Hindawi
Mobile Information Systems
Volume 2020, Article ID 1960368, 6 pages
https://doi.org/10.1155/2020/1960368

Research Article

A Dual Privacy Preserving Algorithm in Spatial Crowdsourcing

Shengxiang Wang, Xiaofan Jia, and Qianqian Sang

School of Information Engineering, Henan University of Science and Technology, 471000 Luoyang, Henan, China

Correspondence should be addressed to Shengxiang Wang; 1084019500@qq.com

Received 15 October 2019; Revised 22 May 2020; Accepted 10 June 2020; Published 27 June 2020

Academic Editor: Paolo Bellavista

Spatial crowdsourcing assigns location-related tasks to a group of workers (people equipped with smart devices and willing to complete the tasks), who complete the tasks according to their scope of work. Since space crowdsourcing usually requires workers' location information to be uploaded to the crowdsourcing server, it inevitably causes the privacy disclosure of workers. At the same time, it is difficult to allocate tasks effectively in space crowdsourcing. Therefore, in order to improve the task allocation efficiency of spatial crowdsourcing in the case of large task quantity and improve the degree of privacy protection for workers, a new algorithm is proposed in this paper, which can improve the efficiency of task allocation by disturbing the location of workers and task requesters through k-anonymity. Experiments show that the algorithm can improve the efficiency of task allocation effectively, reduce the task waiting time, improve the privacy of workers and task location, and improve the efficiency of space crowdsourcing service when facing a large quantity of tasks.

1. Introduction

With the increasing use of various mobile devices, the data collected and shared by smartphones have grown exponentially. Taking advantage of the mobility of a large number of potential users, an efficient and scalable new data collection mechanism emerges, namely, space crowdsourcing (SC). SC has extensive applications in the fields of environmental awareness, smart city, news, and crisis response, such as gMission [1], Didi chuxing [2], and baidu takeaway [3]. Spatial crowdsourcing generally consists of three parts: task requester, crowdsourcing platform, and worker [4]. Figure 1 shows the workflow of general spatial crowdsourcing. The functions of these three components are as follows Task requesters publish spatio-temporal tasks that are assigned to workers by the crowdsourcing platform, and these tasks contain the location information of the task requesters and the deadline to complete the tasks. Workers will also send their temporal and spatial information to the crowdsourcing platform, such as location information and work scope. Depending on the specific application, workers can be assigned tasks or choose their own tasks. The crowdsourcing platform connects task publishers and workers. Its core functions include task assignment, protecting the privacy of workers and task publishers, setting the incentive mechanism for workers, and summarizing the results submitted by workers.

Task requesters and worker are usually registered on crowdsourcing servers by SC, which act as intermediaries between parties and often play a role in assigning tasks to worker. The requester distributes one or more tasks to the server, and then the server assigns the task to the worker, which we call the task assignment.

Workers have the right to decide whether they will complete the nearest task around them. According to Kazemi and Shahabi's classification [5], SC is divided into two completely different SC types, namely, worker-selected tasks (WST) mode and sever-assigned tasks (SAT) mode. In the WST mode, the SC server publishes space tasks online, and the worker can arbitrarily select tasks in their vicinity without sending their specific location to the server. That is, in this mode, the worker does not display its location to the SC server. However, it is difficult to obtain an optimal task assignment strategy without a worker global system view. In contrast, in the SAT mode, the SC server assigns location-based tasks to the nearest worker based on the specific

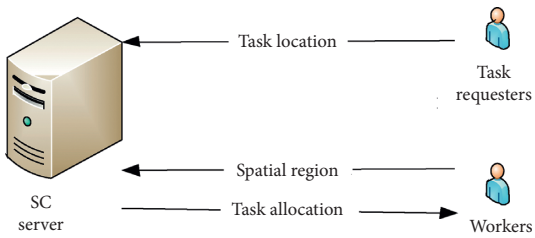

Figure 1: Spatial crowdsourcing scheme.

location of the worker, so we find it necessary to send the worker's surroundings and location to the server in advance. In addition, although it is easy to implement the optimal task allocation scheme, it is inevitable to expose private information to the SC server. Suppose there are reasonable but terrible situations. In order to get a good reward, the SC server may exchange personal privacy with other agents. These agencies may then make full use of the information to take adverse actions against employees, such as personal monitoring, identity theft, and revealing sensitive information. Therefore, the SC server should not be fully trusted to some extent. Corresponding measures must be taken to protect the privacy of employees, and a large number of employees are willing to participate in the crowdsourcing system eventually.

In this article, the application can be regarded as a spatial crowdsourcing platform, taking the didi chuxing as an example. The task publisher (passenger) and the worker (taxi) send their disturbed positions to the SC server. After the server calculates the distance between them, the task is assigned. When the assignment is successful, the taxi goes to the passenger location to complete the task. In this process, the location information obtained by the crowd-sourcing server is all disturbed, and the attacker cannot obtain the detailed location information of taxi and passenger, so as to protect the location information of taxi and passenger.

2. Related Work

2.1. Privacy Protection. The development of mobile networks has produced a huge amount of data, which contains many user's private data, such as user's location information and user's interest preferences. While users are using more services, they are also paying more and more attention to their privacy protection. In spatial crowdsourcing, since both the task publisher and the worker need to send their own location to the SC server, the privacy problem based on the location service has been extensively studied in the literature.

Kido et al. designed some interference locations without disclosing the exact location of each worker, and then they sent the packaged, untrue location to the service provider. In order to provide location-based anonymous services, the predecessors proposed a new algorithm based on spatio-temporal protection. It is proposed in the literature [6] that k-anonymity is concerned with the problem that the user's location cannot be distinguished from other k users. In addition, a trusted third-party location anonymizer is introduced in the k-anonymous technology, which allows the location to be blurred to a hidden spatial area. The work presented by [7] has improved the security of their proposed models further, and they developed a new peer-to-peer anonymity mechanism. Due to the complexity of the application scenario in the SC, the above solution for location privacy is not applicable to the SC.

In recent work, differential privacy has been widely used to protect workers' location privacy [8], and it only protects the worker's location privacy, but reveals the location of the task requester. In the work of Chen et al. [9], the premise that the worker must know the exact location of the requester in advance still exists, and then homomorphic encryption is used to protect the worker's location privacy. The method of Chen et al. poses a privacy threat because there is no trust between the requester and the worker.

2.2. Spatial Crowdsourcing. The concept of "crowdsourcing" has been taken seriously by the business community since it became a standard term, but SC has only attracted attention in recent years. Previous discussion on privacy protection focused on location privacy in location-based services, while precise locations were obscured by obscuration [9] or aggregation. However, they cannot be directly applied to the SC.

A number of research works are proposed for the location privacy issues in SC. The work in [8] introduces an analysis model of task assignments that explores the basic trade-offs between privacy, utility, and overhead and uses differential privacy to sanitize worker positions. In the work of Chen et al. [9] privacy protection for workers based on geolocation and differential privacy is provided. The method in Gong et al. [7] leverages the social tie structure among mobile users to motivate them to participate in pseudonym change. It also takes advantage of the social group utility maximization framework, which allows users to protect location privacy with their specific personalized anonymous settings. If privacy protection scheme is not adopted in spatial crowdsourcing to protect the privacy information of workers, the privacy of workers will be leaked. However, if workers are not required to submit location information, it will lead to an increase in error rate and the number of spammers, thus affecting the quality of spatial crowdsourcing. In the work of Zeng et al. [10] a spatial crowdsourcing quality control model was proposed, in which k-anonymous algorithm was used to protect the location privacy of spatial crowdsourcing workers, extreme learning machine algorithm was used to detect spam senders, and the expectation maximization algorithm was used to estimate the error rate, thereby protecting the privacy of workers without affecting the quality of spatial crowdsourcing. However, this method only considers the privacy protection of workers, not the privacy protection of task publishers.

Spatial crowdsourcing mainly consists of three parts: SC server, worker, and task requester. First, the worker and task requester register on the server. The task publisher then posts the task to the server, which assigns the task based on the worker's location and scope of work. The main advantage

of spatial crowdsourcing service is that the worker can be considered as trusted third party to a certain extent. Existing research has two main drawbacks. First, some studies focus only on protecting the location privacy of worker [11], but the location of the task is open. The attacker can determine the user's location by the location of the task, thereby obtaining the user's location privacy. Second, the existing spatial crowdsourcing privacy protection scheme usually assumes that the server is reliable, but it is difficult to find a fully trusted server.

In order to solve the above two problems, To et al. [8] assumed that the server was untrusted, disturbed the positions of workers and task requester at the same time, and allocated tasks to the nearest workers by calculating the distance between the worker and task sender after disturbance. The method can realize the minimum disclosure of the task position and the nondisclosure of the work position and complete the task according to the priority when the task requester is relatively dense. This method is preferred when the task publisher is highly dense. Lower level tasks will wait longer.

2.3. Location K Anonymous in Spatial Crowdsourcing. In the SC, the worker's location attribute is a quasi-identifier. As shown in Figure 2, in an anonymous space area, the location of any worker cannot be distinguished from the location of K-1 workers at least. The pseudoidentifier is the smallest attribute, which combines other external information and has a higher probability of identifying the location of the target. As shown in the figure, the real position of the space crowder is L, and then the position point L is extended to the hidden area R, replacing the accurate position information of the worker. In this anonymous space area, each worker is hidden in at least K-1 workers. This means that any attacker can only determine the number of workers in the hidden area, but not their exact location. This approach gives workers a degree of privacy protection.

In this paper, we introduce a method for the high density of task requester in space crowdsourcing, Dual Privacy preserving (DPP). This method can reduce the waiting time of passengers (task requester) by setting thresholds, while protecting the location information of taxis (workers) through K-anonymity.

3. System Model

The DPP algorithm is mainly used in areas where task requester is more densely distributed. In this article, we divide the dense area into two cases. As shown in Figure 3, tasks in the dense $L1$ area can be assigned to two taxis, and tasks in the dense $L2$ area can be assigned to three taxis.

The general system model for SC services consists of three parts: worker, task publisher, and SC server. The worker and task publisher send the location information to the SC server, which sends the task to the appropriate worker.

The model we propose consists of four parts: taxi (worker), passenger (task publisher), SC server, and task distributor, as shown in Figure 4. The passenger submits the

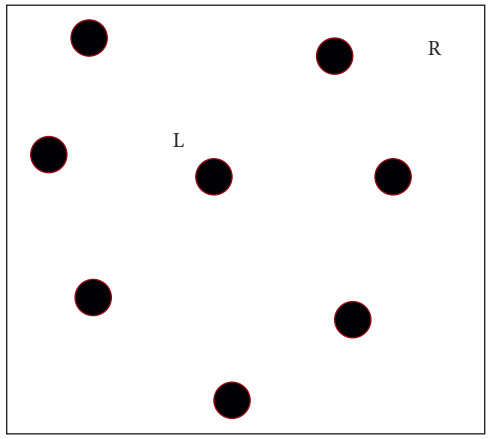

Figure 2: Spatial k-anonymity scheme.

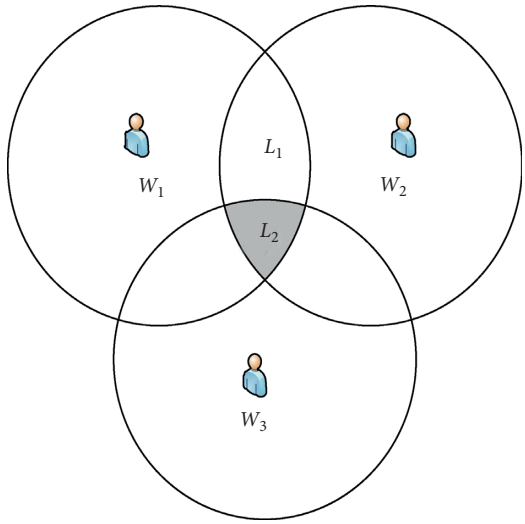

Figure 3: Dense regions.

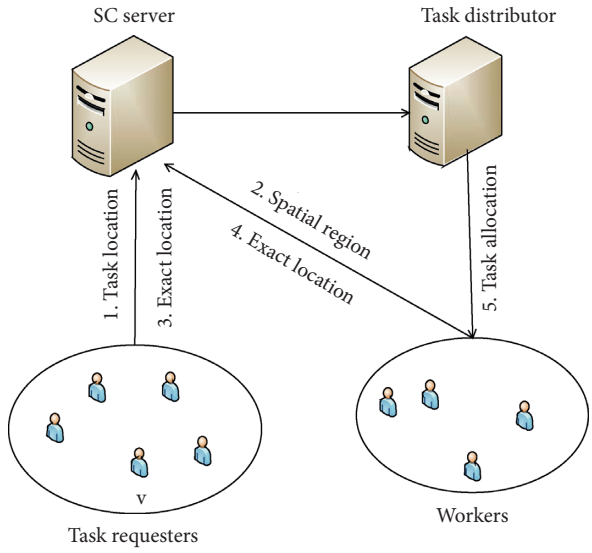

Figure 4: System model.

(1) Input: W, T, $\mathbf{R}_{wi}$, ϵ, r, m, $\mathbf{m_{max}}$
(2) Output: a set of valid worker-task matches
(3) Perturb locations of workers and tasks using K-Anonymity
(4) $\mathbf{l}_{wi} \longrightarrow \mathbf{l}_{w_i'}$, $\mathbf{l}_{tj} \longrightarrow \mathbf{l}_{t_j'}$
(5) For $\mathbf{t_j} \in \mathbf{T}$ do: {assign it to the highest-rank worker}
(6) U2U: Server identifies candidate workers **Nj** for $\mathbf{t_j}$
(7) $\mathbf{N_j} = \{\mathbf{w_i}: \mathbf{d}(\mathbf{w_i'}, \mathbf{t_j'}) \hat{\mathbf{A}} \mathbf{ER_{wi}}\}$
(8) If $\mathbf{N_j} = \boldsymbol{\phi}$ **tj** remains unassigned; go to Line 5
(9) Server forwards candidate workers $\mathbf{N_j}$ to $\mathbf{t_j}$'s requester
(10) U2E: Requester matches $\mathbf{t_j}$ to $\mathbf{w_{mac}}$, where
(11) $\mathbf{W_{max}} = \mathbf{argmax}\{\mathbf{Rank}(\mathbf{W_i}) : \mathbf{W_i} \in \mathbf{N_j}\}$
(12) Rank $(\mathbf{W_i})$ = precomputed random [0, 1]
(13) Requesters sends exact task location $\mathbf{l}_{t_j}$ to $\mathbf{w_{max}}$
(14) Check if the number of tasks $\mathbf{m}$:$\mathbf{m} \leq \mathbf{m_{max}}$
(15) If so, go to Line 17
(16) Otherwise, go to Line 5
(17) E2E: Worker $\mathbf{w_{max}}$ checks if $\mathbf{d}(\mathbf{w_{max}}, \mathbf{t_j}) \leq \mathbf{R_{wmax}}$
(18) If so, match $(\mathbf{t_j}, \mathbf{w_{max}})$ is a valid assignment
(19) Update **W** : **W** = **W**-$\{\mathbf{w_{max}}\}$; go to Line 5
(20) Otherwise, update $\mathbf{N_j}$: $\mathbf{N_j}$ - $\{\mathbf{w_{max}}\}$, go to Line 10

Algorithm 1: Efficient task allocation algorithm.

task to the SC server, which assigns the task to the nearest taxi. Throughout the process, the specific location of the taxi and task requester is not disclosed to the server. Because the server may reveal the location of passenger and taxi.

When the task requester publishes the task, the task requester collaboratively communicates with the surrounding k-1 users to form a k-anonymous zone composed of k users, and then the task publisher sends the location of the disturbed zone to the SC server (Step 1). At the same time, the taxi also sends the disturbed location information to the SC server (Step 2). The SC server calculates the Euclidean metric between the taxi and the passenger's position after the disturbance, determines whether the passenger is within the working range of the taxi, and uses all the taxis capable of completing the task as the candidate taxi (Step 3). The candidate taxi and passenger then send the precise location information to the SC server, which calculates the Euclidean distance between the candidate taxi and the passenger again and finally determines the taxi that can complete the mission (Step 4). The SC server sends the task to the task distributor, which assigns the task to the taxi based on the taxi threshold and the K-anonymity algorithm.

We assume that the taxi handles a task for a constant time and that time is much longer than the task assignment time. The processing time of each task is set to t_w. When there are n tasks to be processed, if the task is assigned to the taxi according to the priority of the task, the task with the lowest priority needs to wait for the time of nt_w. In the DPP algorithm, the waiting time is reduced by designing a threshold $m_{\max}$ for each taxi. When the number of tasks at one taxi reaches a threshold, subsequent tasks are assigned to nearby taxis until the threshold is reached. The waiting time for the lowest priority task in the DPP algorithm is $[n - (n_w - 1) * m_{\max}] * t_w$. The specific algorithm of DPP is shown in Algorithm 1 The parameter meanings in the above pseudocode are shown in Table 1.

Table 1: The meaning of parameters in DPP algorithm.

The meaning of parameters in DPP algorithm	
W	The number of workers
T	The number of tasks
R_w	The scope of a worker's work
ϵ	The level of k-anonymity
r	The radius of the anonymous region
m	The number of tasks that a worker has assigned
$m_{\max}$	The threshold of workers

Using the disturbance algorithm to simultaneously disturb the location of the taxi and the passenger, calculating the Euclidean distance between the taxi and the passenger after the disturbance, and determining whether the passenger is within the working range of the taxi, the server marks the candidate taxi for the task, assigning the task to taxi with the highest level. The passenger releases the specific location to the taxi, calculates the Euclidean distance between the passenger and the actual position of the taxi again, and determines whether the distance is smaller than the working radius of the taxi. If the Euclidean distance is greater than the working radius, the W is updated, and if it is smaller than the working radius, the task is assigned to the taxi. It is calculated whether the number m of pending tasks waiting at the taxi is less than the threshold $m_{\max}$, and the task is assigned to the taxi, and if not, the return is 5, and the task is assigned to the taxi of the lower level.

In order to protect the location privacy of the taxi when sending the taxi location to the SC server, we use the K anonymous algorithm to send the location of multiple taxis

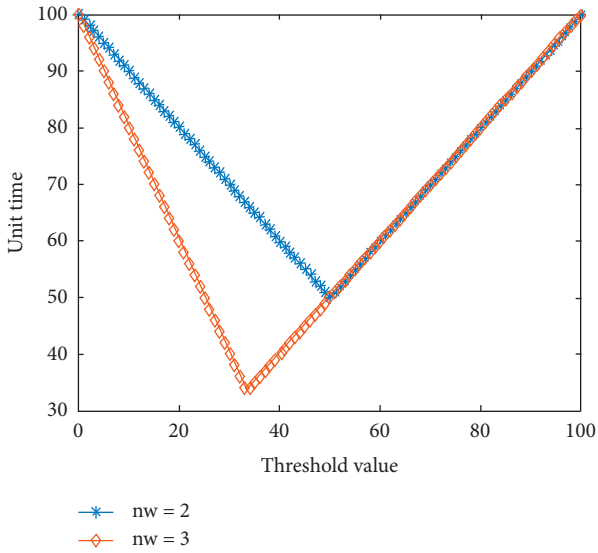

Figure 5: Waiting time for 100 tasks.

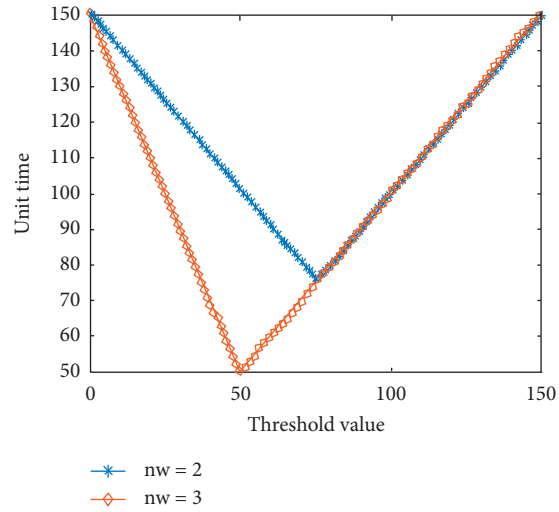

Figure 6: Waiting time for 150 tasks.

to the SC server at the same time so that the attacker cannot obtain the specific location of the taxi to be accepted.

4. Experimental Result

We performed experiments on the *T*-Drive dataset. This dataset contains more than 9019 taxis and thousands of passengers. Our experiment was performed on python. We set drivers who were SC workers and passengers who were SC requesters. We chose typical ranges of values for ϵ, r, R_w as follows, and we assumed the requesters and the workers have the same privacy level (ϵ, r), where $\varepsilon \in \{0.1, 0.4, 0.7, 1.0\}$ and $r \in \{2,000, 1400, 800, 200\}$ in meters, ranging from strict to loose privacy requirements. We set the reachable distance of each worker to a random value in meters, $1000 \leq R_w \leq 3000$. We set the number of tasks to 100 and 150, and the number of taxis that can be reached by tasks is 2 and 3, respectively. If the processing time of the task is constant to t_w, and the time is much longer than the time required for task assignment, the task is processed before it is processed. The required wait time is the product of the number of pending tasks and the individual task processing time. We compared the impact of different thresholds on latency when the number of taxis was different, with a number of missions of 100 and 150, respectively. The results are shown in Figures 5 and 6.

As can be seen from the figure, setting different thresholds can reduce the waiting time. Before the threshold reaches a certain value, the waiting time is linearly decreasing as the threshold increases. When the threshold exceeds a certain limit, the waiting time will increase linearly as the threshold increases. We call this boundary the optimal threshold m. From the comparison of Figures 3 and 4, we can see that the optimal threshold is affected by the number of tasks and the number of taxis that the task can reach. The optimal threshold is typically (n/n_w).

The Gedik algorithm is an algorithm that assigns tasks by prioritizing them. Computation time consists of two parts: SC

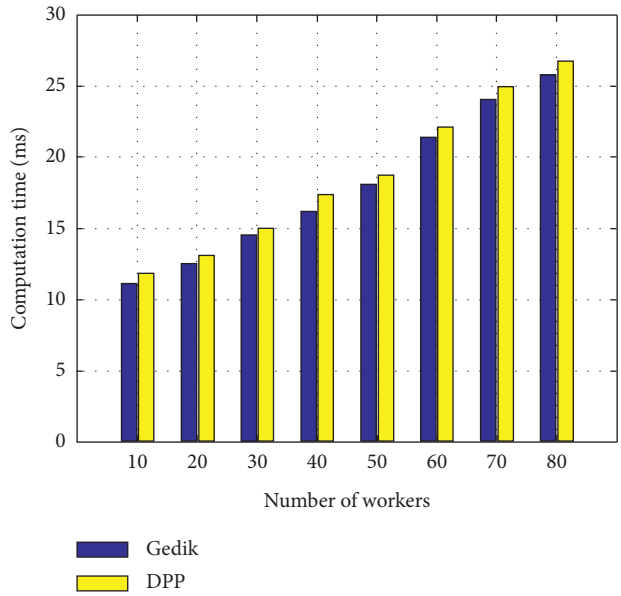

Figure 7: The computation time.

server calculates the Euclidean distance between the positions of the task requester and the worker after the disturbance, determines whether the task requester is within the scope of the worker's work, and the time required for all the workers who can complete the task to be candidates for the job, as well as the task distributor matches the time the task requester precisely locates the worker to allocate the task. We compared the calculation time taken by the DPP algorithm proposed in this paper with that of Gedik, as shown in Figure 7. It can be seen from the figure that the time taken by the calculation increases with the increase in the number of taxis. The DPP algorithm takes more time, which is because the DPP algorithm anonymizes the taxi location K compared with Gedik when sending the taxi location, which results in more computing time spent by the whole system.

5. Conclusion

A new algorithm is proposed in this paper. The algorithm can set the threshold value for the taxi and assign the subsequent task to other taxis when the task to be processed by one taxi reaches the threshold, thereby reducing the task waiting time and improving the efficiency of the entire SC service system. At the same time, the K anonymous algorithm is used to protect the taxi location privacy when the taxi sends its location to the SC server. We have experimentally verified that this method can reduce waiting time, improve system efficiency, and protect taxi privacy.

Data Availability

The T-Drive database can be found at the following website: https://www.microsoft.com/en-us/research/publication/t-drive-driving-directions-based-on-taxi-trajectories/.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] L. N. Jiang, L. Chen, and Z. Chen, "Knowledge base enhancement via data facts and crowdsourcing," in *Proceedings of the 2018 IEEE 34th International Conference On Data Engineering (ICDE)*, Paris, France, April 2018.

[2] K. Zhang, Z. Liu, and L. Zheng, "Short-term prediction of passenger demand in multi-zone level: temporal convolutional neural network with multi-task learning," *IEEE Transactions On Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1480–1490, 2020.

[3] X. L. Liu and D. D. Sun, "A study on the impact of online ta out based on the baidu index on the management of university cafeteria," in *Proceedings of 2016 4th Ieee International Conference on Cloud Computing and Intelligence Systems (IEEE CCIS 2016)*, pp. 182–185, Beijing, China, August 2016.

[4] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *The VLDB Journal*, vol. 29, no. 1, pp. 217–250, 2019.

[5] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving privacy with probabilistic indistinguishability in weighted social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1417–1429, 2017.

[6] S. Zhang, G. Wang, and Q. Li, "A dual privacy preserving scheme in continuous location-based services," in *Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS*, Sydney, Australia, August 2017.

[7] X. Gong, X. Chen, K. Xing, D., H. Shin, M. Zhang, and J. Zhang, "Personalized location privacy in mobile networks: a social group utility approach," in *Proceedings of the 2015 IEEE Conference on Computer Communications*, Kowloon, Hong Kong, April 2015.

[8] H. To, C. Shahabi, and L. Xiong, "Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server," in *Proceedings of the IEEE 34th International Conference on Data Engineering*, Paris, France, April 2018.

[9] J. Chen, K. He, Q. Yuan, M. Chen, R. Du, and Y. Xiang, "Blind filtering at third parties: an efficient privacy-preserving framework for location-based services," in *Proceedings of the IEEE Transactions on Mobile Computing*, Piscataway, NJ, USA, June 2018.

[10] MJ. Zeng, ZL. Cheng, X. Huang, and B. Zheng, "Spatial crowdsourcing quality control model based on K-anonymity location privacy protection and ELM spammer detection," *Mobile Information Systems*, vol. 2019, Article ID 2723686, 10 pages, 2019.

[11] H. To, G. Ghinita, L. Fan, and C. Shahabi, "Differentially private location protection for worker datasets in spatial crowdsourcing," in *Proceedings of the IEEE Transactions on Mobile Computing*, Piscataway, NJ, USA, June 2016.