Spatiotemporal Prediction Based Intelligent Task Allocation for Secure Spatial Crowdsourcing in Industrial IoT

Mengyao Peng, Jia Hu, Hui Lin, Xiaoding Wang, Peng Liu, Kapal Dev, Sunder Ali Khowaja, and Nawab Muhammad Faseeh Qureshi

***Abstract*—With the emergence of spatial crowdsourcing technology, an efficient task allocation is the key to ensure the sustainable development of spatial crowdsourcing. However, previous spatial crowdsourcing task allocation technologies ignore the temporal and spatial continuity between historical task data, thus reducing the efficiency of crowdsourcing task allocation. In addition, spatial crowdsourcing also suffers from the privacy leakage problem. To solve these problems, we propose a Spatiotemporal Prediction based Spatial Crowdsourcing strategy, named SPSC, using both blockchain and artificial intelligence. Specifically, considering the temporal and spatial continuity of crowdsourced task data, SPSC combines both gated recurrent unit and variational autoencoder for crowdsourcing task prediction. In addition, different Laplacian noises are added to crowdsourced task data so as to protect the privacy of crowdsourced workers during the task prediction. Moreover, by classifying crowdsourcing tasks and grouping crowdsourcing workers, SPSC reduces the risk of crowdsourcing workers colluding to steal the privacy data of crowdsourcing tasks using the blockchain technology. The experimental results show that SPSC can improve the privacy protection of spatial crowdsourcing, specifically, the more the number of categories, the higher the degree of privacy protection, and under the premise of predicting value, excellent system performance can be achieved.**

***Index Terms*—Spatial crowdsourcing, task allocation, privacy protection, blockchain, machine learning.**

Manuscript received 30 August 2021; revised 12 May 2022; accepted 8 August 2022. Date of publication 16 August 2022; date of current version 20 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grants U1905211 and 61702103 and in part by the Natural Science Foundation of Fujian Province under Grants 2020J01167 and 2020J01169. Recommended for acceptance by Dr. Varun G Menon. *(Corresponding authors: Hui Lin; Nawab Muhammad Faseeh Qureshi.)*

Mengyao Peng, Hui Lin, and Xiaoding Wang are with the College of Computer and Cyber Security, Engineering Research Center of Cyber Security and Education Informatization, Fujian Normal University, Fuzhou, Fujian 350117, China (e-mail: 18875857995@163.com; linhui@fjnu.edu.cn; wangdin1982@163.com).

Jia Hu is with the University of Exeter, EX4 4RN Exeter, U.K. (e-mail: J.Hu@exeter.ac.uk).

Peng Liu is with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: perryliu@hdu.edu.cn).

Kapal Dev is with the Department of Institute of Intelligent Systems, University of Johannesburg, Johannesburg 2092, South Africa (e-mail: kapal.dev@ieee.org).

Sunder Ali Khowaja is with the Department of Telecommunication Engineering, Faculty of Engineering and Technology, University of Sindh, Jamshoro 66702, Pakistan (e-mail: sandar.ali@usindh.edu.pk).

Nawab Muhammad Faseeh Qureshi is with the Department of Computer Education, Sungkyunkwan University, Jongno-gu, Seoul 35017, Korea (e-mail: faseeh@skku.edu).

Digital Object Identifier 10.1109/TNSE.2022.3198675

I. Introduction

With the rapid development of mobile Internet technology, spatial crowdsourcing has been widely used in many fields, such as transportation, food delivery, and map services. In spatial crowdsourcing, the crowdsourcing platform allocates, scheduling and quality control to crowdsourced tasks with temporal and spatial characteristics, so that crowdsourcing workers can complete tasks in an active or passive manner and eventually get a certain incentive reward [1]. Take the online car-hailing service as an example. The instant ride-hailing demand issued by passengers is used as a spatial crowdsourcing task, which specifies a departure from a certain destination. The online car-hailing platform needs to assign tasks to nearby drivers in a timely manner in the form of crowdsourcing. The driver goes to the designated departure place of the task to pick up the passengers and send them to the designated place and get corresponding rewards. As a bridge between crowdsourcing workers and crowdsourcing tasks, the task allocation strategy is one of the core issues in spatial crowdsourcing [2].

Existing researches on spatial crowdsourcing task allocation divide space crowdsourcing task allocation problems into static offline allocation problems and dynamic online allocation problems [3]. The optimization objectives mainly consider maximizing distribution income, distribution quantity, distribution quality, distribution efficiency, etc. However, most of the task allocation problems of spatial crowdsourcing [4] in real scenes have the characteristics of large-scale and multi-stage, which makes the traditional task allocation strategy have the following limitations in practical applications, that is, the failure to effectively use historical task data and consider the temporal and spatial continuity of historical task data. For the dynamic online allocation problem, the traditional task allocation strategy only knows the local spatiotemporal information of workers and tasks, and generally uses greedy algorithms and heuristic algorithms to obtain approximate optimal solutions. Most of the information contained in historical data is not fully mined in the solution process. The spatial crowdsourcing platform records various historical data in the process of task completion, including task details, task attributes, environmental data and task completion status, etc., through the learning of these historical data, it can provide more auxiliary information to optimize task allocation strategies [5]. Due to the continuity of time and space, the spatial crowdsourcing task allocation problem is also a multi-stage sequential decision-making problem. The current task allocation decision not only affects the current allocation result, but also affects the next decision [6]. For example, whether a worker can receive the next task after completing the task assigned by the current platform depends on the space-time environment in which the task is completed. This means that the crowdsourced task needs to be predicted through historical data and space-time correlation [7].

Forecasting methods based on time series data are traditionally relied on judgments according to experience and rules (i.e., autoregressive moving average model, autoregressive integrated moving average model, etc.), the accuracy and timeliness are insufficient. Now through machine learning methods (e.g., recurrent neural networks, convolutional neural networks, long short-term memory, gated recurrent unit and XGBoost, etc.), combined with scientific logical reasoning algorithms to make predictions, the efficiency and accuracy of predictions are greatly improved. However, the prediction process requires direct access to crowdsourced data, which may cause the privacy of crowdsourced workers to be leaked. This is because crowdsourced data contains private information about crowdsourced workers [8]. For example, a crowdsourced worker needs to obtain the temperature change in a certain sensing area. When this task and the perception interval are released to crowdsourcing workers through authorized agencies or publishing platforms, these crowdsourcing workers located in the perception area will feed back the perception results to the task publisher. In this process, in addition to some necessary information, such as worker's location information, etc., it has to be provided to each other and the authorized agency to complete the corresponding task. Other personal information, such as the worker's ID number and work unit, etc., are at risk of being leaked. Therefore, the privacy protection [9] of crowdsourcing workers in the process of crowdsourcing task prediction is an urgent problem to be solved.

On the other hand, as a public distributed ledger, blockchain has received extensive attention due to its openness and transparency, data immutability, and privacy protection. The blockchain is composed of a set of blocks in a peer-to-peer network, and each block uses a consensus algorithm to maintain data consistency. Each block is composed of a block header and a block body, containing its serialized transaction data, and the transaction hash value of all transactions on each block constitutes each leaf node of the Merkel tree. Due to the stability of the blockchain structure and the uniqueness of the block, the blockchain has been widely used in spatial crowdsourcing [10]. Since malicious crowdsourcing workers will steal private information carried by crowdsourcing tasks from the blockchain by means of collusion, this will seriously hinder the development of space crowdsourcing. Therefore, how to protect the data privacy of crowdsourcing tasks [11] through the blockchain is also an open problem.

Based on the above analysis, this paper first proposes a secure and efficient spatial crowdsourcing architecture (see Fig. 1), which consists of a mobile crowdsourcing worker layer, an intelligent spatial crowdsourcing task scheduling layer, and a spatial crowdsourcing application layer. Specifically, the intelligent crowdsourcing scheduling layer is not only responsible for allocating crowdsourcing tasks to mobile crowdsourcing workers, where the blockchain is used for the storage and release of crowdsourcing tasks, but also responsible for the prediction of crowdsourcing tasks. Intelligent algorithms are implemented on the basis of crowdsourced task data provided by mobile crowdsourced workers. In addition, the smart crowdsourcing task scheduling layer can support applications such as industrial equipment interconnection, industrial equipment optimization, industrial equipment maintenance, and industrial equipment tracking through reasonable prediction and allocation of crowdsourcing tasks.

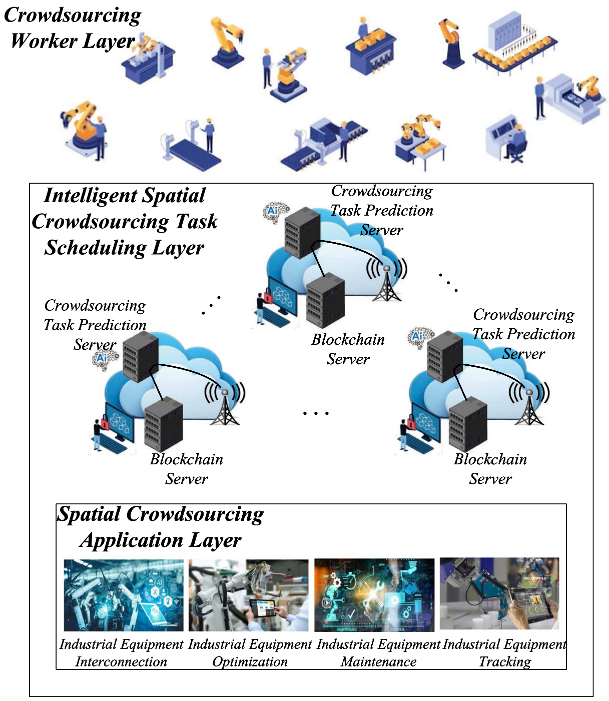

Crowdsourcing Worker Layer
Intelligent Spatial Crowdsourcing Task Scheduling Layer
Crowdsourcing Task Prediction Server
Blockchain Server
Crowdsourcing Task Prediction Server
Blockchain Server
Crowdsourcing Task Prediction Server
Blockchain Server
Spatial Crowdsourcing Application Layer
Industrial Equipment Interconnection
Industrial Equipment Optimization
Industrial Equipment Maintenance
Industrial Equipment Tracking

Fig. 1. The secure and efficient spatial crowdsourcing architecture.

Based on this architecture, we propose a Spatiotemporal Prediction based Spatial Crowdsourcing strategy, named SPSC, using both blockchain and artificial intelligence. The main contributions of this paper are as follows:

1) In order to achieve efficient spatial crowdsourcing, we use machine learning methods to predict the demand of crowdsourcing tasks from historical crowdsourcing data. In this way, the task release according to the predicted value can effectively improve the performance of the system. Specifically, we first analyze the historical task data of crowdsourcing, such as crowdsourcing task details, task attributes, environmental data and task completion, and the continuity in time and space. Then, given the inaccurate time series prediction by the gated recurrent unit (GRU) when the sample sequence contains a linear relationship or contains noise, while considering that the variational autoencoder (VAE) can continuously

eliminate incorrect samples during the encoding, decoding, and reconstruction of samples, both GRU and VAE are combined for the prediction of crowdsourcing tasks.

2) In order to achieve privacy protection in spatial crowdsourcing, we first protect the data privacy of crowdsourcing tasks by classifying crowdsourcing tasks and grouping crowdsourcing workers. Specifically, considering that different crowdsourcing tasks require crowdsourcing workers with different credits, we combine blockchain technology according to crowdsourcing tasks and crowdsourcing workers into multiple categories according to their credit demands, and assign crowdsourcing tasks with the corresponding credits, thereby reducing the risk of crowdsourcing workers conspiring to steal private data of crowdsourcing tasks. Secondly, we add noise to the historical task data of crowdsourcing to realize the privacy protection of crowdsourcing workers in the prediction of crowdsourcing tasks. Specifically, we add a larger Laplacian noise to the crowdsourcing task data with high credit demand, and add a smaller Laplacian noise to the crowdsourcing task data with low credit demand, thus realizing the differential privacy of crowdsourcing task data and meanwhile protecting the privacy of crowdsourcing workers.

3) Under the premise of protecting the privacy of historical data and tasks, the strategy proposed in this paper predicts the appropriate number of tasks to be issued through the analysis of historical data. And we verify the performance of the proposed strategy SPSC on a real dataset. Experimental results show that the proposed SPSC performs well in terms of throughput, latency, CPU utilization, memory utilization and privacy protection.

The organization of this paper is as follows. The related work is introduced in Section II. Both system model and attack model are given in Section III. The implementation of the proposed SPSC is elaborated in Section IV. The performance evaluation is conducted in Section V. We conclude this paper in Section VI.

II. Related Work

For the spatial crowdsourcing task allocation strategy, scholars have carried out a lot of related research. Cheng et al. [12] proposed task-priority greedy approach and game theoretic approach with two optimization methods to quickly solve the problem of how to assign workers to spatial tasks and achieve high total cooperation quality scores. Yu et al. [13] modeled the collaborative software task assignment problem in the crowdsourced environment as the assignment optimization problem, and integrates the three factors of worker capacity, task module complexity and worker active time to establish optimization goals. Based on the Hungarian algorithm, the optimization problem is solved by introducing a collaborative workgroup replacement strategy. Gao et al. [14] adopted truth inference methods to iteratively infer the truth and qualities. Based on the quality inference, this paper proposed a task assignment problem called quality-bounded task assignment with redundancy constraint. Different from traditional task assignment problem, redundancy constraint is added to satisfy the preliminaries of truth inference, which requires that each task should be assigned a certain or more amount of workers. Fan et al. [15] decomposed the task into phased sub-tasks undertaken by different workers on the basis of considering the worker's access rights and task time and space constraints to improve the probability of task completion. Wang et al. [16] based on the Markov model and collaborative filtering model, the similarities, trajectory prediction, dwell time, and trust degree are considered to propose the Markov and Collaborative filtering-based Task Recommendation (MCTR) model. Then, based on the Walrasian equilibrium, the optimum solution is researched to maximize the social welfare of mobile crowdsourcing systems. Liu et al. [17] proposed a utility-aware heuristic algorithm to address the task assignment problem, which can maximize the overall utility with adequate computation overhead. They further proposed a set of optimization techniques to enhance the design.

At present, there are some researches on crowdsourcing task allocation and privacy protection. Zeng et al. [18] proposed a novel bilateral privacy-preserving and accurate task assignment framework in fog-assisted MCS, called BRAKE. Specifically, they utilized the multisecret sharing scheme to preserve location privacy in the MCS task assignment, where tasks and workers only need to provide the secret shares of their real location information to fog nodes. Moreover, they considered distance-oriented and time-oriented tasks for assignment optimization and proposed an adaptive top-k worker selection algorithm to accurately select the most suitable workers. Zhao et al. [19] proposed a bilateral privacy-preserving Task Assignment mechanism for MCS (iTAM), which protects not only the task participants privacy but also the task requesters privacy and can minimize the travel distance. Furthermore, the strategy provides both equality and range constraints of task assignment by utilizing the Paillier cryptosystem. To accommodate the multiple relations between the task participants and the task, they proposed the single/multiple task participants selection problems for a task requiring task participants to compete and cooperate. Yuan et al. [20] proposed a privacy-preserving framework without online trusted third parties. They devised a grid-based location protection method, which can protect the locations of workers and tasks while keeping the distance-aware information on the protected locations such that they can quantify the distance between tasks and workers. They still proposed an efficient task assignment algorithm, which can instantly assign tasks to nearby workers on encrypted data. To protect the task content, they leverage both attribute-based encryption and symmetric-key encryption to establish secure channels through servers, which ensures that the task is delivered securely and accurately by any untrusted server. Wu et al. [21] introduce a fog-assisted SC architecture, in which many fog nodes deployed in different regions can assist the SC-server to distribute tasks and aggregate data in a privacy-aware manner. Specifically, a privacy-aware task allocation and data aggregation scheme (PTAA) is proposed leveraging bilinear pairing and homomorphic encryption. Zhang et al. [22] propose fog-assisted privacy-preserving task allocation in crowdsourcing by means of the advantages of fog computing, which can not only achieve privacy protection of both the task and the worker but also alleviate the workers' computational burden by offloading partial computation to the fog node. By applying threshold secret sharing technology, the proposed scheme enables that only workers satisfying task requirements can decrypt the task content, which achieves the verification of the workers' ability and resists the attacks of the greedy workers. Hao et al. [23] propose a privacy preserving interest-ability based task allocation scheme in crowdsourcing, which protects both task and worker privacy and enables the crowdsourcing server to allocate tasks in a fine-grained way. Specifically, by utilizing attribute-based encryption (ABE) and proxy re-encryption based searchable encryption (PRE-SE) on the task content and task tags respectively, customers are able to enforce fine-grained ability requirements on their tasks, and workers can specify flexible interests to choose their desired tasks. Wang et al. [24] propose a new algorithm, which can improve the efficiency of task allocation by disturbing the location of workers and task requesters through k-anonymity, 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. Yang et al. [25] also used the distributed consensus blockchain to complete the release of the group intelligence perception task of privacy protection, and this kind of collaboration may reduce the success rate of privacy protection due to the willingness of users to collaborate. Wu et al. [26] proposed PETA, a privacy-preserving edge task assignment framework for MCS, leveraging the powerful edge servers deployed between users and the platform to cluster and manage users according to user attributes. Furthermore, group signature is employed by PETA to anonymize and verify user identities for privacy-preserving task assignments.

Although the above research can provide efficient spatial crowdsourcing task allocation, there are still the following problems. First of all, how to use spatial crowdsourced historical task data with their temporal and spatial continuities for prediction of spatial crowdsourced missions should be considered. Second, how to protect the data privacy of crowdsourcing tasks and the privacy of crowdsourcing workers in spatial crowdsourcing? Third, how to combine blockchain technology and machine learning technology to improve the efficiency of spatial crowdsourcing task allocation? In order to solve these problems, this paper proposes a spatiotemporal prediction based spatial crowdsourcing strategy (SPSC) using both blockchain and artificial intelligence.

III. System Model

Crowdsourcing tasks often need to specify the task location, and the crowdsourcing task distribution platform needs to consider the distance between the current location of the crowdsourcing worker and the task location to ensure that the crowdsourcing worker can arrive at the task location within the specified time. If the crowdsourcing task distribution platform does not assign crowdsourcing tasks for a long time or crowdsourcing workers do not arrive at the task location within the specified time, users may cancel the tasks and reduce user satisfaction. Only after the crowdsourcing workers arrive at the task location can they obtain the benefits of task completion, and the current space-time environment will affect whether they can successfully receive the next task in the follow-up process. According to above analysis, we give the following definitions.

1) *Spatial Crowdsourcing Task:* Let T be the task set in the space crowdsourcing and $t_i \in T$ be the crowdsourcing spatial task in the set T represented by a seven tuples, i.e., $t_i =< p_s, p_e, t_s, t_f, t_c, ty, c >$, seven tuples for the, where p_s represents the start position of the task; p_e represents the end position; T_t represents the submission time of the task; t_f represents the failure time of the task; t_c represents the time required to complete the task; ty represents the task type, and different types of tasks need workers with different abilities to complete; c represents the credit requirement of the task, different types of crowdsourcing tasks require crowdsourcing workers of different credits. Workers can perform the task normally when they reach the starting position p_s before the expiration time $t_S + t_f$, and can complete the task only when it takes time to reach the position p_e.
2) *Spatial Crowdsourcing Worker:* Let W denote the set of workers in spatial crowdsourcing and $w_i \in W$ be a worker of set W represented by a seven tuple, i.e., $wi = < Id, p_c, t_r, t_w, sk, r, C >$, where Id represents the worker's identification; p_c represents the worker's current location; t_r represents the time when the worker starts to accept the task; t_w represents the worker's continuous working time; s_k represents the worker's skills indicating the type of task they are good at and the crowdsourcing task allocation platform will prioritize the tasks that the workers are inclined to for the improvement on the quality of task completion; r represents the working radius of the workers, and the workers will accept the tasks within their working radius. C represents the credit level of the crowdsourcing worker, workers of different levels can only receive the crowdsourcing tasks required by the corresponding level. Under this model, workers have working hours, and they need to accept tasks within the specified time. However, tasks require workers to spend time and the completion of tasks is accompanied by the movement of spatial positions.

In view of the above system model, this paper considers the following security threats in spatial crowdsourcing, namely the data privacy leakage of spatial crowdsourcing tasks and the privacy leakage of spatial crowdsourcing workers.

On the one hand, the data privacy leakage of the spatial crowdsourcing tasks is caused by malicious crowdsourcing workers launching a collusion attack. This is because different spatial crowdsourcing tasks contain a small amount of private information about the task publisher. If a large number of malicious crowdsourcing workers conspire, they can almost restore the private information they obtained to the complete

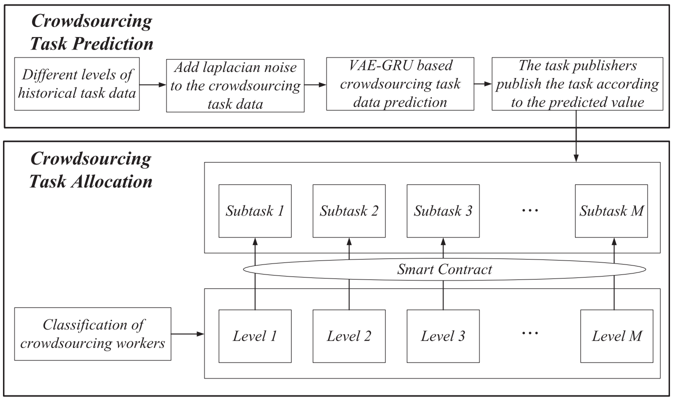

Crowdsourcing Task Prediction
Different levels of historical task data
Add laplacian noise to the crowdsourcing task data
VAE-GRU based crowdsourcing task data prediction
The task publishers publish the task according to the predicted value
Crowdsourcing Task Allocation
Subtask 1
Subtask 2
Subtask 3
···
Subtask M
Smart Contract
Classification of crowdsourcing workers
Level 1
Level 2
Level 3
···
Level M

Fig. 2. The workflow of SPSC.

private information of the task publisher, which will inevitably prevent the task publisher from continuing to publish spatial crowdsourced tasks. On the other hand, the prediction of spatial crowdsourcing tasks requires historical data of crowdsourcing tasks, and these data contain the private information of crowdsourcing workers. Once these data are leaked, it will seriously affect the participation of crowdsourcing workers in crowdsourcing tasks.

IV. The Implementation of the Proposed SPSC

Task allocation is the core issue of spatial crowdsourcing, and crowdsourcing task prediction can greatly improve the efficiency of task allocation [27]. In this paper, the gated recurrent unit and the variational autoencoder are combined to predict the spatial crowdsourcing task based on the spatial crowdsourcing historical task data with temporal and spatial continuity. In response to the leakage of data privacy of crowdsourcing tasks and the privacy of crowdsourcing workers in spatial crowdsourcing, this paper combines the blockchain technology to protect the data privacy of crowdsourcing tasks through the classification of crowdsourcing tasks and the grouping of crowdsourcing workers. Specifically, in our previous work, we added a security level attribute to the block header so that only tasks of the same level can be placed in the block. Similarly, only workers of the same level can receive and view the corresponding task content. Then add different Laplacian noises to crowdsourcing task data with different credit requirements to protect the privacy of crowdsourcing workers. Thereby, as shown in the Fig. 2, the proposed SPSC should include the following two modules, namely the blockchain based crowdsourcing task allocation module and the crowdsourcing task prediction module.

A. Blockchain Based Crowdsourcing Task Allocation

Blockchain is a new application mode of computer technology such as distributed data storage, point-to-point transmission, consensus mechanism, and encryption algorithm. It is essentially a decentralized database, and as the underlying technology of Bitcoin, it is a series of data blocks generated using cryptographic methods. Each data block contains a batch of Bitcoin network transaction information, which is used to verify the validity of the information and generate the next block. As a distributed shared ledger and database, the blockchain has the characteristics of decentralization, non-tampering, full trace retention, traceability, collective maintenance, openness and transparency, etc. These characteristics ensure the "honesty" and "transparency" of the blockchain, to lay the foundation for the blockchain to create trust.

Based on the above analysis, we use the blockchain as a distribution platform for spatial crowdsourcing tasks. That is, the spatial crowdsourced task publisher publishes the crowdsourced task on the blockchain, and the crowdsourced worker applies for the task from the blockchain and finally submits the task data to the crowdsourced task publisher through the blockchain. Considering that malicious crowdsourcing workers may steal private data in crowdsourcing tasks through collusion attacks, the classification of crowdsourcing tasks and grouping of employees has proved to be an effective means to resist collusion attacks of crowdsourcing workers. Based on our previous works [28], we have increased the number of crowdsourced task classifications and corresponding crowdsourced worker groups, that is, increased from the original three categories to six and nine categories. According to the proof given in literature [29] about the use of task classification and employee grouping to achieve data privacy protection, increasing the number of classifications can greatly reduce the chance of malicious crowdsourcing workers colluding. For each crowdsourcing worker, when a crowdsourcing task is posted on the blockchain, if the worker's credit meets the credit requirements of the crowdsourcing task, then the crowdsourcing worker can apply for the task. However, whether the crowdsourcing task can be successfully applied depends on whether the skills of the crowdsourcing worker meets the type of the crowdsourcing task. If it meets the requirements, the publisher of the crowdsourcing task will sign the smart contract on the crowdsourcing task with the crowdsourcing worker.

Smart contract is used to supervise every transaction between task publishers and crowdsourcing workers, which can effectively prevent any party from denying and deceiving related transactions. In our strategy, the smart contract includes contract for publishing tasks, contract for applying for tasks, and contract for submitting task data. First, contract for publishing tasks means that the task publisher specifies various data in the seven tuples t_i through the system, and then the smart contract automatically records the above information and creates a task agreement. Second, contract for applying for tasks means that when a crowdsourcing worker applies for a task that matches his own level, the worker will judge whether the distance between his current position p_c and the starting position p_s of the task is less than his working radius r, and at the same time, the worker will judge whether the type of the task t_y is within the worker's own skill s_k. If all the above conditions are met, the worker will sign the task with his private key $sign(w_i^{t_i})$ to ensure that the payment can be correctly allocated to the worker's account after the task is completed, and the contract to apply for the task will monitor and record the relevant information of the worker in the above process, including basic worker information w_i, worker credit

C, worker's account address A_{w_i}, and worker's private key signature $sign(w_i^{ti})$. Third, contract for submitting task data refers to the upload of the related data D_{w_i} after completing the task. The worker needs to sign it with the private key $sign(D_{w_i})$, and then encrypt it with the public key of the task publisher to ensure the task publisher can use his private key for decryption. The above content is the entire smart contract agreement process.

B. Crowdsourcing Task Prediction

Before using crowdsourced data with temporal and spatial continuity for spatial crowdsourced task prediction, we need to add noise to historical crowdsourced task data to achieve its differential privacy, thereby protecting the privacy of crowdsourcing workers.

1) Privacy Protection of the Historical Crowdsourcing Task Data: Differential privacy adds an appropriate amount of noise to the statistical results to ensure that the modification of an individual record in the data set will not have a significant impact on the statistical results, thereby meeting the requirements of privacy protection. Even if the attacker has control of all other data records except one piece of data, differential privacy can still prevent the attacker from analyzing the piece of data that he does not have, effectively avoiding the problem of privacy leakage caused by data release. We then give the definitions of the differential privacy and the Laplace distribution.

Definition 1: Set a random algorithm M, P_M is the set of all possible outputs of M. For any two adjacent data sets D and D' and any subset S_M of P_M, if algorithm M satisfies:

$$Pr[M(D) \in S_M] \leq e^{\epsilon} \times Pr[M(D') \in S_M], \tag{1}$$

then it preserves ϵ-differential privacy.

Definition 2: Set a random variable x, if the probability distribution function of x satisfies:

$$f(x|\mu, b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right), \tag{2}$$

then it follows the (μ, b) Laplace distribution, that is $x \sim Laplace(\mu, b)$.

It can be proved that if we add independent Laplacian noise to each crowdsourced historical task data by setting $\mu = 0$ and $b = \frac{1}{\epsilon}$, then the crowdsourced historical task data will have ϵ-differential privacy. In the process of adding Laplacian noise, the smaller the ϵ is, the higher the privacy protection degree is, otherwise the privacy protection degree is relatively low. Note that we add different levels of Laplacian noises to different crowdsourcing task data, i.e., a relatively high level of Laplacian noises will be added to the crowdsourcing task data with higher credit requirements.

2) VAE-GRU Based Crowdsourcing Task Prediction: The LSTM network is developed from the recurrent neural network (RNN). The simple RNN network has a memory function, but it also has a long-term dependence problem. The LSTM model is dedicated to solving the problem of long-term dependence. Its basic idea is to introduce a gating device to deal with the memory/forgetting, input level, and output level of the memory unit [30]. Because gate control is also a neural network, the learning of neural network parameters in gate control can let the machine know when to remember certain information and when to discard certain information. Compared with LSTM, the use of GRU can achieve considerable results, and is easier to train in comparison, which can greatly improve training efficiency.

GRU introduces the concept of reset gate and update gate. Therefore, the calculation method of the hidden state in the recurrent neural network is modified. The inputs of the reset gate and the update gate in the GRU are the current time step input X_t and the previous time step hidden state H_{t-1}, and the output is calculated by the fully connected layer whose activation function is the sigmoid function. Specifically, assuming that the number of hidden units is h, the small batch input $X_t \in R^{n\times d}$ (the number of samples is n, the number of inputs is d) at a given time step t and the hidden state at the previous time step $H_{t-1} \in R^{n\times h}$. The calculation of reset gate $R_t \in R^{n\times h}$ and update gate $Z_t \in R^{n\times h}$ is as follows:

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + br), \tag{3}$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + bz), \tag{4}$$

where $W_{xr}, W_{xz} \in R^{d\times h}$ and $W_{hr}, W_{hz} \in R^{h\times h}$ are weights, and $b_r, b_z \in R^{1\times h}$ are biases. If the value of the element in the reset gate is close to 0, it means that the corresponding hidden state element is reset to 0, that is, the hidden state of the previous time step is discarded. If the element value is close to 1, it means that the hidden state of the previous time step is retained. Then, the result of the element-wise multiplication is connected with the input of the current time step, and the candidate hidden state is calculated through the fully connected layer containing the activation function tanh, and the value range of all elements is $[-1, 1]$. Specifically, the candidate hidden state $H_t \in R^{n\times h}$ at time step t is calculated as

$$\hat{H}_t = \tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h), \tag{5}$$

where $W_{xh} \in R^{d\times h}$ and $W_{hh} \in R^{h\times h}$ are weights, and $b_h \in R^{1\times h}$ is the bias. Finally, the calculation of the hidden state $H_t \in R^{n\times h}$ at time step t is based on the update gate Z_t of the current time step, the hidden state H_{t-1} at the previous time step and the candidate hidden state $\hat{H}_t$ at the current time step as:

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \hat{H}_t. \tag{6}$$

VAE is an unsupervised generative network model, which consists of two parts: encoder and decoder. The encoder is used to learn the distribution of training data and generate compression values of the training data, and the decoder reconstructs the compressed data to reconstruct high-quality data to eliminate interference samples. The encoder learns the distribution of the input data, maps the input data to the mean μ and standard deviation σ of the data distribution, and samples ϵ in the standard normal distribution to generate the latent

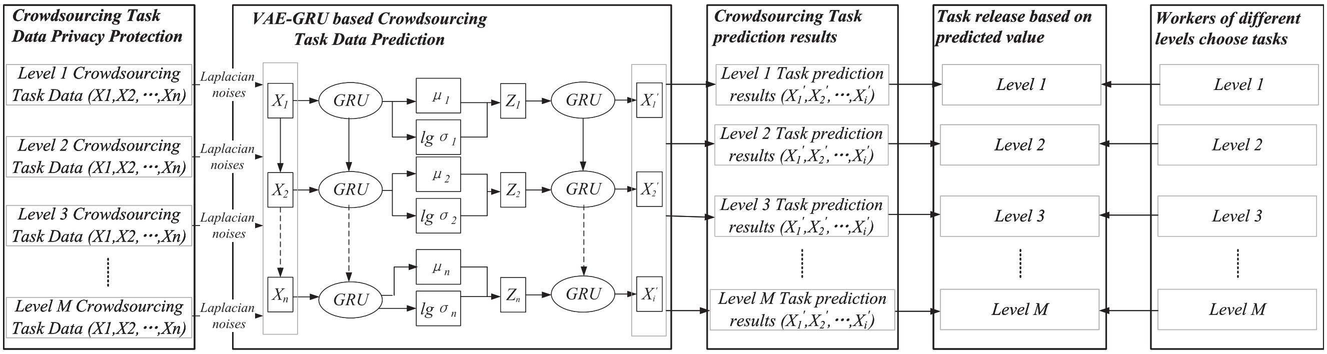

Crowdsourcing Task Data Privacy Protection
Level 1 Crowdsourcing Task Data (X1,X2,···,Xn)
Level 2 Crowdsourcing Task Data (X1,X2,···,Xn)
Level 3 Crowdsourcing Task Data (X1,X2,···,Xn)
Level M Crowdsourcing Task Data (X1,X2,···,Xn)
Laplacian noises
Laplacian noises
Laplacian noises
Laplacian noises
VAE-GRU based Crowdsourcing Task Data Prediction
X1
X2
Xn
GRU
GRU
GRU
μ1
lg σ1
μ2
lg σ2
μn
lg σn
Z1
Z2
Zn
GRU
GRU
GRU
X1'
X2'
Xi'
Crowdsourcing Task prediction results
Level 1 Task prediction results (X1',X2',···,Xi')
Level 2 Task prediction results (X1',X2',···,Xi')
Level 3 Task prediction results (X1',X2',···,Xi')
Level M Task prediction results (X1',X2',···,Xi')
Task release based on predicted value
Level 1
Level 2
Level 3
Level M
Workers of different levels choose tasks
Level 1
Level 2
Level 3
Level M

Fig. 3. VAE-GRU based crowdsourcing task prediction.

variable Z, i.e., $Z = \mu + \sigma \odot \epsilon$. In VAE, the network is optimized by maximizing evidence lower bound L_b as

$$L_b = E_{q_\phi(z|x)} \log p_\theta(x|z) - D_{KL}(q_\phi(z|x)|p_\theta(z)), \tag{7}$$

where $E_{q_\phi(z|x)}(\cdot)$ is the log-likelihood estimate of the posterior probability of x, representing the reconstruction probability; $D_{KL}(\cdot)$ is the KL divergence, used to measure the difference between the approximate posterior distribution and the standard Gaussian distribution. We design a VAE-GRU model for crowdsourcing task prediction based on VAE, and the structure is shown in Fig. 3. The input of the VAE-GRU model in the figure is the historical task data X of each level added with Laplacian noise, and then the GRU is used to predict the time series data, and then the VAE is used to reconstruct the data to improve the data accuracy. Finally, output a more accurate prediction sequence X'. We replace the BP neural network layer in the VAE encoder and decoder with the gate control unit GRU. This not only avoids the disappearance of the gradient in the feature extraction process of the crowdsourced historical task data, but also better encodes and decodes the spatiotemporal sequence information. Therefore, the prediction of time series data can be realized by GRU, and then the data can be reconstructed by VAE, thereby improving the accuracy of the data, so as to realize the prediction of crowdsourcing tasks.

V. Performance Evaluation

A. Experiment Setup

The proposed SPSC is implemented on a Hyperledger Fabric1.2-based simulator, which is available at https://github.com/hyperledger/fabric#releases. The physical machine runs with 16 G of memory, the Intel Core i7 processor with a frequency of 3.2GHZ is equipped with a 64-bit win10 system, and a VMware Workstation 14 Pro with 4 GB running memory and 2 processors of Ubuntu system. We use the Foursquare dataset [31] in this experiment, which contains 2,153,471 users, 1,143,092 places, 1,021,970 check-ins, 27,098,490 social relationships, and 2,809,581 ratings assigned by users to places and all these data including anonymous users' information and geographic locations recorded are extracted by the public API from Foursquare applications. In this dataset, each user has a unique ID and geospatial location of the user's hometown, while each place has a unique ID and Geospatial location. On this basis, each check-in contains a unique ID, a user ID and a place ID. Moreover, there are ratings about how much users like a particular place. We divide the Foursquare dataset to generate a set of spatial crowdsourcing tasks of different levels, in which the highest level of tasks have the highest credit requirements and vice versa.

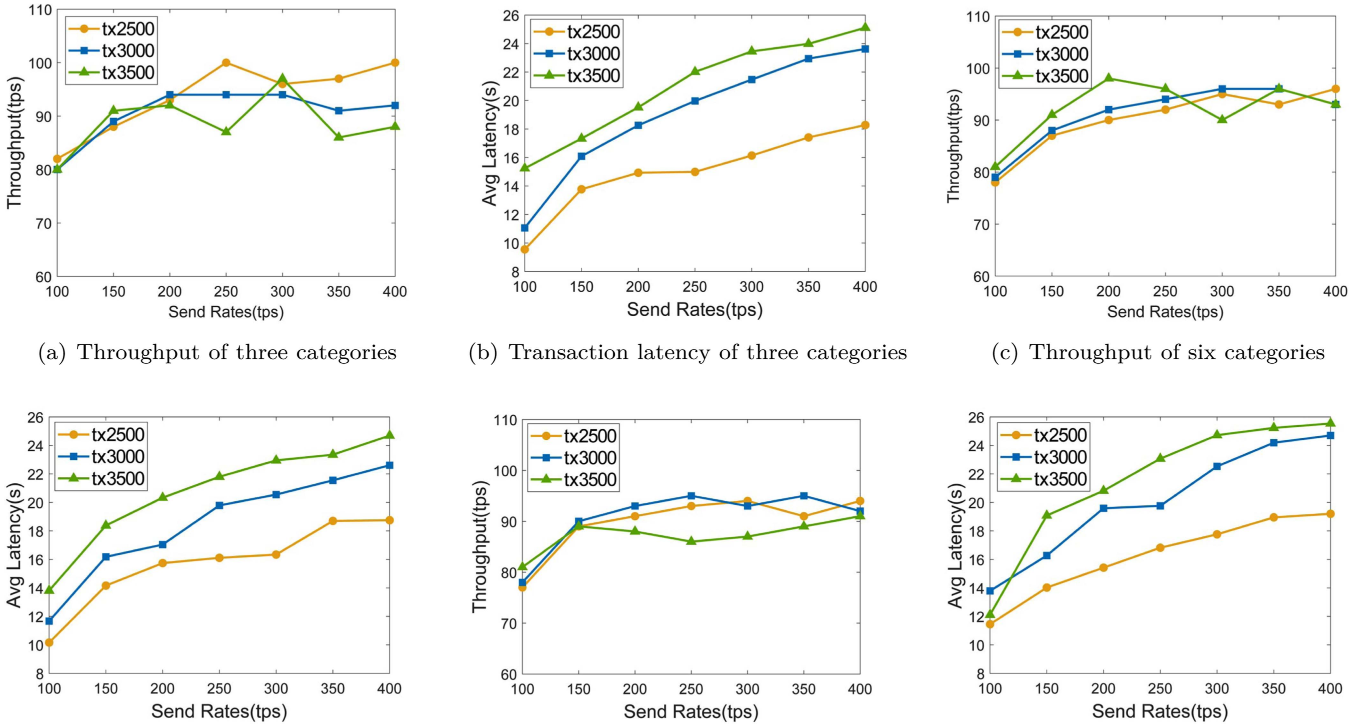
B. Performance Metrics

We first validate the performance of the SPSC in terms of system throughput, transaction latency, CPU usage and memory usage, considering the send rate. Then, we compare the performance of the SPSC with the baseline approaches BPDC [28] in terms of privacy protection, considering the number of crowdsourcing workers and the percentage of malicious crowdsourcing workers.

- *System Throughput:* A better throughput of the crowdsourcing system is attributed to a higher transaction processing speed.
- *Transaction Latency:* A better transaction processing capacity contributes to a lower latency in crowdsourcing.
- *CPU Usage:* A lower CPU usage will decrease the computation burden of crowdsourcing workers.
- *Memory Usage:* The performance of a crowdsourcing system partially depends on the memory usage.
- *Privacy Protection:* A well-designed crowdsourcing mechanism should be of privacy protection.

C. Experiment Results

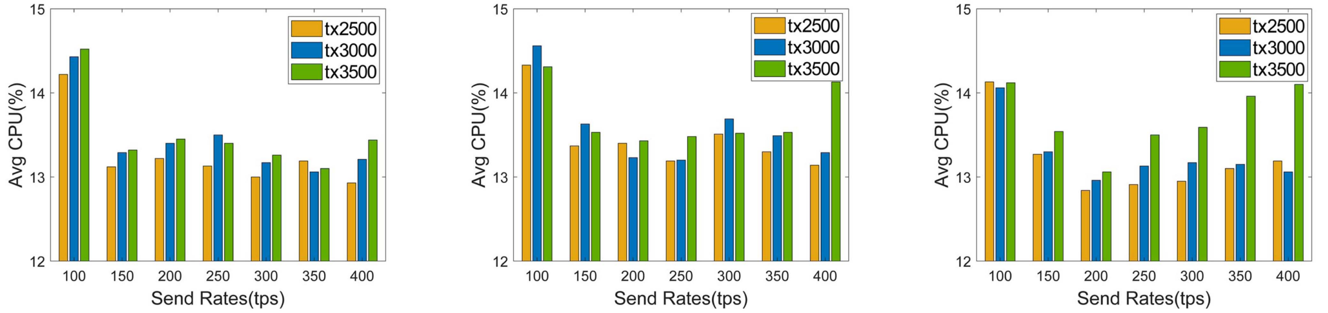
1) System Throughput: We first evaluate the throughput of the proposed strategy SPSC under different transaction sending rates. The throughput of each transaction number tx under different classifications are shown in Fig. 4(a), (c), (e). Seen from the results in Fig. 4(a), we find that when the transaction number is 2500, 3000 and 3500, the system throughput at different send rates is relatively close. Furthermore, the throughput of the system has basically maintained a stable and high level. For example, in Fig. 4(a), when $T = 3$ and the sending rate is 200tps, the throughputs of the transaction number tx of 2500, 3000 and 3500 are 93tps, 94tps and 100tps, respectively. In Fig. 4(c), when $T = 6$ and the sending rate is 250tps, the throughputs of the transaction number tx 2500, 3000 and 3500 are 92tps, 94tps and 96tps, respectively. Moreover, in Fig. 4(e), when $T = 9$ and the sending rate is 150tps, the throughputs of



(a) Throughput of three categories (b) Transaction latency of three categories (c) Throughput of six categories

(d) Transaction latency of six categories (e) Throughput of nine categories (f) Transaction latency of nine categories

Fig. 4. Throughput and transaction latency of SPSC under different classifications.



(a) CPU usage of three categories (b) CPU usage of six categories (c) CPU usage of nine categories

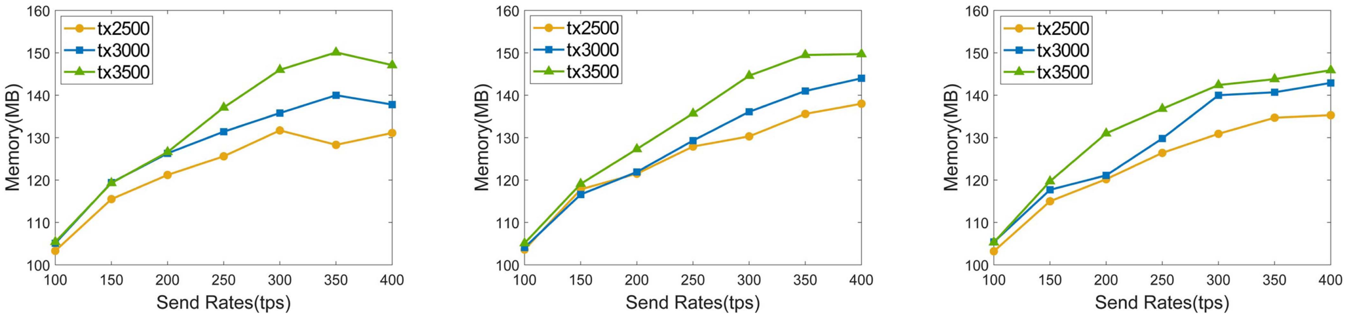
Fig. 5. CPU usage of SPSC under different classifications.

the transaction number tx of 2500, 3000 and 3500 are 89tps, 90tps and 89tps, respectively.

2) Transaction Latency: We compare the transaction latency at different transaction sending rates, and the test results are shown in Fig. 4(b), (d), (f). Observed from the result in Fig. 4(b), we find that the higher the transaction sending rate, the longer the latency. Meanwhile, as the numbers of transactions continues to increase, the corresponding latency is also increase. For example, in Fig. 4(b), when $T = 3$ and the sending rate is 300tps, the transaction latency of the transaction number tx of 2500, 3000, and 3500 are 16.14 s, 21.47 s and 23.45 s, respectively. In Fig. 4(d), when $T = 6$ and the sending rate is 200tps, the transaction latency of the transaction number tx of 2500, 3000, and 3500 are 15.74 s, 17.04 s and 20.33 s, respectively. Furthermore, in Fig. 4(f), when $T = 9$ and the sending rate is 400tps, the transaction latency of the transaction number tx of 2500, 3000, and 3500 are 19.20 s, 24.70 s and 25.54 s, respectively.

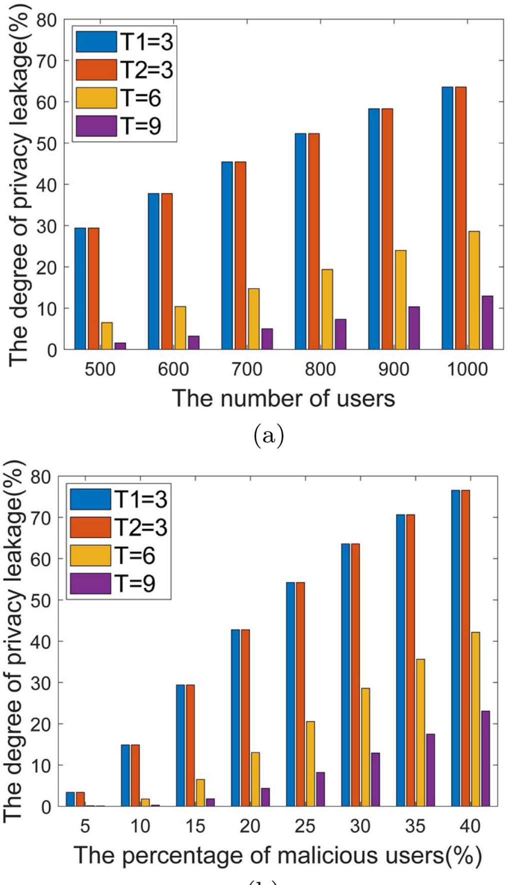
Both system throughput and transaction latency shown in Fig. 4 suggest that the proposed SPSC achieves excellent system performance and it can be applied to various crowdsourcing applications.

3) CPU Usage: We compare the CPU usage at different transaction sending rates, and the test results are shown in Fig. 5(a)–(c). As shown in Fig. 5(a), we know that a higher the transaction number contribute to a higher CPU usage as we expected. In addition, as the sending rate increases, the CPU usage also shows an upward trend. For example, in Fig. 5(a), when $T = 3$ and the sending rate is 100tps, the CPU usage of the transaction number tx of 2500, 3000, and 3500 are 14.22%, 14.43% and 14.52%, respectively. In Fig. 5(b), when $T = 6$ and the sending rate is 350tps, the CPU usage of the transaction



(a) Memory usage of three categories (b) Memory usage of six categories (c) Memorye usage of nine categories

Fig. 6. Memory usage of SPSC under different classifications.



(a)

(b)

Fig. 7. Comparison of privacy protection.

TABLE I
Comparison of Privacy Measures in Different Strategies

Strategies	Different privacy measures
BPDC	Crowdsourced tasks and workers can only be divided into three levels
SPSC	Crowdsourced tasks and workers can be divided into not only three levels, but also six or nine levels, with better privacy protection

number tx of 2500, 3000, and 3500 are 13.30%, 13.49% and 13.53%, respectively. In Fig. 5(c), when $T = 9$ and the sending rate is 200tps, the CPU usage of the transaction number tx of 2500, 3000, and 3500 are 15.42%, 19.59% and 20.82%, respectively.

4) Memory Usage: According to the number of tasks(tx) predicted by SPSC, we also test the memory usage at different transaction sending rates, and the test results are shown in Fig. 6(a)–(c). Observed from the result in Fig. 6(a), we find that as the sending rate and the number of participating crowdsourcing workers increases, the memory usage will also increase. For example, in Fig. 6(a), when $T = 3$ and the sending rate is 250tps, the memory usage of the transaction number tx of 2500, 3000, and 3500 are 125.6 MB, 131.4 MB and 137.1 MB, respectively. In Fig. 6(b), When $T = 6$ and the sending rate is 300tps, the memory usage of the transaction number tx of 2500, 3000, and 3500 are 130.3 MB, 136.1 MB and 144.6 MB, respectively. Furthermore, in Fig. 6(c), when $T = 6$ and the sending rate is 400tps, the memory usage of the transaction number tx of 2500, 3000, and 3500 are 135.3 MB, 142.9 MB and 145.9 MB, respectively.

The memory usage shown in Fig. 6 and the CPU usage shown in Fig. 5 indicate that the proposed SPSC performs excellently in various crowdsourcing scenarios.

5) Privacy Protection: We compare the degree of privacy protection of the proposed SPSC under different classifications with the baseline approach BPDC [28], and the results are shown in Fig. 7. The comparison of the privacy measures in the two strategies mentioned above is also shown in the Table I. First, we use a random function to generate the possibility of a malicious user leaking privacy in each category as test parameters. In Fig. 7(a), we set the proportion of malicious users to 30%, and then compare the degree of privacy leakage of different categories with different numbers of users. We can see clearly from Fig. 7 that as the number of users continues to grow, the degree of privacy leakage is also increasing. In Fig. 7, $T1 = 3$ represents the privacy protection degree of the proposed SPSC, $T2 = 3$ represents that of the baseline approach. We also can see that in the case of $T = 3$, the degree of privacy leakage of either approach is very close. However, SPSC still consider to divide crowdsourcing workers into six categories and nine categories. As shown in this figure, when crowdsourcing workers are classified into six or nine categories, the degree of privacy protection is better than when they are classified into three categories. We also find that the degree of privacy leakage is the highest when crowdsourcing

TABLE II
PERFORMANCE COMPARISON BETWEEN PREDICTING THE NUMBER OF TASKS AND NOT PREDICTING

Send Rates($T = 3$)	100tps	150tps	200tps	250tps	300tps	350tps	400tps
Throughput($tx = 3000$)	80tps	89tps	94tps	94tps	94tps	91tps	92tps
Throughput($tx = 4000$)	67tps	71tps	68tps	73tps	73tps	76tps	72tps
Avg Latency($tx = 3000$)	11.06s	16.09s	18.26s	19.97s	21.47s	22.94s	23.62s
Avg Latency($tx = 4000$)	17.39s	17.97s	22.27s	22.31s	24.28s	25.69s	27.16s
Avg CPU($tx = 3000$)	14.43%	13.29%	13.40%	13.50%	13.17%	13.06%	13.21%
Avg CPU($tx = 4000$)	20.01%	18.02%	18.32%	23.12%	26.02%	28.10%	28.81%
Memory($tx = 3000$)	105.1MB	119.4MB	126.3MB	131.4MB	135.8MB	140.0MB	137.8MB
Memory($tx = 4000$)	114.3MB	124.1MB	132.0MB	144.4MB	147.0MB	156.9MB	158.0MB

workers are classified into three categories, and the degree of privacy leakage is the lowest when users are classified into nine categories. For example, when the number of crowdsourcing workers reaches 600, the degree of privacy leakage approaches 37.76%, 10.39%, and 3.22% with $T = 3$, $T = 6$, and $T = 9$, respectively. In Fig. 7(b), we set the number of participating users to 1,000, and then compare the degree of privacy protection under different proportions of malicious crowdsourcing workers. The degree of privacy protection for each approach, when $T = 3$, is still very close to each other, but that of $T = 6$ and $T = 9$ of SPSC is significantly better than that of $T = 3$ of the baseline. For example, when the proportion of malicious crowdsourcing workers is 20%, the corresponding degree of privacy leakage when $T = 3$, $T = 6$, and $T = 9$ are 29.41%, 6.49%, and 1.81%, respectively. As a result, although SPSC performs no better than the baseline when $T = 3$, the advantage of SPSC over the baseline in terms of privacy protection emerges when crowdsourcing workers are divided into six and nine categories.

6) Performance Comparison Between Predicting the Number of Tasks and Not Predicting: After the strategy proposed above predicts the number of tasks, the task releaser then publishes tasks according to this value accurately. On this basis, the above has carried out corresponding tests on the system performance under different predicted values. Next, we will compare the two cases of predicting the number of tasks in advance and not predicting. As shown in the Table II, this paper compares system performance under the experimental parameters of $T = 3$ and predicted value $tx = 3000$. When the system does not have the function of task number prediction, the task releaser will blindly send different numbers of tasks to the system, resulting in poor task completion. From the Table II, we can clearly see that when there is no predicted value and the task releaser sends 4000 tasks to the system at once. Obviously, the throughput, latency, CPU utilization, and memory utilization of the system at this time are inferior to the situation with predictive value. For example, the throughput of $tx = 3000$ and $tx = 4000$ differs by 20tps at different send rates, and $tx = 3000$ is always better than $tx = 4000$. Next, the latency of $tx = 4000$ is longer than $tx = 3000$. When the send rate is 100tps, the maximum latency gap is 6.33 s. Then, when the send rate is 400tps, the CPU utilization of $tx = 4000$ is almost twice that of $tx = 3000$. Finally, the memory utilization of $tx = 4000$ is almost 1.2 times that of $tx = 3000$. In summary, sending the corresponding number of tasks according to the predicted value can not only improve the work efficiency of the task releasers, but also improve the performance of the system.

What's more, the SPSC proposed in this paper can effectively improve the performance of spatial crowdsourcing systems for the following reasons. The task prediction model proposed in this paper can effectively control the number of tasks in the application after predicting the number of tasks, and the value is an optimal prediction based on the current status of task receivers and task publishers. Therefore, compared with the case of no predicted value, that is, the task publishers publish tasks at will regardless of the actual situation [32], the strategy proposed in this paper can significantly improve the performance of traditional spatial crowdsourcing systems.

VI. Conclusion

Nowadays, spatial crowdsourcing can well complete the task allocation and data transmission in IIoT. It can not only allocate tasks reasonably, but also effectively protect the privacy information in the data transmission process. In this paper, we propose a spatiotemporal prediction based spatial crowdsourcing strategy (SPSC) using both blockchain and artificial intelligence. In SPSC, machine learning methods are used to predict the demand of crowdsourcing tasks from the historical crowdsourcing data. Specifically, the continuity in time and space of the historical task data of crowdsourcing is analyzed. Then, given the problem of inaccurate time series prediction by the GRU when the sample sequence contains a linear relationship or contains noise, while considering that the VAE has the advantage in eliminating incorrect samples during the encoding, decoding, and reconstruction of samples, the GRU and VAE based crowdsourcing task prediction method is developed. To achieve privacy protection in spatial crowdsourcing, the data privacy of crowdsourcing tasks is realized by classifying crowdsourcing tasks and grouping crowdsourcing workers using the blockchain. That is, crowdsourcing tasks are released on the blockchain such that crowdsourcing workers can only apply for the tasks of the corresponding credits to prevent crowdsourcing workers from conspiring to steal private crowdsourcing task data. Moreover, Laplacian noises are introduced to realize the differential privacy of crowdsourcing task data and meanwhile protecting the privacy of crowdsourcing workers. We conduct the validation experiment on a real dataset and the experimental results show that the proposed SPSC performs well in terms of throughput, latency, CPU utilization, memory utilization and privacy protection. To sum up, this paper mainly solves the problem of crowdsourcing task demand forecasting, but in order to achieve privacy protection, the classification process of crowdsourcing tasks and workers cannot be ignored. Based on this, in the follow-up work, we will focus on the classification strategy of crowdsourcing tasks and workers, that is, to achieve the optimal partitioning strategy in different regions, so as to obtain a more efficient crowdsourcing strategy.

References

[1] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Mechanism design for wireless powered spatial crowdsourcing networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 920–934, Jan. 2020.

[2] B. Guo, Y. Liu, L. Wang, V. O. K. Li, J. C. K. Lam, and Z. Yu, "Task allocation in spatial crowdsourcing: Current state and future directions," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1749–1764, Jun. 2018.

[3] Y. Tong, Y. Zeng, B. Ding, L. Wang, and L. Chen, "Two-sided online micro-task assignment in spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2295–2309, May 2021.

[4] L. Wang, Z. Yu, Q. Han, B. Guo, and H. Xiong, "Multi-objective optimization based allocation of heterogeneous spatial crowdsourcing tasks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1637–1650, Jul. 2018.

[5] L. Wang, Z. Yu, D. Zhang, B. Guo, and C. H. Liu, "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 84–97, Jan. 2019.

[6] A. K. Tripathi, K. Sharma, M. Bala, A. Kumar, V. G. Menon, and A. K. Bashir, "A parallel military-dog-based algorithm for clustering Big Data in cognitive industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 17, no. 3, pp. 2134–2142, Mar. 2021.

[7] Y. Yu, F. Li, S. Liu, J. Huang, and L. Guo, "Reliable fog-based crowdsourcing: A Temporal–Spatial task allocation approach," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3968–3976, May 2020.

[8] Z. Wang et al., "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1330–1341, Jun. 2019.

[9] L. Wang, D. Yang, X. Han, D. Zhang, and X. Ma, "Mobile crowdsourcing task allocation with differential-and-distortion geo-obfuscation," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 967–981, Mar./Apr. 2021.

[10] S. Zhu, Z. Cai, H. Hu, Y. Li, and W. Li, "zkCrowd: A hybrid blockchain-based crowdsourcing platform," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 4196–4205, Jun. 2020.

[11] C. Zhang, Y. Guo, X. Jia, C. Wang, and H. Du, "Enabling proxy-free privacy-preserving and federated crowdsourcing by using blockchain," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6624–6636, Apr. 2021.

[12] P. Cheng, L. Chen, and J. Ye, "Cooperation-aware task assignment in spatial crowdsourcing," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 1442–1453, doi: 10.1109/ICDE.2019.00130.

[13] D. Yu, Z. Zhou, and Y. Wang, "Crowdsourcing software task assignment method for collaborative development," *IEEE Access*, vol. 7, pp. 35 743–35 754, 2019.

[14] X. Gao, H. Huang, C. Liu, F. Wu, and G. Chen, "Quality inference based task assignment in mobile crowdsensing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3410–3423, Oct. 2021.

[15] Z. J. Fan et al., "Multi stage task allocation on constrained spatial crowdsourcing," *Chin. J. Comput.*, vol. 42, no. 12, pp. 2722–2741, 2019.

[16] Y. Wang, Z. Cai, Z. -H. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing," *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 4, pp. 1033–1046, Aug. 2020.

[17] Z. Liu, Z. Li, and K. Wu, "UniTask: A unified task assignment design for mobile crowdsourcing-based urban sensing," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6629–6641, Aug. 2019.

[18] B. Zeng, X. Yan, X. Zhang, and B. Zhao, "BRAKE: Bilateral privacy-preserving and accurate task assignment in fog-assisted mobile crowdsensing," *IEEE Syst. J.*, vol. 15, no. 3, pp. 4480–4491, Sep. 2021.

[19] B. Zhao, S. Tang, X. Liu, X. Zhang, and W.-N. Chen, "iTAM: Bilateral privacy-preserving task assignment for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 12, pp. 3351–3366, Dec. 2021.

[20] D. Yuan, Q. Li, G. Li, Q. Wang, and K. Ren, "PriRadar: A privacy-preserving framework for spatial crowdsourcing," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 299–314, 2020.

[21] H. Wu, L. Wang, and G. Xue, "Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 589–602, Jan.–Mar. 2020.

[22] J. Zhang, Q. Zhang, and S. Ji, "A fog-assisted privacy-preserving task allocation in crowdsourcing," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8331–8342, Sep. 2020.

[23] J. Hao, C. Huang, G. Chen, M. Xian, and X. Shen, "Privacy-preserving interest-ability based task allocation in crowdsourcing," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6, doi: 10.1109/ICC.2019.8761188.

[24] S. Wang, X. Jia, and Q. Sang, "A dual privacy preserving algorithm in spatial crowdsourcing," *Mobile Inf. Syst.*, vol. 2020, pp. 1–6, 2020, doi: 10.1155/2020/1960368.

[25] M. Yang, T. Zhu, K. Liang,W. Zhou, and R. H. Deng, "A blockchain-based location privacy-preserving crowdsensing system," *Future Gener. Comput. Syst.*, vol. 94, pp. 408–418, 2019.

[26] D. Wu, Z. Yang, B. Yang, R. Wang, and P. Zhang, "From centralized management to edge collaboration: A privacy-preserving task assignment framework for mobile crowdsensing," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4579–4589, Mar. 2021.

[27] A. Munusamy et al., "Service deployment strategy for predictive analysis of FinTech IoT applications in edge networks," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2021.3078148. .

[28] H. Lin, S. Garg, J. Hu, G. Kaddoum, M. Peng, and M. S. Hossain, "A blockchain-based secure data aggregation strategy using sixth generation enabled network-in-box for industrial applications," *IEEE Trans. Ind. Inform.*, vol. 17, no. 10, pp. 7204–7212, Oct. 2021.

[29] X. Wang, S. Garg, H. Lin, G. Kaddoum, J. Hu, and M. S. Hossain, "A secure data aggregation strategy in edge computing and blockchain empowered Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14 237–14 246, Aug. 2022, doi: 10.1109/JIOT.2020.3023588.

[30] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, "Anomaly detection for time series using VAE-LSTM hybrid model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 4322–4326, doi: 10.1109/ICASSP40776.2020.9053558.

[31] M. Sarwat, J. J. Levandoski, A. Eldawy, and M. F. Mokbel, "LARS*: An efficient and scalable location-aware recommender system," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 6, pp. 1384–1399, Jun. 2014. [Online]. Available: https://sites.google.com/site/yangdingqi/home/foursquare-dataset

[32] X. Hu, Y. Zhang, X. Liao, Z. Liu, W. Wang, and F. M. Ghannouchi, "Dynamic beam hopping method based on multi-objective deep reinforcement learning for next generation satellite broadband systems," *IEEE Trans. Broadcast.*, vol. 66, no. 3, pp. 630–646, Sep. 2020.