Hindawi
Security and Communication Networks
Volume 2018, Article ID 9203076, 13 pages
https://doi.org/10.1155/2018/9203076

WILEY | Hindawi

Research Article

GANs Based Density Distribution Privacy-Preservation on Mobility Data

Dan Yin and Qing Yang

Department of Computer Science and Technology, Harbin Engineering University, Harbin, China

Correspondence should be addressed to Qing Yang; yangqing@hrbeu.edu.cn

Received 7 September 2018; Revised 15 October 2018; Accepted 19 November 2018; Published 2 December 2018

Guest Editor: Liran Ma

With the development of mobile devices and GPS, plenty of Location-based Services (LBSs) have emerged in these years. LBSs can be applied in a variety of contexts, such as health, entertainment, and personal life. The location based data that contains significant personal information is released for analysing and mining. The privacy information of users can be attacked from the published data. In this paper, we investigate the problem of privacy-preservation of density distribution on mobility data. Different from adding noises into the original data for privacy protection, we devise the Generative Adversarial Networks (GANs) to train the generator and discriminator for generating the privacy-preserved data. We conduct extensive experiments on two real world mobile datasets. It is demonstrated that our method outperforms the differential privacy approach in both data utility and attack error.

1. Introduction

With the increasing popularity of mobile devices and GPS, plenty of Location-based Services (LBSs) have emerged in these years. LBSs can be applied in a variety of contexts, such as health, entertainment, and personal life. People can report their locations anywhere and anytime. For example, people release tweets with their current locations on social networks; users share their running routines with their friends on the Internet. The location based data which includes significant personal information is often published for analysing and mining.

The mobility data implies valuable personal information, such as home addresses, occupation, social relations, and interests. Attackers can discover the privacy information of users from the published dataset. For instance, the identities can be interred from the locations where people often visit over a period of time, even their home addresses or occupations.

In order to protect the personal information, there has been some research on the privacy-preservation of mobility data. One of them is proposed in [1], which aggregates the users in each location and publishes the aggregated results (density distribution) instead of the original location distribution. However, attackers can recover the users' mobile trajectories from the density distribution for a period of time. As shown in Figure 1, there are 3 location samples of 6 users $\{u_1, u_2, \ldots, u_6\}$ at timestamp $\{t_1, t_2, t_3\}$. The whole space is divided into 3 blocks $\{b_1, b_2, b_3\}$. If we aggregate the users in each block, we can get the density distribution from times t_1 to t_3. While this aggregate result is vulnerable to reconstruction attack [2], therefore, it is crucial to propose a new method for the density distribution privacy-preservation. As shown in the figure, we can generate similar density distribution under privacy-preservation to publish. It is hard for the adversaries to recover the users' trajectories from the published distribution.

Alternatively, there are other methods by adding noises into aggregated results for privacy-preservation. For example, differential privacy (DP) [3] is proposed aiming to provide means to maximize the accuracy of queries from statistical databases while minimizing the chances of identifying its records. However, the utility of the privacy-preserved data produced by DP method becomes worse significantly.

GANs are a class of algorithms in unsupervised machine learning, which have been widely used to produce samples of photo realistic images for the purposes of visualizing new interior design. Motivated by this, we try to utilize GANs

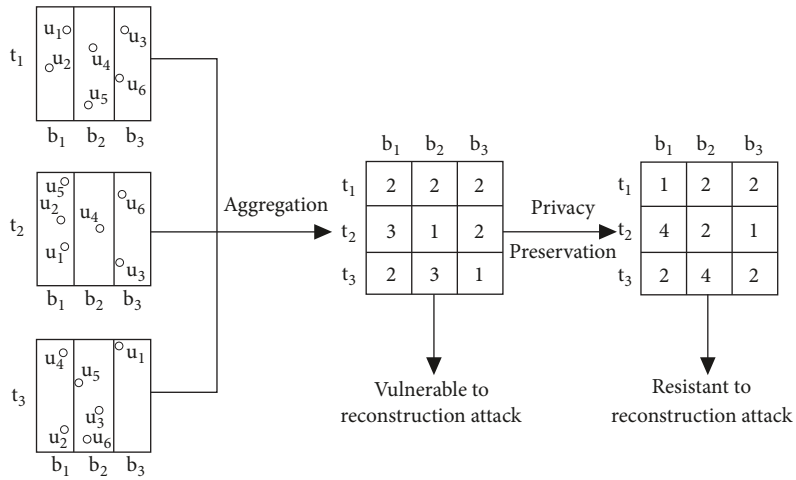


Figure 1: Density distribution privacy-preservation.

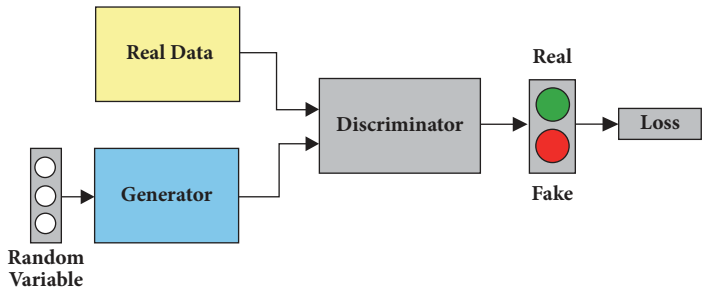


Figure 2: Architecture of GANs.

for generating privacy-preserving data with high utility. In this paper, we propose the density distribution privacy-preservation on mobility data based on GANs. By training the two neural networks, we can generate the privacy-preserved data which can achieve high data utility and low attack error. The main contributions of this paper are as follows.

(i) We investigate the privacy-preservation of density distribution on mobility data against the aggregation attack. Different from adding noises to the original data, we propose a deep learning method based on GANs to solve the problem.

(ii) Motivated by the applications of GANs on image processing, we train the generator and discriminator in GANs by random data and the original data and publish the data generated by the generator instead of the original data. To the best of our knowledge, this is the piece of paper employing GANs on data privacy-preservation.

(iii) We conduct extensive experiments on two real world datasets. The experimental results demonstrate that our method outperforms the differential privacy in both data utility and attack error.

The rest of this paper is organised as follows. Section 2 introduces the preliminaries. Section 3 introduces the proposed methods. Section 4 presents the experiment results. Section 5 describes the related work. Section 6 concludes the whole paper.

2. Preliminary

In this section, we start with the introduction of GANs, which is the basic architecture of our method. Then we describe the recently proposed attack model that recovers individual users' trajectries from density distributions, which will be adopted to measure the privacy-preservation ability of our method.

2.1. Generative Adversarial Networks. Generative Adversarial Networks (GANs) are a class of artificial intelligence algorithms used in unsupervised machine learning, which are composed of two neural networks contesting with each other in a zero-sum game framework. GANs were introduced by lan Goodfellow et al. [4] in 2014 as a novel way to model data distributions. The architecture of the general GANs is shown in Figure 2.

Specifically, the two neural networks are a generator G and a discriminator D. In the original GANs, the generator G accepts a random distribution P_z and generates synthetic data from P_z. While the goal of the discriminator D is to distinguish the synthetic data generated by G from real data $\boldsymbol{x}$, the optimal D would distinguish synthetic data from real data exactly. While the optimal G would generate synthetic

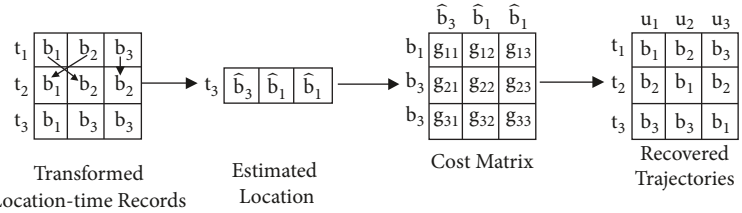


Figure 3: Procedure of reconstruction attack on transformed mobility dataset.

data that are indistinguishable from the real data for D, in the training phase of the original GANs, both the generator G and the discriminator D are iteratively optimized against each other in a minimax game with the value function $V(G, D)$, which can be formalized as

$$\min_{\theta_G} \max_{\theta_D} V(G, D) = \mathbb{E}_{x \sim P_{data}} [\log D(x)] + \mathbb{E}_{z \sim P_z} [\log (1 - D(G(z)))], \quad (1)$$

where θ_G and θ_D are the parameters of G and D, respectively.

To handle the privacy risks of releasing density distributions datasets, we generate the synthetic distributions with GANs as a protected version of the original data.

2.2. Attack Model. Many researches have been done on inference of sensitive information [5]. In order to protect the location information, some researchers aggregate the users in the same locations over a period of time and then publish the density distributions [6]. However, Fengli et al. [2] propose that releasing aggregated results does not preserve users' privacy, since a user's mobility pattern is regular while different from others'. Based on the characteristics of human mobility, they transform the density distribution to a *location-time* format and propose the trajectories reconstruction attack that iteratively associates the same users' mobility records in the neighbouring time slots. They exploit the regularity of mobility data to estimate the next location of the user and choose the location in the transformed dataset with the largest similarity to the estimated next location as the reconstructed next location according to the uniqueness pattern of human mobility data.

To recover trajectories from the density distributions, the first step is to transform the density distribution $P^t = [p_1^t, p_2^t, \ldots, p_i^t, \ldots, p_n^t]$ into a *location-time* record $B^t = [b_1^t, b_2^t, \ldots, b_j^t, \ldots, b_m^t]$, where p_i^t represents the number of users at location i during time slot t, b_j^t represents the location of the j^{th} user at time slot t, n represents the total number of possible locations, and m is the total number of users. To link the *location-time* records that represent the same users across different time slots, the reconstruction attack is modeled as a *Linear Sum Assignment Problem* [7], which can be solved in polynomial time based on *Hungarian algorithm* [8].

Specifically, we assume a set of recovered trajectories until time slot t as $S^t = [s_1^t, s_2^t, \ldots, s_k^t, \ldots, s_m^t]$, where $s_k^t = [b_k^1, b_k^2, \ldots, b_k^t]$ is the k^{th} recovered trajectory and b_k^t is the recovered location at time slot t. For the adversaries, with the abundant kinds of social networks, such as WeChat and MoMo, it is effortless to get some background information of the individuals, such as trajectories in a shot time. To recover the next position b_k^{t+1} from the *location-time* records $B^{t+1} = [b_1^{t+1}, b_2^{t+1}, \ldots, b_m^{t+1}]$, an estimated location $\widehat{b}_k^{t+1}$ is first generated based on the continuity feature of human mobility, and then the location in the *location-time* record B^{t+1} with the largest likelihood to the estimated location $\widehat{b}_k^{t+1}$ will be chosen as the recovered next location, i.e., b_k^{t+1}. In the daytime, users move frequently, and their locations are continuous, which makes it possible to estimate the next location with the current location and the velocity. Formally, for the $k^{th}(1 \leq k \leq m)$ recovered trajectory, the estimated location is

$$\widehat{b}_k^{t+1} = b_k^t + \left(b_k^t - b_k^{t-1}\right). \quad (2)$$

To quantify the likelihood between the estimated location and those in the *location-time* records, Fengli et al. [2] formulate the cost matrix $G^t = \{g_{i,j}^t\}_{m \times m}$, where $g_{i,j}^t$ is the distance between the estimated next location $\widehat{b}_i^{t+1}$ and the actual location $_j^{t+1}$.

Figure 3 presents the process of recovering the trajectories. There are three possible locations and three time slots. We assume that trajectories until time slot t_2 have been recovered, and then the estimations of the t_3 locations are generated based on the continuity feature of mobility data. The distance between the estimated locations and those in *location-time* records is formulated as the cost matrix. In the last step, *Hungarian algorithm* is applied to minimize the cost matrix and find each trajectory's associated location in the *location-time* record. The right part in Figure 3 demonstrates the recovered trajectories.

Generally, the adversaries may have different kinds of background knowledge based on various sources such as social networks. However, in our specific attack model, to ease the presentation, we assume that the adversary has the target users' location information in the first two time slots as the background knowledge.

3. Method

In this section, we first give an overview of our proposed method using GANs to generate private-preserving density distributions. Then, we describe the architecture of discriminator and generator network, respectively. Finally, we introduce the loss function of our method.

3.1. Overview. In GANs, the generator network G accepts the random data and generates the synthetic data that are similar

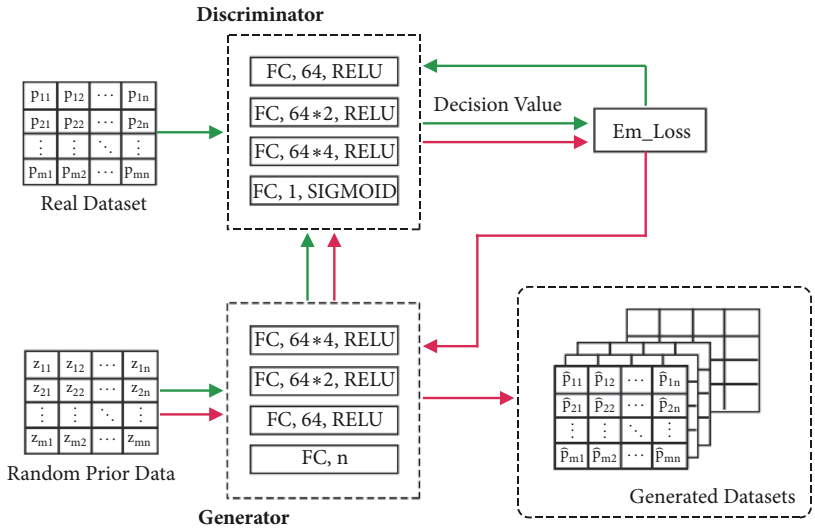


Figure 4: The training process of GANs.

Input: The real dataset P; The learning rate lr;
Output: The privacy-preserved version of the real dataset $\widehat{P}$;
1: Build discriminator network **D**;
2: Build generator network **G**;
3: $Z = randm(size(P))$;
4: **while** $lr \leq loss$ **do**
5: $D = train(D)$;
6: $G = train(G)$;
7: $loss = em\_loss(G(Z), P)$;
8: **end while**
9: $\widehat{P} = G(Z)$;

Algorithm 1: Pseudocode of the proposed privacy-preservation method based on GANs.

to the real data, while the discriminator network D tries to distinguish the generated data from the real one. At the start of the training phase of GANs, the synthetic data generated by G is easy to be distinguished by the discriminator D as G has not learned the features of the real data, which could be regarded as achieving high privacy-preserving ability, while, with the increase of the training steps, the synthetic data generated by the generator network G becomes indistinguishable for the discriminator D, as G has learned the features of the real data, which could be regarded as having less privacy-preserving ability but high utility. Based on the above observation, we propose a framework based GANs to generate privacy-preserving density distributions. We also summarize our proposed method using pseudocode in Algorithm 1.

We assume the real density distribution is $P_{n\times m}$, which is the protected target containing n blocks, i.e., $\{b_1, b_2, \ldots, b_n\}$ during m time slots, i.e., $\{t_1, t_2, \ldots, t_m\}$. We use $p_{i,j}$ to denote the number of users in block b_i at time slot t_j. To protect the real density distribution P, we build GANs to learn the potential features of P and exploit the generator network G to generate the privacy-preserving version of the real dataset that includes most of the real features and excludes the sensitive information (e.g., the individual users' mobility trajectories).

We introduce the training process of GANs in Figure 4. We show the training flow of discriminator and generator with green and red lines, respectively. When training the discriminator, we input the real data and update the parameters in D based on the loss value. Then the generated data is fed into D, which trains D to learn the features of the generated synthetic data.

When training the generator network G, we first provide a random integral matrix $Z_{n\times m}$ with the same size of the real data P. Then the synthetic data is generated by G and fed into the discriminator network D. After that, the discriminator D computes the classification of the generated data as the decision value and sends it to the loss function to compute the loss value. Based on the loss value, the generator G adjusts its parameters via backproppagation. The generated synthetic data becomes more and more similar to the real data with the number of the training rounds increasing. We train the GANs for k rounds and store the synthetic data generated by G as $\widehat{P} = \{\widehat{P_1}, \widehat{P_2}, \ldots, \widehat{P_k}\}$, where $\widehat{P_i}$ is the synthetic data generated in the i_{th} training round. The difference between the generated and real data is computed as the utility loss. With the increasing of training rounds, the utility loss experiences a downward trend via backpropogation, and the generated data becomes more and more similar to the real data. In other words, the utility of the synthetic dataset changes with each training round, and the datasets contained in $\widehat{P}$ can satisfy different utility requirements.

3.2. Discriminator Network. We design the discriminator network D with 4 layers. The first 3 layers are designed to learn the features of the input data, and the last layer is designed to compute the decision value. As the density distribution is one-dimension integral matrix, we set all the

layers in discriminator network to be fully connected with the adjacent layers and choose *Relu* as the activation function for the first 3 layers. We choose *Sigmoid* as the activation function of the last layer, so that the output is limited from 0 to 1. We set 0 as fake and 1 as real. When the input of the discriminator is the real data, the goal of the discriminator is to output a value as close to 1 as possible, and when the input is the generated data, the output should be as close to 0 as possible.

The training stage of the discriminator is composed of 3 parts. (1) Learn the features of the real data. (2) Learn the features of the generated data. (3) Learn to distinguish the real and fake data.

3.3. Generator Network. The input of the generator is a random intergal matrix Z. We set the size of Z the same as the real data. The output of the generator is the synthetic density distributions data. The generator is trained to learn the features of the real data. The generated data becomes more and more similar to the real data. In the early training stage, the generated dataset is of high privacy-preserving ability as the generator has not learned the features of the real data. With the training rounds increasing, the generated data is more similar to the real data, but less privacy-preserving.

We design the generator network G with 4 layers. In original GANs, the generator and discriminator are opponents, so we reverse the first 3 layers in the discriminator network D as the first part of the generator. Similar to the discriminator network, we fully connect all the adjacent layers and choose *Relu* as the activation function. For the last layer of the generator, we set its node number the same as the block number of the real data. All nodes are fully connected with the third layer's nodes. So the output of the generator has the same size as the real data.

We first train the discriminator D with the real data P and the generated data $\widehat{P}$ from G for a certain training rounds. Then we train the generator by combining the generator G and the discriminator D. In this stage, the parameters in D are fixed, while G adjusts its parameters based on the decision value computed by D via backpropagation. The generator's goal is to cheat the discriminator, so the generator tries to adjust it parameters that could receive the decision value from the discriminator as close to 1 (which means real data) as possible, which means the discriminator cannot distinguish the generated data from the real data.

At each training round of generator, we save the generated data $\widehat{P}$ and compute the mean square error (MSE) between $\widehat{P}$ and P as the utility loss, so that we can provide the suitable generated privacy-preserving data satisfying different utility requirements.

3.4. Loss Function. We employ the *Wasserstain distance* [9] $W(q, p)$ as the loss function, which measures the difference between the decision value and the real classification value (1 for real data and 0 for fake data). $W(q, p)$ indicates the minimum cost of transforming from the distribution q to p. The classification value of the real/generated data is $Y = [y_1, y_2, \ldots, y_m]$, where y_i is the classification value of the i_{th} record and m is the number of records in the dataset. The corresponding decision value computed by discriminator is $\widehat{Y} = [\widehat{y_1}, \widehat{y_2}, \ldots, \widehat{y_m}]$. The loss value is

$$em\_loss\left(Y, \widehat{Y}\right) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i \cdot \widehat{y_j}. \tag{3}$$

In the training phase of the discriminator, when the input is the real data, we set the classification value as 1, and when the input is the generated data, we set the classification value as 0. In the training phase of the generator, we set the classification value of the generated dataset as 1 to cheat the discriminator.

4. Evaluation

In this section, we evaluate the performance of our method. We also introduce another two privacy-preserving methods: geo-indistinguishable and exponential mechanism-based methods. Finally we compare our method with them on the trade-off between the utility loss and privacy-preservation, which is measured by the attack model.

4.1. Evaluation Metrics. In the privacy-preserving research area, the trade-off between the privacy and utility is the focus attention. In this part, we demonstrate the performance of our method on privacy-preservation under the attack model and utility loss compared with the real world datasets.

We quantify the privacy-preservation performance by the *reconstruction error* under the attack model, which is defined as the Euclidean distance between the reconstructed individual users' trajectories and the ground truth. A larger *reconstruction error* indicates that the density distributions protected by our method are not vulnerable to be attacked and our method achieves a better privacy-preservation performance.

We employ the *mean square error* (MSE) to measure the difference between the generated density distributions and the real world dataset. We quantify the *utility loss* by MSE. A smaller MSE means the density distributions generated by our method are more similar to the real datasets.

4.1.1. Reconstruction Error. The attack model introduced in Section 2.2 aims to reconstruct the individual users' trajectories from the density distributions. We compute the average Euclidean distance (reconstruction error) between the reconstructed trajectories and the ground truth to measure the privacy-preservation performance.

We assume the number of users in the real density distribution is u, and the trajectory of the $i_{th}(0 \le i \le u)$ user is $s_i = [l_i^1, l_i^2, \ldots, l_i^m]$, where m is the number of the time slots and each element is the location block of the user at a specific time slot. The corresponding reconstructed trajectory of the user is $\widehat{s_i} = [\widehat{l_1^i}, \widehat{l_2^i}, \ldots, \widehat{l_m^i}]$. We compute the reconstruction error as

$$Reconstruction\ Error = \frac{\sum_{i=1}^{u} \left\|\left(\widehat{s_i} - s_i\right)\right\|_2^2}{u}, \tag{4}$$

where $\|(\widehat{s}_i - s_i)\|_2^2$ is the Euclidean distance between the reconstructed and ground truth trajectories. A larger reconstruction error indicates the attack is unsuccessful, while the privacy is protected better.

4.1.2. Utility Loss. We measure the utility loss of protected version of the real world dataset by computing the mean square error between the real world dataset P and its privacy protected version $\widehat{P}$.

Formally, we denote the real density distributions as $P_{n\times m}$, where n is the total number of the location blocks and m is the total number of the time slots. We use $p_{i,j}(0 \le i \le n, 0 \le j \le m)$ to represent the number of users at block i, in time slot j. We denote the corresponding protected version of the real density distributions as $\widehat{P}_{n\times m}$ with the same size of the real world dataset, and $\widehat{p}_{i,j}$ represents the number of users in block i at time slot j in the protected dataset. The utility loss can be computed as

$$Utility\ Loss = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(p_{i,j} - \widehat{p}_{i,j}\right)^2}{m \times n}. \tag{5}$$

A smaller utility loss indicates the data in the protected dataset is more similar to the real dataset and better practical usability.

4.2. Compared Methods. In this part, we introduce another two privacy-preserving methods commonly used in the recent research: the geo-indistinguishability method [1] and exponential mechanism [3] based method.

Geo-Indistinguishability Method. This method is proposed by Andrés et al. to protect the location-based data with a differentially private mechanism. In this method, Laplacian noises [1] are employed to generate a radius, and the real location data is remapped by the radius with a random angle. We call the dataset protected by this method Geo-MDA.

Exponential Mechanism Based Method. The exponential mechanism [3] is one of the most renowned tools used in differential privacy. The general idea of exponential mechanism is sampling an output from the output space according to a utility function. In our experiments, we employ the proportion of users in each location block as the utility function and sample the user numbers of each location under different parameter settings. We call the dataset protected by this method Exp-MDA.

4.3. Datasets. We use two real world mobility datasets, MoMo mobile app dataset and San Francisco cabs dataset.

MoMo Mobile App Dataset (MoMo) [10]. MoMo is a mobile social networking application in China. This dataset was collected from the GPS of the mobile devices using MoMo from 21 May, 2012, to 26 June, 2012, in Beijing, China. Each record in the dataset contains the user ID, timestamp, latitude, and longitude.

San Francisco Cabs Dataset (SFC) [11]. This dataset contains the mobility trajectories of taxi cabs in San Fancisco, USA. This dataset was collected over 30 days in the San Francisco bay area. Each record has four attributes: cab ID, timestamp, latitude, and longitude.

After the preprocessing of the raw datasets, we choose 198 users' trajectory records from MoMo dataset and set the spatial resolution 2km and the temporal resolution 30 minutes. For the SFC dataset, we choose 127 users' trajectories and set the temporal resolution 2 minutes, as the mobility speed of human beings is much slower than the taxi cabs. The size of the area for both the datasets is $50km \times 50km$, and the location blocks number is 625. In the preprocessing stage, we transform the individual users' mobility records into the density distributions P, that is, counting the users number in each location at each time slot.

4.4. Privacy-Preserving Performance against the Attack Model. We first apply the attack model on the real density distributions datasets of MoMo and SFC. The average reconstruction errors obtained by the reconstruction attack on MoMo and SFC are 4.32km and 1km, respectively. We regard these reconstruction errors as the baseline in our evaluation experiments and represent them as horizontal lines in Figures 5 and 7.

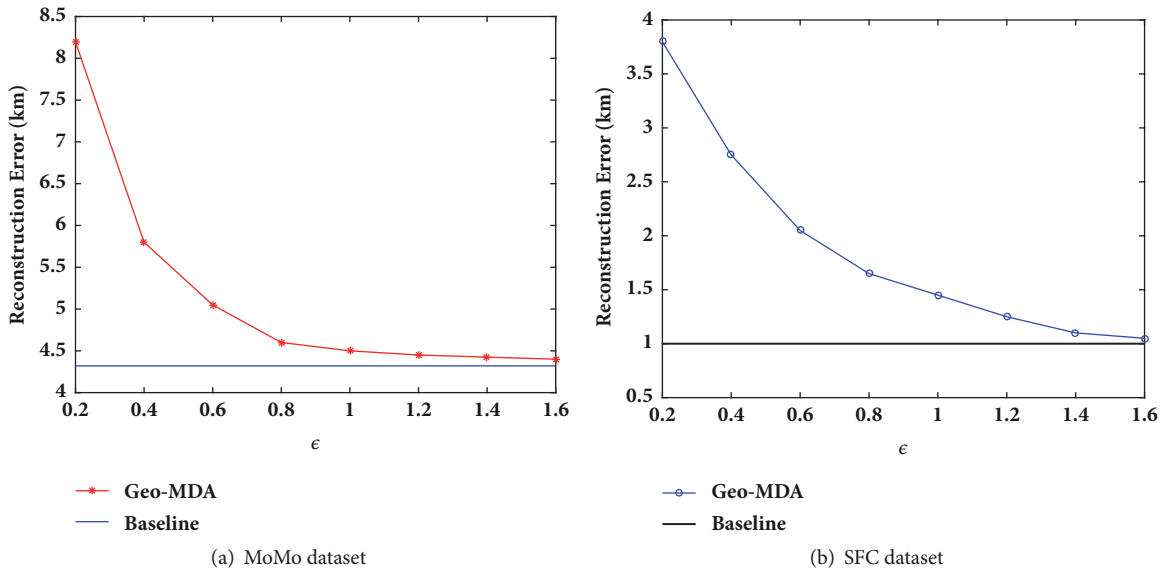
We evaluate the privacy-preservation ability of the Geo-MDA against the attack model. The results are shown in Figure 5. The x-axis shows the parameter ϵ of this method, which controls the noise level, and the y-axis stands for the reconstruction error. We evaluate the Geo-MDA with the parameter ϵ from 0.2 to 1.6. As the results show in Figure 5, for both the MoMo and SFC datasets, the reconstruction error decreases when ϵ increases, which indicates that when the value of ϵ increases, the privacy-preservation ability of Geo-MDA becomes weak.

Then we evaluate the privacy-preservation ability of the exponential mechanism method (Exp-MDA). The results are shown in Figure 6. In the exponential mechanism, we test the value of ϵ from 0; however, for MoMo dataset, when we vary ϵ from 0 to 5, the reconstruction error is stable, and for both datasets, when ϵ is larger than 13.2, the reconstruction error starts to increase. We only show the results with ϵ ranging from 5 to 13.2 for MoMo dataset, and for the SFC dataset, we show the results with ϵ ranging from 5.8 to 13.2.

Figure 6 shows that the reconstruction errors of both datasets are decreasing when ϵ value is increasing. And the minimum reconstruction error for MoMo is about 19.2 km and for SFC is 5.3 km; both are larger than the baseline of the real dataset. Besides, we observe that when the value of ϵ is around 6 and 7, for MoMo and SFC, respectively, the reconstruction error remains stable, because, in the exponential mechanism, the output changes slowly when ϵ is small, and with the increase of ϵ, the changes become quicker as shown in Figure 6. The Exp-MDA could provide protection to the real dataset.

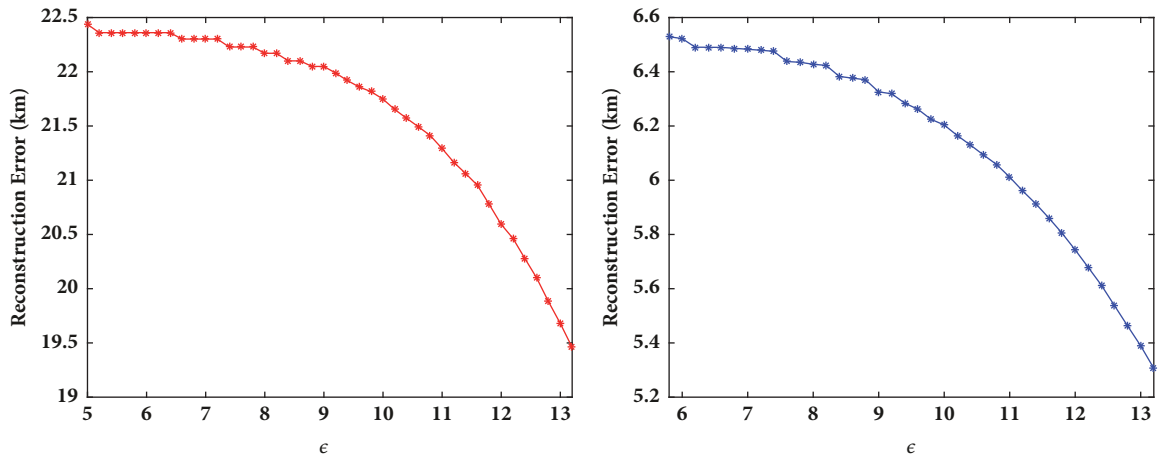
In our method, we save the generated dataset at each training time. To evaluate the privacy-preservation performance against the attack model of our method, we conduct the reconstrction attack on the fake datasets generated by each training time, and the results are shown in Figure 7.

In Figure 7, the x-axis represents the training times, and the y-axis is the corresponding reconstruction error.



(a) MoMo dataset (b) SFC dataset

Figure 5: The reconstruction error of Geo-MDA under different noise factor ϵ.



(a) MoMo dataset (b) SFC dataset

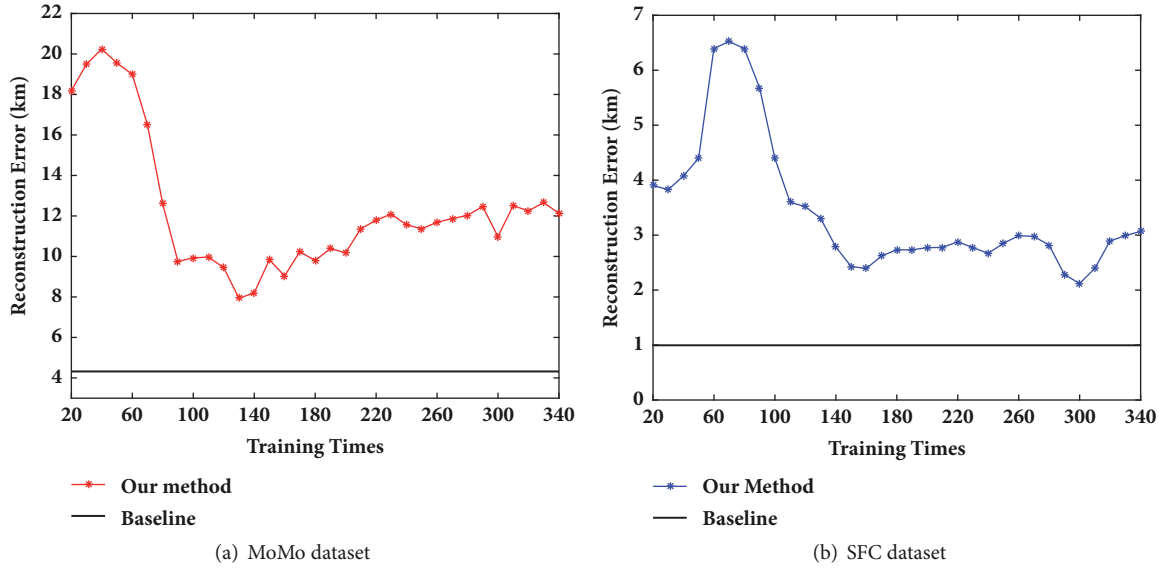
Figure 6: The reconstruction error of Exp-MDA under different noise factor ϵ.

We observe that the shape of the results is wavy; that is, because the training phase of GANs is adversarial, the generator and discriminator are trained in turn. However, the trend of the results is decreasing, which indicates that the privacy-preserving ability is decreasing with the training time growing. For MoMo dataset, the reconstruction error is limited within 10 ~ 12km (larger than the baseline 4.32km) when the GANs are trained more than 220 times. And for the SFC dataset, when the training time is larger than 180, the reconstruction error is constrained between 2km and 3km, which is also larger than the baseline value (1km).

We cannot compare our method with Geo-MDA and Exp-MDA by now, because their parameters are different, and we need to consult the utility-preservation ability of these methods as well.

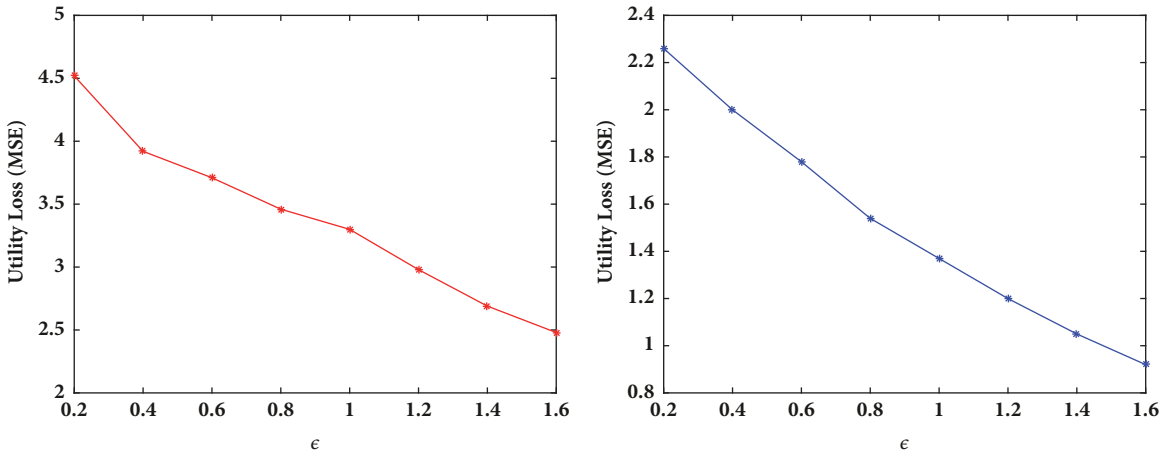
4.5. Performance on the Utility-Preservation. In this section, we evaluate the utility-preservation ability of the Geo-MDA, Exp-MDA, and our method. The utility-preservation performance is quantified by the utility loss, which has been introcduced in Section 4.1.2. A smaller utility loss indicates that the difference between the protected dataset and the real dataset is small, and the protected dataset generated by the privacy-preserving methods is of high practical usability.

The utility-preservation performance of the Geo-MDA is shown in Figure 8. The x-axis represents the parameter of Geo-MDA, ϵ, which controls the noise level of this method. And the y-axis is the utility loss. We observe that when we vary ϵ from 0.2 to 1.6, the values of the utility loss for both MoMo dataset and SFC dataset represent a downtrend. When ϵ is equal to 1.6, the utility loss for MoMo and SFC datasets



(a) MoMo dataset (b) SFC dataset

Figure 7: The reconstruction error of our method.



(a) MoMo dataset (b) SFC dataset

Figure 8: The utility loss of Geo-MDA under different noise factor ϵ.

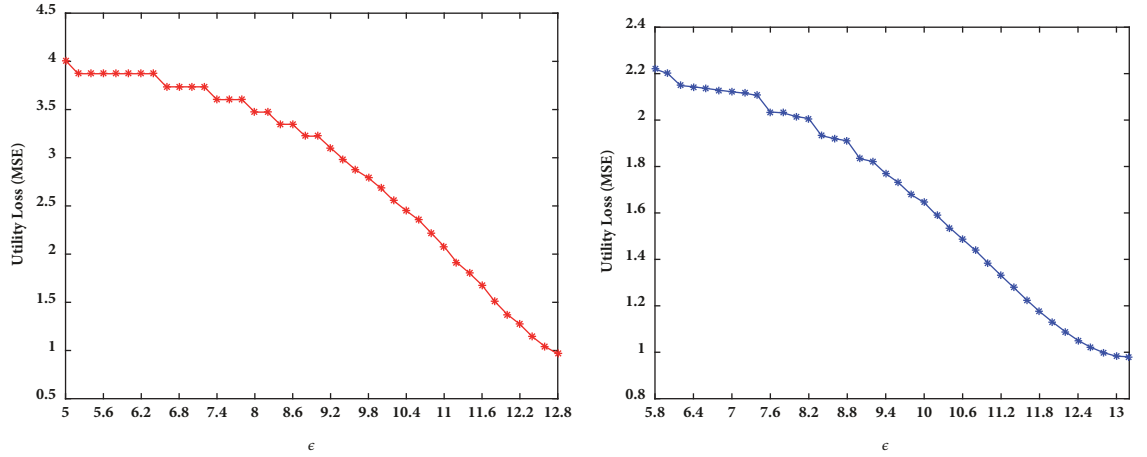
is 2.5 and 0.92, repectively, which shows that the utility-preservation ability of both MoMo and SFC is improving with the growth of ϵ.

We then evaluate the utility-preservation ability of the Exp-MDA under different parameter settings. The results are shown in Figure 9.

The x-axis is the parameter ϵ of this method, and the y-axis indicates the utility loss of the protected dataset under different parameter settings. For both the MoMo and SFC datasets, the utility loss decreases with the increase of ϵ. For MoMo dataset, when we increase ϵ from 5 to 12.8, the utility loss decreases from 4 to 1. The condition of the SFC dataset is similar to that of the MoMo dataset. Similar to the Geo-MDA, the utility-preservation ability of the Exp-MDA method improves with the increase of ϵ.

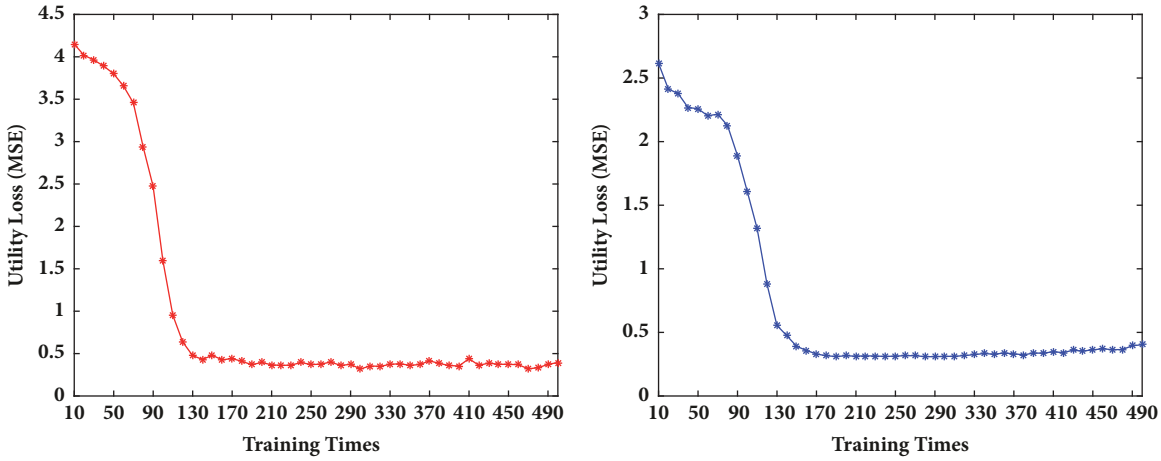
We then evaluate the utility loss that existed in our method under different training times. The results are shown in Figure 10. For both MoMo and SFC datasets, we train the GANs 500 times and save the generated datasets every 10 training times. We observe that as the training times increase, the utility loss decreases and becomes stable. For the MoMo dataset, the utility loss stays about 0.4 when the training times are larger than 130. On the other side, the utility-preservation ability will not increase when the train times are larger than 130 and 170 for MoMo and SFC datasets, respectively.

In Figure 11, we present some actual examples protected by our method to illustrate the utility loss value and the protected dataset in practice. We use colours with different gradations to show different numbers of users on the map, and the lighter the colour is, the larger the number of users in that location is. Straightforwardly, we can observe that when the utility loss is 2.25, the practical usage of the dataset generated by our method is weak and when the utility loss decreases to 0.4, the dataset generated by our method is very



(a) MoMo dataset (b) SFC dataset

Figure 9: The utility loss of Exp-MDA under different noise factor ϵ.



(a) MoMo dataset (b) SFC dataset

Figure 10: The utility loss of our method.

similar to the real dataset, which achieves better practical usage.

4.6. Privacy-Utility Trade-Off Comparison. We compare the performance of our method with the Geo-MDA and Exp-MDA by the privacy-utility trade-off in this part. The comparison is conducted by combining the privacy-preservation evaluation and the utility-preservation evaluation. The results are shown in Figure 12.

The x-axis represents the reconstruction error, which denotes the privacy-preservation ability, and the y-axis is the utility loss. We observe that, for both the MoMo and SFC datasets, under the same reconstruction error, the utlity loss of our method is smaller than the other two methods. For example, in Figure 12(b), for the SFC dataset, when the reconstruction error is 2.8km, the utility loss of the Geo-MDA and the Exp-MDA is 2 and 1.5, respectively, while the utility loss of our method is only 0.5 under the same reconstruction error. Besides, in Figure 12(a), even when the reconstruction error is as high as 12.5km, our method still preserves the utility loss less than 0.5.

5. Related Work

In this section, we start with the introduction about the services of the mobility dataset. Then we present the applications of GANs. Finally, we summarize the existing privacy-preserving methods for mobility datasets releasing.

5.1. Services for the Mobility Datasets. With the fast development of the mobile smart devices and the Internet technology, a huge number of services are developed based on the mobility datasets to provide useful information to the users. These services support human daily life by studying mobility patterns from trillions of trails and footprints [12]. Urban planning [13], face recognition [14], classification [15], traffic forecasting [16], marker campaign [17], prediction of epidemics [18, 19], latent data privacy [20], and designing

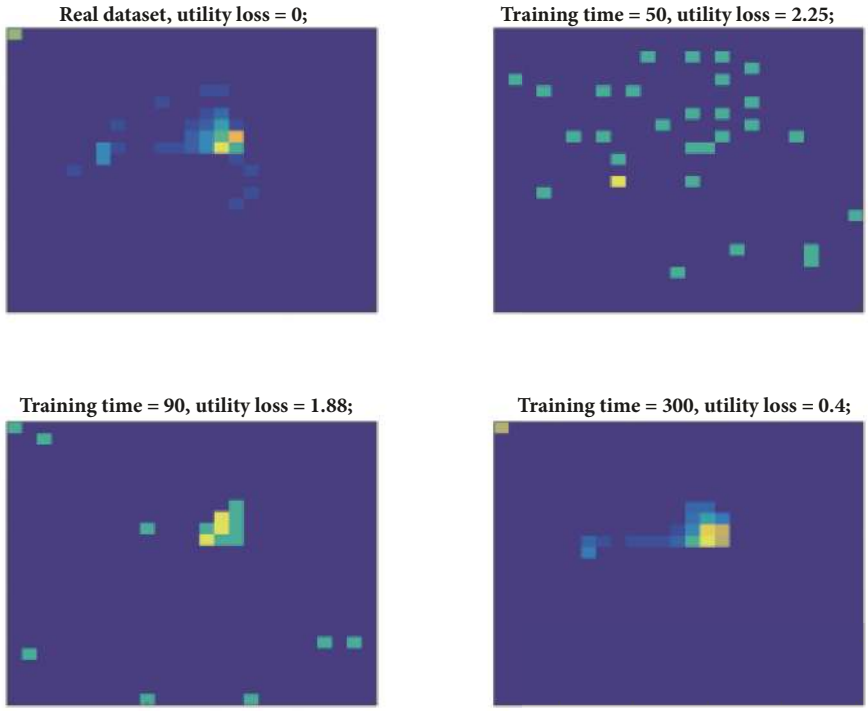
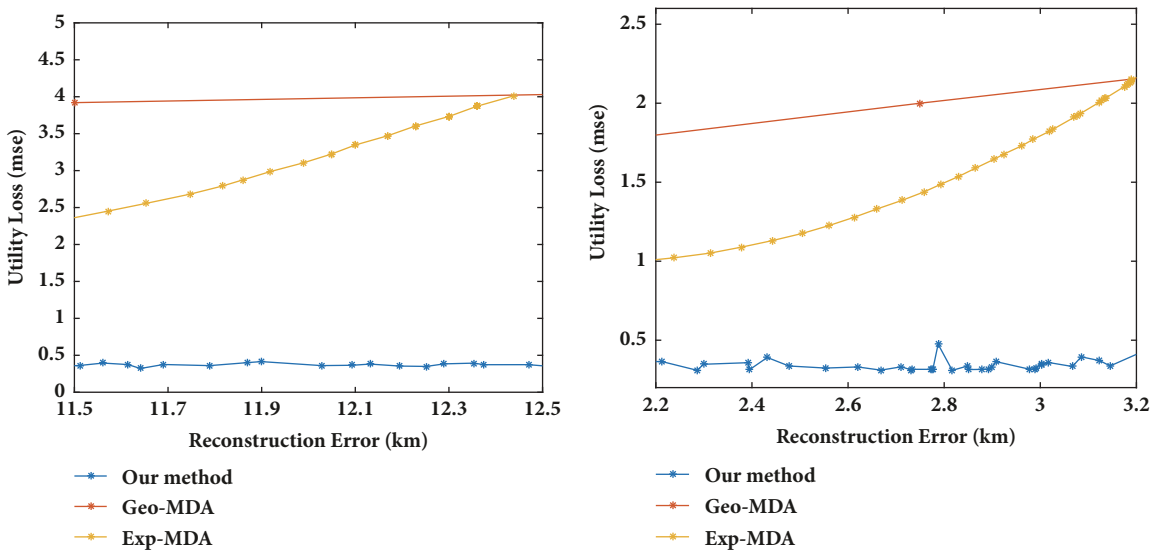


Figure 11: Actual examples of our method and the corresponding utility loss.



(a) MoMo dataset

(b) SFC dataset

Figure 12: The privacy-utility trade-off comparison.

of mobile network protocols [21] are all powered by human mobility trajectories. Some other services exploit the users' daily mobility datasets finding the mobility patterns and mining users' activities to provide useful extensive services [22]. Gonzlez et al. [23] discovered that the human mobility has spatiotemporal regularity, which indicates people are very likely to return to a few frequently visited locations. However these services [24] endanger users' security and privacy as the datasets mining algorithms can link the mobility datasets to a variety of sensitive information as studies in [25, 26].

5.2. Applications of Generative Adversarial Networks (GANs). Researches on GANs are mainly in two directions. One direction is about the variants of GANs, which is aiming to solve the problems of the original GANs. For example, WGANs [27, 28] and DCGANs [29] are proposed to improve the stability of training and to alleviate mode collapse.

The other research direction of GANs focuses on the applications of GANs, and most of such researches are in the area of image processing. Radford et al. [29] use convolutional neural networks to improve image processing capacity.

The authors in [30–32] design the GANs as a conditional architecture to generate higher quality images. Reed et al. [33] combine GANs with the natural language processing technology and propose the text-to-image generation.

5.3. Privacy-Preserving Methods for Mobility Datasets Releasing. Researches on privacy-preserving mobility datasets releasing [34, 35] are becoming popular, as the mobility datasets contain sensitive individual information [36–39]. One popular method in solving the privacy issue is releasing the statistics of the mobility datasets instead of the individual trajectories. For example, the French XData project [6] only reports the density of each region in the area, which could conceal the individual information. However, the recently proposed attack method in [2] shows that such aggregation method is not safe, and they propose an approach that could recover the individual trajectories from such aggregation by exploiting the uniqueness and regularity of human mobility.

Encrypting or encoding the mobility datasets before releasing is another research direction to protect the datasets. In [1], the authors add Laplacian noise to the data, which achieves ϵ-differential privacy. There also have been many researches following other principles [40]: position dummies [41], rumor spreading [42], data aggregation [43], spatial obfuscation [44], coordinate transformation [45], and position sharing [46]. Approaches following these principles could provide privacy protection, but the trade-off between the privacy-preservation ability and the utility loss is another main focus for them.

6. Conclusions

This paper investigates the density distribution privacy-preservation on mobility data. We design a deep learning framework based on GANs. To the best of our knowledge, this is the piece of paper employing GANs on data privacy-preservation. We train the generator and discriminator in GANs by random data and the original data and publish the data generated by the generator. Adversaries cannot easily recover the users' trajectories from the published density distribution. We conduct plenty of experiments on the real world datasets. It is demonstrated that our method performs better than the compared approaches on data utility and privacy-preservation.

Data Availability

The San Francisco Cabs data have been deposited in the CRAWDAD dataset and can be downloaded from https://crawdad.org/epfl/mobility/20090224 (2009). The MoMo mobile application data are from previously reported studies and datasets. The prior study has been cited at relevant places within the text as [10].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Dan Yin and Qing Yang contributed equally to this work.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China under Grant 61702132, Natural Science Foundation of Heilongjiang province under grant QC2017071, the Fundamental Research Funds for the Central Universities Grant No. HEUCFM 180603, and the China Postdoctoral Science Foundation No. 2018M631913.

References

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 901–914, ACM, Berlin, Germany, November 2013.

[2] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory Recovery From Ash," in *Proceedings of the the 26th International Conference*, pp. 1241–1250, Perth, Australia, April 2017.

[3] C. Dwork, "Differential privacy: a survey of results," in *Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, April 25–29, 2008. Proceedings*, vol. 4978 of *Lecture Notes in Computer Science*, pp. 1–19, Springer, Berlin, Germany, 2008.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*, pp. 2672–2680, Canada, December 2014.

[5] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[6] G. Acs and C. Castelluccia, "A case study: privacy preserving release of spatio-temporal density in Paris," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014*, pp. 1679–1688, ACM, August 2014.

[7] "Linear Programming and Extensions," *Students Quarterly Journal*, vol. 34, no. 136, p. 242, 1964.

[8] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[9] S. Vallender, "Calculation of the wasserstein distance between probability distributions on the line," *Theory of Probability & Its Applications*, vol. 18, no. 4, pp. 784–786, 1974.

[10] T. Chen, M. A. Kaafar, and R. Boreli, "The where and when of finding new friends: Analysis of a location-based social discovery network," in *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media, ICWSM 2013*, pp. 61–70, USA, July 2013.

[11] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, CRAWDAD dataset epfl/mobility (v. 2009-02-24)," Downloaded from https://crawdad.org/epfl/mobility/20090224, 2009.

[12] X. Liang, X. Zheng, W. Lv, T. Zhu, and K. Xu, "The scaling of human mobility by taxis is exponential," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 5, pp. 2135–2144, 2012.

[13] H. D. Rozenfeld, D. Rybski, J. S. Andrade Jr., M. Batty, H. E. Stanley, and H. A. Makse, "Laws of population growth," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 105, no. 48, pp. 18702–18707, 2008.

[14] W. Xu, Y. Shen, N. Bergmann, and W. Hu, "Sensor-Assisted Multi-View Face Recognition System on Smart Glass," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 197–210, 2018.

[15] Y. Shen, C. Luo, D. Yin, H. Wen, R. Daniela, and W. Hu, "Privacy-preserving sparse representation classification in cloud-enabled mobile applications," *Computer Networks*, vol. 133, pp. 59–72, 2018.

[16] B. Jiang, J. Yin, and S. Zhao, "Characterizing the human mobility pattern in a large street network," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 2, Article ID 021136, 2009.

[17] E. Agliari, R. Burioni, D. Cassi, and F. M. Neri, "Word-of-mouth and dynamical inhomogeneous markets: an efficiency measure and optimal sampling policies for the pre-launch stage," *IMA Journal of Management Mathematics*, vol. 21, no. 1, pp. 67–83, 2010.

[18] L. Hufnagel, D. Brockmann, and T. Geisel, "Forecast and control of epidemics in a globalized world," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 101, no. 42, pp. 15124–15129, 2004.

[19] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS '15)*, pp. 205–214, July 2015.

[20] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1-1, 2017.

[21] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: A new mobility model for human walks," in *Proceedings of the INFOCOM 2009, IEEE*, pp. 855–863, 855-863. IEEE. doi, 2009.

[22] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.

[23] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[24] X. Zheng, Z. Cai, G. Luo, L. Tian, and X. Bai, "Privacy-preserved community discovery in online social networks," *Future Generation Computer Systems*, 2018.

[25] A. LaMarca, M. Langheinrich, and K. N. Truong, *Pervasive Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[26] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "De-anonymization attack on geolocated data," *Journal of Computer and System Sciences*, vol. 80, no. 8, pp. 1597–1614, 2014.

[27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the International Conference on Machine Learning*, pp. 214–223, 2017.

[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

[29] A. Radford, L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, 2015.

[30] C. Ledig, L. Theis, F. Huszár et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 105–114, USA, July 2017.

[31] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, p. 5, 2018.

[32] T. Karras, T. Aila, S. Laine, and J. Lehtinen, *Progressive growing of gans for improved quality, stability, and variation*, 2017.

[33] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proceedings of the 30th Annual Conference on Neural Information Processing Systems, NIPS 2016*, pp. 217–225, Spain, December 2016.

[34] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems>," *EEE Transactions on Network Science & Engineering*, 2018.

[35] Y. Shen, H. Wen, C. Luo et al., "GaitLock: Protect Virtual and Augmented Reality Headsets Using Gait," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[36] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable Secure Computing*, vol. 15, no. 4, p. 1, 2018.

[37] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, "Location prediction on trajectory data: A review," *Big Data Mining and Analytics*, vol. 1, no. 2, pp. 108–127, 2018.

[38] L. Shi, Y. Wu, L. Liu, X. Sun, and L. Jiang, "Event detection and identification of influential spreaders in social media data streams," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 34–46, 2018.

[39] X. Zheng, Z. Cai, and Y. Li, "Data Linkage in Smart Internet of Things Systems: A Consideration from a Privacy Perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[40] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.

[41] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services (ICPS '05)*, pp. 88–97, IEEE Press, July 2005.

[42] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2017.

[43] Y. Huo, C. Yong, and Y. Lu, "Re-adp: Real-time data aggregation with adaptive w-event differential privacy for fog computing," *Wireless Communications and Mobile Computing*, pp. 1–13, 2018.

[44] C. Reynold, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving user location privacy in mobile data management infrastructures," in *Privacy Enhancing Technologies*, G. Danezis and P. Golle, Eds., vol. 4258 of *Lecture Notes in Computer Science*, pp. 393–412, Springer, Berlin, Germany, 2006.

[45] M. L. Yiu, C. S. Jensen, J. Møller, and H. Lu, "Design and analysis of a ranking approach to private location-based services," *ACM Transactions on Database Systems (TODS)*, vol. 36, no. 2, p. 10, 2011.

[46] F. Dürr, P. Skvortsov, and K. Rothermel, "Position sharing for location privacy in non-trusted systems," in *Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications, PerCom 2011*, pp. 189–196, IEEE, Seattle, Wash, USA, March 2011.