, there is a strong connection between user comments and their identities. Thus, "Doc2Vec" came into place to vectorize the text information, making it impossible to convert the vector back. Additionally, the evaluation method for such vectorization is difficult to establish considering the already complicated system. Different text datasets were tested on the training of the "Doc2Vec" model, such as comments from the same city, general words, or nationwide POI comments. Eventually, we decided to experiment using the non-private POIs' similarities based on vectors to regularize the private features in our loss function.

Subsequently, the system has many moving parts, and an optimal controller becomes immediately necessary, which will be discussed in the later sections.

3. Proposed Model and Methodology

This section introduces the model from its overall structure, to the notations, as well as problem definitions. Meanwhile, the details of the framework is depicted as well and the formulation is explained.

3.1. Overall Structure

As Figure 2 shows, physically, there are three significant components in this model. The first part is the users' mobile devices where raw data are preprocessed and final recommendations are generated. In each user's POI recommendation application, the device keeps records of their private information, such as the locations they visited and the comments they left. Specifically, each device needs to carry out three data processing tasks corresponding to each type of private data. (1) Each user's comments are vectorized using the embedding tool "Doc2Vec". (2) The mobile device uploads user feedback through secure ad hoc P2P Wi-Fi Direct to the nearby aggregator server. (3) The user's device uploads a package containing recent anonymized ratings and matching vectorized comments. In the first component, the environment is comparatively secure and the data process level is low, meaning that it is closer to raw private user data.

Additionally, it is worth mentioning that mobile devices are also in charge of generating random user IDs when the connection is established. In other words, even when multiple connections were initiated by the same user at the same place over certain time, the user cannot be identified.

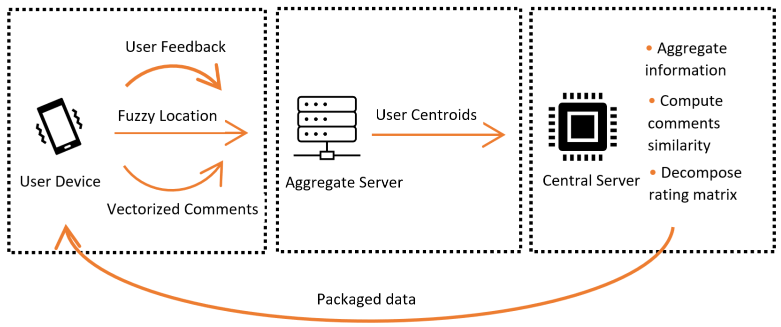

Figure 2. The overall model structure and the tasks of each component as well as the data flow.

The next component's primary task is the user clustering on the aggregator server. The clustering is based on each user's uploaded ratings and their estimated GPS locations using the locations of the POIs they visited. Note that we do not require the user's address nor does the framework require any information on the real-time user location. When an aggregator server collects enough ratings, e.g., 1000, it then performs the clustering and transfers the results to the central server. Moreover, user ratings and their comments are detached and uploaded separately. In this component, the data are further processed to cope with the less secure environment.

In the third component, its task is similar to a conventional non-privacy preserving RS. The centroids collected from all the aggregator servers are treated as physical users, and the model is then trained with "user" ratings and comments. At this point, the trained model should have generated all the necessary fragments, such as lower-ranked decomposed matrices from the rating matrix factorization, to reconstruct the rating matrix. In order to receive recommendations, users need to download these fragments to reconstruct the imputed rating matrix on their devices. In the last component, the central server, sensitive data such as user ratings, locations and reviews are substantially processed to minimize the potential data breach risks.

3.2. Notations and Problem Definitions

Suppose we use u to denote a user (customer) and i an item (POI); then U and I are the user and item sets, where $u \in U$ and $i \in I$. A rating r_{ui} indicates the preference of user u over item/POI i. In our datasets, each rating $r_{ui} \in [1, 5]$ where 1 indicates the least favored and 5 indicates the most favored. Furthermore, we use $\hat{r}_{ui}$ for predicted ratings, and r_{ui} for their observed counterparts. All the observed and unobserved r_{ui} together form the matrix R. Now, if we decompose it into two lower-ranked matrices, the column vectors p_u and q_i of each matrix represent the user and item latent factors, respectively.

A generic matrix factorization procedure can be regarded as an optimization problem, as shown below:

$$\min_{p_u,q_i} \sum_{r_{ui}\in R} (r_{ui} - \hat{r}_{ui})^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{1}$$

In our framework, however, user comments are taken into consideration on the central server and we use centroid ratings instead of user ratings. The rationale behind our changes to the above objective function is that there may be potential patterns that each user follows when reading other users' comments to make their decisions to visit. Therefore, we want to reflect the weight of such effects by introducing POI similarities calculated from the vectorized user comments. If we define s_{ij} as the similarity between two POIs i and j based on the user impression of their comment section, we have the similarity matrix S_c and the updated objective function is as follows:

$$\min_{p_u,q_i} \sum_{u,i\in R} (r_{ui} - \hat{r}_{ui})^2 + \alpha(\|b_u\|^2 + \|b_i\|^2 + \|p_u\|^2 + \|q_i\|^2) + \beta \sum_{j\in N(i)} s_{ij}(q_i - q_j)^2 \tag{2}$$

where $s_{ij} \in S_c$; b_u, b_i and $N(i)$ represent the user bias, item/POI bias and the set of item i's neighbors, respectively; α and β are two different learning rates of the regularization terms. If we define $e_{ui} = r_{ui} - \hat{r}_{ui}$, then we have the following update functions. Suppose we let $\mathcal{L}$ represent the loss function above; then, the gradients with respect to p_u and q_i can be calculated as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial p_u} &= \frac{\partial}{\partial p_u}((r_{ui} - p_u \cdot q_i)^2 + \alpha \|p_u\|^2) \\ &= -2q_i(r_{ui} - p_u \cdot q_i) + 2\alpha p_u \end{aligned} \tag{3}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial q_i} &= \frac{\partial}{\partial q_i}((r_{ui} - p_u \cdot q_i)^2 + \alpha \|q_i\|^2) + \beta \sum_{j \in N(i)} s_{ij}(q_i - q_j)^2 \\ &= -2p_u(r_{ui} - p_u \cdot q_i) + 2q_i(\alpha + \beta \left(\sum_{j \in N(i)} s_{ij} \right)) - 2\beta \sum_{j \in N(i)} s_{ij} q_j \end{aligned} \tag{4}$$

We use θ to globalize all terms' learning rate; then, the corresponding update protocols are as follows:

$$b_u \leftarrow b_u + \theta(e_{ui} - \alpha b_u) \tag{5}$$

$$b_i \leftarrow b_i + \theta(e_{ui} - \alpha b_i) \tag{6}$$

$$p_u \leftarrow p_u + \theta(e_{ui} \cdot q_i - \alpha p_u) \tag{7}$$

$$q_i \leftarrow q_i + \theta(e_{ui} \cdot p_u - (\alpha + \beta \left(\sum_{j \in N(i)} s_{ij} \right))q_i + \beta \sum_{j \in N(i)} s_{ij} q_j) \tag{8}$$

After settling the update protocol, we can use the classic stochastic gradient descent (SGD) to optimize the aforementioned objective function. In addition, it is important to mention that even though we use centroids' ratings on the central server, all centroids are still referred to as users to prevent excessive notations. However, in order to obtain all the centroids, a clustering task must be conducted on the aggregator server, as mentioned previously. We tested various clustering methods and chose kernel spectral clustering due to its better performance in terms of prediction accuracy.

Furthermore, in order to perform such clustering, we first construct the affinity matrix S by calculating the users' similarities. Suppose S_r is the user similarities based on user ratings, while S_l is based on estimated user locations. We then have the following equation:

$$S = \alpha' S_l + (1 - \alpha') S_r \tag{9}$$

where α' is the weight ratio of the two similarities and S_r is imputed based on the Pearson correlation coefficient (PCC) of users' ratings. Specifically, if we use μ to denote user mean rating, for any pair of users a and b, the similarity between the two is defined as:

$$s_{ab\_PCC} = \frac{\sum_{i \in I_{ab}} (r_{ai} - \mu_a) \cdot (r_{bi} - \mu_b)}{\sqrt{\sum_{i \in I_{ab}} (r_{ai} - \mu_a)^2} \cdot \sqrt{\sum_{i \in I_{ab}} (r_{bi} - \mu_b)^2}} \tag{10}$$

On the other hand, to complete the construction of S_l, the users' locations need to be estimated using the GPS locations of their previously visited POIs. If we let Lat_u and Lon_u represent the latitude and longitude of a user, respectively, and I_u represents the set of all POIs that a user visited, then we have the following equations:

$$Lat_u = \frac{1}{|I_u|} \sum_{i \in I_u} Lat_i \tag{11}$$

$$Lon_u = \frac{1}{|I_u|} \sum_{i \in I_u} Lon_i \tag{12}$$

where I_u denotes all the POIs that the user u has visited.

Finally, after the clustering, the aggregator servers can upload the centroids to the central server to finish the previously mentioned optimization according to Equation (2). Afterwards, the users can download b_u, b_i, p_u and q_i to their personal devices to reconstruct the complete centroid rating matrix $\hat{R}$. To receive personalized recommendations, a user's personal device needs to calculate the weighted average ratings of the top similar centroids by checking its private rating records against all centroids, thus applying the following equation:

$$\hat{r}_{\bar{u}i} = \sum_{v \in U} \bar{s}_{\bar{u}v} r_{vi} \bigg/ \sum_{v \in U} \bar{s}_{\bar{u}v} \tag{13}$$

where $\bar{u}$ is the target user and $\bar{s}$ is the vector containing similarities between the target user and all centroids. The similarities were also calculated according to Equation (10).

Furthermore, we use T_e and T_r to denote the test set and training set in later sections. For a complete summary of the notations, please refer to Table 1.

Table 1. Notation summary.

Notation	Description
$\boldsymbol{U}$	User (centroid) set
$\boldsymbol{I}$	Item (POI) set
$\boldsymbol{R}$	Rating matrix
r	Observed rating
$\hat{r}$	Predicted rating
p_u	The column vector representing the latent factors of user u
q_i	The column vector representing the latent factors of item i
b_u	User (centroid) bias
b_i	Item (POI) bias
$N(i)$	Neighbor set of item i
$\mathcal{L}$	Loss function
e	The error between the observed rating and predicted rating
θ	Global learning rate
s_{ij}	Similarity between item (POI) i and item (POI) j
S_c	Similarity matrix based on comments
S_r	Similarity matrix based on ratings
S_l	Similarity matrix based on locations
μ_a	The mean rating of all ratings from user a
Lat_u	The estimated latitude of user u
Lon_u	The estimated longitude of user u
$\bar{u}$	A real user (cannot be considered a centroid)
$\bar{s}$	The similarity vector between a real user and all centroids
$\boldsymbol{T_r}$	Training set
$\boldsymbol{T_e}$	Test set

3.3. Implementation

In order to better illustrate the implementation of the theory mentioned at the beginning of the article, we compare hypothetical real-world scenarios with our conducted experiment.

First of all, we sorted all datasets in temporal order from the perspective of time awareness. In reality, one cannot depend on interactions that happen in the future to find the affinities between current users to perform personalization. Thus, we put user records into a timely sorted "buckets" list, where each bucket represents a time window.

In the previously described theory, the users share anonymized personal information via a secure ad hoc P2P network with the nearest aggregator server, which naturally reflects the user's active visiting area. Since there are no physical user devices in the experiment, this step is reproduced by estimating the users' GPS locations using the geographical information of the visited POIs. At the same time, we vectorize the text comments left by device owners, removing any trace before leaving their host devices. In the implementation, "Doc2Vec" is used to convert user text comments into various vectors. It is worth noting that since the user IDs are anonymized, geographically or temporally separated data sharing will be considered from different users.

The proposed theory devotes a lot of effort to the clustering of users. This is not only to further obfuscate the users' identities but also to find the "trend" of the users of the present location. To implement this part in the experiment, we use users' shared ratings in the current "bucket" and their estimated locations to calculate the similarities among the users. After the clustering, a set of centroids will be generated on the aggregator servers and sent to the central server. In the experiment, we treat the centroids as new users with index-based IDs.

Then comes the matrix factorization on the central server, where there is no difference between the real-world scenario and the experiment. This step focuses on improving the imputation of the rating matrix and dimensionality reduction. It will be faster when users download the smaller-sized required "fragments" to rebuild a complete rating matrix.

Finally, the users can use their protected private information to compare against the imputed centroids rating matrix to see what "trends" they belong to more. In the experiment, the top similar centroids ratings will be extracted, and a weighted mean is taken to impute the current user′s rating vector.

4. Datasets and Experiment Results

In this section, we first introduce the datasets we use for training and testing the cascade recommender system (CRS) model. After the introduction of the metrics and evaluation methods, the proposed CRS, another two privacy model, as well as a non-privacy model are empirically compared by the same standards and procedures. Lastly, we show the tuning methods and the effects of parameters on model performance.

4.1. Datasets

Two areas are chosen from Yelp's real-world POI feedback dataset, Champaign–Urbana metropolitan area and the city of Phoenix [41] (https://www.yelp.com/dataset (accessed on 1 September 2021)). Both sets fetched the information between January 2007 and December 2017. The Champaign-Urbana area was chosen for its small scale and concision. In contrast, Phoenix is one of the areas with the largest amount of feedback data in the entire Yelp business review dataset. However, both are significantly sparse compared to conventional RS datasets such as MovieLens [42]. The statistics of the two areas can be found in Table 2.

Table 2. Datasets statistics.

Area	Users	POIs	Ratings	Density
Champaign-Urbana	11,953	1579	33,990	0.1802%
Phoenix	204,887	17,213	576,700	0.0163%

Moreover, there is no repeated feedback meaning that each user can only rate a place once. Figure 3 shows the distribution of the user rating numbers from Phoenix with a base 2 logarithmic scale on the y axis.

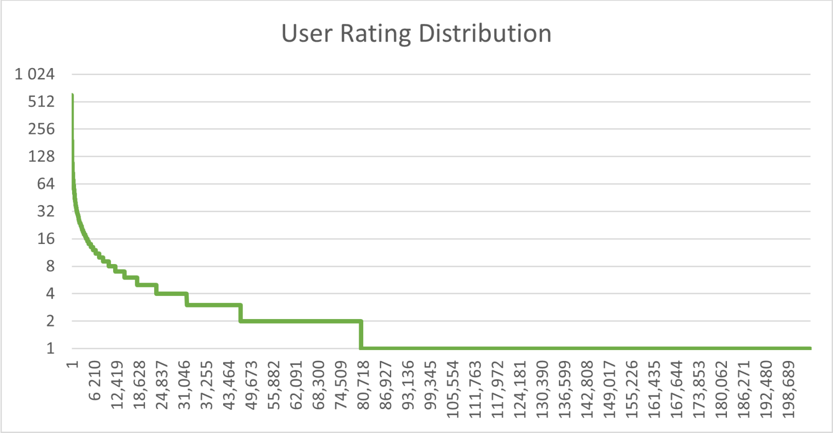

Figure 3. The user rating distribution (Phoenix). The X axis represents the numerical user IDs starting from 1. For example, if we have 10 users, they will be numbered from 1 to 10. The Y axis is the number of ratings they have left and it is on log scale to show a clearer distribution.

The datasets contain features such as user IDs, POI IDs, explicit rating feedback, timestamps, user text comments and POI GPS information. As previously noted, an explicit rating can only be $\{1, 2, 3, 4, 5\}$. Since Yelp does not collect user GPS information in real time, we can only estimate their active visiting area. For privacy reasons, the estimation is necessary both in the experiment and in practice. Assuming that all users decided to disable GPS tracking, the location of POIs can be utilized instead. Since the GPS information of each POI is not private and publicly available, it is more than viable to be utilized as a source to estimate each user's fuzzy position without privacy concerns. For example, if we determine that a user is only active in the downtown area, it is helpful to eliminate some unnecessary recommendations in suburban areas. Figure 4 shows the plot of the POI locations from which we can almost see the city's shape.

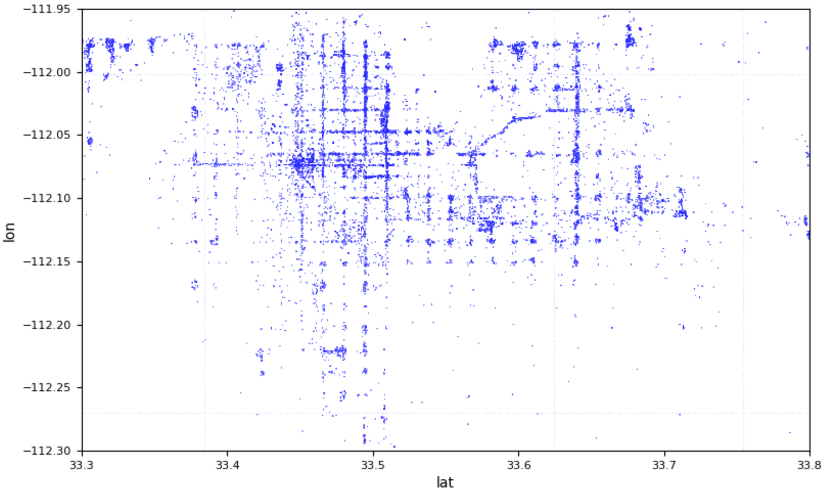

Figure 4. The user visiting locations (Phoenix). Each point represents a user visiting a real-world geographical location. (Abscissa: the latitude; Ordinate: the longitude).

4.2. Metrics and Evaluation Methods

We use both root mean square error (RMSE) and mean absolute error (MAE) to evaluate the model's prediction accuracy to better study the trade-off between privacy preservation and the RS model's accuracy. The following Equations (14) and (15) show the details of the definition:

$$RMSE = \sqrt{\frac{\sum_{u,i \in T_e} (r_{ui} - \hat{r}_{ui})^2}{n}} \tag{14}$$

$$MAE = \frac{\sum_{u,i \in T_e} |r_{ui} - \hat{r}_{ui}|}{n} \tag{15}$$

where T_e is the entire test set and n is the number of ratings compared.

Meanwhile, the experiment was designed in a way to reflect real-world scenarios. First, a time-dependent cross-validation method is conducted, ensuring the time dependencies among all the training–test pairs. For any training set, the ratings given by the users are in an earlier time state than any ratings in the corresponding test set. Moreover, the window size of each training set is built through the number of ratings uploaded by the aggregator servers.

Second, the user GPS information is estimated by the locations of POIs that this user visited to avoid the need for private user information. Since the locations of POIs are not private, an estimation of a user's active area using such information is sufficient considering the correlation between a user's physical location and their active visiting areas.

Third, all comments in the experiment can be collected beforehand due to its non-private nature, better assisting the "Doc2Vec" neural network's training. Unlike the older centroids' preference information, the information from comments can be accumulated due to their less significant temporal effect on semantics. In reality, even comments from different areas can be collected and utilized simultaneously to collaboratively build a more robust word embedding model. Algorithm 1 describes the procedure of model training and testing.

Algorithm 1: CRS Model Training/Testing Algorithm.

```
Input: Preprocessed training sets $\{Tr_1, Tr_2, \ldots Tr_n\}$, preprocessed test sets $\{ Te_1, Te_2, \ldots Te_n \}$, the POI similarity matrix $S_c$, and the POIs' GPS information
Output: MAE, RMSE
for each training set $Tr_i$ do
    Calculate each user's longitude $Lon_u$ and latitude $Lat_u$ according to Equations (11) and (12)
    Construct affinity matrix $S$ based on Equations (9) and (10)
    Perform clustering to generate the centroids' rating matrix $R$
    for each $r_{ui}$ in $R$ do
        Calculate the gradients of $\mathcal{L}$ with respect $b_u$, $b_i$, $p_u$, and $q_i$
        Update iteratively according to Equations (5)–(8)
    end
    Reconstruct the centroids' rating matrix $\hat{R}$
    for each $r_{\bar{u}i}$ in the corresponding $Te_i$ do
        Calculate the corresponding $\hat{r}_{\bar{u}i}$ according to Equation (13)
    end
    Calculate $RMSE_i$ and $MAE_i$ according to Equations (14) and (15) (We calculate $RMSE$ for the convenient averaging)
end
Calculate the average $RMSE$ and $MAE$
return $RMSE$ and $MAE$
```

4.3. Results and Comparison

This section compares our framework with three other models: a baseline model, a purely decentralized model and a federated RS model. Afterwards, we briefly show the overall improvement in contrast with our preliminary work in [11].

- We start by choosing the well-known biased matrix factorization (MF) [4] as the baseline model. It is a non-privacy centralized RS model with simple structures and reliable performance, and this model was thoroughly studied and constantly compared.
- The federated recommender system, MetaMF [23], has a distinct advantage in terms of privacy protection and providing exceptionally accurate predictions. Despite requiring

a global model on a central server, this method distributes its training to multiple local devices, reducing the risk of leaking private user data.

- The decentralized recommender system (DMF) [8] is a purely distributed privacy-preserving recommender system that relies on matrix factorization and gradient exchange. Due to its distributive framework, the only leakage risk of data is the loss function's gradient. Each user has a dynamic global and personal model combined, making the hacking even harder.

We compare our CRS with the three aforementioned models with the classic biased MF as the baseline. For fairness, the data filtering methods and preprocessing are the same for all models involved. The overall results are presented in Table 3. It is worth noting that even though the scales of the two areas are different, we strive to show the same detail in our results presenting the results from each training–test pair. The fold size is 1000 ratings for the dataset of Champaign-Urbana and 3000 ratings for the city of Phoenix. The RMSE and MAE of each fold are shown in Figures 5–8.

Table 3. Average Result Comparison (Champaign-Urbana).

		Champaign-Urbana		
Model Name	**Biased MF**	**CRS**	**MetaMF**	**DMF**
RMSE	1.1966	**1.1685**	1.3942	1.4984
MAE	0.9537	**0.9494**	1.1268	1.2018
		Phoenix		
Model Name	**Biased MF**	**CRS**	**MetaMF**	**DMF**
RMSE	**1.0463**	1.0482	1.3647	1.4534
MAE	0.8247	**0.8235**	1.0647	1.0872

Among the four compared models, the CRS has the best performance in terms of accuracy. The CRS is close to the baseline, the non-private centralized recommendation model. In specific folds, the performance for the CRS is even better. All the models use the same training and test datasets. The following subsection will discuss the tuning of models and hyper-parameters.

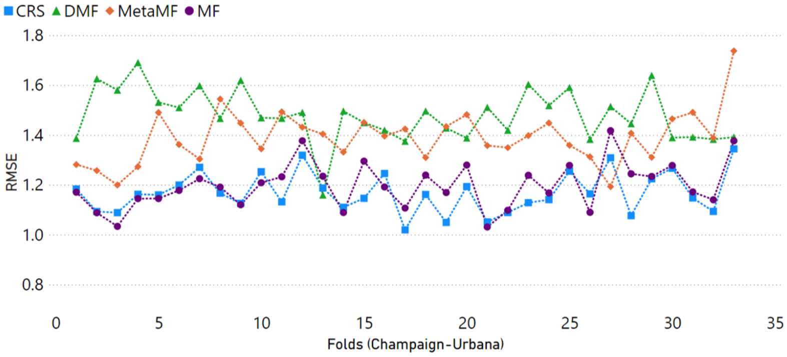

Figure 5. The RMSE comparisons among the four models for each training–test dataset pair (Champaign-Urbana). The dataset is partitioned chronologically. (Abscissa: value of RMSE; ordinate: index of folds).

The testing procedure mimics real-world scenarios where recommendations are temporally given in succession. In other words, only old data can be used to predict newer data. The previous test set will be the training set in the subsequent round of testing. As the result figures have shown, each inflection point reflects a result from a training–test pair. To maintain a steady curve of accuracy, we choose to maintain the same number of ratings in all of the training and test set instead of having the same time span from user ratings. This is also to ensure that we have enough records for each time period for generating recommendations when simulating the real-world scenario.

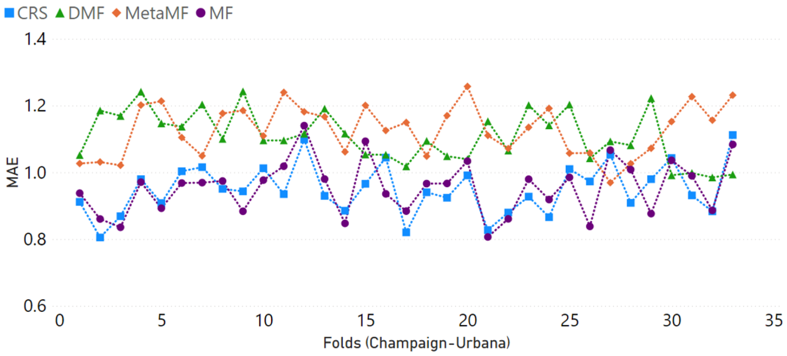

Figure 6. The MAE comparisons among the four models for each training–test dataset pair (Champaign-Urbana). The dataset is partitioned chronologically. (Abscissa: value of MAE; Ordinate: index of folds).

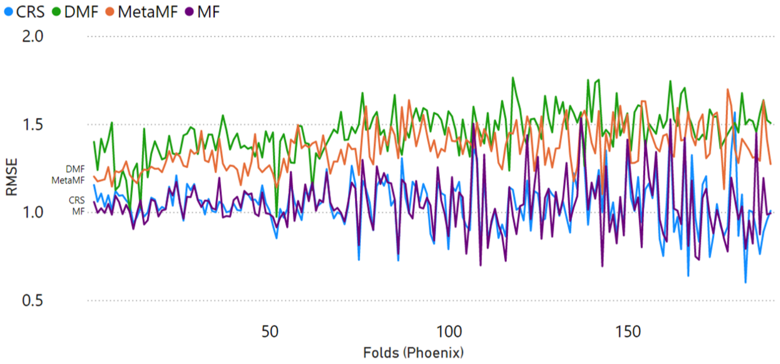

Figure 7. The RMSE comparisons among the four models for each training–test dataset pair (City of Phoenix). The dataset is partitioned chronologically. (Abscissa: value of RMSE; ordinate: index of folds).

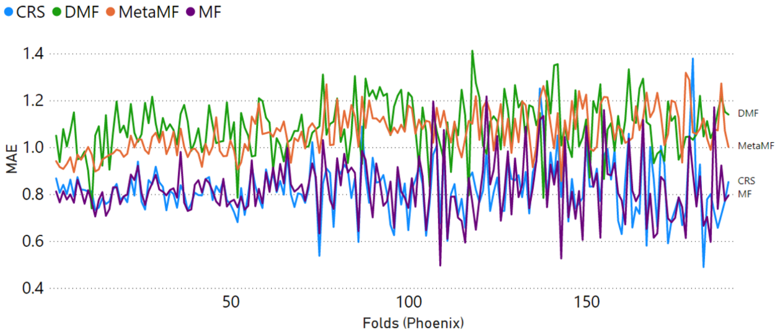

Figure 8. The MAE comparisons among the four models for each training–test dataset pair (City of Phoenix). The dataset is partitioned chronologically. (Abscissa: value of MAE; ordinate: index of folds).

For both RMSE and MAE, the CRS shows a very steady and good performance on both datasets in an environment that is very close to real-world settings. Compared to DMF, the CRS does not push all the computing burdens onto the users' end, ensuring an easier, more light-weighted way of generating recommendations. At the same time, the CRS's central server relies on the data that are already processed by the aggregator server, making the model's training process more accessible and faster. The CRS is also more modulized and detachable compared to the federated learning-based recommender systems. In comparison to its previous prototype [11], the average value of MAE falls from 1.118 to 0.949.

In terms of privacy preserving, unlike other privacy RS models, users in CRS anonymously and dynamically share private information with the aggregator server on an ad hoc P2P network via Wi-Fi Direct. In other words, each time a user shares information, their IDs are temporary and different. Moreover, contextual information such as text comments and GPS information is protected and preserved. Other privacy-preserving models such as DMF and MetaMF require that users have a constant identity in order to maintain a long-term communication.

4.4. Parameters and Tuning

The previous sections introduced the CRS model from three aspects: the users' mobile devices, aggregator servers and the central server. Since various parameters can directly affect the result in CRS, we connect the discussion of parameters with these three parts correspondingly for easier understanding.

- During the aggregation stage, there are two crucial parameters. First, the performing of the clustering is based on the affinity matrix according to the Equation (9). In this equation, α' denotes the ratio or weight of either part in building the final affinity matrix. Second, the number of centroids, $n\_clusters$, is required due to the clustering method we chose.
- On the central server, we have the number of latent factors k, the global learning rate θ, and each specific learning rate α and β of each term from the loss function. We used n_epochs to denote the number of epochs used in SGD for each training process.
- On users' mobile devices, after users reconstruct the imputed rating matrix, they will compute their personalized predictions according to Equation (13). During our experiment, we found that not including all centroids yields more accurate results. Here, we use $n\_centroids$ to denote the number of centroids involved.

Due to the number of parameters, broad intervals and contentiousness, we seek an automatic approach to our model tuning instead of brute-force searching or heuristic random guessing based on human experiences. In the field of optimal controls, researchers have made significant progress during the past decade on adaptive control schemes and solving nonlinear systems [43,44]. However, since this is our initial attempt to adopt such methods, our strategy is to combine the use of a developed library with stable performance [45] and manual intervention. Therefore, we adopted a Bayesian optimization framework wherein the model's performance is treated as a sample from a Gaussian process. This approach, proposed by Snoek et al. [46], considers the entire training–test process as a continuous function.

According to the authors, the Bayesian optimization method reduces the heavy computation task of finding the close-to-optimal combinations of parameters by using proxy optimization. In the beginning, we construct a function that wraps up our entire algorithm (Algorithm 1). In other words, instead of optimizing each part of our framework, we include all parts of the algorithm and put them into one "function." The number of observations keeps growing as our program runs, and a posterior distribution is gradually built. Each time a parameters–error pair is plotted, exploration strategies such as the upper confidence bound and expected improvement are used to determine the next set of parameters to try. The number of steps to find optimal parameters is claimed to be minimized compared to a brute-force strategy. Specifically, in our experiment, we use the method to estimate the range of optimal parameters in the tuning process and to perform manual fine-tuning. For example, we would combine the observed parameters from different top-ranked results and increase or decrease specific parameters to see whether a better performance can be achieved.

In CRS, we set α' to 0 to maximize the effect of user location. However, this does not mean it is always a better choice to exclude all contributions from the similarities based on user feedback. For example, Figure 9 shows the comparison of the final RMSE by selecting different α' in 1.0, 0.9, 0.8 ... 0.0. $n\_clusters$ is set to 6. It is worth noting that the best result was achieved when k is 1. This may be caused by the sparsity of the dataset and that the data are clustered. For the coefficients of the loss function, θ, α and β are set to 0.12, 0.02

and 0.6, respectively. For each round of training, *n_centroids* is as small as 20 to achieve convergence. Finally, *n_centroids* is set to 9.

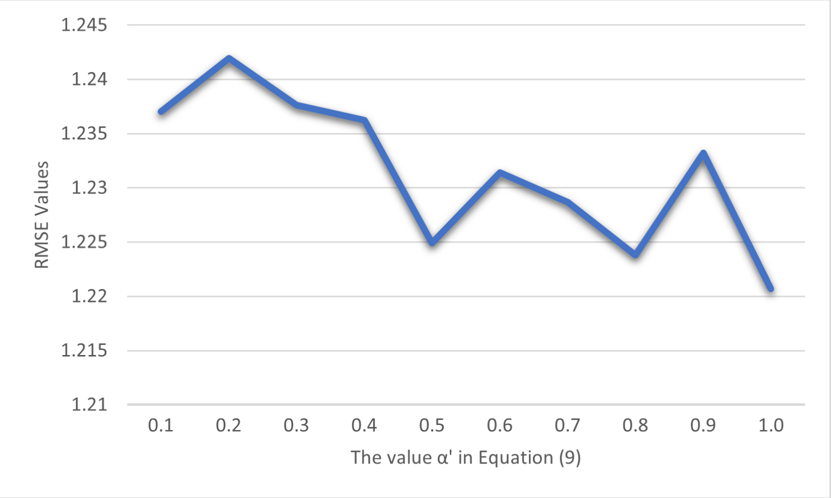

Figure 9. The impact on the average RMSE value from changes on α' in Equation (9) (Champaign-Urbana). The α' indicates the weight of geographical similarities among the users played in the affinity matrix S.

5. Conclusions and Future Work

In this paper, we propose a privacy-preserving POI recommendation framework CRS that utilizes ad hoc wireless peer-to-peer communication and user clustering. This framework, while having the advantage of automatically estimating users' location, can also obfuscate users' text comments. By adopting a cascade structure, the CRS processes risky data under secure connections and uploads processed secure data when facing a potentially malicious environment. Separated into three parts, the CRS keeps the most vulnerable data on the users' end while pushing the well-prepared data to the central server. Because of the nature of clustering, CRS has excellent scalability as well as geolocational awareness. Furthermore, we mimic a real-world scenario by preparing the cross-validation folds temporally sequential. Under such a testing strategy alongside real-world datasets, the CRS shows a performance close to a centralized non-private model.

Meanwhile, the CRS framework does face some difficulties when facing different datasets or uncertain real-world situations. Two major challenges remain the need for ample input and its severance on the connections between the users and their historical information. The former one is caused by the nature of CRS where constant clustering is required. The latter one, even though protecting the user from being traced, will cap the personalization performance since personal historical records cannot be accumulated to further improve the accuracy of the predictions.

In the future, we would like to investigate the impact of more advanced contextual information processing tools in CRS. In addition, we will develop more on the communication among the aggregator severs, possibly with transfer learning and gradient exchanges.

Author Contributions: Conceptualization, L.C. and X.W.; methodology, L.C. and X.W.; software, L.C.; validation, X.W.; formal analysis, L.C.; investigation, L.C. and X.W.; resources, X.W.; data curation, L.C.; writing—original draft preparation, L.C.; writing—review and editing, X.W.; visualization, L.C.; supervision, X.W.; project administration, X.W.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a Committee on Organized Research (COR) grant from Northeastern Illinois University.

Data Availability Statement: The data are available from the corresponding author on request.

Acknowledgments: We would like to express our gratitude to our colleagues from the Department of Computer Science at the University of Kentucky and Northeastern Illinois University whose insights and expertise inspired us. Particularly, I would also like to extend my sincere thanks to Zongming Fei, who is my advisor for his support.

Conflicts of Interest: The authors claim that there is no conflict of interest.

References

1. Voigt, P.; Von dem Bussche, A. *The EU General Data Protection Regulation (GDPR)*; A Practical Guide; Springer International Publishing: Cham, Switzerland, 2017; Volume 10.
2. Goldman, E. An introduction to the California consumer privacy act (CCPA). Santa Clara Univ. Legal Studies Research Paper. 2020. Available online: https://ssrn.com/abstract=3211013 (accessed on 1 September 2021).
3. Determann, L.; Ruan, Z.J.; Gao, T.; Tam, J. China's draft personal information protection law. *J. Data Prot. Priv.* **2021**, *4*, 235–259.
4. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
5. Koren, Y. Collaborative filtering with temporal dynamics. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June 2009–1 July 2009; pp. 447–456.
6. Rendle, S. Factorization Machines. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, Australia, 13 December 2010; pp. 995–1000.
7. Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1583–1592.
8. Chen, C.; Liu, Z.; Zhao, P.; Zhou, J.; Li, X. Privacy preserving point-of-interest Recommendation using decentralized Matrix Factorization. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
9. Wei, R.; Tian, H.; Shen, H. Improving k-anonymity based privacy preservation for collaborative filtering. *Comput. Electr. Eng.* **2018**, *67*, 509–519. [CrossRef]
10. Wang, X.; Zhang, J.; Wang, Y. Trust-aware privacy-preserving recommender system. In Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, Xi'an, China, 18–20 June 2016; pp. 107–115.
11. Wang, X.; Nguyen, M.; Carr, J.; Cui, L.; Lim, K. A group preference-based privacy-preserving POI recommender system. *ICT Express* **2020**, *6*, 204–208. [CrossRef]
12. Lau, J.H.; Baldwin, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv* **2016**, arXiv:1607.05368.
13. Pazzani, M.J.; Billsus, D. Content-based recommendation systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 325–341.
14. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*, 5–53. [CrossRef]
15. Agarwal, D.; Chen, B.C. Regression-based latent factor models. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June 2009–1 July 2009; pp. 19–28.
16. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
17. Xue, H.J.; Dai, X.; Zhang, J.; Huang, S.; Chen, J. Deep matrix factorization models for recommender systems. In Proceedings of the 2017 International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; Volume 17, pp. 3203–3209.
18. Raza, S.; Ding, C. Progress in context-aware recommender systems—An overview. *Comput. Sci. Rev.* **2019**, *31*, 84–97. [CrossRef]
19. Levandoski, J.J.; Sarwat, M.; Eldawy, A.; Mokbel, M.F. Lars: A location-aware recommender system. In Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, Arlington, VA, USA, 1–5 April 2012; pp. 450–461.
20. Camacho, L.A.G.; Alves-Souza, S.N. Social network data to alleviate cold-start in recommender system: A systematic review. *Inf. Process. Manag.* **2018**, *54*, 529–544. [CrossRef]
21. Sojahrood, Z.B.; Taleai, M. A POI group recommendation method in location-based social networks based on user influence. *Expert Syst. Appl.* **2021**, *171*, 114593. [CrossRef]
22. Chen, C.; Zhou, J.; Wu, B.; Fang, W.; Wang, L.; Qi, Y.; Zheng, X. Practical privacy preserving POI recommendation. *ACM Trans. Intell. Syst. Technol. (TIST)* **2020**, *11*, 1–20. [CrossRef]
23. Lin, Y.; Ren, P.; Chen, Z.; Ren, Z.; Yu, D.; Ma, J.; Rijke, M.d.; Cheng, X. Meta matrix factorization for federated rating predictions. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 11–15 July 2020; pp. 981–990.
24. Shokri, R.; Pedarsani, P.; Theodorakopoulos, G.; Hubaux, J.P. Preserving privacy in collaborative filtering through distributed aggregation of offline profiles. In Proceedings of the Third ACM Conference on Recommender Systems, New York, NY, USA, 23–25 October 2009; pp. 157–164.
25. Shin, H.; Kim, S.; Shin, J.; Xiao, X. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1770–1782. [CrossRef]

26. Cui, L.; Wang, X.; Zhang, J. Vendor-Based Privacy-Preserving POI Recommendation Network. In Proceedings of the International Conference on Mobile Multimedia Communications, Online, 23–25 July 2021; pp. 477–490.
27. Kikuchi, H.; Kizawa, H.; Tada, M. Privacy-preserving collaborative filtering schemes. In Proceedings of the 2009 International Conference on Availability, Reliability and Security, Fukuoka, Japan, 16–19 March 2009; pp. 911–916.
28. Erkin, Z.; Beye, M.; Veugen, T.; Lagendijk, R. Privacy-preserving content-based recommendations through homomorphic encryption. In Proceedings of the Information Theory in the Benelux and The 2nd Joint WIC/IEEE Symposium on Information Theory and Signal Processing in the Benelux, Boekelo, The Netherlands, 24–25 May 2012; p. 71.
29. Badsha, S.; Yi, X.; Khalil, I. A practical privacy-preserving recommender system. *Data Sci. Eng.* **2016**, *1*, 161–177. [CrossRef]
30. Ravi, L.; Subramaniyaswamy, V.; Devarajan, M.; Ravichandran, K.; Arunkumar, S.; Indragandhi, V.; Vijayakumar, V. SECRECSY: A secure framework for enhanced privacy-preserving location recommendations in cloud environment. *Wirel. Pers. Commun.* **2019**, *108*, 1869–1907. [CrossRef]
31. Ng, A.; Jordan, M.; Weiss, Y. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **2001**, *14*, 849–856.
32. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the KDD, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
33. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [CrossRef]
34. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
35. Zhang, Y.; Tan, Y.; Zhang, M.; Liu, Y.; Chua, T.S.; Ma, S. Catch the black sheep: Unified framework for shilling attack detection based on fraudulent action propagation. In Proceedings of the The Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
36. McAuley, J.; Leskovec, J. Hidden factors and hidden topics: Understanding rating dimensions with review text. In Proceedings of the 7th ACM International Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 165–172.
37. Ling, G.; Lyu, M.R.; King, I. Ratings meet reviews, a combined approach to recommend. In Proceedings of the 8th ACM Conference on Recommender Systems, Foster City, CA, USA, 6–10 October 2014; pp. 105–112.
38. Zhang, Y.; Lai, G.; Zhang, M.; Zhang, Y.; Liu, Y.; Ma, S. Explicit factor models for explainable Recommendation based on phrase-level sentiment analysis. In Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, New York, NY, USA, 6–11 July 2014; pp. 83–92.
39. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
40. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
41. Yelp Dataset. Available online: https://www.yelp.com/dataset (accessed on 1 March 2020).
42. Harper, F.M.; Konstan, J.A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* **2015**, *5*, 1–19. [CrossRef]
43. Roman, R.C.; Precup, R.E.; Petriu, E.M. Hybrid data-driven fuzzy active disturbance rejection control for tower crane systems. *Eur. J. Control* **2021**, *58*, 373–387. [CrossRef]
44. Zhu, Z.; Pan, Y.; Zhou, Q.; Lu, C. Event-triggered adaptive fuzzy control for stochastic nonlinear systems with unmeasured states and unknown backlash-like hysteresis. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 1273–1283. [CrossRef]
45. Nogueira, F. Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python. 2014. Available online: https://github.com/fmfn/BayesianOptimization (accessed on 1 October 2021).
46. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 2960–2968.