DOI: 10.1111/tgis.12769

RESEARCH ARTICLE

Transactions in GIS WILEY

A privacy-preserving framework for location recommendation using decentralized collaborative machine learning

Jinmeng Rao[1] | Song Gao[1] | Mingxiao Li[1,2] | Qunying Huang[3]

[1]Geospatial Data Science Lab, Department of Geography, University of Wisconsin-Madison, Madison, WI, USA

[2]Shenzhen Key Laboratory of Spatial Information Smart Sensing and Services, Shenzhen University, Shenzhen, China

[3]Spatial Computing and Data Mining Lab, Department of Geography, University of Wisconsin-Madison, Madison, WI, USA

Correspondence
Song Gao, Geospatial Data Science Lab, Department of Geography, University of Wisconsin-Madison, Madison, WI, USA.
Email: song.gao@wisc.edu

Funding information
American Family Insurance Funding Initiative; Wisconsin Alumni Research Foundation, Grant/Award Number: 135-VCRGE

Abstract

The nowadays ubiquitous location-aware mobile devices have contributed to the rapid growth of individual-level location data. Such data are usually collected by location-based service platforms as training data to improve their predictive models' performance, but the collection of such data may raise public concerns about privacy issues. In this study, we introduce a privacy-preserving location recommendation framework based on a decentralized collaborative machine learning approach: federated learning. Compared with traditional centralized learning frameworks, we keep users' data on their own devices and train the model locally so that their data remain private. The local model parameters are aggregated and updated through secure multiple-party computation to achieve collaborative learning among users while preserving privacy. Our framework also integrates information about transportation infrastructure, place safety, and flow-based spatial interaction to further improve recommendation accuracy. We further design two attack cases to examine the privacy protection effectiveness and robustness of the framework. The results show that our framework achieves a better balance on the privacy–utility trade-off compared with traditional centralized learning methods. The results and ensuing discussion offer new insights into privacy-preserving geospatial artificial intelligence and promote geoprivacy in location-based services.

1 | INTRODUCTION

The prevalence of mobile devices and communication network infrastructure connects people worldwide, leading to a boom in location-based services and location-based social networks (Cho, Myers, & Leskovec, 2011; Gao, Janowicz, & Couclelis, 2017; Huang, Gartner, Krisp, Raubal, & Van de Weghe, 2018). Such location-based networks enable people to share their social activities, preferences, and online reviews for different types of places (e.g., points of interest [POIs]), all of which are highly valuable for location-based marketing. According to a report by Fortune Business Insights (https://www.fortunebusinessinsights.com/industry-reports/location-based-services-market-101060), the market value of location-based services was $18.54 billion in 2019, and is predicted to reach $66.61 billion by 2026. Although location data benefit business analysis (Liang, Gao, Cai, Foutz, & Wu, 2020), POI recommendation (Bao, Zheng, Wilkie, & Mokbel, 2015; Gao, Tang, Hu, & Liu, 2015; Ye, Yin, & Lee, 2010), urban informatics and human dynamics research (Cho et al., 2011; Gao & Mai, 2018; Huang et al., 2018; McKenzie, Janowicz, Gao, Yang, & Hu, 2015; Shaw & Sui, 2020; Sun, Fan, Li, & Zipf, 2016), it also raises public concerns about privacy issues (Armstrong & Ruggles, 2005; Keßler & McKenzie, 2018; McKenzie, Keßler, & Andris, 2019; Seidl, Jankowski, & Tsou, 2016). For example, user data are usually collected and stored in centralized databases owned by enterprise platforms. Users worry that someone may re-identify them by analyzing their check-in history and mobility behavior, which can be done with location clustering or data mining algorithms (Huang, Cao, & Wang, 2014; Liu, Huang, & Gao, 2019). Furthermore, users worry that their home location and other private locations may be disclosed, thus making them vulnerable. Privacy protection techniques such as de-identification, geomasking, and differential privacy are commonly used by the platforms to add more uncertainty (e.g., noise) to the data (Duckham & Kulik, 2005; Gruteser & Grunwald, 2003; Krumm, 2009; Kwan, Casas, & Schmitz, 2004; Seidl et al., 2016; Xiao & Xiong, 2015). Still, the trade-off between privacy and data utility is hard to control (Rao, Gao, Kang, & Huang, 2020). Besides, privacy exposure risks also exist in the process of uploading and storing user data (Bertino, Thuraisingham, Gertz, & Damiani, 2008).

To this end, we present a novel privacy-preserving framework for location recommendation based on federated learning, a decentralized collaborative machine learning technique. It keeps users' check-in data on their own devices rather than on a centralized database so that users' data will not be uploaded or shared. A location recommendation model is then deployed on each user's device to locally train the data and model location preferences. Apart from the users' profiles and check-in POI category, we also integrate the spatio-temporal features (i.e., check-in location and time-stamp) and auxiliary public data (i.e., transportation infrastructure, place safety, and flow-based spatial interaction) to further improve the location recommendation accuracy. The weighted average of all local model parameters is derived through secure multiple party computation to support privacy-preserving collaborative learning among users.

The following two research questions are investigated in this research.

> RQ1 From the privacy perspective, how and to what extent can the proposed decentralized collaborative learning framework protect users' privacy?
>
> RQ2 From the utility perspective, what is the performance of the proposed framework in the location recommendation task on the premise of preserving privacy?

In RQ1, privacy represents users' right to prevent the disclosure of their private data (e.g., private user profiles, private check-in history) and the information that can be derived from the data (e.g., home locations of users). In RQ2, performance means the accuracy of the recommendations made by the model. The contribution of our work is fourfold. First, we propose a privacy-preserving framework for location recommendation based on federated learning, which keeps users' data on their own devices and performs collaborative learning in a decentralized manner. Second, we introduce a data encoding module based on spatial indexing to encode various continuous and discrete data. Third,

we evaluate the framework on a real-world social media check-in data set and explore the trade-off between privacy protection effectiveness and model utility. Fourth, we design two attack cases to further verify the framework's ability to protect location privacy and examine its robustness.

The remainder of the article is organized as follows. Section 2 introduces related work, including some privacy protection techniques and privacy-preserving recommendation frameworks. Section 3 describes the structure and components of our framework, including a data encoding module, a private modeling module, a secure aggregation module, and a multi-source public database. In Section 4 we evaluate the performance of our framework using a real-world social media check-in data set and conduct two attack cases to examine both the effectiveness of privacy protection and the utility of the model. In Section 5 we explore the factors affecting model performance, the privacy–utility trade-off, and the limitations of our framework. Finally, Section 6 summarizes the research and outlines our future work.

2 | RELATED WORK

2.1 | Privacy protection techniques

The increasing public concern about privacy issues has prompted the emergence of a number of privacy protection techniques. An intuitive method is de-identification, in which the data are anonymized by deleting or masking users' identity from the data set (i.e., identifiers such as name or ID). This common practice ensures the de-identified data set does not contain any identifiers, but other attributes in the data set may still serve as strong "quasi-identifiers" (gender, date of birth, check-in location and time, etc.) which can still be used to re-identify users and result in serious privacy disclosure (Chow & Mokbel, 2011). Even though such quasi-identifiers are also sanitized, one may link the data set to some external data sets to de-anonymize the data and re-identify users (Narayanan & Shmatikov, 2008). To further enhance the effectiveness of privacy protection, many privacy guarantees and protection methods are proposed, such as k-anonymity (Niu, Li, Zhu, Cao, & Li, 2014; Zhu, Yang, Wang, Wang, & Lee, 2019), ℓ-diversity (Machanavajjhala, Kifer, Gehrke, & Venkitasubramaniam, 2007), t-closeness (Li, Li, & Venkatasubramanian, 2007), and differential privacy (Dwork, 2008). Among them, k-anonymity is particularly popular in protecting location privacy (Gruteser & Grunwald, 2003; Palanisamy & Liu, 2011). The key idea of k-anonymity is to ensure the information for each user cannot be perceived from at least $k-1$ users whose information shows up in the data set (Jain, Gyanchandani, & Khare, 2016). Another group of techniques, geomasking, focuses more on the spatial dimension. They mask or blur the locations so that the original locations of the data will not be revealed, while still preserving some spatial characteristics (Gao et al., 2019; Hampton et al., 2010; Rao et al., 2020; Swanlund, Schuurman, Zandbergen, & Brussoni, 2020). A common practice is to directly aggregate the locations into geographic or administrative units. However, many recent studies point out that in many scenarios the aggregation (to fine units) is proved to provide only little anonymity (De Montjoye, Hidalgo, Verleysen, & Blondel, 2013), and some important information can still be revealed from the aggregated results (Xu et al., 2017). Increasing the level of the aggregation geographic unit to make it a coarser unit may mitigate such a situation, but this will in turn result in loss of spatial resolution for analytics, resulting in limited data utility (Gao et al., 2019). Other commonly used geomasking techniques include random perturbation (Kwan et al., 2004; Seidl et al., 2016) and Gaussian geomasking (Gao et al., 2019).

The aforementioned techniques were originally designed for centralized data sets. In this mode, users have to share their data to a central server in some way, running the risk of leaking privacy (Vaudenay, 2020). While these techniques can mitigate the privacy risk to a certain degree, they cannot fundamentally address this issue with a centralized data storage and processing mechanism. Hence, another group of methods utilizes decentralization to mitigate the privacy leakage risk. The key idea is to keep users' data on their own devices and prevent access from others. The decentralized data set greatly preserves privacy, while the decentralized nature of data may

complicate data usage and model updates (Kaissis, Makowski, Rückert, & Braren, 2020). As a decentralized collaborative learning technique, federated learning is able to collaboratively train a model on the data set distributed over a large number of devices (Konečný et al., 2016). Each device computes an update based on its local data, sends the update to a central server, and receives the aggregated new parameters from the server. Federated learning provides an infrastructure for privacy protection, but it does not guarantee privacy (Kaissis et al., 2020). Common privacy techniques for federated learning include differential privacy, secure multi-party computation (SMPC), and homomorphic encryption (HE) (Yang, Liu, Chen, & Tong, 2019). In this research, our framework utilizes SMPC and HE to achieve secure aggregation and privacy-preserving location inference capability.

2.2 | Privacy-preserving recommendation

Recommendation systems have been widely applied in plenty of real-world industrial scenarios. These systems make personalized recommendations based largely on users' profiles, preferences, check-in, and browsing history, which raises public concerns about privacy issues. Thus, some of the above-mentioned privacy protection techniques are integrated into recommendation systems to make privacy-preserving recommendations. For instance, some recommendation systems utilize random perturbation (Polatidis, Georgiadis, Pimenidis, & Mouratidis, 2017), *k*-anonymity (Casino, Domingo-Ferrer, Patsakis, Puig, & Solanas, 2015; Zigomitros, Papageorgiou, & Patsakis, 2016), and differential privacy (McSherry & Mironov, 2009; Meng et al., 2018) to preserve privacy. While these methods are straightforward, the trade-off between the privacy protection and model utility is hard to control (Rao et al., 2020). Some studies utilize cryptography-based techniques such as SMPC and HE to preserve privacy effectively. For instance, Nikolaenko et al. (2013) present a privacy-preserving protocol that combines SMPC and HE for efficient matrix factorization on user ratings. Although the utilized privacy protection techniques vary, what these methods have in common is that they are designed for centralized training. Instead, our framework is designed for a decentralized collaborative learning environment, in which users' data are kept only on their local devices.

An early work similar to ours is PriRec (Chen et al., 2020), where they propose a distributed privacy-preserving POI recommendation framework. However, our work differs from PriRec in at least four ways. First, our framework follows the classic federated learning procedure, distributing users' data and models to their own devices and using the server to perform secure aggregation after the local model training process. PriRec divides the model into two parts and deploys them on the client side and server side, respectively, and the secure aggregation protocol and procedure it utilizes are also different. Second, our framework introduces a spatial-indexing-based data encoding module to encode and integrate georeferenced features and spatio-temporal features. Auxiliary public data such as transportation infrastructure, place safety, and flow-based spatial interaction are also encoded and integrated into our modeling process. PriRec, by contrast, utilizes user–POI interaction and a geographic adjacency network. Third, our framework does not collect or store any user data. PriRec collects user–POI interaction data from users, although the data are processed by a local differential privacy technique. Fourth, and finally, we design two attack cases to further examine the effectiveness of privacy protection and the robustness of our framework, which are not investigated in PriRec.

3 | METHODS

Figure 1 depicts the overall workflow of the proposed framework, which consists of four components: a data encoding module; a private modeling module; a secure aggregation module; and a multi-source public database. Our method aims to learn user location preferences from users' check-in data and auxiliary public data (e.g., transportation infrastructure, place safety, and flow-based spatial interaction) without disclosing privacy. Check-in data in

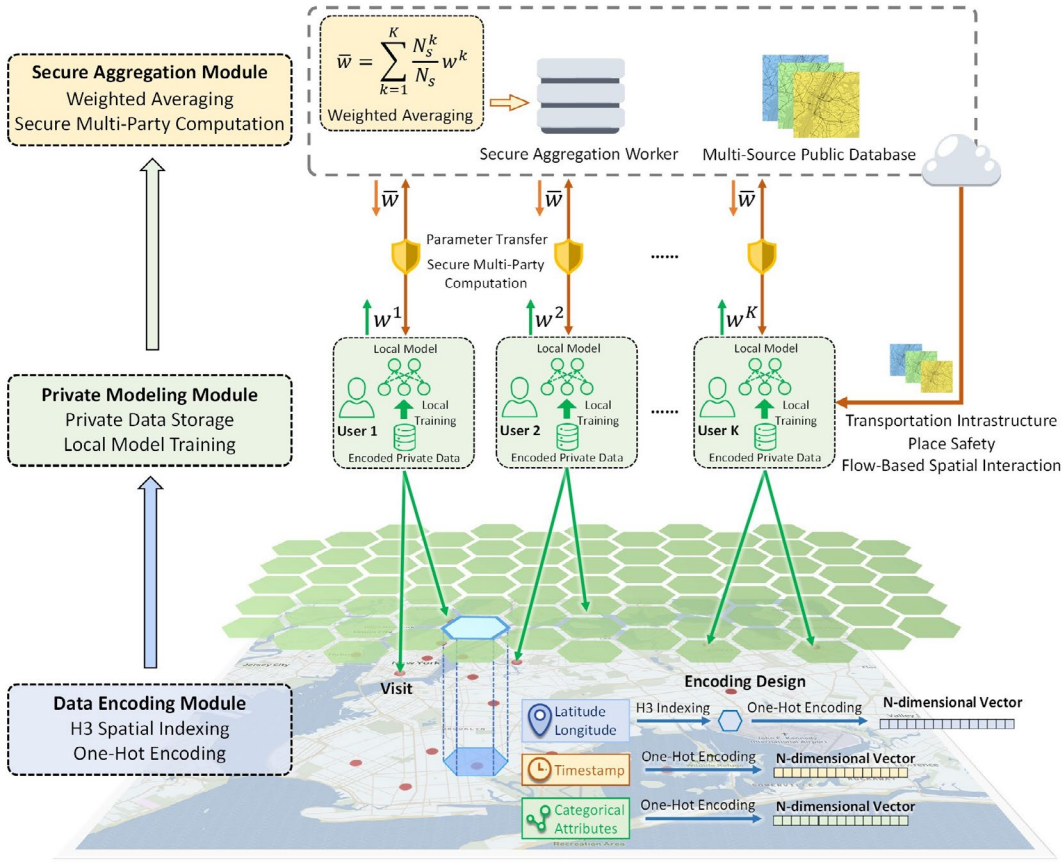

FIGURE 1 The overall workflow of the proposed framework

this study are data that contain location information and are purposefully recorded by the user. Such check-in data could be traditional social platform check-ins (e.g., Twitter check-ins) or any other data following our definition (e.g., Yelp online reviews). One property of such data is that they can be either public or anonymous. For example, some people do not want others (even the platform) to know some places they visit, but they still write online anonymous reviews for these places and may also need similar venue recommendations when traveling to new places. In such cases, their check-in data are private and their identity needs to be protected, and thus our framework is beneficial to such users. When users visit some specific locations, the check-in data will be encoded by the data encoding module and stored in users' local devices privately. After collecting sufficient data, the private modeling module deployed on users' local devices acquires auxiliary public data from a multi-source public database and then merges them with users' private data to train the local model. During the model update, local model parameters among all users are aggregated through the secure aggregation module so that the models are in fact trained on all the data while no one is able to see the data or model parameters of others. This training process will repeat until the models converge to the optimum (i.e., minimizing the inference errors).

3.1 | Data encoding module

As shown in Figure 1, the data encoding module is used to convert certain data attributes to valid numerical representations. Such representations, especially the spatial representations for the location attribute, can serve

as the model input for downstream machine learning tasks and pave the way for many geospatial applications (Mai et al., 2020). For the location attribute, we use Uber's H3 hexagonal hierarchical spatial index (https://eng.uber.com/h3/) to aggregate and encode the latitude and longitude. As a grid indexing system, H3 divides the target area into contiguous hexagonal cells of the same size based on a specific resolution level. By transforming latitude–longitude pairs into a 64-bit index (i.e., a string), H3 can be easily used for indexing locations. One nice property of the H3 is that, compared with other grid systems such as a triangle or square grid system like Geohash, the distance between the center point of a hexagon and its neighbors is consistent, making the H3 suitable for preserving the spatial relationship.

As to the time attribute, we encode it into high-dimensional binary vectors based on their vocabulary sizes using one-hot encoding (OHE, a representation process using dummy variables) so that it can serve as the input for the model. For example, one week has seven days, and thus one day of the week can be encoded into a seven-dimensional binary vector (e.g., $(1, 0, 0, 0, 0, 0, 0)$ represents Monday). Similarly, we can also encode the hour attribute into 24-dimensional binary vectors. We can also use OHE to encode categorical attributes, such as gender and POI category, based on their vocabulary sizes. In addition, for attributes with continuous values (e.g., friend count, bus stop density in transportation infrastructure), we first reclassify them into five different classes (levels) based on the data range and distribution and then encode them using OHE. The encoded data structure is shown in Figure 2. Details of the auxiliary public data are introduced in Section 3.4.

In general, OHE introduces data sparsity issues when storing data, which could be quite heavy for a mobile device. To handle such issues, in practice we only store the index of "1" in an OHE vector as a dense representation of that vector; it can still be used for further computation, but saves storage space.

3.2 | Private modeling module

The private modeling module learns users' location preferences from their profiles and check-in information and recommends the POI locations that users will most likely visit. We regard location recommendation as a classic task in a recommendation system: click-through rate (CTR) prediction—estimating the probability that a user will click on a recommended item (Zhou et al., 2019) (In our context, it is the probability that a user will visit a recommended location.) In this research we use a CTR prediction model named Deep Factorization Machine (DeepFM; Guo, Tang, Ye, Li, & He, 2017) to make location recommendations. A simplified structure diagram of the DeepFM model is shown in Figure 3.

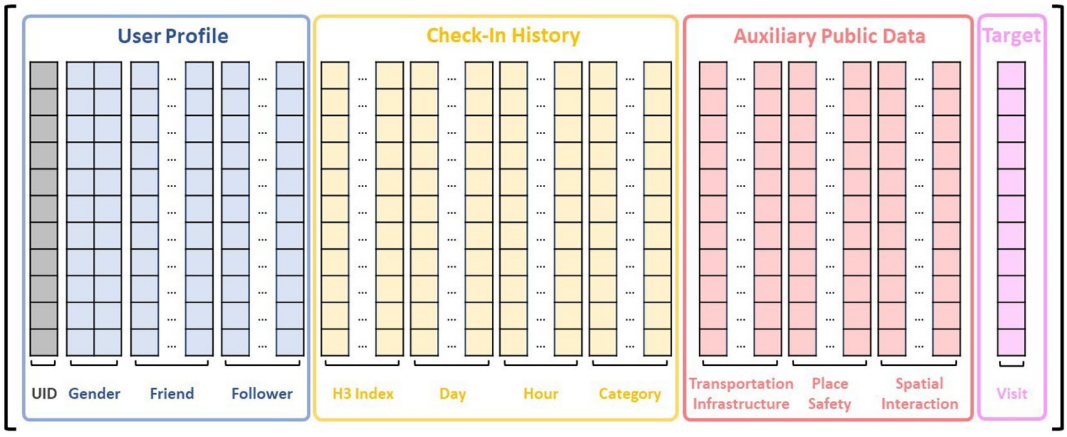

FIGURE 2 The encoded data structure

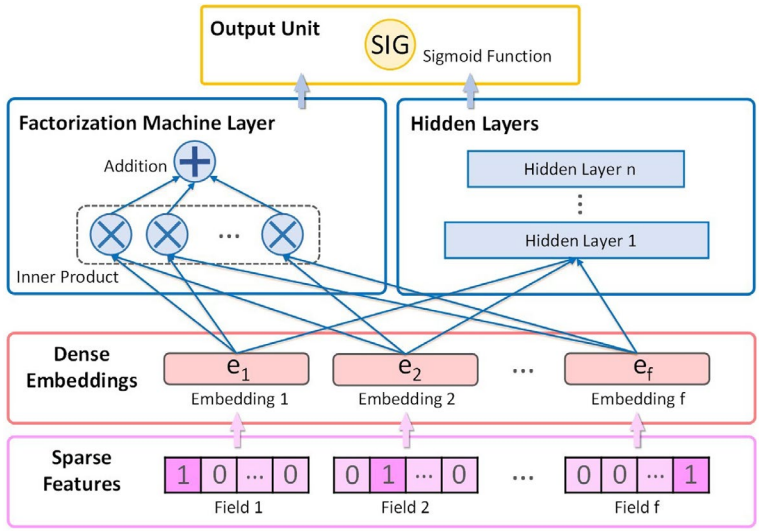

FIGURE 3 The structure of the DeepFM model (simplified from Zhou et al., 2019)

The DeepFM model consists of two components: the factorization machine (FM) component and the deep component, which share the same input. The FM component in the model is the factorization machine proposed by Rendle (2010). The model equation of a factorization machine is given by:

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j \tag{1}$$

where x_i is the ith feature variable, w_0 is the bias item, w_i is the weight of the ith feature variable (order-1 feature interactions), and $< v_i, v_j >$ is the weight of the interaction between the ith and jth feature variables (order-2 feature interactions), where v_i and v_j are the latent feature vectors of the ith and jth feature variables, respectively. As is shown in Equation (1), the factorization machine captures higher-order (i.e., order-2, pairwise) feature interactions by considering the inner product of the feature latent vectors. Compared with a linear regression model, the factorization machine can better model the feature interactions and also better handle the data sparsity issue.

The deep component is a feedforward deep neural network. The high-dimensional sparse input vectors are first embedded as low-dimensional dense real-value vectors by an embedding layer, and then fed into the hidden layer for learning higher-order feature interactions. The DeepFM model combines the FM and deep components together as an end-to-end deep learning structure, which improves the feature learning ability and makes it able to learn both low-order and high-order feature interactions. The private modeling module is deployed on each user's local device and run locally, and thus private user data are not uploaded or exposed.

3.3 | Secure aggregation module

The secure aggregation module, deployed on a cloud computing server, guarantees a privacy-preserving collaborative machine learning process. The module collects model parameters from each local model, calculates the weighted average, and then iteratively returns the aggregated parameters to each local model as new parameters. Here we further explain how the module works. Assume that there are K users in total, and they produce and keep their private data on their own devices, respectively. If the size of the kth user's private data is denoted by N_s^k, then the size of all the data among all users is $N_s = \sum_{k=1}^{K} N_s^k$. We assign the kth user a weight β_k based on the

size of their private data, which can be calculated as $\beta_k = N_s^k / N_s$, $\beta_k \in [0, 1]$. The more data the user has, the larger weight the user is assigned. Before the kth user sends the model parameters $w^k = \{\theta_0, \theta_1, \theta_2, \ldots, \theta_{T_k}\}$ (where T_k is the size of the kth user's model parameters) to the server for aggregation, the assigned weight should be multiplied by each value in the parameters. Thus, the weighted model parameters are $\hat{w}^k = \beta_k w^k = \{\beta_k\theta_0, \beta_k\theta_1, \beta_k\theta_2, \ldots, \beta_k\theta_{T_k}\}$. On the server side, all the weighted parameters collected from the users are aggregated (i.e., averaged), and the aggregated parameters $\overline{w} = \sum_{k=1}^{K} \hat{w}^k$ are returned to each local device as new model parameters. Such a process will be repeated until the global model (i.e., the model with the aggregated parameters) converges.

The above process can be summarized as a weighted averaging formula:

$$\overline{w} = \sum_{k=1}^{K} \frac{N_s^k}{N_s} w^k \tag{2}$$

where K is the number of users, w^k is the model parameters of the kth user, and N_s and N_s^k represent the sample size of all the users and of the kth user, respectively.

In practice, we use SMPC, a technology that allows multiple parties to jointly perform computation over their input data while keeping the data private, to protect the parameter transfer between local devices and the server. Specifically, we use the additive secret sharing (ASS) scheme to break the parameter data (i.e., secret) into many shares, so that others cannot infer the data without knowing all its parts (Xiong, Zhou, Xia, Gu, & Weng, 2020). ASS is an additively homomorphic encryption approach, which ensures that we can derive the parameters' average value using all the shares without knowing what value each parameter is, thereby avoiding parameter leakage and better preserving user privacy.

3.4 | Multi-source public database

The multi-source public database, also deployed on the cloud server, hosts auxiliary public data such as transportation infrastructure, place safety, and flow-based spatial interaction. Recent studies reveal that these data are of great importance to users' travel preferences and destination choices (Xu, Li, Belyi, & Park, 2021; Yan & Zhou, 2019; Yao, Peng, Xiao, & Guan, 2017; Zhu & Diao, 2020). The transportation infrastructure includes the distribution of subway stations, bus stops, and parking lots. The place safety data we use are the complaint records released by the local police department. The flow-based spatial interaction reflects people's travel preference between places and can be extracted from multiple sources such as transportation surveys and aggregated taxi trip records (Liu, Gong, Gong, & Liu, 2015).

For the transportation infrastructure data and the place safety data, we first use the H3 spatial index to aggregate them into corresponding hexagonal cells. Then the size of each type of data aggregated in each cell is a density value (e.g., the number of parking lots in each cell). For each type of data, we reclassify the density values into five levels based on their data range and distribution. For example, the density values of subway stations across the cells vary from 0 to 31 (mostly less than 20), and we use an arithmetic sequence $\{5, 10, 15, 20\}$ as the breaking points to reclassify the values into five levels. After the reclassification, the data can be further encoded using OHE.

For the flow-based spatial interaction data, we calculate the transition probability from one hexagonal cell to another hexagonal cell based the flow data. We take taxi trip records as an example. We first aggregate the pick-up locations and drop-off locations into hexagonal cells using the H3 index, and then we derive the origin–destination (OD) flow matrix among the hexagonal cells. Thus, the transition probability P_{ij} from hexagonal cell i to hexagonal cell j can be represented as the total amount of flows from hexagonal cell i to hexagonal cell j divided by the total amount of outflows of hexagonal cell i. The formula is:

$$P_{ij} = \frac{f_{ij}}{\sum_{k=1}^{K} f_{ik}} \tag{3}$$

where P_{ij} is the transition probability from hexagonal cell i to hexagonal cell j, K is the total number of hexagonal cells, and f_{ij} stands for the OD flow from hexagonal cell i to hexagonal cell j. Finally, we get a transition probability matrix that reveals the flow-based spatial interaction among all the hexagonal cells. Since OD flow data at different times of the day usually have different patterns (Ahas, Aasa, Silm, & Tiru, 2010; Xu et al., 2016), in practice we divide the OD flow data into two time periods: stay-at-home time (before 7 a.m. and after 8 p.m.) and stay-outside time (7 a.m. to 8 p.m.). We then deal with these two parts of flow data and produce two transition probability matrices, one for stay-at-home time and the other for stay-outside time. In different time periods, the spatial interaction data that users inquire will be different.

4 | EXPERIMENTS AND RESULTS

4.1 | Data set

We choose New York City (NYC) as the experiment area and extract the check-in data in NYC from the Foursquare global-scale social media check-in data set collected from April 2012 to September 2013 (Yang, Zhang, & Qu, 2016; Yang, Zhang, Chen, & Qu, 2015). We perform data cleaning to ensure that each user has at least ten check-in records, and each location has at least 10 visits. We then link the data with the user profile data set (Yang, Qu, & Cudré-Mauroux, 2018; Yang, Zhang, Qu, & Cudré-Mauroux, 2016) and select the data of the top 500 active users as our experiment data set. A summary of the processed social media check-in data set is shown in Table 1. There are 118,316 check-in records in the NYC data set, containing 3,628 unique POIs, 500 users, and 294 POI categories. Figure 4 shows the geographic distribution of the check-in data.

In the multi-source public database, we store the distribution of subway stations, bus stops, and parking lots in NYC as transportation infrastructure data (https://data.cityofnewyork.us/), and the complaint report in 2011 and 2012 released by the New York Police Department as the place safety data (https://catalog.data.gov/dataset/nypd-complaint-data-historic). For the flow-based spatial interaction, we acquire the Taxi and Limousine Commission trip record data in September 2013 from the NYC government website (https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page).

Our next step is to determine the spatial resolution (i.e., the Uber H3 spatial indexing level) for encoding. Figure 5 shows the comparison between different spatial resolution settings when encoding the NYC check-in data using Uber H3. As seen in the figure, when level = 8 (or greater) the encoding results are too sparse to cover

TABLE 1 The attributes in the NYC social media check-in data set

Attribute	Type	Description
VenueId	String	Unique ID for the POI
UserId	Integer	Unique ID for the user
Gender	String	Gender of the user
Friend	Integer	Twitter friend count of the user
Follower	Integer	Twitter follower count of the user
Latitude	Float	The latitude of the POI
Longitude	Float	The longitude of the POI
Timestamp	String	The time when the user checks in at this POI
Category	String	The category of the POI

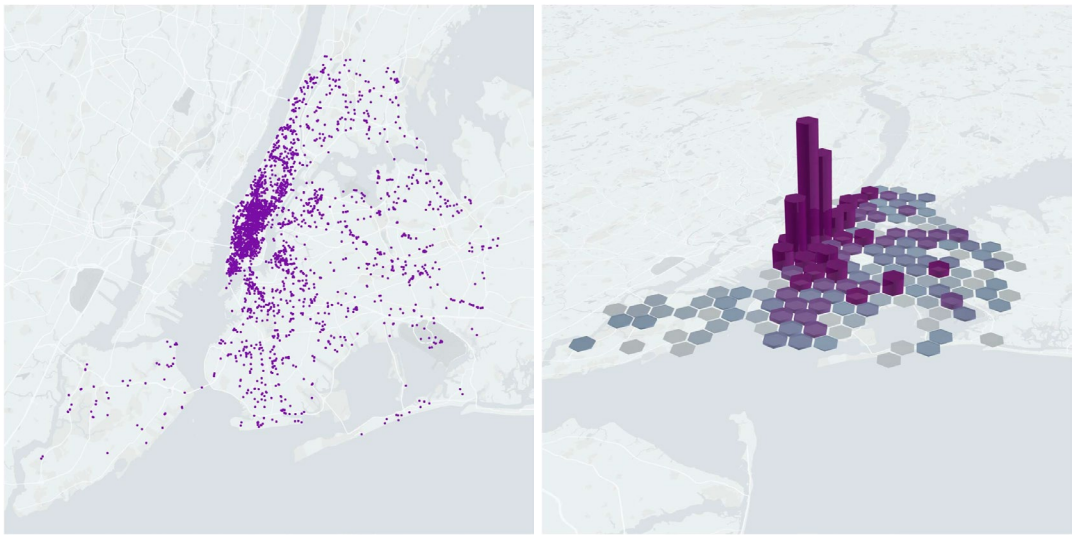

FIGURE 4 (a) Geographic distribution; and (b) check-in frequency statistics

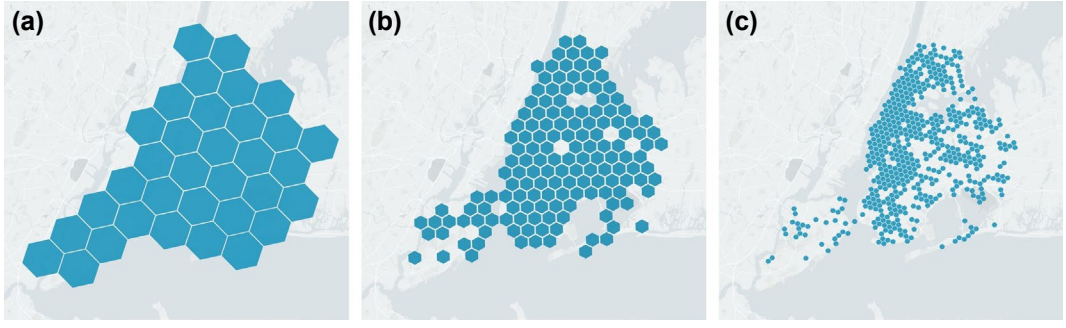

FIGURE 5 Comparison between the encoding results of NYC check-in data using Uber H3 with different indexing levels: (a) level = 6; (b) level = 7; and (c) level = 8

the entire NYC area, since there are lots of "holes" that contain no check-in data. For level = 6 (or less), the hexagonal cell size is too large and many locations that are far away from each other are contained in the same cell, losing some location details. Compared with level = 6 and level = 8, in our case, the encoding results with level = 7 cover basically the entire NYC area (i.e., dense representations) while preserving a relatively small cell size (i.e., preserving location details). Thus, we find that H3 level 7 (5.161 km^2 per hexagonal cell) better balances the trade-off between the spatial coverage and location details than other H3 indexing levels. We then encode the data via the data encoding module based on that indexing level. In particular, for the flow-based spatial interaction data, we derive the OD flow matrix among all the hexagon cells, and then calculate the transition probability from one arbitrary cell to another. Since our check-in data set does not contain users' previous locations before checking in, we extract the two most frequently visited locations in different time periods (i.e., stay-at-home time and stay-outside time) for each user. We are then able to retrieve the transition probability from these locations to the recommended locations. Figure 6 shows the geographic distribution and hexagonal cell density of the transportation

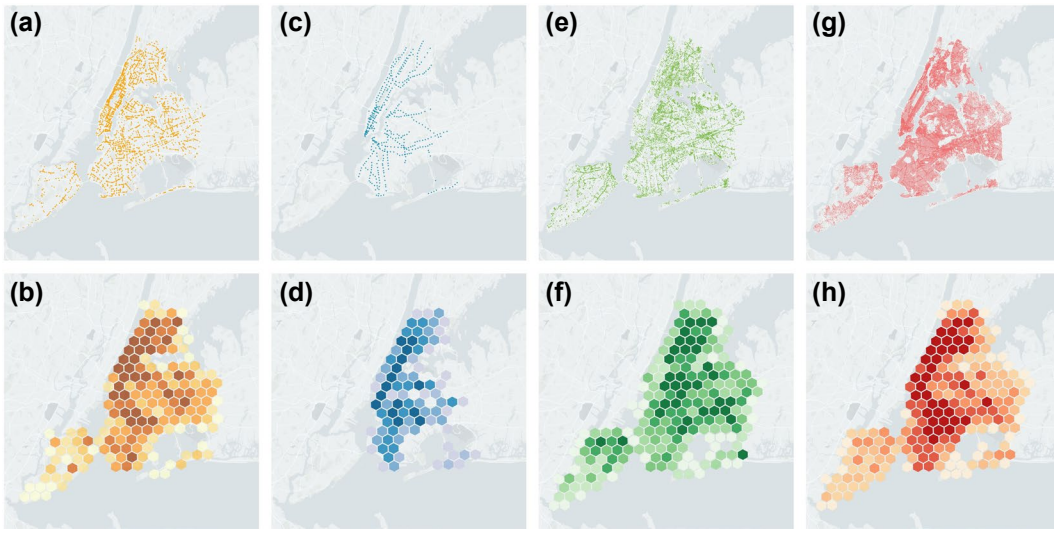

FIGURE 6 Geographic distribution and hexagonal cell density (deeper colors represent higher density) of transportation infrastructure data and place safety data: (a) and (b) bus stops; (c) and (d) subway stations; (e) and (f) parking lots; (g) and (h) complaint records

infrastructure data and place safety data. The visualization of NYC taxi trip OD flow can be seen in Figure 7. From these figures, we can see obvious spatial heterogeneity of flow distributions.

Since each check-in record represents a positive sample (e.g., a user checks in at the recommended location), we randomly generate negative samples (e.g., a user does not check in at the recommended location) so that the ratio between positive and negative samples is 1: 1, which establishes a more balanced data set. Specifically, for each user, we create a POI set that contains all the POIs (including the category and location) that the user never visits historically. Then we randomly sample a group of POIs with a sample size equal to the size of the user's check-in records. These sampled POIs are labeled as negative samples and their other attributes such as check-in day and hour information will be randomly generated. After preprocessing, the data set contains 236,632 records in total. We randomly split the data set into a training set, validation set, and test set in the ratio 8: 1: 1.

4.2 | Training and evaluation

We set up a federated learning environment for the location recommendation task using PySyft, a Python library for secure and private deep learning. PySyft implements the SPDZ secure protocol (Damgård, Pastro, Smart, & Zakarias, 2012) multiparty, which achieves secure multiple party computation. For each user, we initialize one worker for storing private data and deploying the private modeling module (with DeepFM model) for local training. Another worker is created to serve as the secure aggregation module. Each model's parameters are sent to the secure aggregation module using the SPDZ protocol after each training epoch. The parameters are aggregated and then returned to each local model. For training the hyperparameters, we set the training epoch number as 200 and batch size as 4,096. For the DeepFM model, we set the embedding dimension as 16, the multi-layer perception dimension as (16, 16), the dropout rate as 0.2, and the learning rate as 0.001, the weight decay as 10^{-6}, and the optimizer as Adam (Kingma & Ba, 2014).

We use the area under the receiver operating characteristic (AUC) as the accuracy metric. The receiver operating characteristic (ROC) curve demonstrates the classification ability of a binary classifier given different

FIGURE 7 Visualization of NYC taxi trip OD flow between hexagonal cells, September 2013

discrimination thresholds. The *x*-axis of an ROC curve is the false positive rate (FPR), and the *y*-axis is the true positive rate (TPR, also known as the sensitivity). The FPR and TPR can be defined as follows:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{4}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

where TP, FP, TN, and FN are respectively the numbers of true positive, false positive, true negative, and false negative test results of recommended POI locations. For each discrimination threshold, we can calculate a pair of FPR and TPR, and accordingly draw a ROC curve given a series of thresholds. The area under the ROC curve is the AUC, revealing the probability that a classifier ranks a randomly chosen positive sample higher than a randomly chosen negative sample. The AUC is a common metric for evaluating the CTR prediction. The higher the AUC value, the better the model can distinguish between two classes.

Since CTR prediction is a binary classification problem, we use binary cross-entropy (BCE) as the loss metric during the neural network model training. The BCE is defined as:

$$\mathrm{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log_2(\hat{y}_i) + (1 - y_i)\log_2(1 - \hat{y}_i)] \tag{6}$$

where N is the total number of samples, y_i is the ground-truth label (0 or 1) for the ith sample, and $\hat{y}_i$ is the predicted result of the ith sample, showing its probability of being 1.

4.3 | Performance comparison

In our experiment, we train two models on the data sets and compare their recommendation accuracy. One model is based on our privacy-preserving framework (referred to as FL-DeepFM), where we deploy the DeepFM model on users' local devices (i.e., decentralization) and use federated learning to collaboratively train the model. The other model is an original DeepFM model, and we train this original model on a centralized data set (e.g., all users' private data are stored in one data set) as a baseline. By comparing the recommendation accuracy of the two models, we can see how much impact the switch from centralized learning to decentralized learning has on accuracy. we also investigate how the different features (apart from the basic features, i.e., user profiles and POI category) in the data set affect the recommendation accuracy. Specifically, we consider the influence of the spatial and temporal features (denoted by ST, i.e., location and time-stamp) and the auxiliary public data (TS represents the transportation infrastructure and place safety features; SI represents the flow-based spatial interaction features). By combining different features, we can see how much impact a specific set of features has on accuracy.

Figure 8 shows the AUC values on the validation data set and test data set over the training epochs. We find that the validation and test AUC values all increase as training proceeds, and all the models converge after around 100 epochs, indicating a successful training process. The original DeepFM model trained on the basic training data set (including only user profiles and POI category) reaches a test AUC of 0.761. After involving the ST features (e.g., DeepFM+ST), the test AUC is increased to 0.952. Further involving the TS and SI features increases the test AUC to 0.954 and 0.963, respectively. Likewise, the FL-DeepFM model trained on the basic training data set reaches a test AUC of 0.651. Further integrating the ST, TS, and SI features brings the test AUC to 0.834, 0.840, and 0.879, respectively. Overall, we can see the original centralized DeepFM model outperforms the privacy-preserving FL-DeepFM model on recommendation accuracy. In fact, such an accuracy decrease by our framework is expected since the original DeepFM model is completely trained on the centralized data set while the FL-DeepFM model is trained locally on each device. Although local model parameters can be aggregated via the secure aggregation

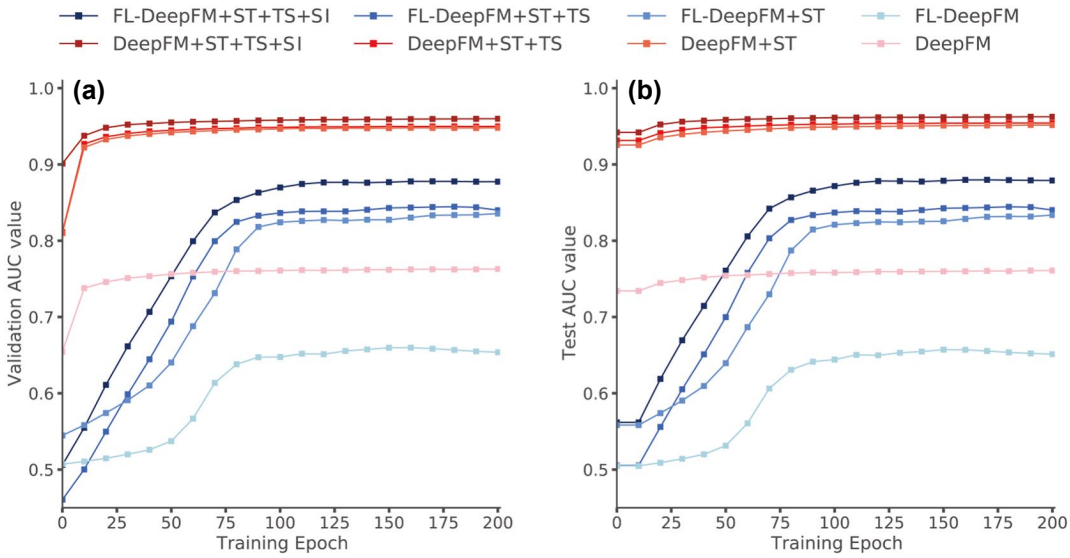

FIGURE 8 (a) Validation; and (b) test AUC values by epoch

module, data insufficiency and imbalance issues may still exist on each local side. Nevertheless, the FL-DeepFM model with all features involved can still achieve a competitive test AUC value of 0.879.

4.4 | Attack cases

Having carried out a performance comparison, we design two attack cases against the proposed privacy-preserving framework to further investigate the aforementioned research questions of privacy protection effectiveness and model utility. One case is a type of inference attack (Krumm, 2007), which is designed to examine the privacy protection effectiveness of our framework, especially for location privacy. The other is a type of distributed poisoning attack (Cao, Chang, Lin, Liu, & Sun, 2019), designed to examine the robustness of the federated learning model, that is, how stable the recommendation accuracy is when under poisoning attack.

4.4.1 | Home location clustering attack

The potential exposure of sensitive locations (e.g., home location) and personal identities through a unique connected string of locations raises public concerns about location privacy (De Montjoye et al., 2013). Our framework is based on a decentralized data set, and the check-in data are locally distributed in users' devices. In principle, no one is able to cluster the home location of specific users without access to their devices, which guarantees location privacy. Nevertheless, in our framework, users need to retrieve the corresponding auxiliary public data (with location information) from the multi-source public database according to their historical check-in locations. One potential threat is that when an attacker hijacks the network pipeline and monitors the requested auxiliary public data, the attacker will be able to know the rough activity zone of users and be able to try to identify the home locations from such information. Here we investigate whether and to what degree our framework can prevent users' home location from being identified, and how our framework outperforms the traditional centralized methods (e.g., the original DeepFM) in protecting location privacy.

For comparison, we first evaluate the location privacy protection effectiveness on the centralized data set as a baseline. For the traditional centralized methods, all the users' data are stored in a centralized server, making home location clustering easy to carry out. We involve three kinds of privacy protection techniques that are commonly used on centralized data sets, namely de-identification, random perturbation, and Gaussian geomasking. De-identification directly removes all the labels from the data set; Random perturbation and Gaussian geomasking further blur the latitude and longitude of each location using random values following uniform and Gaussian distributions, respectively. The parameter for random perturbation is the maximum perturbation threshold, which is set to 1,000 (1 km) in our experiment. The two parameters for Gaussian geomasking are the mean and standard deviation of the displacement from the original location, which are set to 0 and 1 km in our experiment, respectively. We choose 1 km as the perturbation threshold and the Gaussian geomasking standard deviation since a previous work investigated the relationship between the displacement distance and the underlying population density and found that the majority of masked points were displaced within 1 km (Hampton et al., 2010). A threshold of 1 km is also used in our recent works as a baseline (Gao et al., 2019; Rao et al., 2020). We then evaluate the location privacy protection effectiveness of our framework on a decentralized data set. Our framework requests auxiliary public data that contain location information, so we conduct the attack on such location information. The granularity of the request (also the returned data), however, is at the whole historical check-in data set level, and such data do not contain the check-in time and frequency for each location, which greatly reduces the chance to identify users' home/work location using spatial clustering algorithms.

We employ density-based spatial clustering with noise (DBSCAN; Ester, Kriegel, Sander, & Xu, 1996) to identify users' home location from the check-in locations. As a spatial clustering algorithm, DBSCAN has been used

frequently in human activity location identification (Huang & Wong, 2015; Luo, Cao, Mulligan, & Li, 2016). The accuracy metrics we use are overall accuracy, precision, recall, and F_1 score. *Overall accuracy* is the ratio between the number of correctly identified home and not-home locations and the total number of locations. *Precision* is the ratio between the number of correctly identified home locations and the number of the locations that are identified as home locations. *Recall* is the ratio between the number of correctly identified home locations and the number of the true home locations. F_1 *score*, which provides a balance between precision and recall, shows how effective a method is at detecting a user's home location. We also use the centroid shift and medoid shift to measure the shift distance from the representative centers of the home location clusters to ground-truth home locations.

We extract the nighttime (8 p.m. to 7 a.m.) check-in records and run the DBSCAN algorithm on them to detect home locations. For the location data from the requested auxiliary public data, we run DBSCAN directly on all of them since they do not contain time attributes. For the parameters of DBSCAN, we set the minimum number of points (MinPts) to 4 and set the search radius in the range 50–1,000 m in steps of 50 m. The boxplots of F_1 score and overall accuracy using different privacy protection techniques are shown in Figure 9. We can see that these metrics change as the service radius (Eps) changes, and overall our framework produces the lowest accuracy metrics, indicating the best privacy protection effectiveness. We report and analyze the mean or median of the accuracy metrics in Table 2 (Eps = 300 m) as is suggested by Huang and Wong (2015), Luo et al. (2016).

As shown in Table 2, the privacy protection effectiveness of different techniques varies. When there is no privacy protection, one can easily identify users' home locations using labels. When using de-identification, home locations can be protected to some extent while the mean overall accuracy and the mean F_1 score can still reach 0.493 and 0.832, respectively. The median distance shifts to the centroid and medoid are also relatively small (106.014 and 0.000 m). Random perturbation and Gaussian geomasking effectively lower the accuracy and increase the distance shifts, while the mean overall accuracy and the mean F_1 score can still reach above 0.404 and above 0.642. In contrast, our framework significantly reduces the accuracy (the mean overall accuracy and the mean F_1 score are only 0.041 and 0.316, respectively), and the distance shifts are increased to around 7 km. Such performance ensures that it is almost impossible to identify users' home locations under our framework, thereby demonstrating the powerful location privacy protection effectiveness.

4.4.2 | Model failure poisoning attack

The goal of the poisoning attack is to make the model produce incorrect predictions. A model failure poisoning attack can damage the performance (e.g., accuracy) of a distributed system (e.g., a federated-learning-based framework). Thus, such attacks can be used to examine the robustness of our framework. Note that in our case, the poisoning attack does not compromise location privacy. Instead, it tries to compromise recommendation accuracy (e.g., model utility). Specifically, in such an attack case, the attacker manages to take control of some devices that participate in the federated learning process and orders the devices to send "incorrect" model parameters (e.g., a set of random vectors) to the secure aggregation module, which affects the normal model training and update process, resulting in reduced model accuracy and even making the recommendation service unavailable in practice.

Since we use the weighted averaging method to aggregate the local parameters from users, a user with a larger data size is assigned a larger weight and has more influence on the model update. We simulate five scenarios where the attacker controls a group of devices each time, and each group of devices has a different total weight (from 0.1 to 3.3%; see Figure 10a). During each iteration, the attacker sends randomly generated model parameters from these devices to the server for secure aggregation (e.g., poisoning). The test AUC curves of the normal model (not poisoned) and the poisoned models are shown in Figure 10b. Compared with the normal model, the test AUC of the model poisoned by group 1 drops by 0.029 (from 0.879 to 0.850), but the model still converges. As the weight increases to 1.2%, the test AUC decreases successively to 0.740 while the model still converges. The model training starts to fail when the weight increases to 1.9% and higher. We can clearly see the total weight of the controlled

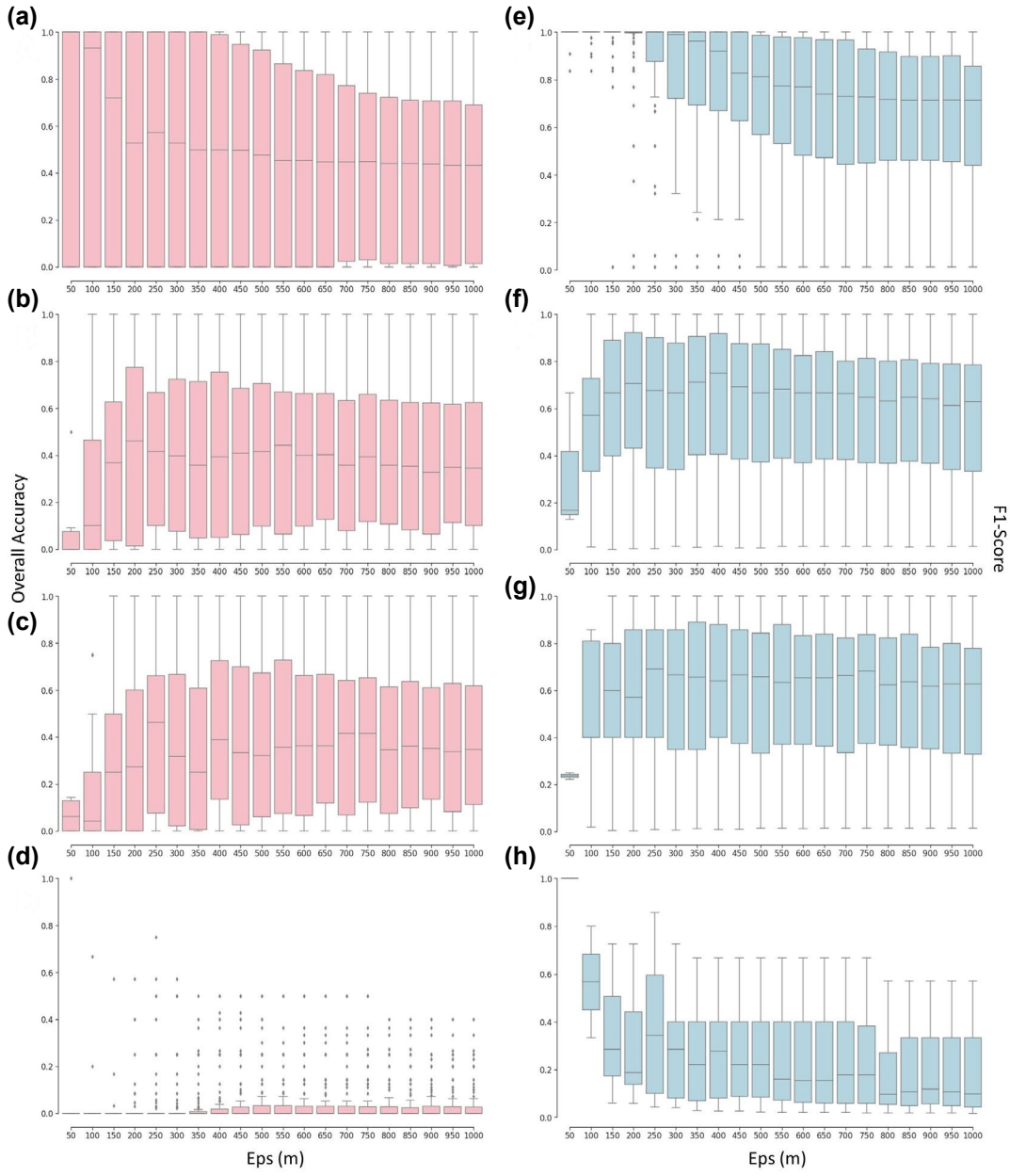

FIGURE 9 Boxplots of accuracy metrics using different privacy protection techniques: (a)–(d) are the overall accuracy for de-identification, random perturbation, Gaussian geomasking, and our framework, respectively; and (e)–(h) are the F_1 score for de-identification, random perturbation, Gaussian geomasking, and our framework, respectively

devices can greatly affect the poisoning efficiency. To conclude, the success of the poisoning attack depends heavily on whether the attacker can control a number of devices with sufficient weight to affect the model update.

This attack case shows that our framework has some resilience to poisoning attacks with relatively lower weights. Our framework is proposed for location-based service providers, which usually have a large number of users (e.g.,

TABLE 2 Home location clustering accuracy metrics using different privacy protection methods

Measures	None	DI	RP	GG	Ours
Mean overall accuracy	1.000	0.493	0.436	0.404	**0.041**
Mean precision	1.000	0.493	0.436	0.404	**0.041**
Mean recall	1.000	1.000	1.000	1.000	1.000
Mean F_1	1.000	0.832	0.657	0.642	**0.316**
Median centroid shift (m)	0.000	106.014	599.710	692.777	**7,065.464**
Median medoid shift (m)	0.000	0.000	623.311	726.030	**7,074.129**

The bold values indicate the best location privacy protection effectiveness.

Abbreviations: DI, de-identification; GG, Gaussian geomasking with 0 km mean and 1 km standard deviation; none, no privacy protection; ours, our privacy-preserving framework on a decentralized data set; RP, random perturbation with 1 km threshold.

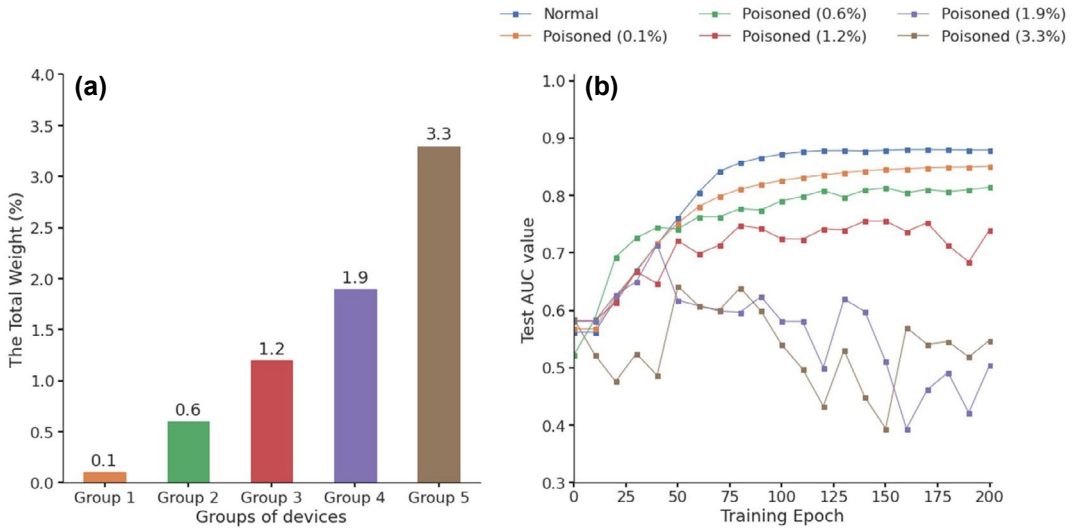

FIGURE 10 (a) The total weight of different groups of devices; and (b) The test AUC values by epoch in the poisoning attack experiment

Twitter, Yelp, and Foursquare currently have around 353, 178, and 50 million monthly active users, respectively; Brightkite, before shutting down, had 2 million monthly active users), making controlling sufficient devices with a total weight of greater than 1% on such platforms hard to achieve in practice. In fact, as long as the number of users is larger than 100, the attacker usually has to control multiple users' devices to ensure that they gain enough weight to affect the model update. In the event that the attacker manages to control enough devices, monitoring the validation accuracy can help detect the attack and take defensive measures (e.g., roll back to the previous model and investigate the attack).

5 | DISCUSSION

In this section we discuss the factors that may affect recommendation accuracy and privacy protection effectiveness. We also discuss the trade-off between the privacy protection effectiveness and the model utility. Finally, some limitations of this research are listed.

5.1 | Factors affecting model performance

We first investigate how some factors affect recommendation accuracy. These factors include spatio-temporal attributes, auxiliary public data (transportation accessibility, place safety, and flow-based spatial interaction), random initialization, the poisoning attack, and spatial resolution. As shown in Table 3, richer features bring higher recommendation accuracy. When the model trains on the basic training data that only include user profiles and POI category, the test AUC is only 0.651. Adding more features eventually increases the accuracy up to 0.879. The model parameters are randomly initialized based on a random seed. Given different random seeds, the final recommendation accuracy remains almost unchanged. The poisoning attack is another important factor greatly affecting recommendation accuracy. When the total weight assigned to the devices controlled by the attacker reaches 0.1%, the recommendation accuracy drops by around 0.029. As the total weight continues to increase, the recommendation accuracy will continue to decrease until the model fails to converge.

We also investigate what impact an increase in spatial resolution has on model performance. Generally, an increase in spatial resolution will result in changes in the encoding results of check-in data and auxiliary public data, thereby influencing the location recommendation accuracy. As shown in Figure 5 and Table 3, due to the data sparsity issues caused by a higher spatial resolution (level = 8), the test AUC value dropped from 0.879 to 0.808. We therefore conclude that for different areas and tasks, the suitable spatial resolution may vary, but in our cases the spatial resolution with a level of 7 is a good choice.

We also discuss how some factors affect privacy protection effectiveness. These factors include data storage, SMPC, and auxiliary public data. When the check-in data are uploaded and stored in a centralized data set, user profiles are at risk of leakage, and sensitive locations such as home locations can be identified via location clustering algorithms even if the data were preprocessed with de-identification or geomasking. Decentralized data storage can minimize privacy leakage risk since users' private data will not be shared with anyone else. SMPC is the cornerstone of the secure aggregation module, which guarantees that the collection and aggregation of the model parameters go smoothly and privately. This is also why our framework can perform model training and update under the premise of preserving privacy. Since users' check-in data are stored locally, requesting the auxiliary public data with location information might be the only way to leak users' location privacy. However, such data do not contain the check-in time and frequency for each location, which greatly reduces the chances of identifying users' home location using location clustering algorithms. As mentioned in the home location clustering attack case, running DBSCAN on the requested auxiliary public data gives a mean overall accuracy of only 0.04.

TABLE 3 Test AUC values measured on the model trained via federated learning with different settings

Setting	Influence	Test AUC
Basic	Data	0.651
Basic + ST	Data	0.834
Basic + ST + TS	Data	0.840
Basic + ST + TS + SI	Data	0.879
Different random seed	Initialization	0.878
Higher resolution (level = 8)	Encoding	0.808
Poisoning (0.1%)	Update	0.850
Poisoning (0.6%)	Update	0.814
Poisoning (1.2%)	Update	0.740
Poisoning (3.3%)	Update	Failed to converge

Note: Basic represents the basic training data set involving only users' profiles and POI category; ST represents the spatial and temporal features (i.e., check-in locations and time-stamps); TS represents the transportation infrastructure and place safety features; and SI represents the flow-based spatial interaction features.

5.2 | Privacy–utility trade-off

This research clearly shows a trade-off between the privacy protection effectiveness and model utility. As is shown in Figure 8, our privacy-preserving model trained on a decentralized data set via federated learning has overall lower recommendation accuracy compared with the original model trained on a centralized data set, which reflects the fact that we sacrifice a small amount of model utility in order to preserve privacy. Further evidence of this trade-off is that our framework requires network transfer and cryptography techniques for collecting and aggregating the model parameters, and these would directly limit the training speed and affect the total time for training. Likewise, the price paid for the high recommendation accuracy of the original model is the high privacy leakage risk. According to the home location clustering accuracy metrics shown in Table 2, lots of home locations in the centralized data set can be correctly identified despite being processed by de-identification, random perturbation, or Gaussian geomasking. Compared with centralized data, our framework based on decentralized data storage significantly minimizes the risk of location privacy leakage.

Overall, the relationship between privacy and utility is somewhat contradictory: if we want to improve one, we might have to sacrifice the other, which results in a "catch-22 situation" (Rao et al., 2020). With this in mind, this research also explores how to improve recommendation accuracy as much as possible while preserving privacy. Successful attempts include involving spatio-temporal features and auxiliary public data (transportation accessibility, place safety, flow-based spatial interaction). In general, our framework protects user profiles and location privacy by keeping users' data on their own devices while providing a competitive recommendation accuracy via decentralized collaborative learning, which strikes a better balance between privacy protection effectiveness and model utility.

5.3 | Limitations

Our framework has several limitations. First, our federated-learning-based framework collects and aggregates model parameters privately, which relies heavily on network communication and cryptography techniques. In practice, it will take a much longer time to complete model training compared with the traditional process on a centralized data set. Second, such a framework requires computing resources and storage while training the model, which may lead to higher consumption of power and storage on the local devices. Third, although our framework deals better with the trade-off between the privacy protection effectiveness and model utility, the recommendation accuracy is still sacrificed slightly and should be further improved. Finally, we do not consider users' current location and time when making location recommendations, due to the lack of such data. In fact, users' spatio-temporal context would have a great impact on the next locations they likely visit (Huang, 2017). We plan to explore and solve these limitations in our future work.

6 | CONCLUSIONS

This research presents a new privacy-preserving framework for location recommendation based on decentralized collaborative machine learning. Key innovations of this method include: keeping users' private data on their own devices; using federated learning to achieve decentralized collaborative learning for local models; encoding and utilizing the spatio-temporal features and auxiliary public data (transportation infrastructure, place safety, and flow-based spatial interaction) to further improve the recommendation accuracy; and designing and performing two attack cases to examine the location privacy protection effectiveness and model robustness. To address our research question RQ1, users' private data will not be shared or uploaded elsewhere, and the parameter transfer is protected by secure multiple party computation, which ensures that users' data and models' parameters are

well protected. In addition, the home location clustering attack case indicates that our framework can efficiently protect users' location privacy. As to RQ2, the results show that, although having slightly lower recommendation accuracy than the original centralized model, our privacy-preserving model still achieves reasonably high accuracy in the POI recommendation task and demonstrates some resilience to distributed poisoning attacks; in other words, our model achieves a nice balance between privacy protection effectiveness and model utility. Our method offers new insights into the development of privacy-preserving artificial intelligence technologies for promoting geoprivacy in recommendation services and geospatial data security more broadly.

Our future work will focus on improving training efficiency and mitigating the communication/computation overhead of the framework, improving the location recommendation accuracy by better balancing the privacy–utility trade-off, designing a strategy to further improve the resilience to distributed poisoning attacks, and integrating users' spatio-temporal context into the location recommendation modeling process.

ACKNOWLEDGMENT

We acknowledge the funding support provided by the American Family Insurance Data Science Institute Funding Initiative at the University of Wisconsin-Madison. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funder. We also acknowledge Jake Kruse at the University of Wisconsin-Madison for his valuable suggestions and generous help during the revision of this manuscript.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in the Github Repository https://github.com/GeoDS/FL-LocRec.

ORCID

Jinmeng Rao https://orcid.org/0000-0003-2370-5129
Song Gao https://orcid.org/0000-0003-4359-6302
Qunying Huang https://orcid.org/0000-0003-3499-7294

REFERENCES

Ahas, R., Aasa, A., Silm, S., & Tiru, M. (2010). Daily rhythms of suburban commuters' movements in the Tallinn metropolitan area: Case study with mobile positioning data. *Transportation Research Part C: Emerging Technologies*, *18*, 45–54.

Armstrong, M. P., & Ruggles, A. J. (2005). Geographic information technologies and personal privacy. *Cartographica*, *40*, 63–73.

Bao, J., Zheng, Y., Wilkie, D., & Mokbel, M. (2015). Recommendations in location-based social networks: A survey. *GeoInformatica*, *19*, 525–565.

Bertino, E., Thuraisingham, B., Gertz, M., & Damiani, M. L. (2008). Security and privacy for geospatial data: Concepts and research directions. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, Irvine, CA (pp. 6–19). New York, NY: ACM.

Cao, D., Chang, S., Lin, Z., Liu, G., & Sun, D. (2019). Understanding distributed poisoning attack in federated learning. In *Proceedings of the 25th IEEE International Conference on Parallel and Distributed Systems*, Tianjin, China (pp. 233–239). Piscataway, NJ: IEEE.

Casino, F., Domingo-Ferrer, J., Patsakis, C., Puig, D., & Solanas, A. (2015). A *k*-anonymous approach to privacy preserving collaborative filtering. *Journal of Computer and System Sciences*, *81*, 1000–1011. https://doi.org/10.1016/j.jcss.2014.12.013

Chen, C., Wu, B., Fang, W., Zhou, J., Wang, L., Qi, Y., & Zheng, X. (2020). *Practical privacy preserving POI recommendation*. Preprint, arXiv:2003.02834.

Cho, E., Myers, S. A., & Leskovec, J. (2011). Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA (pp. 1082–1090). New York, NY: ACM.

Chow, C.-Y., & Mokbel, M. F. (2011). Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter*, *13*, 19–29. https://doi.org/10.1145/2031331.2031335

Damgård, I., Pastro, V., Smart, N., & Zakarias, S. (2012). Multiparty computation from somewhat homomorphic encryption. In R. Safavi-Naini & R. Canetti (Eds.), *Advances in cryptology: CRYPTO 2012* (Lecture Notes in Computer Science, Vol. *7417*, pp. 643–662). Berlin, Germany: Springer.

De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., & Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, *3*, 1376. https://doi.org/10.1038/srep01376

Duckham, M., & Kulik, L. (2005). A formal model of obfuscation and negotiation for location privacy. In H. W. Gellersen, R. Want, & A. Schmidt (Eds.), *Pervasive computing: Pervasive 2005* (Lecture Notes in Computer Science, Vol. *3468*, pp. 152–170). Berlin, Germany: Springer. https://doi.org/10.1007/11428572_10

Dwork, C. (2008). Differential privacy: A survey of results. International Conference on Theory and Applications of Models of Computation. In M. Agrawal, D. Du, Z. Duan, & A. Li (Eds.), *Theory and applications of models of computation: TAMC 2008* (Lecture Notes in Computer Science, Vol. *4978*, pp. 1–19). Berlin, Germany: Springer.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, OR (pp. 226–231). Menlo Park, CA: AAAI.

Gao, H., Tang, J., Hu, X., & Liu, H. (2015). Content-aware point of interest recommendation on location based social networks. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, Austin, TX (pp. 1721–1727). Menlo Park, CA: AAAI.

Gao, S., Janowicz, K., & Couclelis, H. (2017). Extracting urban functional regions from points of interest and human activities on location-based social networks. *Transactions in GIS*, *21*, 446–467. https://doi.org/10.1111/tgis.12289

Gao, S., & Mai, G. (2018). Mobile GIS and location-based services. In B. Huang (Ed.), *Comprehensive geographic information systems* (pp. 384–397). Oxford, UK: Elsevier.

Gao, S., Rao, J., Liu, X., Kang, Y., Huang, Q., & App, J. (2019). Exploring the effectiveness of geomasking techniques for protecting the geoprivacy of Twitter users. *Journal of Spatial Information Science*, *19*, 105–129. https://doi.org/10.5311/JOSIS.2019.19.510

Gruteser, M., & Grunwald, D. (2003). Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the First International Conference on Mobile Systems, Applications and Services*, San Francisco, CA (pp. 31–42). New York, NY: ACM.

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). *DeepFM: A factorization-machine based neural network for CTR prediction*. Preprint, arXiv:1703.04247.

Hampton, K. H., Fitch, M. K., Allshouse, W. B., Doherty, I. A., Gesink, D. C., Leone, P. A., … Miller, W. C. (2010). Mapping health data: Improved privacy protection with donut method geomasking. *American Journal of Epidemiology*, *172*, 1062–1069. https://doi.org/10.1093/aje/kwq248

Huang, H., Gartner, G., Krisp, J. M., Raubal, M., & Van de Weghe, N. (2018). Location based services: Ongoing evolution and research agenda. *Journal of Location Based Services*, *12*, 63–93. https://doi.org/10.1080/17489725.2018.1508763

Huang, Q. (2017). Mining online footprints to predict user's next location. *International Journal of Geographical Information Science*, *31*, 523–541. https://doi.org/10.1080/13658816.2016.1209506

Huang, Q., Cao, G., & Wang, C. (2014). From where do tweets originate? A GIS approach for user location inference. In *Proceedings of the Seventh ACM SIGSPATIAL International Workshop on Location-based Social Networks*, Dallas/Fort Worth, TX (pp. 1–8). New York, NY: ACM.

Huang, Q., & Wong, D. W. (2015). Modeling and visualizing regular human mobility patterns with uncertainty: An example using Twitter data. *Annals of the Association of American Geographers*, *105*, 1179–1197. https://doi.org/10.1080/00045608.2015.1081120

Jain, P., Gyanchandani, M., & Khare, N. (2016). Big data privacy: A technological perspective and review. *Journal of Big Data*, *3*, 25. https://doi.org/10.1186/s40537-016-0059-y

Kaissis, G. A., Makowski, M. R., Rückert, D., & Braren, R. F. (2020). Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, *2*, 305–311. https://doi.org/10.1038/s42256-020-0186-1

Keßler, C., & McKenzie, G. (2018). A geoprivacy manifesto. *Transactions in GIS*, *22*, 3–19. https://doi.org/10.1111/tgis.12305

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. Preprint, arXiv:1412.6980.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). *Federated learning: Strategies for improving communication efficiency*. Preprint, arXiv:1610.05492.

Krumm, J. (2007). Inference attacks on location tracks. In A. LaMarca, M. Langheinrich, & K. N. Truong (Eds.), *Pervasive computing: Pervasive 2007* (Lecture Notes in Computer Science, Vol. *4480*, pp. 127–143). Berlin, Germany: Springer.

Krumm, J. (2009). A survey of computational location privacy. *Personal and Ubiquitous Computing*, *13*, 391–399. https://doi.org/10.1007/s00779-008-0212-5

Kwan, M.-P., Casas, I., & Schmitz, B. (2004). Protection of geoprivacy and accuracy of spatial information: How effective are geographical masks? *Cartographica*, *39*, 15–28. https://doi.org/10.3138/X204-4223-57MK-8273

Li, N., Li, T., & Venkatasubramanian, S. (2007). *t*-closeness: Privacy beyond *k*-anonymity and *l*-diversity. In *Proceedings of the 23rd IEEE International Conference on Data Engineering*, Istanbul, Turkey (pp. 106–115). Piscataway, NJ: IEEE.

Liang, Y., Gao, S., Cai, Y., Foutz, N. Z., & Wu, L. (2020). Calibrating the dynamic Huff model for business analysis using location big data. *Transactions in GIS*, *24*(3), 681–703. https://doi.org/10.1111/tgis.12624

Liu, X., Gong, L., Gong, Y., & Liu, Y. (2015). Revealing travel patterns and city structure with taxi trip data. *Journal of Transport Geography*, *43*, 78–90. https://doi.org/10.1016/j.jtrangeo.2015.01.016

Liu, X., Huang, Q., & Gao, S. (2019). Exploring the uncertainty of activity zone detection using digital footprints with multi-scaled DBSCAN. *International Journal of Geographical Information Science*, *33*(6), 1196–1223. https://doi.org/10.1080/13658816.2018.1563301

Luo, F., Cao, G., Mulligan, K., & Li, X. (2016). Explore spatiotemporal and demographic characteristics of human mobility via Twitter: A case study of Chicago. *Applied Geography*, *70*, 11–25. https://doi.org/10.1016/j.apgeog.2016.03.001

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). *l*-diversity: Privacy beyond *k*-anonymity. *ACM Transactions on Knowledge Discovery from Data*, *1*(1), 1–52.

Mai, G., Janowicz, K., Yan, B., Zhu, R., Cai, L., & Lao, N. (2020). *Multi-scale representation learning for spatial feature distributions using grid cells*. Preprint, arXiv:2003.00824.

McKenzie, G., Janowicz, K., Gao, S., Yang, J.-A., & Hu, Y. (2015). POI pulse: A multi-granular, semantic signature-based information observatory for the interactive visualization of big geosocial data. *Cartographica*, *50*, 71–85.

McKenzie, G., Keßler, C., & Andris, C. (2019). Geospatial privacy and security. *Journal of Spatial Information Science*, *19*, 53–55. https://doi.org/10.5311/JOSIS.2019.19.608

McSherry, F., & Mironov, I. (2009). Differentially private recommender systems: Building privacy into the Netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France (pp. 627–636). New York, NY: ACM.

Meng, X., Wang, S., Shu, K., Li, J., Chen, B., Liu, H., & Zhang, Y. (2018). Personalized privacy-preserving social recommendation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA (pp. 1–8). Menlo Park, CA: AAAI.

Narayanan, A., & Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, Oakland, CA (pp. 111–125). Piscataway, NJ: IEEE.

Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., & Boneh, D. (2013). Privacy-preserving matrix factorization. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, Berlin, Germany (pp. 801–812). New York, NY: ACM.

Niu, B., Li, Q., Zhu, X., Cao, G., & Li, H. (2014). Achieving *k*-anonymity in privacy-aware location-based services. In *Proceedings of the 2014 IEEE Conference on Computer Communications*, Toronto, Canada (pp. 754–762). Piscataway, NJ: IEEE.

Palanisamy, B., & Liu, L. (2011). Mobimix: Protecting location privacy with mix-zones over road networks. In *Proceedings of the 27th IEEE International Conference on Data Engineering*, Hannover, Germany (pp. 494–505). Piscataway, NJ: IEEE.

Polatidis, N., Georgiadis, C. K., Pimenidis, E., & Mouratidis, H. (2017). Privacy-preserving collaborative recommendations based on random perturbations. *Expert Systems with Applications*, *71*, 18–25. https://doi.org/10.1016/j.eswa.2016.11.018

Rao, J., Gao, S., Kang, Y., & Huang, Q. (2020). LSTM-TrajGAN: A deep learning approach to trajectory privacy protection. In *Proceedings of the 11th International Conference on Geographic Information Science*, Poznań, Poland (Part 1, pp. 12:1–12:17).

Rendle, S. (2010). Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, Sydney, Australia (pp. 995–1000). Piscataway, NJ: IEEE.

Seidl, D. E., Jankowski, P., & Tsou, M.-H. (2016). Privacy and spatial pattern preservation in masked GPS trajectory data. *International Journal of Geographical Information Science*, *30*, 785–800. https://doi.org/10.1080/13658816.2015.1101767

Shaw, S.-L., & Sui, D. (2020). Understanding the new human dynamics in smart spaces and places: Toward a splatial framework. *Annals of the American Association of Geographers*, *110*, 339–348. https://doi.org/10.1080/24694452.2019.1631145

Sun, Y., Fan, H., Li, M., & Zipf, A. (2016). Identifying the city center using human travel flows generated from location-based social networking data. *Environment and Planning B*, *43*, 480–498.

Swanlund, D., Schuurman, N., Zandbergen, P., & Brussoni, M. (2020). Street masking: A network-based geographic mask for easily protecting geoprivacy. *International Journal of Health Geographics*, *19*, 1–11. https://doi.org/10.1186/s12942-020-00219-z

Vaudenay, S. (2020). *Centralized or decentralized? The contact tracing dilemma* (Technical Report 2020/531). Retrieved from https://eprint.iacr.org/2020/531.pdf

Xiao, Y., & Xiong, L. (2015). Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver, CO (pp. 1298–1309). New York, NY: ACM.

Xiong, L., Zhou, W., Xia, Z., Gu, Q., & Weng, J. (2020). *Efficient privacy-preserving computation based on additive secret sharing*. Preprint, arXiv:2009.05356.

Xu, F., Tu, Z., Li, Y., Zhang, P., Fu, X., & Jin, D. (2017). Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia (pp. 1241–1250). New York, NY: ACM.

Xu, Y., Li, J., Belyi, A., & Park, S. (2021). Characterizing destination networks through mobility traces of international tourists: A case study using a nationwide mobile positioning dataset. *Tourism Management*, *82*, 104195. https://doi.org/10.1016/j.tourman.2020.104195

Xu, Y., Shaw, S.-L., Zhao, Z., Yin, L., Lu, F., Chen, J., Fang, Z., & Li, Q. (2016). Another tale of two cities: Understanding human activity space using actively tracked cellphone location data. *Annals of the American Association of Geographers*, *106*, 489–502. https://doi.org/10.4324/9781315266336-2

Yan, X.-Y., & Zhou, T. (2019). Destination choice game: A spatial interaction theory on human mobility. *Scientific Reports*, *9*, 1–9. https://doi.org/10.1038/s41598-019-46026-w

Yang, D., Qu, B., & Cudré-Mauroux, P. (2018). Privacy-preserving social media data publishing for personalized ranking-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *31*, 507–520.

Yang, D., Zhang, D., Chen, L., & Qu, B. (2015). NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *Journal of Network and Computer Applications*, *55*, 170–180. https://doi.org/10.1016/j.jnca.2015.05.010

Yang, D., Zhang, D., & Qu, B. (2016). Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology*, *7*, 1–23.

Yang, D., Zhang, D., Qu, B., & Cudré-Mauroux, P. (2016). PrivCheck: Privacy-preserving check-in data publishing for personalized location based services. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Heidelberg, Germany (pp. 545–556). New York, NY: ACM.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, *10*, 1–19.

Yao, Y., Peng, Z., Xiao, B., & Guan, J. (2017). An efficient learning-based approach to multi-objective route planning in a smart city. In *Proceedings of the 2017 IEEE International Conference on Communications*, Paris, France (pp. 1–6). Piscataway, NJ: IEEE.

Ye, M., Yin, P., & Lee, W.-C. (2010). Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, San Jose, CA (pp. 458–461). New York, NY: ACM.

Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., … Gai, K. (2019). Deep interest evolution network for click-through rate prediction. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI (pp. 5941–5948). Menlo Park, CA: AAAI.

Zhu, H., Yang, X., Wang, B., Wang, L., & Lee, W.-C. (2019). Private trajectory data publication for trajectory classification. In W. Ni, X. Wang, W. Song, & Y. Li (Eds.), *Web information systems and applications: WISA 2019* (Lecture Notes in Computer Science, Vol. *11817*, pp. 347–360). Cham, Switzerland: Springer.

Zhu, Y., & Diao, M. (2020). Crowdsourcing-data-based dynamic measures of accessibility to business establishments and individual destination choices. *Transportation Research Part D: Transport and Environment*, *87*, 102382.

Zigomitros, A., Papageorgiou, A., & Patsakis, C. (2016). A practical *k*-anonymous recommender system. In *Proceedings of the Seventh International Conference on Information, Intelligence, Systems and Applications*, Chalkidiki, Greece (pp. 1–4). Piscataway, NJ: IEEE.

How to cite this article: Rao J, Gao S, Li M, Huang Q. A privacy-preserving framework for location recommendation using decentralized collaborative machine learning. *Transactions in GIS*. 2021;00:1–23. https://doi.org/10.1111/tgis.12769