IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xx.xxxx/ACCESS.2021.DOI

Mobility-aware Differentially Private Trajectory for Privacy-preserving Continual Crowdsourcing

Guoying Qiu[1] and Yulong Shen[1], Member, IEEE
[1]School of Computer Science and Technology, Xidian University, Xi'an Shaanxi 710071, P.R.China

Corresponding author: Yulong Shen (e-mail: ylshen@mail.xidian.edu.cn).

This work is partially supported by the National Key R&D Program (Grant No.2018YFE0207600), National Natural Science Foundation of China (Grant No.U1736216, 61571352 and 61602364), Project of Hainan Province Key R&D Program (Grant No.ZDYF2019202).

ABSTRACT Participating in mobile services by synthesizing trajectories with consistent lifestyle and meaningful mobility as actual traces are the most popular way to protect location privacy. However, recent trajectory synthesizing techniques are still threatened by the information that the attacker inevitably obtains, such as the locations of the accepted tasks in the crowdsourcing application. With this information and the spatiotemporal correlation hidden in the user's mobility, the attacker can infer the user's actual location and even future behaviors. It remains open to defend against such inferential attacks in the continual crowdsourcing scenarios.

In this paper, we propose a mobility-aware differentially private solution, ConCrowd-DP, for achieving the privacy-preserving continual crowdsourcing application. Specifically, before starting the application, we first construct a spatiotemporal mobile model, STMarkov, to model the spatiotemporal correlation in users' mobility. Then, a perturbed location is generated for the user to participate in the crowdsourcing application, according to STMarkov and K-norm DP. Finally, we eliminate the privacy threat brought by the accepted task based on K-norm DP and Bayesian posterior theorem. With ConCrowd-DP in place, a mobility-aware differentially private trace is generated for the user to participate in the application continually. Extensive experiments with real-world datasets demonstrate that ConCrowd-DP guarantees the usability of the synthesized trajectory effectively, while providing the DP protection for defending against the inferential attacks which stem from the multiple accepted tasks.

INDEX TERMS Location privacy preservation; trajectory prediction; inferential attacks; differential privacy; continual location sharing.

I. INTRODUCTION

The increasing popularity of smartphones, mobile Internet, and cloud computing has pushed human society into a new perception and service model of the Internet of Things (IoT). The most notable phenomenon is the vigorous development of location-based service (LBS), and crowdsourcing is one of its typical applications. The crowdsourcing platform (service provider) launches location-related tasks and then recruits the mobile participants. The mobile users participate in the application and upload the location-related working reports for obtaining the rewards. Mobile crowdsourcing provides massive low-cost high-flexibility multi-source data. According to the perceived big data, the cloud platform provides users with various data services [1], [2]. As a new type of data perception and service model of IoT, mobile crowdsourcing and cloud platform has been widely implemented in many fields, including environmental monitoring, smart medical treatment, intelligent transportation, social services, etc.

A. MOTIVATION AND CHALLENGES

The mobile users participate in the application as shown in Fig. 2, where our ConCrowd-DP plays the role as a location anonymizer. We will describe its detailed dataflow in Section II-A. *The continual crowdsourcing means that mobile users participate in the application multiple times continually.*

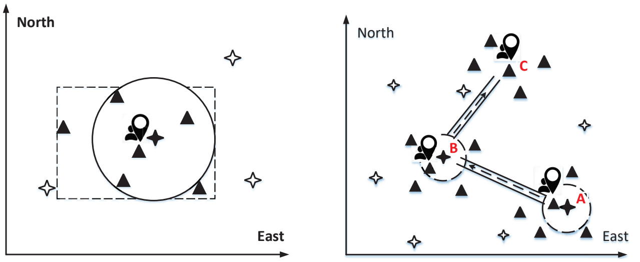
Privacy Threats Analysis. This paper considers the service platform as the adversary, assuming that he (she) is honest but curious. We do not take the cybersecurity issues into consideration in this paper. To improve the quality of service, the crowdsourcing platform tries to infer the user's real-time actual position or even the future mobile behavior based on the observed locations, such as the locations of the LBS queries and the accepted tasks. As the obtained data accumulates in the continual crowdsourcing application, the platform can even analyze the users' behavioral patterns and become a strong attacker [3]. Because of various inferential attacks that the adversary always launches, the information of the user's whereabouts directly or indirectly threatens the user's identity or other location-sensitive information [4]. Customized advertisements or other personalized services may be pushed to the user without permission. It may even result in serious threats to the safety of users' life and property.

Privacy-preserving Challenges for the Continual Crowdsourcing. Inspired by the above privacy-threat analysis, we know that the main challenges come from the following two aspects. The first one is the user-related location information observed by the platform in real-time application, such as the locations of queries and the accepted tasks. The second one is the spatiotemporal correlation hidden in the users' mobile behaviors which can be derived from the historical mobile data. Generally, the users always perturb their actual positions with existing privacy-preserving techniques, and participate in the application with the perturbed locations. As for the privacy risks brought by the perturbed locations of the users' queries, please refer to [5] for the detailed privacy-preserving solution.

To achieve the spatiotemporal correlation-based privacy protection, various techniques, such as statistical analysis, hidden Markov model, Bayesian theorem, conditional random field, etc., have been proposed to model the users' dynamic mobilities. Statistical analysis [6] only represents primary spatiotemporal correlations. The traditional Markov model [7] reflects the spatial transfer relationship, but not the temporal correlation, while the conditional random field [8] has poor compatibility with the existing location privacy-preserving mechanism (LPPM). Bayesian theorem considers the influence of a specific factor in the system in the form of conditional probability [9].

The differential privacy (DP) [10] provides provable privacy guarantees by limiting the individual's impact on the output. K-norm DP [11]–[13] constructs the geometric convex hull on the difference set of its anonymity set to formalize the DP's sensitivity, which quantifies the noise that needs to be introduced into the system. This method is suitable for the mobile application scenarios. More importantly, the sensitivity hull K-norm DP builds provides finer noise control than the l_1-norm sensitivity [5], which achieves better performance on a balance between location privacy protection and data availability. Literature [5] provides a differentially private solution to eliminate the privacy risks brought by the perturbed locations of the users' mobile queries. DPSense [4] applies it in a crowdsourcing scenario.

To the best of our knowledge, there is no work studying the privacy risks brought by the accepted tasks in crowdsourcing applications. The positions of the accepted tasks may weaken or even break the above-mentioned privacy protection. Consider the following examples. *We refer to the solid stars as the accepted tasks in Fig. 1, the hollow stars as the unaccepted tasks and the solid triangle as the locations of an anonymity set.*



(a) The location of the accepted task may reveal the user's actual location.

(b) The inferential attack brought by the locations of multiple accepted tasks.

FIGURE 1. The privacy risks brought by the accepted tasks in continual crowdsourcing.

I. *In a general crowdsourcing application, mobile users accept the tasks nearby, as shown in Fig. 1(a). According to an accepted task, we can draw a circle to represent the effective area where the user may appear, as described in the first inferential attack shown in Fig. 4(a). We know that the user's actual position must be included in the circular area, and may even be the closest one to the accepted task (i.e., the center of the circle) in the anonymity set. If the anonymity set is not well considered, the actual location may be the only position contained within the circular area.*

II. *In the continual crowdsourcing scenario, the multiple accepted tasks may even expose the user's future mobile behavior, as shown in Fig. 1(b). After a user takes tasks at locations A,B, the attacker may have obtained the user's actual locations at these two points. He (she) can then infer that the user may appear at the location C after analyzing the user's dynamic behavioral pattern, as stated in the inferential attack shown in Fig. 4(b).*

B. OUR CONTRIBUTIONS

With these issues in mind, we aim to design a privacy-preserving solution for the continual crowdsourcing application. In the solution, we should fully consider the user's mobile spatiotemporal association and eliminate the privacy risks brought by the accepted tasks.

The major contributions of this paper can be summarized as follows:

- STMarkov. We introduce the time factor into the traditional Markov model, proposing an improved spatiotemporal Markov, STMarkov. It perceives and models the spatiotemporal association hidden in the user's mobility. In our solution, STMarkov contributes the user's time-related transfer pattern and the steady-state distribution,

TABLE 1. Parameter settings

Parameter	Setting
U/u_i	The mobile user set / the i^{th} user
S	Location set
TP	Time partitions set
LBS	Location based service
x, y, z	Single locations
t	Timestamps
π	Steady state distribution
M	Transfer matrix
p_t	The probability distribution of the user at time t^{th}
p^-	The prior probability distribution
p^+	The posterior probability distribution
δ *set*	Anonymity set of DP
ε	Privacy degree of DP
K	Sensitivity hull
K_I	Sensitivity hull in the isotropic position
T_r/T	PIM transformation matrix

which outputs the locations that the user is most likely to visit at the corresponding time.

- ConCrowd-DP. We build the mapping relationship between the accepted-task's location and the location set according to K-norm DP, and eliminate the privacy risks caused by the accepted task based on the Bayesian posterior theorem. Finally, connected by STMarkov, the above methods form a closed-loop, formalizing the ConCrowd-DP model.
- DP trace. We execute our ConCrowd-DP iteratively for the user to participate in crowdsourcing application securely and continually, generating a mobility-aware DP trajectory. Our ConCrowd-DP achieves the privacy-preserving continual crowdsourcing application.

The rest of the paper is structured as follows. Section II clarifies the research problem of this paper. Section III introduces the system architecture of our solution. Our ConCrowd-DP is introduced in detail in Section IV. Section V analyzes the performance of our solution on privacy and complexity. Section VI demonstrates extensive experiments. The suitable application modes of our ConCrowd-DP and future work are discussed in Section VII. Section VIII reviews the related works, and Section IX concludes this paper.

II. PROBLEM FORMALIZATION

This section presents a general application system, discusses the adversary model, and describes our design goals to clarify our research problem further. Table 1 illustrates the settings of some important symbols.

A. PRIVACY-PRESERVING CROWDSOURCING SYSTEM

Participating in location-based crowdsourcing threatens the mobile users' location privacy, as analyzed in Section I previously. Therefore, the privacy-preserving system is widely needed in various crowdsourcing applications. Here, we present a general architecture of the privacy-preserving crowdsourcing in Fig. 2. To protect the location privacy, the mobile user sends the crowdsourcing platform a perturbed position or anonymity set to apply for the crowdsourcing task. Afterwards, he (she) corrects the reward's deviation caused by privacy protection according to the actual location. We describe its detailed dataflow as follows.

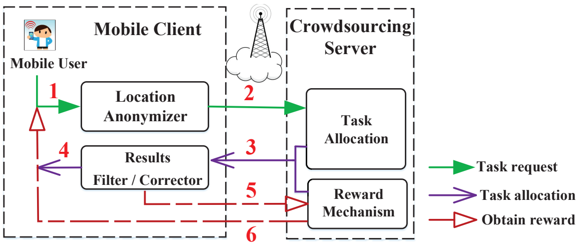


FIGURE 2. The application scenario. It aims to achieve the privacy-preserving crowdsourcing.

Dataflow. 1. The mobile user sends a location-related query; 2. The Anonymizer protects the query's position by generating a perturbed location or anonymity set, and sends it to the crowdsourcing platform; 3. The Server responds to the query with crowdsourcing tasks and corresponding rewards; 4. If an anonymity set is sent out by the mobile user in Step 2, the Result Filter picks out the exact response corresponding to the actual location; or the Result Corrector corrects the response's deviation, if the perturbed position is selected; 5. The accepted task or the reward's deviation is fed back to the Platform; 6. Finally, the mobile user obtains the corresponding reward.

Note. The anonymity set is composed of the actual position and several fake locations. The crowdsourcing platform responds one task to each location of the anonymity set. Therefore, the user only needs to feedback on the accepted task in Step 5. While, if the user sends out a perturbed location, the platform responds only one task and the corresponding reward. Then, the reward's deviation needs to be fed back.

B. THE ADVERSARY MODEL

This paper assumes the attacker is a normal adversary or even the crowdsourcing platform [14]. Generally speaking, the platform owns more users' information, such as the user's historical trajectory data (may be noisy and incomplete), than normal adversaries. It means that the platform has much stronger power to launch attacks. Therefore, we take the platform as the major defensive target in this paper. Typically, we assume that the platform is honest but curious. He (she) seeks to get the users' location information as accurately as possible to improve the service quality.

We exhibit a general architecture of the adversary model, as shown in Fig. 3. It presents the crowdsourcing platform's detailed knowledge and the way he (she) launches the attacks. During the application process, the platform observes the mobile user's location-related information, such as the service queries and the accepted tasks, and accumulates constantly. Therefore, he (she) can model the user's spatiotemporal mobile-behavioral pattern based on historical mobile data. According to the mobile pattern and the real-time information, such as the accepted tasks, the platform may launch the inferential attacks to infer the user's actual location and even

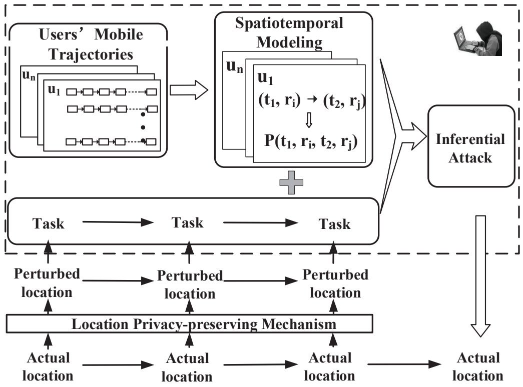


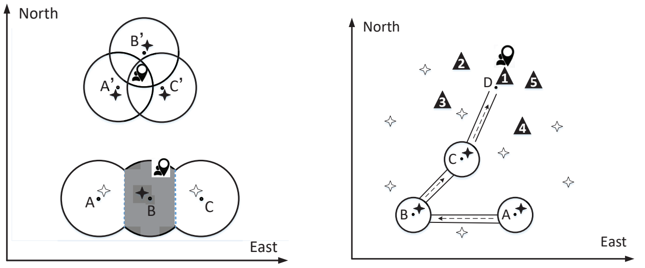
FIGURE 3. The adversary attack. It models the user's spatiotemporal mobility based on the historical traces, and launches the inferential attacks to infer the user's following action according to the locations of the tasks accepted by the user in the application process.

the following mobile behavior.

In the following, we illustrate the inferential attacks that the adversaries commonly adopt in mobile crowdsourcing, as shown in Fig. 4. Such as the inferential attacks based on the crowdsourcing elements and the spatiotemporal association in mobility. We refer to the solid stars as the accepted tasks and the hollow stars as the unaccepted tasks.

The inferential attacks based on crowdsourcing elements [4]. The attacker takes the location of an accepted task as the center and draws a circle with the maximum distance radius (i.e., the maximum distance for accepting tasks). The area within the circle represents the effective area for the user to participate in the crowdsourcing. It also means that the user must appear in this area while taking the task. The attacker can perform such inferential attack by combining the impacts of multiple tasks together, as shown in Fig. 4(a). The above subfigure presents the attack launched from the multiple tasks accepted at the same time. The intersection of the three circles is the area where the user is most likely to appear. The bottom subfigure corresponds to the scenario in which the user participates in the application with an anonymity set. The Server dispatches one task to each location of the anonymity set, and the user accepts one task but rejects the others. The shaded part represents the area where the user may appear.

The inferential attacks based on the spatiotemporal association in user's mobility [15]. In the mobile crowdsourcing, the strong attacker owns a large amount of users' mobile trajectories and can perform the dynamic behavioral analysis to infer the user's coming behavior. As shown in Fig. 4(b), the actual location 1 may be picked out by the adversary, although protected with the anonymity set $\{1-5\}$. Because it may be the location where the user is most likely to visit given the previously accepted tasks $\{A, B, C\}$, after performing the mobile behavioral analysis. Several methods can be used to perform such mobile behavioral analysis, including the Hidden Markov model [16], Conditional Random Field [8], etc.



(a) Inferential Attacks vs Crowdsourcing Elements

(b) Inferential Attacks vs Spatiotemporal Correlation of user's mobility

FIGURE 4. The inferential attacks in crowdsourcing scenarios.

C. OUR DESIGN GOALS

As described in the above adversary model and the examples in Section I-A, two challenges need to be overcome to achieve the privacy-preserving continual crowdsourcing. In the application at single timestamps, the attacker may launch the first inferential attack to discover the user's actual location based on the information of crowdsourcing elements. In a continual crowdsourcing scenario, he (she) may infer the user's following mobile behavior according to the multiple accepted tasks. We name it the second inferential attack.

For the application at single timestamps, Fig. 4(a) has shown that the ways of accepting multiple tasks at the same time and participating in the application with an anonymity set both bring serious location-privacy risks. Therefore, we suggest the mobile user participate in the application with a perturbed location, and the Server signs one task to each query. During this process, the platform observes no other positions except the accepted task and the perturbed location. According to the accepted task, he (she) can only have a circular area with a diameter of 10 km, assuming 5 km as the maximum distance for taking the tasks [4]. It is a fairly large area. Besides, the perturbed location should also be well designed to avoid bringing in the direct privacy risks to the user's actual location.

To defend against the second inferential attack, we should prevent the multiple accepted tasks from contributing to the spatiotemporal-mobility modeling which can be operated by the adversary in the continual-crowdsourcing scenario. For this purpose, we should eliminate the effect of the accepted task on the user's time-related distribution on location-set, which is the information carrier to model the user's dynamic behavioral pattern.

III. SYSTEM ARCHITECTURE

For achieving the privacy-preserving continual crowdsourcing, we propose our ConCrowd-DP solution. In this section, we present its system architecture and discuss its rationality.

A. ARCHITECTURE DESIGN

We first construct a STMarkov model to perceive the user's dynamic mobile patterns. Then, according to STMarkov and K-norm DP, a mobility-related perturbed location is generated for the user to participate in the crowdsourcing.

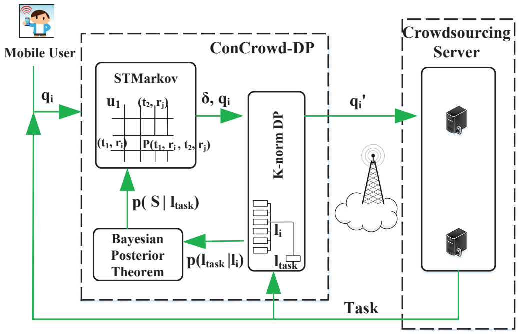


FIGURE 5. A glimpse of ConCrowd-DP.

After completing the task, we build a mapping relationship between the location-set and the accepted task based on K-norm DP, and further eliminate the impact of the accepted task on the user's distribution with the help of STMarkov and Bayesian theorem. Finally, these steps form a circular operation for the user to continually participate in the application by executing ConCrowd-DP iteratively.

B. A GLIMPSE OF CONCROWD-DP

We show the overview of ConCrowd-DP vividly in Fig. 5. Before a mobile user participates in the application, we first build the STMarkov model. When the user sends an LBS request q_i, STMarkov outputs the δ-location set from the user's time-related distribution on location-set (named the **user's distribution** for short). According to q_i and δ-location set, K-norm DP generates a perturbed position q_i', and sends it to the platform. The Server responds to the request and assigns the crowdsourcing task. Then, K-norm DP is reused to generate the mapping relationship between the position-set and the assigned-task's location, $p(l_{task}|l_i)$. The Bayesian posterior theorem updates the user's distribution based on the accepted task, $p(L|l_{task})$. Finally, the STMarkov transforms the user's posterior distribution into the prior values corresponding to the next moment, getting ready for the user to participate in crowdsourced applications again.

C. RATIONALITY ANALYSIS

To fully simulate the attacker's capability on the spatiotemporal-association perception, we first need to model the user's mobile behavioral patterns. As we all know, the traditional Markov model has its limitations. We propose a spatiotemporal Markov model, STMarkov, by bringing the time factor into the Markov model to tackle these limitations. The STMarkov models the user's dynamic behavioral pattern, constructing a spatiotemporal-transfer matrix and outputting a time-related steady-state distribution of the user on location-set. Its transfer matrix not only has the probability of the spatial transition but also records the time when the transition happens. Furthermore, the user's probability distribution on location-set can be driven from the time-related steady-state distribution at each corresponding time.

We redesign the PIM based K-norm DP [5], [11] and formalize the ConCrowd-DP, combining with the STMarkov and Bayesian posterior theorem. When a user sends an LBS query, STMarkov outputs a time-related δ-location set, composed of the places that the user is most likely to visit at the corresponding time. According to the DP principle, K-norm DP generates a perturbed location from δ-location set, making the perturbed position indistinguishable from the above places [5]. After the user completes the application, we need to eliminate the impact of the accepted task on the user's time-related probability distribution, getting ready for the user to participate in the application again. In order to control the system's computational complexity, K-norm DP is reused to build the mapping relationship between the location-set and accepted task. Then, Bayesian posterior theorem updates the user's distribution, taking the accepted task as the conditional probability. After STMarkov transforms the user's posterior distribution to be the prior distribution at the next corresponding time, the user can then participate in the application securely and continually.

To improve our solution's practical feasibility, we can further reduce the computational burden of the local client, by handing over the task of building STMarkov to the platform's server. The platform is our assumed adversary. He (she) can model the users' spatiotemporal mobility. It is for this reason that we incorporate this attack capability into our privacy-preserving solution. ConCrowd-DP constructs DP protection based on this function module. It anonymizes the perturbed position with the user's locations most likely to visit, making them indistinguishable. Therefore, it doesn't matter who builds the STMarkov. In other words, the STMarkov model produced by the Server will not threaten the privacy-preserving performance of ConCrowd-DP. Furthermore, there is another advantage of building STMarkov on the server-side. It can build different kinds of STMarkovs, building popular STMarkov based on public users' mobile data or producing a personal one according to the individual data. We will describe it in detail below.

In order to further expand the suitable scope of ConCrowd-DP, we design two ways of building the STMarkov. It can be trained based on the individual mobile data, creating a personal spatiotemporal transfer matrix. It can also be produced from the public users' data, constructing a popular transfer model. These two matrices make different senses. The personal one offers individual-related mobile patterns. The DP protection, based on it, introduces more considerable noise, generating much stronger perturbance. Therefore, the perturbed location is relatively far away from the actual site, leading to low data availability. On the other hand, the public transfer matrix provides DP with the popular mobile patterns. It introduces the weaker disturbances, resulting in the higher data availability. We will evaluate the different performances of these two matrices in the experimental section.

IV. MOBILITY-AWARE PRIVACY PROTECTION

This section presents the detailed solution of our ConCrowd-DP, as shown in Fig. 6. It describes how ConCrowd-DP models the user's spatiotemporal mobility and how to further

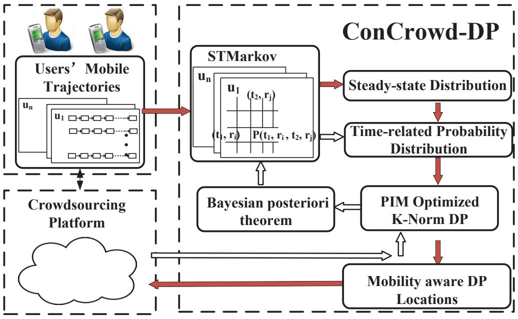

ConCrowd-DP
Users' Mobile Trajectories
u_n
u_1
STMarkov
u_n
u_1
(t_2, r_j)
(t_1, r_i)
P(t_1, r_i, t_2, r_j)
Steady-state Distribution
Time-related Probability Distribution
Crowdsourcing Platform
Bayesian posteriori theorem
PIM Optimized K-Norm DP
Mobility aware DP Locations

FIGURE 6. The detailed design of ConCrowd-DP.

eliminate the impact of the accepted tasks.

We divide it into three parts. The first one is STMarkov's spatiotemporal-mobility perception. The second one is the generation of the perturbed location for participating in the application at a specific moment, as shown in the process marked by the solid arrows in Fig. 6. The third one is eliminating the privacy risk caused by the accepted task after completing the participation, and preparing for the next participation, as shown in the process marked by the hollow arrows in Fig. 6.

A. SPATIOTEMPORAL MOBILITY PERCEPTION, STMARKOV

We propose the STMarkov model for perceiving users' mobile behaviors, as shown in Fig. 7.

1) A New Markov Transfer Model

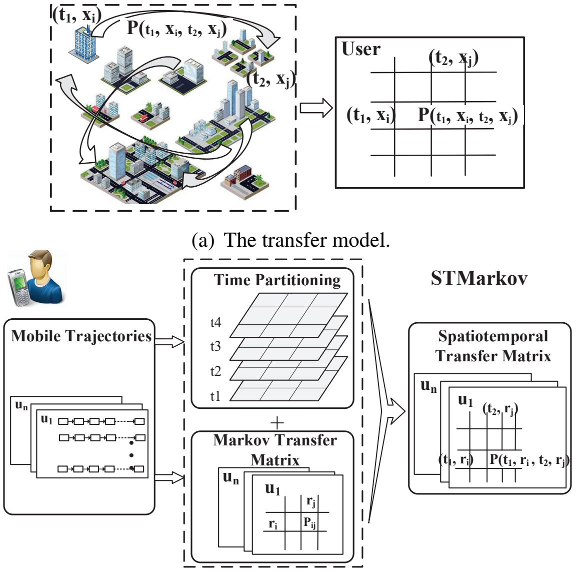
To break the Markov model's limitations, we redesign it by introducing the time factor into the traditional one and propose STMarkov, a time-related spatial transfer model. Here, we characterize the time factor by making use of the time-partitioning concept according to LPM^2 [17].

We add the time element into the state variable of the Markov model in the manner of time-partitioning. As shown in Fig. 7(a), STMarkov replaces the state variable (x) with (t,x), representing the state of the location x with the corresponding timestamp t. The transfer model turns out to be the transfer probability from the state (t_1,x_i) to (t_2,x_j), replacing P_{ij} with $P(t_1,x_i,t_2,x_j)$. A spatiotemporal transfer matrix is generated. It not only represents the probability of spatial transition, but also records the time when the transition happens.

2) STMarkov: Spatiotemporal-mobility Modeling

To build the STMarkov model, we need to do some preparatory work. Determine the target geographic area and time-period. That is, to determine that in what area and within what time-period the user's transfer mobility occurs, which we want to model.

We first select an active area and collect the involved PoIs, S. The area can be set in multiple ways, such as business circles, administrative regions, etc. Second, according to the requirements of mobility modeling, an appropriate time-period is picked out as the target time-period. Finally, we should further determine the granularity for dividing the effective time-period and area, forming a reasonable time-partition set, TP, and geographic grids, as shown in Fig. 7(b). The granularity of the time-partitioning and area dividing reflects the accuracy of the mobility analysis. We refer n,m to represent the sizes of the time-partition set, TP, and the grids, S, respectively. Therefore, the transfer matrix's scale is determined to be $(n*m)*(n*m)$.



(a) The transfer model.

(b) Spatiotemporal-mobility modeling.

FIGURE 7. STMarkov. Spatiotemporal-mobility perception.

After the above preparations, we construct the mobile model, STMarkov. However, an inevitable fact is that the training data is incomplete, and we need to estimate the missing data. We refer TT to represent the real training trajectory set and ET to describe the hypothetical complete dataset. Then they have the following probabilistic relationship:

$$P(M \mid TT) = \sum_{ET} P(M, ET \mid TT). \tag{1}$$

However, it is infeasible to sample directly from $Pr(M,ET \mid TT)$. The computation increases exponentially as the trajectory becomes longer. Therefore, we introduce one kind of Monte Carlo sampling method, Gibbs Sampling [7], for the unbiased training of STMarkov. The above joint distribution is achieved in polynomial time by sampling from the following two conditional distributions iteratively. An iteration execution of the following sampling pairs (M^l, ET^l) yields a pair of sampling data.

$$M^l \sim P\left(M \mid ET^{l-1}, TT\right); \tag{2}$$

$$ET^l \sim P\left(ET \mid M^l, TT\right). \tag{3}$$

Algorithm 1 shows its sampling process. We can pre-set the maximum number of iterations or approximate value

Algorithm 1 Two-dimensional unbiased training for the STMarkov model

```
Input  Data Set $S = (M, ET)$. $M$ the transfer matrix, $ET$ hypothetical complete dataset.
Output  Sample Set $S' = (m, et)$.
Initialize $M_0 = m_0$, $ET_0 = et_0$;
Set $Max(SamplingNum)$, loop sampling;
for $i = 0$; $i \le Max(SamplingNum)$; $i++$ do
    Sample $ET^{i+1} \sim P(ET \mid M^i)$ as shown in Eq. 2;
    Sample $M^{i+1} \sim P(M \mid ET^i)$ as shown in Eq. 3;
end for
Return the sample set $S' = (M, ET)$.
```

between the last two samplings as the condition for ending the iteration.

STMarkov provides the ConCrowd-DP with the user's time-related probability distribution on location-set (the user's distribution) and the spatiotemporal transfer matrix. The user's distribution constructs the anonymity set for K-norm DP, protecting the actual location with the locations that the user is most likely to visit. While the transfer matrix transforms the user's distribution into the values corresponding to the following timestamp.

B. DP PROTECTION FOR CROWDSOURCING APPLICATION AT SINGLE TIMESTAMPS

For the crowdsourcing application at single timestamps, we need to perturb the actual positions based on the DP principle and generate the perturbed locations. Here, we adopt the improved DP solution, the PIM based K-norm DP [5], for this propose. It synthesizes the DP noise based on the δ-location set, which is driven from the user's probability distribution on location-set output by the STMarkov model.

Definition 1 δ-location Set [5]. Let p_t be the prior probability of a user on-location set at timestamp t. δ-location set is a set containing minimum number of locations that have prior probability sum no less than 1 -δ.

$$\Delta X_t = min\{x_i | \sum_{x_i} p_t[i] \ge 1 - \delta\}, \tag{4}$$

where x_i denotes a location, $p_t[i]$ refers to the probability with which the user visits the location x_i.

Definition 2 K-norm (Sensitivity Hull). In the two-dimensional mobile scenarios, the K-norm is equivalent to the sensitivity hull. It can be formulized as follows.

$$K = Convex(\Delta x),\ \Delta x = (x_1 - x_2), x_1, x_2 \in \delta, \tag{5}$$

where x_1, x_2 are any two of the locations in δ set, and Δx is the distance vector between x_1 and x_2. Then, K-norm is a convex hull formed by the distances.

In order to further control the DP noise, PIM-based K-norm DP also implement an isotropic transformation to the above-mentioned sensitivity hull.

Corollary 1 Isotropic Transformation [5]. For any convex body K in R^2, any integer $p \ge 1$, there is an absolute constant c such that if $\varsigma \ge 4cp^2$ with probability at least 1˘2^{-p}, $K_I = T_r K$ is in isotropic position.

$$T_r = \left(\frac{1}{\varsigma} \sum_{i=1}^{\varsigma} y_i y_i^T \right)^{-\frac{1}{2}}, \tag{6}$$

where $y_1, y_2, \cdots, y_l$ are independent random points uniformly distributed in K.

The major execution process of the PIM based K-norm DP is described as follows, which is also shown in the process marked by the solid arrow in Fig. 6.

- First, we extract the δ-location set from the user's time-related probability distribution, and construct the sensitivity hull of the δ-location set.
- Second, we train a transformation matrix, T_r, and transform the sensitivity hull onto the isotropic position, $T_r * K$, forming the isotropic sensitivity hull.
- Third, two random samplings are performed respectively in the isotropic sensitivity hull and the Gamma function, obtaining the samples z and r.
- Finally, the perturbed location is generated according to the Eq. 7. The perturbed position satisfies DP in δ-location set [5] and is indistinguishable from the locations of δ-location set.

$$x_t^* = x_t + r * T_r^{-1} * z. \tag{7}$$

The mobile user sends the perturbed location to the crowdsourcing platform, and the Server dispatches one task as the response. The user chooses to accept or reject the task based on a certain criteria. If he (she) takes and completes the task, he (she) can get the corresponding reward.

C. DP PROTECTION FOR THE CONTINUAL CROWDSOURCING

In the following, we seek to eliminate the privacy risks caused by the accepted tasks for achieving the privacy-preserving continual crowdsourcing. To achieve the above goal, the main challenges come from the following two aspects. One is to formulize the mapping relationship between the location set and the accepted task. The second is to eliminate the privacy risk brought by the accepted task and prepare for participating in the application in the following time-partition.

The releasing probability for the location of the accepted task. As shown by the process marked by the hollow arrows in Fig. 6, we need to generate the mapping relationship between the position set and the accepted task. To maintain the consistency of system operation, we still seek to achieve the DP protection for hiding the location of the accepted task in the δ-location set. We formalize the mapping probability from the location set to the position of the accepted task, referring to it as the releasing probability of the accepted task. This releasing probability is formalized according to the principle of K-norm Mechanism, taking the location of the accepted task as the output.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3058211, IEEE Access

Definition 2 K-norm Mechanism [11]. Given a linear function $F : R^N \to R^d$ and $\varepsilon > 0$, we let $K = FB_1^n$ and define the mechanism $\mathrm{KM}(F,d,\varepsilon)$, so that each measure is given by the probability density function defined over R^d.

$$p(o) = Z^{-1} exp(-\varepsilon \|Fx - o\|_K),$$
$$Z = \int_{R^d} exp(-\varepsilon \|o - Fx\|_K)da = \Gamma(d+1) * Vol(\varepsilon^{-1}K). \tag{8}$$

here, Z denotes the normalized constant. The mechanism $KM(F,d,\varepsilon)$ is K-norm Mechanism, and satisfies Differential Privacy.

As shown in Eq. 8, the releasing probability of the output location is exponentially related to the distance between the output location x^* and the actual location x, i.e., $\Delta x = (x^* - x)$. However, there is a prerequisite for the establishment of this formula. That is, the distance Δx must be within the range of K-norm.

In the following, we analyze the distance relationship between the Δx and K-norm. When $\Delta x = (x^* - x) \in K$, K-norm mechanism provides DP protection effectively, as shown in its definition. We prove that this situation accounts for the vast majority in the total by the experimental results. When $\Delta x = (x^* - x) \notin K$, we need a surrogate for Δx in K-norm to seek an approximative DP protection. The point closest to Δx in K-norm, $\eta_{agent} = Nearby(\Delta x, K)$, can be used as the surrogate. We will evaluate this drift phenomenon in the experiments.

Besides, we also transform the K-norm to the isotropic position to improve the DP performance. Then, we formalize the releasing probability of the location of the accepted task as follows.

$$p(x^*|x) = \begin{cases} \dfrac{\varepsilon^2}{2 * Area(K_I)} exp(-\varepsilon \|T * (x^* - x)\|_{K_I}), & \Delta x \in K \\ \dfrac{\varepsilon^2}{2 * Area(K_I)} exp(-\varepsilon \|T * (\eta_{agent})\|_{K_I}), & \Delta x \notin K. \end{cases} \tag{9}$$

Eliminate the privacy risk brought by the accepted task. As shown by the process marked by the hollow arrows in Fig. 6, the Bayesian posterior theorem is then introduced to update the user's distribution, taking the accepted task as the condition. According to the principle of the Bayes-Markov inference model, there are two steps required for eliminating the privacy risk brought by the accepted task. First, according to the releasing probability of the accepted task $p(x^*|x)$, we update the probability distribution of the user on-location set based on the Bayesian posterior theorem. It transforms the prior probability distribution $p_t^-(x)$ to be the posterior probability distribution $p_t^+(x|x^*)$, taking the accepted task as the condition.

Finally, the user's posterior distribution need to be transformed into the prior distribution in the following time-partition. For achieving this purpose, the spatiotemporal transfer matrix, $M(t, x_t, t+1, x_{t+1})$, generated by the STMarkov, is recalled back. It transforms the user's posterior probability distribution in the current time-partition, $p_t^+(x|x^*)$, to be the prior probability distribution in the following time-partition, $p_{t+1}^-(x|x^*)$, i.e., $p_{t+1}^-(x) = p_t^+(x) * M(t, x_t, t+1, x_{t+1})$. The probability $p_{t+1}^-(x|x^*)$ is the target distribution which needs to be prepared for the user to participate in the application in the following time-partition.

The process marked by these hollow arrows can form a loop operation combined with the process characterized by the solid arrows as shown in Fig. 6. Performing circular operations iteratively can generate a mobility-aware perturbed location sequence, allowing users to participate in the crowdsourcing securely and continually. Up to now, we have eliminated the privacy risk caused by the accepted tasks and achieved the privacy-preserving continual crowdsourcing application.

V. PERFORMANCE ANALYSIS

In the following, we analyze the performance of our ConCrowd-DP in terms of privacy preservation and computational complexity.

A. PRIVACY ANALYSIS

Here, we analyze its privacy-preserving characteristics and the perturbation degree caused by the introduced noise.

Theorem 1. ConCrowd-DP satisfies ε-DP in the space of the isotropic sensitivity hull, K_I.

Proof. In the mobile crowdsourcing scenario, the linear mapping is $F : R^2 \to R^2$. Given $\varepsilon > 0$ and $K_I = TB_1^d$, the releasing probability $p(x^*|x)$ satisfies the condition of K-norm mechanism defined in Eq. 8. So the mechanism $K(TB_1^d, d = 2, \varepsilon)$ defined in this paper satisfies ε-DP in the space of K_I. That is, ConCrowd-DP provides ε-DP protection in the space of the isotropic sensitivity hull K_I. For the detailed proof of the DP in K-norm mechanism, please refer to Literature [11].

Theorem 2. ConCrowd-DP satisfies ε-DP on δ-location set.

Proof. The isotropic transform is a unique linear mapping: $R^2 \to R^2$, and it is easy to replace the isotropic sensitivity hull K_I with the original sensitivity hull K. That is, there is also a corresponding releasing probability in the space of the original sensitivity hull of K. It is almost in the same form as the releasing probability in Eq. 9. Therefore, ConCrowd-DP is ε-DP in the space of K. That also means ConCrowd-DP satisfies ε-DP on the δ-location set.

Differential privacy is a perturbation-based privacy-preserving method. Therefore, we evaluate the error caused by the introduced noise from the ConCrowd-DP and give the upper and lower bounds in the following.

Theorem 3. The introduced noise brought by ConCrowd-DP causes the errors with Lower Bound $\Omega\left(\frac{1}{\varepsilon}\sqrt{AREA(K)}\right)$, and Upper Bound $O\left(\frac{c}{\varepsilon}\sqrt{AREA(K)}\right)$, where c is the approximation degree of the isotropic position.

Proof. In this paper, we seek the DP protection in a two-dimensional isotropic plane. Thus, we evaluate the error brought by the ConCrowd-DP in the space of K_I, by making use of the results from Literature [11].

Lower Bound. To answer a linear function $F : R^n \rightarrow R^d$ under the definition of DP, every ε-DP mechanism must have

$$ERROR \geq \Omega\left(\frac{d}{\varepsilon}\left(\frac{VOL(K)}{VOL(B_2^d)}\right)^{d^{-1}}\right), \tag{10}$$

where $VOL(\boldsymbol{.})$ is the volume and B_2^d is the unit l_2 ball.

Upper Bound. Let $\varepsilon > 0$. Suppose $F : R^n \rightarrow R^d$ is a linear mapping such that $K = FB_1^n$ is in c-approximately isotropic position. Then, the K-norm mechanism is ε-DP with the error at most $O(cL_K)\boldsymbol{.}LB(F,\varepsilon)$.

In a two-dimensional isotropic plane, we can obtain the Lower Bound: $\Omega\left(\frac{1}{\varepsilon}\sqrt{AREA(K_I)}\right)$ and Upper Bound: $O\left(\frac{c}{\varepsilon}\sqrt{AREA(K_I)}\right)$. After transforming back into the original space, it is easy to show that $AREA(K) = det(T_r^{-1}) * AREA(K_I)$. Therefore, the Lower Bound and Upper Bound of **Theorem 3** are obtained.

In summary, the perturbed location generated by our ConCrowd-DP is geographically indistinguishable from the positions where the user is most likely to visit after completing the previously accepted tasks. The perturbation degree caused by the introduced noise is proportional to the area of the K-norm constructed by the δ-anonymity set, and inversely proportional to the DP's privacy degree.

B. COMPLEXITY ANALYSIS

In the following, we analyze our ConCrowd-DP's complexity from the perspectives of the implementations at single timestamps and participating in the application continually.

Theorem 4. In the implementations at single timestamps, our ConCrowd-DP takes $O\left(nmd + klog(h) + h^2log(h)\right)$ or $O\left(klog(h) + h^2log(h)\right)$ time. Here, n, m, d are the numbers of time-partitions, divided geographic grids and iteratives of Gibbs sampling, and k, h refer to the size of ΔX and number of vertices on $Conv(\Delta X)$.

Proof. The implementation of our Concrowd-DP contains the following three core components. STMarkov perceptives the user's mobility, PIM-based K-norm DP generates the perturbed location, and the Bayesian posterior theorem eliminates the impact of the accepted task. We denote their computational complexities as $T(STMakov)$, $T(K - norm\ DP)$ and $T(Bayes)$, respectively.

$T(STMakov)$. Because of the data's incompleteness, we train the STMarkov model by implementing the Gibbs Sampling method in two dimensions, i.e., the target matrix M and the estimated completion ET. Both of their matrices' scales are $(nm) * (nm)$ and we produce the samples for each row separately. Therefore, this operation needs to be executed $2nmd$ times, i.e.,$T(STMakov) \sim O(nmd)$.

$T(K - norm\ DP)$. As described in Section IV-C, we have $T(K - norm\ DP) \sim T(PIM) + O(1)$. Meanwhile, $T(PIM) \sim O(klog(h) + h^2log(h))$ [5]. Therefore, $T(K - norm\ DP) \sim O(klog(h) + h^2log(h))$. Besides, Bayesian posterior takes $2m$ operations, i.e., $T(Bayes) \sim O(m)$.

In addition, the mobile user only needs to train the STMarkov model before participating in the application for the first time, which is not needed at other timestamps. Therefore, **Theorem 4** is obtained.

Theorem 5. When the mobile user participates in the application continually, our ConCrowd-DP takes $O\left(nmd + N[klog(h) + h^2log(h)]\right)$ time, where N is the times of participation.

Proof. When the mobile user participates in the application multiple times continually, our ConCrowd-DP first trains the STMarkov matrix, and then generates the perturbed locations for the user to participate in the application, iteratively. Based on the analysis in Theorem 4, we have **Theorem 5** easily.

VI. EXPERIMENTAL EVALUATION

In this section, we present the experimental evaluation results of ConCrowd-DP, including the accuracy of the users' spatiotemporal-mobility perception and the performance of the DP protection.

A. DATASET AND SETUP

We adopt the Geolife dataset [18]–[20] conducted by Microsoft Research Asia to verify the performance of ConCrowd-DP. Microsoft Research Asia organizes hundreds of people and publishes the dataset with a large number of trajectories covering a distance of millions of kilometers, after years of accumulation. The data in Geolife covers more than 30 cities in China and even some cities in USA and Europe. But most of them are recorded in Beijing. We take the area within the Third-ring road of Beijing as the target. It has an area of $12.8 * 12.8$ [km*km], which is divided into $20 * 20$ equal sizes.

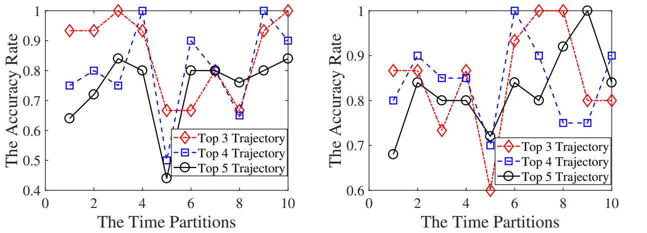
The platform is assumed to publish the crowdsourcing tasks at the working time on weekdays. We should perceive and model the user's mobile behavior during the time-period from $8:00$ am to $6:00$ pm. The time-period is divided into ten time-partitions with the time granularity of an hour in each slot. We sample time-location pairs from the trajectories in every 10 minutes and construct **two** training datasets, as the comparative experiments. We simulate the path of the accepted tasks with the route most often taken by the user. The experiments on each training set are repeated 20 times.

To fully verify the DP protection performance offered by ConCrowd-DP, we also need to construct two kinds of databases, a personal database and a public database. We randomly select a person from individuals with relatively abundant data, regarding his dataset as a personal database. The remaining dataset can be considered as a public database.

B. MOBILITY PERCEPTION

1) Evaluation Criteria

We evaluate the perceived performance of the STMarkov model by taking the statistic method as a baseline. We



(a) The accuracy ratio of each time partition from the first training set.

(b) The accuracy ratio of each time partition from the second training set.

FIGURE 8. Dynamic Accuracy Ratio on the Time-sequence.

extract Top n PoIs from the user's time-related probability distribution in the descending order in each time-partition, forming a perceptual trajectory. Meanwhile, we divide the user's historical mobile data set according to time partition. Then, Top n PoIs, with the highest user's access rate, are picked up from the sub-dataset corresponding to each time-partition, composing a statistical trajectory. We take it as a comparison of the perceptual trajectory. We select the evaluation indicators as follows.

- **Dynamic Accuracy Ratio on the Time-sequence.** In each time-partition, we compare the number of PoIs, n', that the above two trajectories have in common, and construct the accuracy rate of the STMarkov's mobility perception in each time-partition, $\zeta = n'/n$. We refer to the accuracy ratio on time-sequence as the similarity of the above two trajectories.
- **P-L Indicator.** We define a $P-L$ indicator to indicate the probability weights of ToP n PoIs. First, we note P probability to depict the probability occupied by Top n PoIs among the total possibility, p^*, in each corresponding time-partition. Further more, $P-L$ indicator divides the P probability by the average chance, $P-L = P*L/(p^*)$, where L is the size of the location set composed of all locations where the user has appeared in the corresponding time-partition.
- **Proportion Distribution of the Time-partitions on the Coverage Ratio.** We refer to this indictor as the proportion of the time-partitions, TP_t, in the total, TP_{total}, corresponding to a certain coverage rate, c.

$$P = \{p_c\},\ p_c = TP_c/TP_{total},\ \ c \in \{1,2,\cdots,n\}, \quad (11)$$

where the coverage ratio c refers to the number of PoIs shared by the above two comparative tracks in the same time-partition.

2) Dynamic Accuracy Ratio on Time-sequence

We refer to the accuracy ratio as the proportion of the PoIs shared by the two comparative trajectories. Fig. 8(a) 8(b) present the accuracy ratios of each time partition in the two groups of comparative experiments, respectively. They compare the perceptual results obtained when n is taken from $\{3, 4, 5\}$ respectively. Fig. 10(a) shows the average accuracy rate obtained when n is taken from 1 to 5.

Fig. 8(a) 8(b) indicate some spike-wave phenomena. It is determined by the user's behavioral pattern. During some time-partitions, the user's behavioral pattern is relatively fixed. The predicted results have higher accuracy ratios. However, the situation is just the opposite of some other time-partitions. Such as the user's mobile behaviors at noon or after getting off work in the afternoon. Uncertain user's behaviors result in lower predicted accuracy ratios.

3) Proportion Distribution of the Time-partitions on the Coverage Ratio

Fig. 9 presents the proportion distribution of time-partitions as defined above. It indicates that time-partitions are distributed over the high coverage ratio with a large probability. For example, Fig. 9(a) shows that the proportion of time partitions reaches 96%(100%) on the coverage ratio of 2 and 3. The proportion accounts for 94%(89%) for the coverage ratio of 3 and 4, as shown in Fig. 9(b).

Besides, it also indicates that the proportion of time-partitions with the full coverage decreases as n increases, as shown in Fig. 9(b) 9(c).

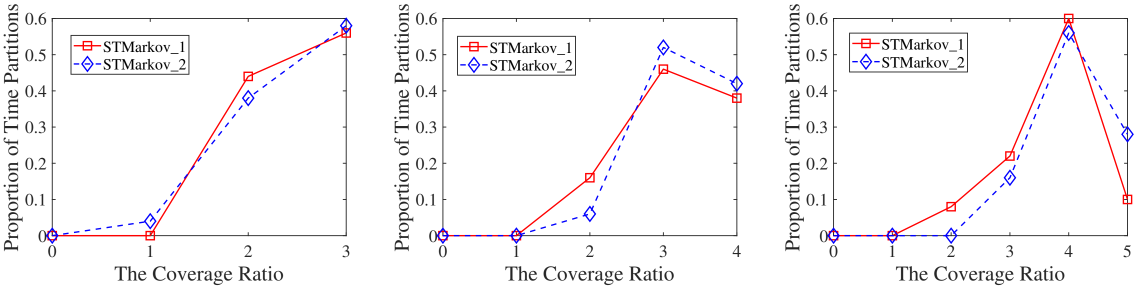
4) Comparative Experiments

We compare the experimental performances of our STMarkov, traditional Markov, and statistic method, in terms of the average accuracy and the indicator of $P-L$. The traditional Markov models the user's mobile behavior based on the Markov Chain. The statistic method reflects the historical fact of user mobility.

The Average Accuracy. Fig. 10(a) shows the performance of the average accuracy obtained from STMarkov and Markov Chain, respectively. We first analyze the corresponding curves of STMarkov. When n takes the value 1, the accuracy ratio is relatively low. Then, the accuracy ratio increases as n becomes larger. When n takes the value of 3, the average accuracy ratio reaches the highest value, up to 85%. After that, it begins to decrease as n increases. When n takes the value 5, the average accuracy rate falls to 82.4%. The phenomenon that the accuracy ratio decreases as n increases is also reflected in the experiments of Fig. 8(a) 8(b).

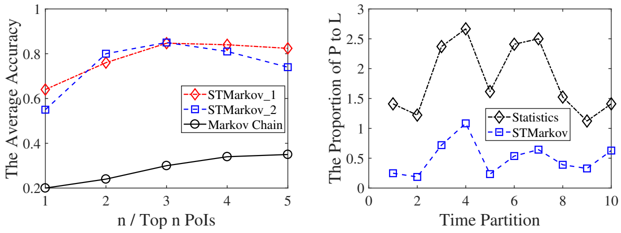
To explain the above phenomenon, we had better start from the definition of the Top n PoIs. Top n PoIs are extracted from the probability distribution of the user's appearance in a decreasing manner. Therefore, the initial increasing trend is due to the introduction of high-frequency PoIs. These PoIs successfully improve the coverage rate of the PoIs shared by the perceptual trajectory and the statistic trajectory. The subsequent decreasing trend is due to the introduction of low-frequency PoIs.

In comparison, we find that the accuracy of the Markov Chain is low. It is because it only records the sequence of spatial transition, but does not model the time-varying mobile pattern. That is, the user's transfer probability is time-varying.



(a) The top 3 temporal PoI trajectory (b) The top 4 temporal PoI trajectory (c) The top 5 temporal PoI trajectory

FIGURE 9. The proportion distribution of the time partitions on the coverage rate.



(a) The average accuracy of the predicted Top n PoIs. (b) P-L Indicator.

FIGURE 10. The Comparative Experiments on the Average Accuracy and P-L Indicator.

P-L Indicator. Fig. 10(b) presents the comparative performance of our STMarkov and Statistic method on the $P-L$ indicator. The two curves have a similar trend, and derive out a consistent conclusion with the result of the dynamic accuracy ratio. That is, the probability weights in the $P-L$ curves are relatively high during working time-periods, but low at other time-partitions, as shown in Fig. 10(b). The reason is that the places where the user works are relatively fixed, while the moving behaviors in other time-periods have more uncertainty.

You may be confused that the curve of the predicted trajectory is below, as shown in Fig. 10(b). Does it mean our prediction results are not good? Actually, no. It is a little necessary sacrifice for the global optimization. The steady-state probability distribution of STMarkov generalizes the probability over the entire (TP,S) state space.

However, we adopt STMarkov to perform mobility perception, rather than the statistic method. It's because the statistic method does not take the spatiotemporal association into consideration, which is hidden in a user's mobility. The absence of the spatiotemporal association makes it impossible to predict the user's mobile behavior in real-time accurately.

C. DP PROTECTION

We go deep into the ConCrowd-DP mechanism and evaluate its performance by taking the original K-norm DP [5] as a baseline. The parameters δ and ε are considered as the variables to explore the DP performance of ConCrowd-DP, including how δ-set changes, how often drift happens, and how accurate the perturbed location is. Besides, we also construct experiments on personal and public datasets respectively to ensure the objectivity and accuracy of the experimental results.

Variables. Besides the training dataset, the input of ConCrowd-DP requires the user to pre-provide two parameters, δ, and ε.

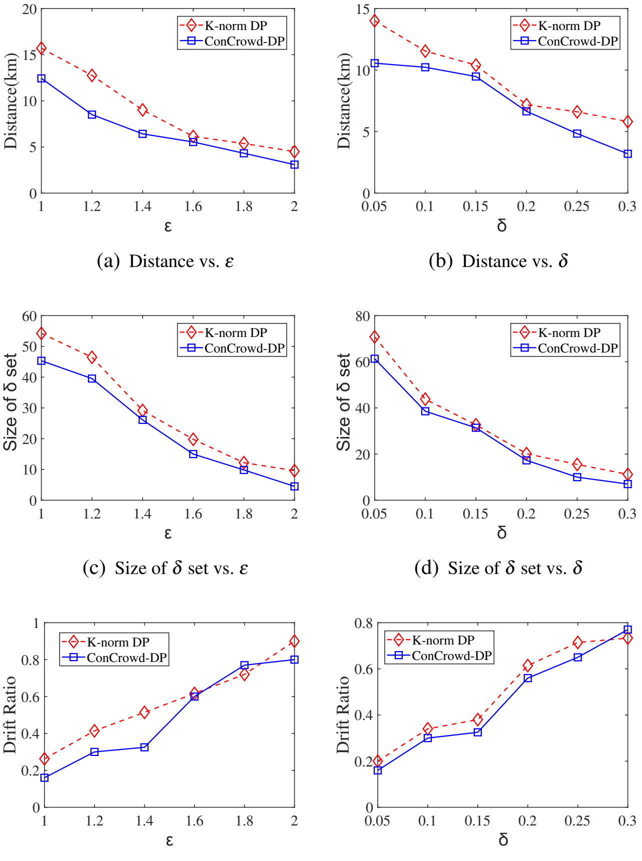
- δ. The parameter δ is required for constructing the δ-location set. The δ-location set consists of the potential locations where the user is most likely to appear. It is used to provide DP protection for the predicted PoIs.
- ε. It is introduced by the definition of K-norm DP. It indicates the privacy degree of DP protection.

Metrics. We select the following three internal indicators to demonstrate the performance of our experimental results, i.e., the distance between the perturbed location and the predicted PoI, the size of δ-location set, and the drift ratio.

- **Distance between the perturbed location and the actual PoI.** This kind of distance reflects the degree of the error caused by the introduced noise. It is a general usability metric in the mobile privacy-preserving scenario.
- **The size of δ-location set.** ConCrowd-DP generates DP protection based on the δ-location set, which consists of the potential locations. Therefore, it indicates the scope of privacy protection, which performs as an anonymity set. The size of δ-location set is determined by both the parameter δ and the probability distribution of the user on-location set.
- **Drift ratio of the accepted tasks.** We refer to 'drift' to indicate that the vector, formed by the actual position and the location of the accepted task, exceeds the range of K-norm. In this scenario, ConCrowd-DP just provides approximate DP. Thus, the drift ratio calculates the proportion of such situations among the total accepted tasks.

The impacts of ε. The parameter ε is introduced by the K-norm DP and indicates the privacy degree of the DP protection as described above. A larger ε value means a more relaxed privacy requirement. In this situation, less noise is introduced, and the perturbed location is closer to the protected object. That is, the output result is more accurate. Otherwise, the situation will be just the opposite.

- **Distance vs ε.** Fig. 11(a) 12(a) present the impacts of the parameter ε on the distance between the released location and the actual position. It shows the distance



(a) Distance vs. ε

(b) Distance vs. δ

(c) Size of δ set vs. ε

(d) Size of δ set vs. δ

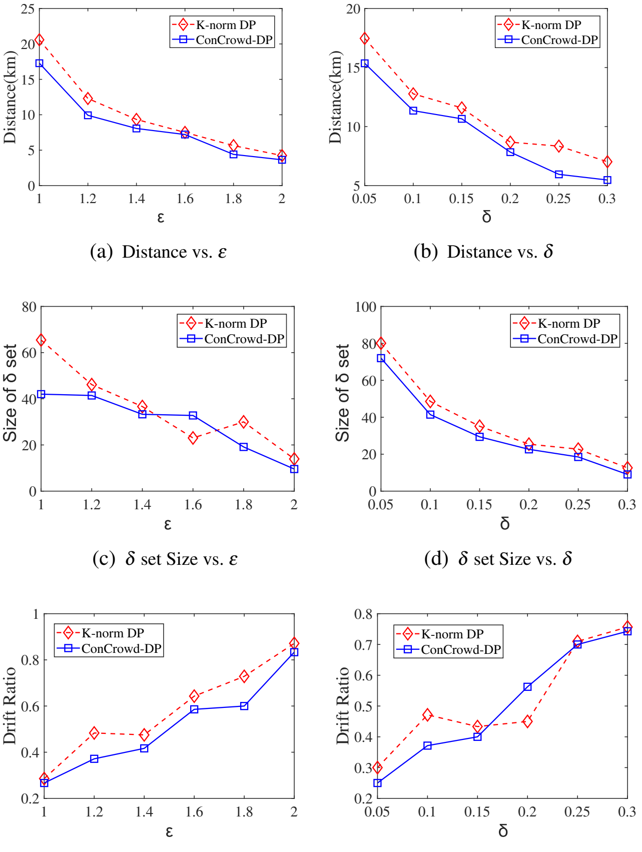
(e) Drift Ratio vs. ε

(f) Drift Ratio vs. δ

FIGURE 11. The DP protection performance of ConCrowd-DP with the personal training set.

decreases while ε increases. It indicates that the increase of ε improves the accuracy of the output location.

- **Size of δ-set vs ε.** Fig. 11(c) 12(c) show the impacts of ε on the size of δ-location set. The size of δ-location set decreases as ε increases. The reason for this phenomenon is that a larger ε results in a more accurate output. An accurate output position optimizes the user's probability distribution on-location set in the subsequent time-partition. The potential locations correspond to a higher probability in the optimized distribution, leading to a smaller size of the δ-set finally.
- **Drift ratio vs ε.** Fig. 11(e) 12(e) present the impacts of ε on the drift ratio of the accepted tasks. It indicates that a larger ε increases the risk of the drift. That's because a larger ε leads the δ-set to become smaller, as mentioned above. Fewer locations, which are used to provide DP protection, result in a smaller K-norm. The smaller K-norm makes the drift more likely to happen.



(a) Distance vs. ε

(b) Distance vs. δ

(c) δ set Size vs. ε

(d) δ set Size vs. δ

(e) Drift Ratio vs. ε

(f) Drift Ratio vs. δ

FIGURE 12. The DP protection performance of ConCrowd-DP with the public training set.

The impacts of δ. The parameter δ directly determines the size of the δ-location set. A larger value of δ means that δ-set contains a smaller number of locations. δ-set selects locations from the probability distribution in the reverse order, with which the user is likely to appear at the site. Therefore, smaller size of δ-set also indicates that there is a more significant correlation between the δ-set and the actual position. Then, ConCrowd-DP generates a more accurate perturbed location.

- **Distance vs δ.** Fig. 11(b) 12(b) show the impact of δ on the DP performance of the distance between the perturbed location and the actual position. The distance decreases as δ increases. It just reflects that a large value of δ improves the accuracy of the perturbed location, as analyzed above.
- **Size of δ-set vs δ.** Fig. 11(d) 12(d) present the impact of δ on the size of the δ-set. The parameter δ determines the size of δ-set directly with the opposite trend as mentioned above.
- **Drift ratio vs δ.** Fig. 11(f) 12(f) present the impact of δ on the drift of the accepted tasks. As mentioned above, a larger δ means a smaller size of δ set, while a smaller size of δ-set means a smaller size of the K-norm. Therefore, a large value of δ increases the risk of drift.

The impacts of the different training data set. Fig. 11 shows the DP performance on the personal training dataset, and Fig. 12 presents the performance on the public training dataset. The personal training set consists of individual trajectories. While the public training set consists of the trajectories from public users. Therefore, compared with the public training dataset, the training trajectories in the personal dataset are more relevant to the user's mobile behavioral pattern. Thus, STMarkov generates a more accurate probability distribution of the user on-location set and spatiotemporal transfer matrix. It results in a smaller size of δ-set and a shorter distance between the perturbed location and the actual position, as shown in Fig. 11 12.

The comparison of the performances between ConCrowd-DP and original K-norm DP. ConCrowd-DP optimizes the original K-norm DP by taking the accepted tasks into consideration. The accepted task is close to the actual location, optimizing the users' probability distribution on-location set. The optimized probability further results in more accurate perturbed locations. Finally, it leads to the advantages on the size of δ-set and the distance, and also the disadvantage on the drift of the accepted tasks, as shown in Fig. 11 12.

VII. DISCUSSION

In this section, we summarize the suitable application scenarios and modes of our ConCrowd-DP, and discuss the directions in which it can be optimized.

A. SUITABLE APPLICATION MODES OF CONCROWD-DP

We discuss various suitable application modes of ConCrowd-DP, according to different real-world scenarios. We do not take the cybersecurity issues into consideration in this paper.

- In a secure and trusted scenario, mobile users can participate in the crowdsourcing application directly with the actual trajectory or the perceptual PoI sequence generated from the STMarkov model.
- In the scenario with untrusted platform, mobile users can participate in the application with a perturbed location as described in Section II-A. The Server assigns a task according to the user's perturbed position. The user then corrects the deviation caused by the perturbance. Literature [4] provides a straightforward solution in such a manner.
- Four kinds of trajectories are introduced in this paper, i.e., the actual trace, the statistical trajectory, the perceptual PoI sequence generated by STMarkov, and the perturbed trajectory from ConCrowd-DP. They can be used not only individually but also in a combination manner. For example, we can set a route-selection option for the mobile users. If the user selects a secure path, we provide him (her) the perturbed DP trace for participating in the application. If he (she) is taking a regular route, the statistical or perceptual trajectory can be selected; if taking a new way, the user can be allowed to set a new route.

B. FUTURE WORK

We further discuss the optimization space of our ConCrowd-DP and take the following ideas as the research directions of our future work.

- **Personalized privacy preservation**. We achieve our privacy-preserving solution by modeling the user's mobility. In the future work, we can seek to achieve further personalized privacy preservation, based on not only the mobility but also the user's identity and the location's sensitivity. For instance, a multi-level DP mechanism can be customized according to the user's identity. It provides different-level DP protections to the users with different privacy-preserving requirements. We can also adaptively protect a location's privacy according to its sensitivity to the user.
- **User's mobility perception**. This paper perceives the user's dynamic mobile behavioral patterns by modeling the spatiotemporal association in mobility. In the future work, we can consider modeling the spatiotemporal association and the social network jointly, considering that social friends or corresponding communities also impact the users' mobility.

VIII. RELATED WORK

In this section, we review prior works that are most relevant to our ConCrowd-DP from the perspective of the spatiotemporal mobility perception and the privacy-preservation in the continual crowdsourcing scenarios.

A. SPATIOTEMPORAL MOBILITY PERCEPTION

Modeling users' mobile behaviors in the mobile applications is an open issue [15], [21]. Literature [22] proposed a utility-aware synthesis of DP trace, which took the trip's distribution, length and start-end points into consideration in the usability perception. Li et al. studied the users' mobility-modeling issues in the mobile social network [23]. These techniques model the users' mobilities simply just based on statistical characteristics. Literature [24] inferred the vehicle's driving path based on speed and other auxiliary information, such as real-time traffic and traffic rules. This technique is mainly limited to modeling the road network.

Literature [25] generated decoys to provide the dummy privacy protection for anonymizing the actual user based on the social and travel behavior patterns. PLP [8] took the continual transfer into consideration, according to the Conditional Random Fields. Literatures [26], [27] modeled users' mobile behaviors based on the Markov method. However, they failed to overcome the limitations of traditional Markov. The above-mentioned techniques only model the spatial transfer of users' mobilities, failing to achieve the time-related transfer model. Our STMarkov realizes this goal by introducing the time-partitioning concept to improve the Markov model. It provides the ConCrowd-DP with time-related steady-state distribution and spatiotemporal transfer matrix.

B. LOCATION PRIVACY-PRESERVATION IN CONTINUAL CROWDSOURCING SCENARIOS

Plenty of location privacy-preserving techniques provided anonymous or uncertain privacy-preservation for mobile applications generally [28], such as the clustering [29], generalization [30], obfuscation [31], perturbation [32]. Literature

[33] quantified the privacy risks brought by the co-location in the social network. Co-location refers to the information that two users meet together. Literature [34] manipulated the graph structure to avoid safety detection. Zuo et al. explored the causes of data leakage in cloud [35]. Jin et al. proposed a solution for private-data transactions in the mobile crowdsensing in [36]. However, these single-location-based protections are vulnerable to the inferential attacks, due to the spatiotemporal association hidden in the users' mobility [7], [37].

Solutions, such as θ-secure area [6], DPSence [4], PLP [8], introduced the spatiotemporal correlation into the location privacy-preservation. θ-secure area [6] assessed whether the clustering area was secure, by comparing the Earth Mover's Distance between the prior and posterior distributions. It relies more on statistical calculations and doesn't dig the spatiotemporal mobility sufficiently. DPSence [4] provided a crowdsourced spectrum-sensing solution with the DP principle based on the Markov model. However, the spatial transfer, it modeled, did not consider the temporal correlation. Literature [8] proposed the PLP solution to model the continual transfer according to Conditional Random Fields (CRF). While the CRF method has poor compatibility with other privacy-preserving mechanisms.

These technologies are still threatened by the re-identification attack proposed in literatures [7], [38]. This attack infers the user's identity by modeling his (her) specific mobile patterns based on the user's shared locations, such as the locations of the service queries and the accepted tasks in the crowdsourcing. Literature [5] proposed a DP solution for defending against the attacks driven from the locations of LBS queries. However, it failed to take the accepted tasks into consideration. According to the design goals of our solution in Section II-C, our ConCrowd-DP can prevent the multiple accepted tasks from contributing to the spatiotemporal-mobility modeling in the continual crowdsourcing scenario. Therefore, our ConCrowd-DP can effectively defend against the mobility-modeling attacks, achieving the privacy-preserving continual crowdsourcing.

IX. CONCLUSION

Targeting at achieving the privacy-preservation continual crowdsourcing, we proposed the ConCrowd-DP, a mobility-aware differentially private solution. According to a combination of spatiotemporal-mobility modeling and the DP principle, it eliminates the privacy risks brought by the multiple accepted tasks, enabling the users to participate in the crowdsourcing securely and continually. Extensive experiments confirmed that our ConCrowd-DP well balances the tradeoff between privacy protection and data usability.

REFERENCES

[1] Y. Chen, D. Guo, P. Lv, T. Zhou, and M. Xu, "Prosc: Profit-driven participant selection in compressive mobile crowdsensing," in *Proc.of IEEE/ACM IWQOS*, 2018.

[2] Y. Chen, P. Lv, D. Guo, T. Zhou, and M. Xu, "Trajectory segment selection with limited budget in mobile crowd sensing," *Elsevier Journal of Pervasive and Mobile Computing*, vol. 40, pp. 123–138, 2017.

[3] M.Gotz, S. Nathn, J. Gehrke, and Maskit, "Privately releasing user context streams for personalized mobile applications," in *Proc.of ACM SIGMOD*, 2012.

[4] X.Jin, R.Zhang, and Y.Chen, "Dpsense: Differentially private crowdsourced spectrum sensing," in *Proc.of ACM CCS*, 2016.

[5] Y.Xiao and L.Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc.of ACM CCS*, 2015.

[6] B. Lee, J. Oh, H. Yu, and J. Kim, "Protecting location privacy using location semantics," in *Proc.of ACM SIGKDD*, 2011.

[7] R.Shokri, G.Theodorakopoulos, J. L. Boudec, and J. Hubaux, "Quantifying location privacy," in *Proc.of IEEE SP*, 2011.

[8] Q. Ma, S. Zhang, and T. Zhu, "Plp: Protecting location privacy against correlation analyze attack in crowdsensing," *IEEE transactions on mobile computing*, vol. 16(9), pp. 2588–2598, 2016.

[9] V. Primault, A. Boutet, S. B. Mokhtar, and L. Brunie, "The long road to computational location privacy," *IEEE Communications Surveys and Tutorials*, 2018.

[10] C.Dwork, "Differential privacy," in *Proc.of ICALP*, 2006.

[11] M.Hardt and K.Talwar, "On the geometry of differential privacy," in *Proc.of ACM STOC*, 2010.

[12] A. Bhaskara, D. Dadush, R. Krishnaswamy, and K. Talwar, "Unconditional differentially private mechanisms for linear queries," in *Proc.of ACM STOC*, 2012.

[13] A. Nikolov, K. Talwar, and L. Zhang, "The geometry of differential privacy: The sparse and approximate cases," in *Proc.of ACM STOC*, 2013.

[14] Y. Chen, P. Lv, D. Guo, T. Zhou, and M. Xu, "A survey on task and participant matching in mobile crowd sensing," *Journal of Computer Science and Technology*, vol. 33, no. 4, pp. 123–138, 2018.

[15] K. Ouyang, R. Shokri, and D. Rosenblum, "A non-parametric generative model for human trajectories," in *Proc.of IJCAI*, 2018.

[16] S.Oya, C.Troncoso, and F.P.Gonzalez, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *Proc.of ACM CCS*, 2017.

[17] R.Shokri and G.Theodorakopoulos, "Location privacy meter tool," in *https://github.com/rzshokri/quantifying location privacy*, 2011.

[18] Y.Zheng, L.Zhang, X.Xie, and W.Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proc.of ACM WWW*, 2009.

[19] Y.Zheng, Q.Li, Y. Chen, X. Xie, and W.Ma, "Understanding mobility based on gps data," in *Proc.of ACM Ubicomp*, 2008.

[20] Y.Zheng, X. Xie, and W.Ma, "Geolife: A collaborative social networking service among user, location and trajectory," *IEEE Data Engineering Bulletin*, vol. 33, no. 2, pp. 32–40, 2010.

[21] M. Bhakti, D. Shelar, and D.K.Chitre, "Hidden markov model with biclustering cache replacement policy for location based services in manet," *International Journal Of Engineering And Computer Science*, vol. 4, no. 5, pp. 12 000–12 004, 2015.

[22] M. E. Gursoy, L. Liu, S. Truex, L. Yu, and W. Wei, "Utility-aware synthesis of differentially private and attack-resilient location traces," in *Proc.of ACM CCS*, 2018, pp. 196–211.

[23] H. Li, H. Zhu, S. Du, X. Liang, and X. Shen, "Privacy leakage of location sharing in mobile social networks: Attacks and defense," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 646–660, 2018.

[24] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong, "Location privacy in usage-based automotive insurance: Attacks and countermeasures," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 196–211, 2019.

[25] J. Kang, D. Steiert, D. Lin, and Y. Fu, "Movewithme: Location privacy preservation for smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 711–724, 2020.

[26] B. Guo, Y. Liu, L. Wang, O.K.Li.Victor, C.K.LAM.Jacqueline, and Z. Yu, "Task allocation in spatial crowdsourcing current state and future directions," *IEEE Internet of Things Journal*, 2018.

[27] J. Jiang, C. Pan, H. Liu, and G. Yang, "Predicting human mobility based on location data modeled by markov chains," in *Proc.of IEEE UPINLBS*, 2016.

[28] G.Ghinita, "Privacy for location-based services," in *Proc.of Synthesis Lectures on Information Security, Privacy, and Tru. Morgan Claypool*, 2013.

[29] R.Chen, B. Fung, B.C.Desai, and N. Sossou, "Differentially private transit data publication: a case study on the montreal transportation system," in *Proc.of ACM KDD*, 2012.

[30] J.Krumm, "A survey of computational location privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[31] M. Nergiz, M. Atzori, Y. Saygin, and B. Guc, "Towards trajectory anonymization: a generalization-based approach," *Transactions on Data Privacy*, vol. 2, no. 1, 2009.

[32] G. Poulis, S. Skiadopoulos, G. Loukides, and A. Gkoulalas-Divanis, "Apriori-based algorithms for k^m-anonymizing trajectory data," *Transactions on Data Privacy*, vol. 7, no. 2, 2014.

[33] A. M. Olteanu, K. Huguenin, and R. Shokri, "Quantifying interdependent privacy risks with location data," *IEEE transactions on mobile computing*, vol. 16, no. 3, pp. 829–842, 2016.

[34] B. Wang and N. Gong, "Attacking graph-based classification via manipulating the graph structure," in *Proc.of ACM CCS*, 2019.

[35] C. Zuo, Z. Lin, and Y. Zhang, "Why does your data leak? uncovering the data leakage in cloud from mobile apps," in *Proc.of IEEE SP*, 2019.

[36] W. Jin, M. Xiao, M. Li, and L. Guo, "If you do not care about it, sell it: Trading location privacy in mobile crowd sensing," in *Proc.of IEEE INFOCOM*, 2019, pp. 1045–1053.

[37] R. Shokri, M. Stronati, and C. Song, "Membership inference attacks against machine learning models," in *Proc.of IEEE SP*, 2017.

[38] Y. Dai, J. Shao, and D. Zhang, "Personalized semantic trajectory privacy preservation through trajectory reconstruction," *World Wide Web*, vol. 21(4), pp. 875–914, 2018.

GUOYING QIU received the B.S. and M.S. degrees from Xinjiang University in 2010 and Chongqing Normal University in 2015, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Xidian University, China. His research mainly focuses on mobile computing, security and privacy in mobile data and privacy-enhancing technologies.

YULONG SHEN received the B.S. and M.S. degrees in computer science and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2002, 2005, and 2008, respectively. He is currently a Professor with the School of Computer Science and Technology, Xidian University, where he is also an Associate Director of the Shaanxi Key Laboratory of Network and System Security and a member of the State Key Laboratory of Integrated Services Networks. His research interests include wireless network security and cloud computing security. He has also served on the technical program committees of several international conferences, including ICEBE, INCoS, CIS, and SOWN.