Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9191605, 12 pages
https://doi.org/10.1155/2022/9191605

Research Article

A Task Recommendation Model in Mobile Crowdsourcing

Yinglei Ji,[1,2] Chunxiao Mu,[1,2] Xiuqin Qiu,[1,2] and Yibao Chen[3]

[1]*School of Computer and Control Engineering, Yantai University, Yantai 264005, China*
[2]*Yantai Key Laboratory of High-End Ocean Engineering Equipment and Intelligent Technology, Yantai University, Yantai 264005, China*
[3]*School of Electromechanical and Automotive Engineering, Yantai University, Yantai 264005, China*

Correspondence should be addressed to Chunxiao Mu; mcx@ytu.edu.cn

Received 10 April 2022; Revised 26 May 2022; Accepted 24 June 2022; Published 14 July 2022

Academic Editor: Xu Zheng

With the development of the Internet of Things and the popularity of smart terminal devices, mobile crowdsourcing systems are receiving more and more attention. However, the information overload of crowdsourcing platforms makes workers face difficulties in task selection. This paper proposes a task recommendation model based on the prediction of workers' mobile trajectories. A recurrent neural network is used to obtain the movement pattern of workers and predict the next destination. In addition, an attention mechanism is added to the task recommendation model in order to capture records that are similar to candidate tasks and to obtain task selection preferences. Finally, we conduct experiments on two real datasets, Foursquare and AMT (Amazon Mechanical Turk), to verify the effectiveness of the proposed recommendation model.

1. Introduction

With the popularity of smart mobile devices, mobile crowdsourcing [1] is gradually emerging as a new type of data acquisition method, which uses smart mobile devices carried by people to sense their surroundings and obtain sensed data [2]. Since most of the smart mobile devices support GPS location services [3], people are able to easily engage in some location-based tasks, such as collecting information about a specific location (map application services), monitoring traffic conditions, assessing air pollution conditions [4], promoting nearby merchant stores, and checking supermarket merchandise shelves [5]. With the emergence of these task requirements, mobile crowdsourcing technologies [6, 7] have emerged and brought new challenges. Task allocation, privacy protection [8–10], and quality control [11, 12] are three core research issues in mobile crowdsourcing. The main research of this paper is task recommendation strategy in task allocation.

Unlike traditional crowdsourcing methods, workers can complete tasks online, and mobile crowdsourcing requires workers to complete location-based tasks offline before the deadline [13]. At present, the problem of rational allocation of tasks is still the focus of research in mobile crowdsourcing. The task allocation methods in mobile crowdsourcing systems are mainly divided into two ways: platform-centric active allocation and workers' independent choice [14, 15]. The system assigns tasks to workers with the platform's interests as the primary goal, ignoring the interest of workers and their willingness to perform the tasks. Therefore, the task completion rate of this allocation method is generally low and affects the task completion quality. In contrast, workers' independent choice often produces better results. However, as the number of workers and tasks in mobile crowdsourcing platforms continues to increase, it is difficult for workers to find tasks suitable for them from the huge amount of data. Lack of accurate description of task requirements makes it more difficult for workers to select tasks [16]. In order to reduce the search cost, workers mostly choose tasks that are recently released or ranked in the first two pages of the task list [17], and if they choose tasks that are not suitable for them, it will reduce workers' willingness to participate and eventually affect the quality of task completion. Therefore, how to recommend suitable tasks for workers from the massive task information becomes an urgent problem in mobile crowdsourcing.

In recent years, recommendation systems have been widely used to guide users to discover products that may be of interest to them from a large number of alternatives [18]. Compared with mainstream e-commerce and video push platforms, mobile crowdsourcing systems have unique features, such as mobile crowdsourcing tasks are not repeatable and exist in the crowdsourcing system for a short period of time, the task requirements posted by the requester and the services provided by the workers are limited by their interests and skills, and mobile crowdsourcing is an integrated online and offline data acquisition method that requires workers to move to a specific location at a specific time to complete the task and upload the sensed data [19].

Data in mobile crowdsourcing performs strongly spatial and temporal characteristics [20]. Whether a mobile crowdsourcing task can be successfully completed by a worker depends mainly on the worker's spatiotemporal behavior and interest preferences. In addition, worker mobility predictions have a significant impact on task completion rates [21]. In this paper, we propose a mobile crowdsourcing task recommendation method based on analyzing users' historical trajectories and task execution records. The main contributions of this paper are as follows:

(i) Based on the historical trajectory information of workers, we design a mobile prediction model to predict where the workers will appear at the next point in time or in which area. The movement pattern of workers varies over time. To achieve a comprehensive and accurate prediction, we first divide the workers' historical movement information into two parts, historical trajectory and current trajectory, and then extract the life patterns from historical trajectory that are similar to the current pattern based on the attention mechanism to interpret the current trajectory

(ii) To calculate the worker-task match, we design a neural network recommendation model by considering worker features such as next possible location, task completion records, and types of historical tasks, as well as task features such as type of task, location information, and payoff

(iii) An attention mechanism is added to the task recommendation model in order to capture the history of similarity to candidate tasks and to obtain the task selection preferences of workers

(iv) We conduct the comparison experiments on two real datasets, Foursquare as well as AMT (Amazon Mechanical Turk), to validate the effectiveness of the proposed task recommendation model

The rest of the paper is organized as follows. The related works are introduced in Section 2. Section 3 presents the mobile prediction model and introduces the task recommendation model for mobile crowdsourcing system. The comparison experiments, as well as the analysis for the experimental results, are introduced in Section 4. Finally, Section 5 concludes this paper.

2. Related Works

With the rapid growth in the number of workers and tasks [22], it has become increasingly difficult for workers to select the right task. The task assignment for workers and tasks is often a time-consuming process [23, 24]. The time cost by a worker to select a suitable task is even comparable to the time cost to complete a task [25]. That is why a personalized task recommendation mechanism is particularly important [23]. How to recommend suitable tasks for workers has become a research hotspot in current crowdsourcing systems.

Some of the existing task recommendation models are implemented based on a probabilistic matrix factorization approach. Yuen et al. [26] found that the previously proposed classification-based task recommendation failed to consider the dynamic scenarios in which new employees and new tasks appear in the crowdsourcing system. They proposed task recommendation (TaskRec) framework based on a uniform probabilistic matrix factorization, which aims to recommend tasks for workers in dynamic scenarios. The framework does not require workers to provide task ratings, which can be inferred from workers' interaction behaviors. Later, they found that different categories of tasks require workers with different skills. They proposed an active learning probabilistic matrix factorization model (ActivePMFv2) based on the TaskRec framework [27]. The model takes into account worker preferences as well as historical performance when recommending tasks, and they actively select the most uncertain tasks for the most reliable workers to retrain the classification model. Safran and Che [28] proposed two Top-N recommendation algorithms (Top-N-Tasks and Top-N-Workers) for crowdsourcing systems by extending matrix factorization and kNN. Then it outputs the most suitable tasks to workers and identifies the best workers for the requester. Dai et al. [29] proposed a feature-based Bayesian task recommendation (BTR) scheme. BTR learns the task selection preferences of workers through their historical behavior and analyzes the hidden information of tasks based on task features instead of task ID.

Social relationships are a critical factor influencing workers' task selection. Some scholars have provided task lists by analyzing the social attributes of workers and referring to the task choices of similar workers. Wang et al. [30] proposed a universal recommendation method based on personal social-collaboration preferences, which was used to recommend the social-collaboration tasks that individuals can participate in. Li et al. [31] obtained behavioral information of users from social networks as well as crowdsourcing platforms. After that, they analyzed the suitability of the candidate tasks in a comprehensive manner in terms of worker preferences, historical records, and social relationships. After the recommendation is completed, requesters and workers are asked to fill out feedback questionnaires. The effectiveness of the proposed recommendation mechanism is evaluated by analyzing the participants' feedback. Li et al. [32] formally defined the Accurate Recommendation Problem (ARP) and proved the problem was NP-Hard. To address this problem, they designed a task recommendation system, Pioneer-Assisted Task Recommendation (PATRON) framework. The framework first selects a set of pioneer workers to collect initial knowledge about the new task and then uses a k-medoids clustering algorithm to divide the workers into subsets based on worker similarity. The efficiency of worker selection is improved through this method. Pan et al. [33] established a semantic tags similarity matrix database based on the Word2vec deep learning method. Through computing the similarity of tags, the correlation between task and worker and the similarity between workers are obtained, which achieves personalized task recommendations for workers. Wang et al. [34] used the cosine similarity theorem to calculate the similarity between participants and obtained the ranking model through learn-to-rank algorithm. Finally, a task recommendation list is generated according to the ranking model.

Some scholars believe that there is a risk of data leakage in the process of task recommendation, and they incorporate privacy-preserving algorithms in the recommendation framework. Shu et al. [35] argued that the existing task recommendation schemes may reveal private and sensitive information about tasks and workers. To protect privacy, they proposed a privacy-preserving task recommendation scheme (PPTR) that enables task-worker matching while protecting task and worker privacy. Tang et al. [36] proposed a privacy-preserving task recommendation scheme with win-win incentives in a crowdsourcing environment by developing advanced attribute-based encryption techniques combined with preparation/online encryption and outsourced decryption techniques. Achieving a balance between privacy, profit, and utility of data is a major challenge for data sharing in the industrial IoT [37]. Qi et al. [38] proposed a privacy-aware data fusion and prediction method based on locally sensitive hashing technique in order to ensure data privacy along with data availability. Gong et al. [39] proposed a flexible recommendation framework by balancing the three metrics of utility, privacy, and efficiency in mobile crowdsourcing systems. In order to protect worker locations in the absence of a trusted database owner, Zhang et al. [40] proposed a task recommendation scheme for location privacy-preserving based on geometric range queries.

In addition to the above recommendation methods, some other scholars have proposed their recommendation structures by considering worker preferences, historical records, multiparty gains, and workers' reputation. Kurup et al. [41] collected workers' preferences and motivational factors from crowdsourcing platform and analyzed that the success rate of bidding is the main factor to affect workers' willingness to participate in the tasks. Therefore, they proposed a task recommendation scheme that considered the participation probability and winning probability of workers. It designs probability models by analyzing workers' performance, historical behavior, metadata, and participation logs. Wang et al. [42] considered the effect of potential interest of workers as well as the trust of tasks. The potential recommendation probability of the task is predicted based on the worker's dwell-time, and the trust of tasks is obtained by analyzing the reputations of tasks and the participation frequencies of workers. Aldahari et al. [43] assumed that each crowd participant aims to maximize their own profit. Because of this assumption, they presented a mechanism based on multiobjective recommendation system to enhance holistic satisfaction. They considered three objectives: (1) find suitable tasks that fit the workers' interests and skills to increase workers' rewards and willingness to participate, (2) provide the proper solution to the requester at a lower cost of time and labor, and (3) raise the acceptance rate of the task to increase the revenue of the crowdsourcing platform. Miao et al. [44] designed a task-worker matching model (Budget-TASC) that balances quality and budget by considering the reputation of the worker and the proximity to the task execution location. The algorithm reduces the budget by considering the travel cost spent by the worker to perform the task.

Most of the above researches on task recommendation are only applicable to crowdsourcing systems that complete tasks online. Considering the characteristics of task sensing in mobile crowdsourcing systems, this paper summarizes the following problems of the existing recommendation models.

(i) Most models failed to consider the worker's mobile preferences, such as predicting the next location or area the worker will visit by analyzing historical trajectory information

(ii) When computing the match between workers and tasks, most recommendation models failed to consider the task execution records of workers or do not consider the different importance of different historical tasks for candidate tasks

(iii) Each worker has different task selection preferences, which means that the importance of each task feature is different for every worker. However, few recommendation models trained different task feature selection weights for workers

3. The Proposed Recommendation Model

Workers are a critical part of the mobile crowdsourcing system. With the rise of mobile crowdsourcing and the popularity of mobile wireless devices, interest-based crowd workers occupy an increasing share of the workforce [45]. The willingness of these workers to perform their tasks depends on their availability. They do not depend on performing tasks to make ends meet and prefer to do so without interfering with their normal lives. For this group of workers, accurate location prediction is the key to improving the success rate of recommendations.

In this paper, reasonable tasks are recommended to workers by uncovering their spatiotemporal behavior patterns. The next location workers may prefer to go to can be predicted in advance, allowing the assignment of spatiotemporal tasks that are consistent with or adjacent to the predicted location, which not only saves workers' travel costs but also increases the attractiveness of tasks to workers.

The flow chart of the proposed task recommendation method is shown in Figure 1.

Considering the spatiotemporal characteristics of mobile crowdsourcing systems, the recommendation model consists of two main components.

(1) Identify candidate regions for the recommendation model, based on the historical mobile trajectory of workers to discover their life patterns and predict the next location and area they are likely to visit.

This allows workers to complete their tasks without disrupting their normal lives

(2) Compute the degree of matching between candidate tasks and workers. The task selection preferences of the workers are obtained by analyzing the tasks they have performed. Whether a worker will perform a candidate task depends on if the features of that task match the worker's task selection preferences. The candidate tasks are taken from the possible visit areas obtained in the previous step

3.1. Problem Definition. In the mobile crowdsourcing system, workers $W = \{w_1, w_2, \cdots, w_m\}$ are randomly online, while there are tasks $T = \{t_1, t_2, \cdots, t_n\}$ posted by requesters in the system. The feature information of task t_i is represented by $infor_{t_i} = \{id, type, t_{\text{cost}}, loc, sal, title, dem\}$, where $type$ denotes the type of the task, t_{cost} denotes the time required to complete the task, loc denotes the location where the task is performed, sal denotes the reward that can be obtained by completing the task, and dem denotes the specific requirements of the task.

The feature information of worker w_j is represented by $infor_{w_j} = \{traj, task, task_{\text{type}}, loc\}$, where $traj$ denotes the historical trajectory sequence of the worker, $task$ denotes the list of tasks performed in the past, $task_{\text{type}}$ denotes the type of the performed task, and loc denotes the next location that the worker is likely to visit.

The trajectories of each worker are arranged in chronological order. The trajectory of worker w_x is defined as

$$T^{w_x} = \left(l_{t_1}^{w_x}, l_{t_2}^{w_x}, l_{t_3}^{w_x}, \cdots, l_{t_n}^{w_x}\right), x \in [1, m], \tag{1}$$

where m denotes the number of workers and $l_{t_i}^{w_x}$ denotes the ith location visited by w_x at time t_i. Then all trajectories of workers are divided into several subtrajectories, $P_1^{w_x}, P_2^{w_x}, P_3^{w_x}, \cdots, P_k^{w_x}$, where k represents the number of subtrajectories. Since the time of location recording is random, the time interval of each subtrajectory and the length of the subtrajectory are fixed at a uniform value. The time interval of each subtrack is set to 7 days, and the length of subtracks is capped at 10.

Let $T_{\text{history}}^w = (P_1^w, P_2^w, P_3^w, \cdots, P_n^w)$ be the historical trajectory of worker w and $T_{\text{current}}^w = P_{n+1}^w = (l_1^w, l_2^w, l_3^w, \cdots, l_v^w)$ be the current trajectory. The problem to be solved by the mobile prediction model can be described as predicting the next location l_{v+1}^w based on the worker's historical trajectory T_{history}^w and the current trajectory T_{current}^w. What the task recommendation model needs to do is to compute the matching degree between $infor_{w_j}$ and $infor_{t_i}$ and recommend suitable tasks for workers.

3.2. RNN and LSTM. Neural network (NN) consists of a large number of interconnected artificial neurons, which are capable of adaptively changing their internal structure based on external input information. The tasks that NN can handle are classified into classification and regression problems, and different problems can be solved by adjusting the structure and the number of neurons. The common network structures in NN are convolutional neural network (CNN), which have achieved excellent results in image and speech recognition, and recurrent neural network (RNN), which are used to process sequential data.

In a general RNN structure, the neurons in the hidden layer are connected by weights (the later neurons are affected by the earlier neurons), so RNN has memory. However, when the sequence is too long, the early inputs have less influence on the later results, which leads to the problem of gradient disappearance and directly affects its memory capability. To solve this problem, variants of RNN, LSTM, and GRU, were proposed. Nowadays, LSTM is widely used for long input problems. The mobile prediction model is constructed based on the LSTM structure, and Equations (2)–(7) explain the principle of LSTM.

$$f_t = \sigma\left(x_t W_x^{(f)} + h_{t-1} W_h^{(f)} + b^{(f)}\right), \tag{2}$$

$$i_t = \sigma\left(x_t W_x^{(i)} + h_{t-1} W_h^{(i)} + b^{(i)}\right), \tag{3}$$

$$o_t = \sigma\left(x_t W_x^{(o)} + h_{t-1} W_h^{(o)} + b^{(o)}\right), \tag{4}$$

$$g_t = \tanh\left(x_t W_x^{(g)} + h_{t-1} W_h^{(g)} + b^{(g)}\right), \tag{5}$$

$$c_t = f_t \odot c_{t-1} + g_t \odot i_t, \tag{6}$$

$$h_t = o_t \odot \tanh(c_t), \tag{7}$$

where $W_x^{(f)}, W_x^{(i)}, W_x^{(o)}, W_x^{(g)}$ are the weight matrices of each control gate, $b^{(f)}, b^{(i)}, b^{(o)}, b^{(g)}$ are the bias vectors, c_t is the memory stored in the neuron at time t, h_t is the hidden state, and $\odot$ is the dot product operation.

3.3. Attention. In recommendation system, local feature extraction based on regions of interest has proven to be very successful [46]. The attention mechanism is essentially similar to the human selective visual attention mechanism. It can use limited attention resources to quickly filter out the high-value parts from a large amount of information. It removes the limitation that traditional encoder-decoder structures rely on an internal fixed-length vector for both encoding and decoding. The input information is selectively learned by the model and will be associated with the output sequence [47].

In the mobile prediction model, historical mobile patterns that are similar to the worker's current trajectory are extracted through the attention mechanism. These patterns can assist in predicting the next possible location to be visited by the worker. The principle of the attention mechanism is shown by Equation (8).

$$c_j = \sum_{i=1}^{T} \alpha_{ij} h_i, \tag{8}$$

where T is the number of historical trajectories. The parameter α_{ij} denotes the importance of historical visit location i to

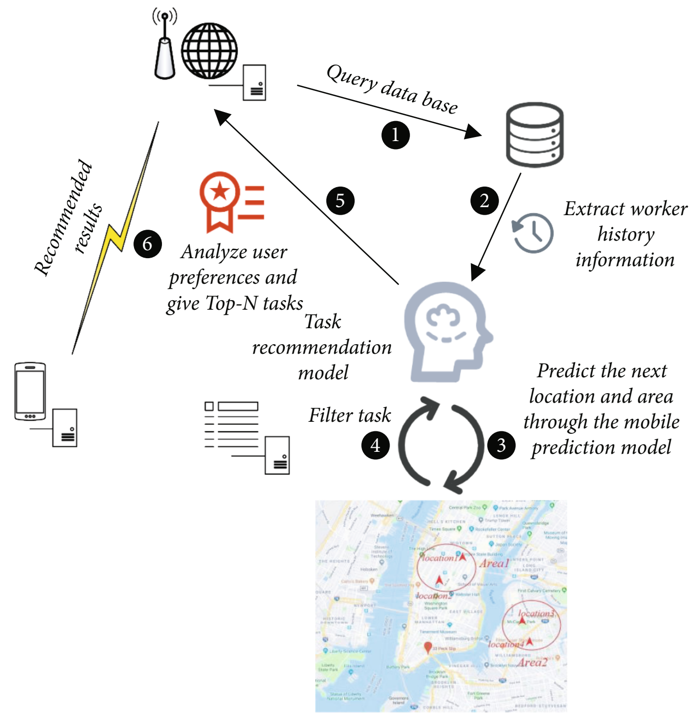

Figure 1: The process of task recommendation.

current location j, h_i is the hidden state of the encoder, which holds the historical memory, and c_j is the weighted sum of historical trajectory memory. In this way, the model can dynamically extract historical records that are useful for the current decision through the attention mechanism.

3.4. Mobile Prediction. The proposed mobile prediction model is built based on the encoder-decoder structure. The structure diagram is shown in Figure 2.

As shown in Figure 2, the historical trajectory sequence as well as the current trajectory sequence is input to the model, and each location in the sequence is encoded with one-hot. If there are 10,000 locations, each location would have to be represented by a 10,000-dimensional vector. The data would be sparse, so we need to complete a dimensionality reduction process. A variable matrix is embedded to achieve the dimensionality reduction, which can be 10000×200. The vector of location is multiplied with the embedding matrix to obtain a dense vector of 200 dimensions. The elements in the matrix will change as the model is trained, and the dense vector after dimensionality reduction represents the corresponding location more accurately.

After dimensionality reduction by embedding layer, a trajectory sequence pair $(x_1, x_2, \cdots, x_n)$ and $(y_1, y_2, \cdots, y_v)$ consisting of dense vectors is obtained, where n and v are not necessarily equal. The encoder processes $(x_1, x_2, \cdots, x_n)$ through Equations (2)-(7). After n time steps of processing, the encoder generates multiple semantic vectors h_t. The computation of h_t can be expressed by

$$h_t = f(x_t, h_{t-1}, c_t), \tag{9}$$

where f is the runtime function of RNN, x_t is the current input, h_{t-1} is the hidden state information at the previous time step, and c_t is the cell state at the current time. Subsequently, the hidden state vector h_t at each time step is integrated into a two-dimensional matrix h_s and input to the attention layer in the prediction stage of the decoder. At the same time, the decoder inputs the hidden state vector h_t formed at each time step to the next memory unit and the upper layer of the attention network. After that, the hidden state vector of the current time step and the weighted sum of h_s are jointly input to the affine layer of the upper layer. Finally, the Softmax layer is added to calculate the probability distribution of the predicted locations based on the output of the previous affine layer.

3.5. Task Recommendation. This subsection introduces the task recommendation model proposed in this paper. To recommend tasks to workers more effectively, an attention-based task recommendation model is designed. We use several attributes of tasks and workers to learn their representations instead of just using their ids. Given the characteristics of mobile crowdsourcing, we comprehensively analyze task characteristics such as task type, time spent, execution location, reward, title, and requirement description, as well as worker characteristics such as completed tasks, task type, and next location, in order to fully explore workers' task selection preferences.

The model structure is shown in Figure 3. The input of the model is divided into two parts: one part is the feature group of workers, and the other part is the feature group of candidate tasks. Both feature groups contain two attributes, task ID and task type. The task ID in the worker's feature group is a sequence that represents the tasks that worker

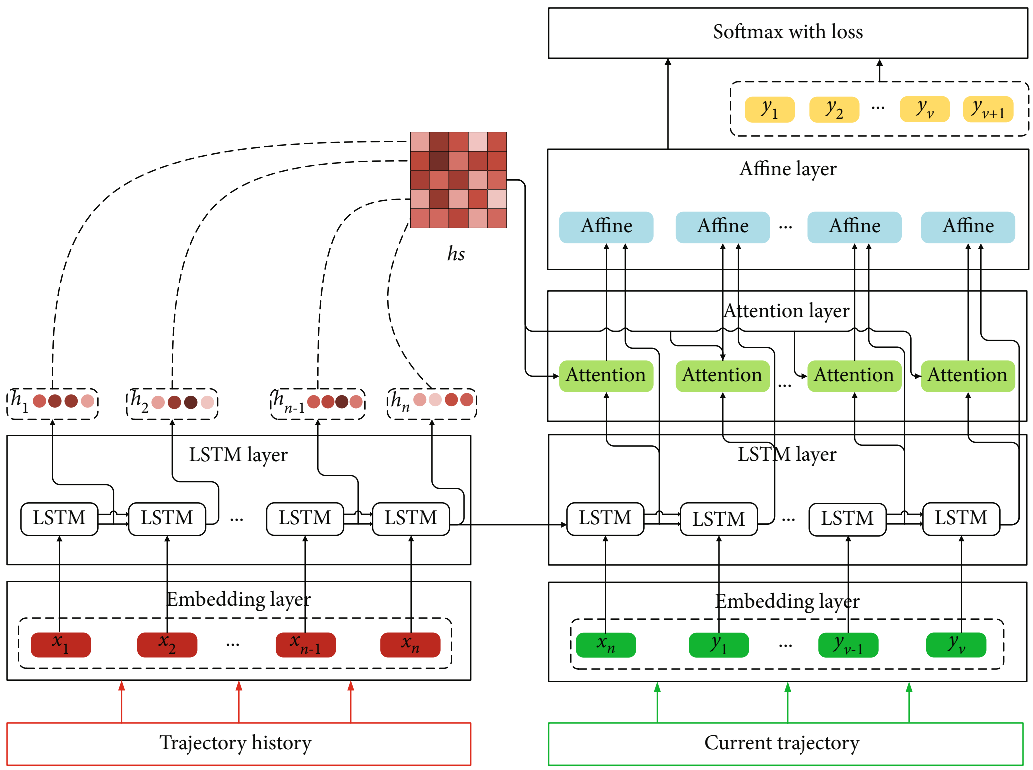

FIGURE 2: Mobile prediction model structure diagram.

has performed before. Firstly, the sparse features are transformed into dense vectors by embedding layer. Word2vec has been verified to have better textual representation [48], so textual features like titles as well as task descriptions are transformed into vectors by Word2vec. Then the two feature groups are stitched together to get the feature vector of the candidate task and the feature vector of the worker, respectively.

Given that the tasks completed by a worker have a different impact on whether a candidate task is executed or not, we add the attention mechanism to the extraction of worker features. The correlation between the candidate tasks and the historical tasks is used to calculate a weight, which represents the strength of attention. The formal expression of the attention component is shown by

$$V_w = f(V_t) = \sum_{i=1}^{N} w_i \bullet V_i = \sum_{i=1}^{N} g(V_i, V_t) \bullet V_i, \qquad (10)$$

where the vector V_w is the weighted sum of historical tasks, V_t is the embedding vector of candidate tasks, V_i is the embedding vector of the worker's ith task, w_i is the attention weight of the ith task, and g is the weight calculation function.

Finally, the obtained worker feature vector is spliced with the task feature vector and fed into a deep neural network (DNN) with ReLU as the activation function. The DNN forms an output vector after three feature crossovers. The vector is mapped by the Sigmoid function to a value between 0 and 1, which indicates the matching degree between the worker and the candidate task.

4. Experiments and Analysis

4.1. Datasets. We conduct comparison experiments on two real datasets, Foursquare-NYC and Amazon Mechanical Turk (AMT), to verify the validity of the model. The Foursquare-NYC dataset is a long-term check-in data collected by Foursquare in New York City from April 12, 2012, to February 16, 2013. Foursquare is a service based on user location information, and they encourage users to share information about their current location with others. The format of the dataset is as follows: user ID, location ID, category ID, category name, longitude, latitude, and timestamp. The Foursquare-NYC dataset contains 227247 check-in data from 1083 users. The location in the dataset is used to represent the task execution location in mobile crowdsourcing system. The user check-in location represents the current location of the worker, and a successful check-in at a specified location means that the worker has arrived at the task execution location and completed the task.

AMT is a well-known crowdsourcing platform. In order to collect the data needed for a task recommendation system, tasks can be posted to the AMT, and information about workers can be collected for analysis. However, the data is difficult to obtain because collecting it often requires backend access. Existing research typically uses synthetic data or small-scale data collected internally. The datasets are collected from the NAACL 2010 Working Group (http://sites.google.com/site/amtworkshop2010/data-1), which is the most comprehensive crowdsourcing dataset published to date. These datasets are collected by different crowd workers to perform various types of tasks, mainly speech-text tasks.

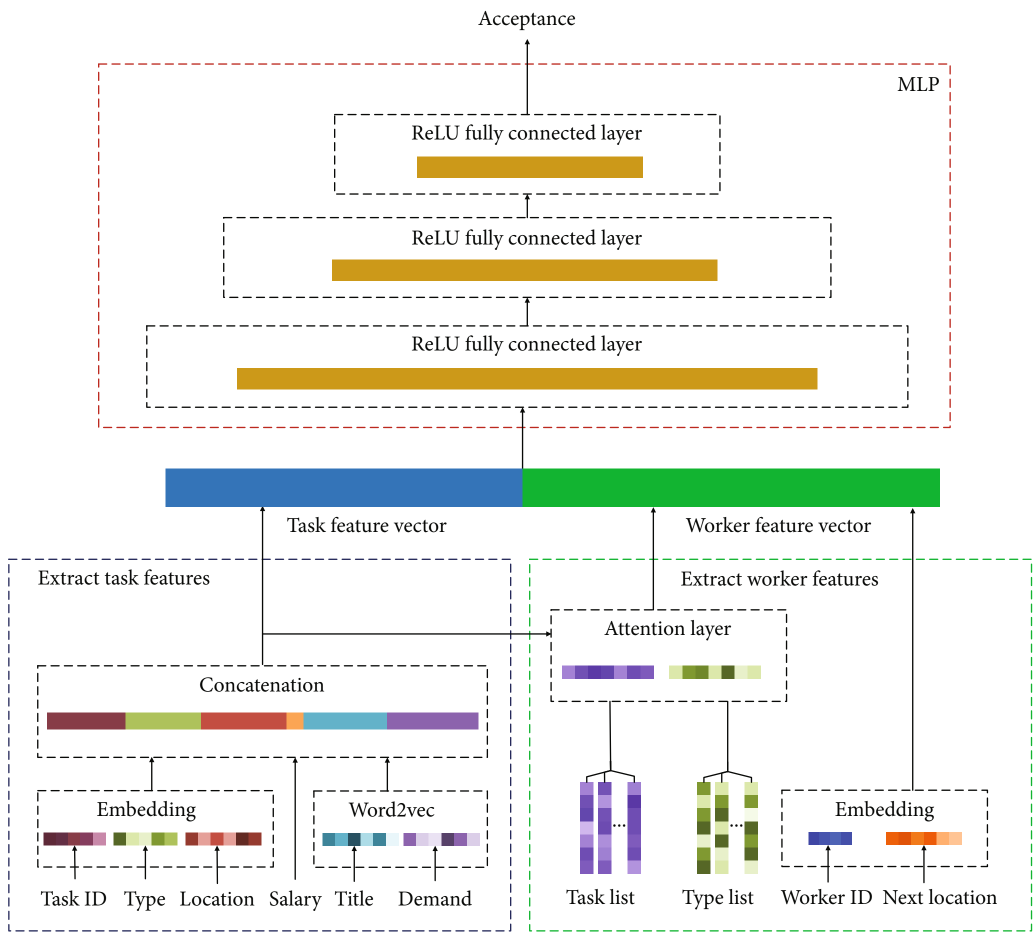

Figure 3: Task recommendation model structure diagram.

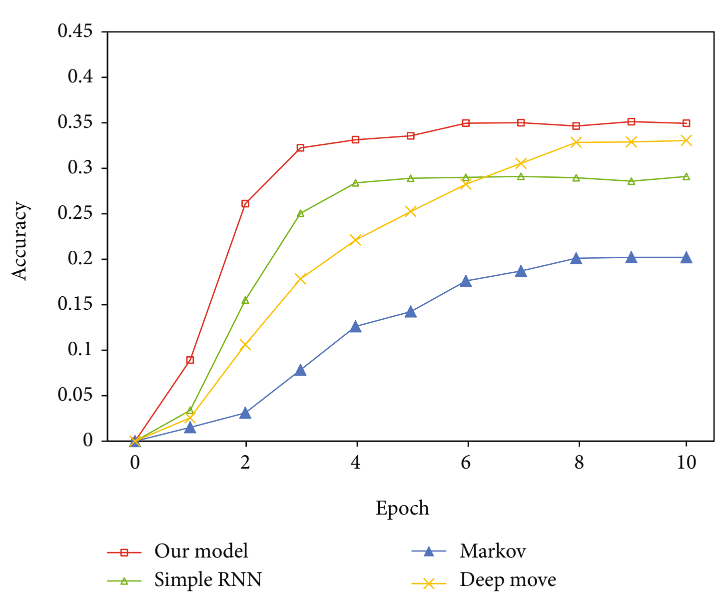

Figure 4: Foursquare-NYC Top@5.

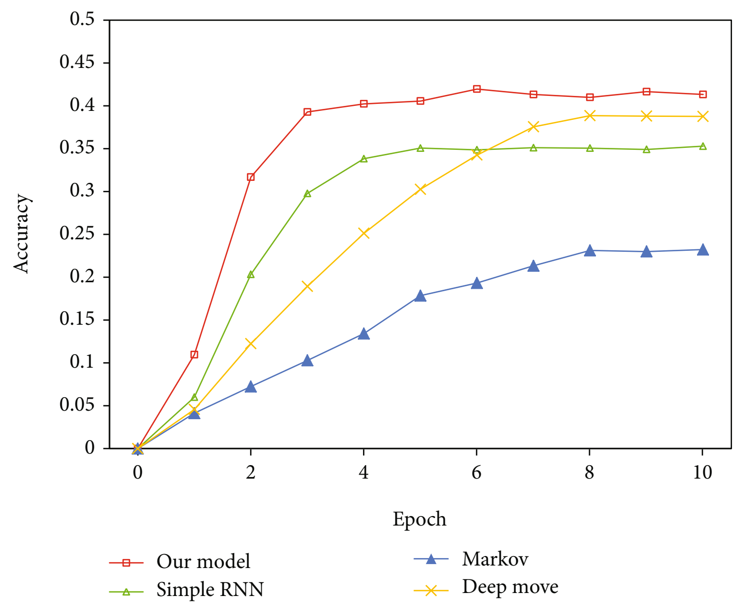

Figure 5: Foursquare-NYC Top@10.

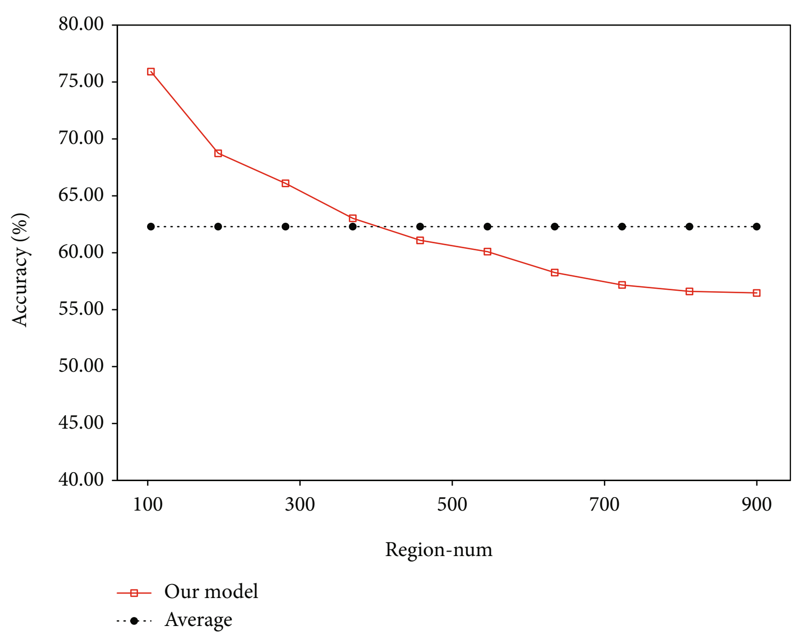

Figure 6: Effect of number of regions on accuracy.

Table 1: Recall and F1-score.

	Markov	simpleRNN	DeepMove	OurModel
Recall	0.213	0.257	0.286	0.324
F1-score	0.235	0.276	0.311	0.358

There are 21 subdatasets available from NAACL 2010 Working Group. However, the format of some datasets does not meet the requirements; 9 usable subdatasets are filtered from them. The selected datasets all contain the following attributes, user ID, HIT ID, HITTYPE ID, title, task description, task keyword, reward, maximum number of assignments, and the ID of the worker who completed the task. After data processing, the 25170 tasks of 1135 workers are obtained.

In addition, for each dataset, 80% of the data from each user is used as the training set, and 20% of the data is used as the test set.

4.2. Evaluation Metrics. For the mobile prediction model, the effectiveness of the model is evaluated by the Top@k as well as the F1-score. Top@k can be described as the hit rate of the top k predicted outcomes with the highest probability and is

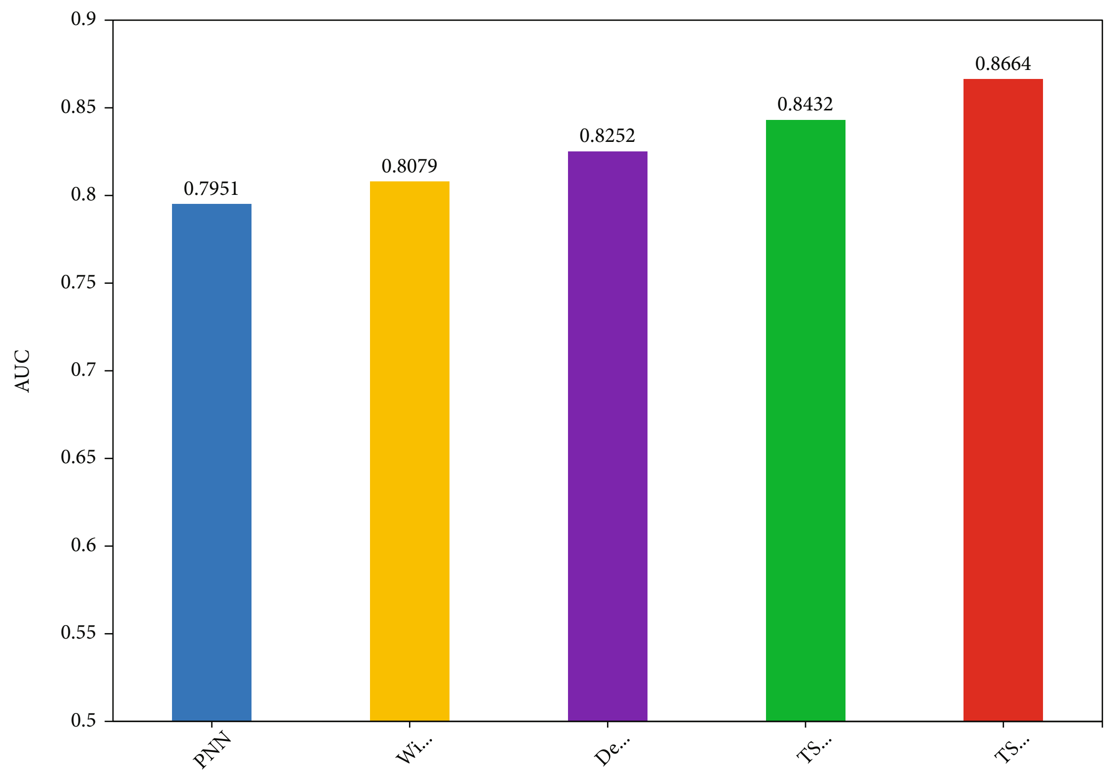

Figure 7: AUC comparison result on AMT datasets.

calculated as shown by

$$Top@k = \frac{|\{n \in N : f(n) \in F_k(n)\}|}{|N|}, \tag{11}$$

where $f(n)$ is the standard value, $F_k(n)$ is the k highest probability predictions, and $|N|$ can be expressed as the total number of workers in the system.

Precision and Recall are the two main metrics for evaluating Top-N recommendations. Precision indicates the likelihood that the user will select the recommended item, and Recall indicates the probability that the item the user wants to select will be successfully recommended. The F1-score is determined by both Precision and Recall, and a larger F1-score indicates a better overall prediction. The calculation methods for Precision, Recall, and F1-score are shown as follows.

$$Precision = \frac{1}{N}\sum_{n=1}^{N}\frac{|R_n \cap L_n|}{|L_n|}, \tag{12}$$

$$Recall = \frac{1}{N}\sum_{n=1}^{N}\frac{|R_n \cap L_n|}{|R_n|}, \tag{13}$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}, \tag{14}$$

where N is the total number of workers, R_n is the list of workers' bid records, L_n is the recommendation list generated by the model for workers, and $|L_n|$, $|R_n|$ are the lengths of the two lists.

AUC is used to evaluate the effectiveness of the task recommendation model. Unlike F1-score, AUC evaluates the probability that given a random positive sample and a negative sample, the model's prediction probability for the positive sample is greater than the model's prediction probability for the negative sample. It can reflect the ranking ability of the model. AUC is calculated by

$$AUC = \frac{\sum P_{\text{positive}} > P_{\text{negative}}}{\text{positiveNum} \times \text{negativeNum}}, \tag{15}$$

where P_{positive} is the predicted probability of a positive sample and P_{negative} is the predicted probability of a negative sample.

4.3. Analysis of Experimental Results. In order verify the effectiveness of the mobile prediction model, we compare the experimental results of Markov, simpleRNN, and the recently proposed DeepMove.

The Top@k results of each model on the Foursquare-NYC dataset are shown in Figures 4 and 5. By analyzing the experimental results, it can be found that the Markov model has the worst results. Because its feature is that the next predicted location is only related to the previous location, it cannot discover the life pattern from the workers' trajectories, which affects its prediction effect. The simpleRNN model with a single RNN neural network structure has the fastest convergence speed because of its simple structure. It has some memory ability, so the effect is much better than Markov. However, when the sequence length is too long, the simpleRNN model cannot remember the earlier inputs, so the effect is not satisfactory compared to the complex model. The DeepMove model takes into account the multilayer periodicity of historical trajectories by applying the attention mechanism and implements modularity. From the experimental results, we can see that its effect is better than Markov and simpleRNN models, but it has the slowest convergence speed due to its complex design.

The proposed model outperforms the other models, while the convergence speed is only slightly worse than that of the simpleRNN model. From Figures 4 to 5, it can be seen that the Top@5 and Top@10 accuracy rates of our proposed model are improved by 20.6% and 18.9% over the simpleRNN model. The improvement is 6.2% and 8% compared to the DeepMove model. Unlike DeepMove, our model adopts encoder-decoder structure and LSTM as the basic unit of recurrent neural network. The DeepMove model only filters out the parts related to the current trajectory from the historical trajectory information and does not extract the movement patterns of workers. In contrast, our model applies recurrent neural network to the historical trajectories, which makes the model more comprehensive in mining the movement patterns of workers from the historical information. So our model can analyze the movement patterns of workers more thoroughly and with higher accuracy.

We also analyzed the correlation between the Top@5 accuracy and the number of divided regions. Figure 6 shows that as the number of divided regions increases, the accuracy rate decreases. When there are fewer regions, each region contains a larger range, so the workers' regional locations do not change much and the model can easily predict the next region. As the regions are divided more and more finely, the workers' locations become more frequently changed and the difficulty of location prediction increases. When the number of regions is 100, the accuracy reaches a maximum of 76%. When the number of regions increases to a maximum of 1000, the model can also guarantee a prediction accuracy of more than 55%. The average prediction accuracy of the model can reach more than 60%.

The comparison results of Recall and F1-score of each model are shown in Table 1. It can be seen that the Recall and F1-score of the proposed model in this paper are the highest compared with other baseline models, which also indicates that its classification accuracy is the best.

We conducted comparison experiments on the AMT dataset to compare with several advanced recommendation models in the recommendation system. The results of the experiments are shown in Figure 7. From the experimental results, we can see that the TSKReco model proposed in this paper significantly outperforms all the other models on the AMT dataset. In order to verify the effect of worker location prediction on the task recommendation model, we train the model with and without the location prediction module, respectively. From Figure 7, it can be seen that the AUC improves from 0.8432 to 0.8664 after adding the location prediction module, which is about 2.8%. The results show that it is successful to add location prediction to task recommendation and also verify the importance of worker location prediction in mobile crowdsourcing system.

5. Conclusions

For the spatiotemporal characteristics of mobile crowdsourcing, this paper proposes a task recommendation model that incorporates worker's mobile location prediction. First, we construct an effective mobile prediction model, which can uncover the mobile patterns of workers by analyzing their trajectory information and predict the future location areas based on the patterns. Then, in order to more accurately recommend user-preferred tasks for users, we analyze the task selection preferences of workers from their historical task records, including task type preferences and reward expectations. The model integrates workers' selection preferences and candidate task features to calculate the match between them and finally gives the most suitable task recommendation list. The candidate tasks can be filtered out based on prediction regions. The results of comparison experiments on two real datasets, Foursquare-NYC and AMT, show that the recommendation model proposed in this paper has high accuracy and efficiency and also prove that the method based on worker trajectory analysis can indeed improve the success rate of task recommendation.

In future works, we will collect larger scale data from mobile crowdsourcing platforms to improve the generalization capability of the model. In addition, in order to have a more accurate portrayal of workers, we will consider introducing more worker characteristics, including worker skills and reputation.

Data Availability

The dataset used in this paper are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The study was supported by the School and Local Integration Development Project: collaborative innovation platform for intelligent monitoring technology development and product application for smart ocean (No. JS19JY08).

References

[1] Y. Wu, J. R. Zeng, H. Peng, H. Chen, and C. P. Li, "Survey on incentive mechanisms for crowd sensing," *Journal of Software*, vol. 27, no. 8, pp. 2025–2047, 2016.

[2] 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 Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 1033–1046, 2020.

[3] S. Wang, Q. Li, M. Xu, X. Ma, A. Zhou, and Q. Sun, "Tiansuan constellation: an open research platform," http://arxiv.org/abs/2112.14957.

[4] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*, pp. 144–153, IEEE, Dallas, TX, USA, 2019.

[5] Y. Wang, Y. Gao, Y. Li, and X. Tong, "A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems," *Computer Networks*, vol. 171, article 107144, 2020.

[6] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *Proceedings of the 20th international conference on advances in geographic information systems*, pp. 189–198, Redondo Beach, California, 2012.

[7] Y. Tong, L. Chen, and C. Shahabi, "Spatial crowdsourcing: challenges, techniques, and applications," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1988–1991, 2017.

[8] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 1–590, 2016.

[9] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[10] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–38, 2021.

[11] F. Li, Y. Wang, Y. Gao, X. Tong, N. Jiang, and Z. Cai, "Three-party evolutionary game model of stakeholders in mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, pp. 1–12, 2021.

[12] C. Chi, Y. Wang, X. Tong, M. Siddula, and Z. Cai, "Game theory in Internet of Things: a survey," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12125–12146, 2022.

[13] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "SLADE: a smart large-scale task decomposer in crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1588–1601, 2018.

[14] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 635–644, IEEE, Atlanta, GA, USA, 2017.

[15] Y. Wang, Z. Cai, Z. H. Zhan, Y. J. Gong, and X. Tong, "An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 414–429, 2019.

[16] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, and S. Yu, "Compatibility-aware web api recommendation for mashup creation via textual description mining," *ACM Transactions on Multimidia Computing Communications and Applications*, vol. 17, no. 1s, pp. 1–19, 2021.

[17] J. H. Feng, G. L. Li, and J. H. Feng, "A survey on crowdsourcing," *Chinese Journal of Computers*, vol. 38, no. 9, pp. 1713–1726, 2015.

[18] C. Shi, X. Han, L. Song et al., "Deep collaborative filtering with multi-aspect information in heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1413–1425, 2021.

[19] Z. Lu, Y. Wang, Y. Li, X. Tong, C. Mu, and C. Yu, "Data-driven many-objective crowd worker selection for mobile crowdsourcing in industrial IoT," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[20] F. Kou, J. Du, Z. Lin et al., "A semantic modeling method for social network short text based on spatial and temporal characteristics," *Journal of Computational Science*, vol. 28, pp. 281–293, 2018.

[21] Y. Ding, L. Zhang, and L. Guo, "Dynamic delayed-decision task assignment under spatial-temporal constrains in mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2418–2431, 2022.

[22] Z. Sun, Y. Wang, Z. Cai, T. Liu, X. Tong, and N. Jiang, "A two-stage privacy protection mechanism based on blockchain in mobile crowdsourcing," *International Journal of Intelligent Systems*, vol. 36, no. 5, pp. 2058–2080, 2021.

[23] D. Geiger and M. Schader, "Personalized task recommendation in crowdsourcing information systems-current state of the art," *Decision Support Systems*, vol. 65, pp. 3–16, 2014.

[24] M. C. Yuen, I. King, and K. S. Leung, "Task recommendation in crowdsourcing systems," in *Proceedings of the first international workshop on crowdsourcing and data mining*, pp. 22–26, Beijing, China, 2012.

[25] L. B. Chilton, J. J. Horton, R. C. Miller, and S. Azenkot, "Task search in a human computation market," in *Proceedings of the ACM SIGKDD workshop on human computation*, pp. 1–9, Washington, DC, 2010.

[26] M. C. Yuen, I. King, and K. S. Leung, "Taskrec: a task recommendation framework in crowdsourcing systems," *Neural Processing Letters*, vol. 41, no. 2, pp. 223–238, 2015.

[27] M. C. Yuen, I. King, and K. S. Leung, "An online-updating algorithm on probabilistic matrix factorization with active learning for task recommendation in crowdsourcing systems," *Big Data Analytics*, vol. 1, no. 1, pp. 1–29, 2016.

[28] M. Safran and D. Che, "Efficient learning-based recommendation algorithms for top-n tasks and top-n workers in large-scale crowdsourcing systems," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 1, pp. 1–46, 2019.

[29] W. Dai, Y. Wang, J. Ma, and Q. Jin, "BTR: a feature-based Bayesian task recommendation scheme for crowdsourcing system," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 780–789, 2020.

[30] J. Wang, Z. Wang, and J. Li, "A recommendation method for social collaboration tasks based on personal social preferences," *IEEE Access*, vol. 6, pp. 45206–45216, 2018.

[31] Y. M. Li, C. Y. Hsieh, L. F. Lin, and C. H. Wei, "A social mechanism for task-oriented crowdsourcing recommendations," *Decision Support Systems*, vol. 141, article 113449, 2021.

[32] Z. Li, B. Cheng, X. Gao, H. Chen, and G. Chen, "A unified task recommendation strategy for realistic mobile crowdsourcing system," *Theoretical Computer Science*, vol. 857, pp. 43–58, 2021.

[33] Q. Pan, H. Dong, Y. Wang, Z. Cai, and L. Zhang, "Recommendation of crowdsourcing tasks based on Word2vec semantic tags," *Wireless Communications and Mobile Computing*, vol. 2019, 10 pages, 2019.

[34] J. Wang, J. Liu, Z. Zhao, and G. Zhao, "A task recommendation framework for heterogeneous mobile crowdsensing," *The*

Journal of Supercomputing, vol. 77, no. 10, pp. 12121–12142, 2021.

[35] J. Shu, X. Jia, K. Yang, and H. Wang, “Privacy-preserving task recommendation services for crowdsourcing,” *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 235–247, 2018.

[36] W. Tang, K. Zhang, J. Ren, Y. Zhang, and X. (Sherman) Shen, “Privacy-preserving task recommendation with win-win incentives for mobile crowdsourcing,” *Information Sciences*, vol. 527, pp. 477–492, 2020.

[37] X. Zheng and Z. Cai, “Privacy-preserved data sharing towards multiple parties in industrial IoTs,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[38] L. Qi, C. Hu, X. Zhang et al., “Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2021.

[39] Y. Gong, L. Wei, Y. Guo, C. Zhang, and Y. Fang, “Optimal task recommendation for mobile crowdsourcing with privacy control,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 745–756, 2016.

[40] C. Zhang, L. Zhu, C. Xu, J. Ni, C. Huang, and X. S. Shen, “Location privacy-preserving task recommendation with geometric range query in mobile crowdsensing,” *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[41] A. R. Kurup, G. P. Sajeev, and M. I. Senior, “A task recommendation scheme for crowdsourcing based on expertise estimation,” *Electronic Commerce Research and Applications*, vol. 41, article 100946, 2020.

[42] Y. Wang, X. Tong, K. Wang, B. Fan, Z. He, and G. Yin, “A novel task recommendation model for mobile crowdsourcing systems,” *International Journal of Sensor Networks*, vol. 28, no. 3, pp. 139–148, 2018.

[43] E. Aldahari, V. Shandilya, and S. Shiva, “Crowdsourcing multi-objective recommendation system,” in *Companion Proceedings of the The Web Conference 2018*, pp. 1371–1379, Lyon, France, 2018.

[44] C. Miao, H. Yu, Z. Shen, and C. Leung, “Balancing quality and budget considerations in mobile crowdsourcing,” *Decision Support Systems*, vol. 90, pp. 56–64, 2016.

[45] T. Liu, Y. Wang, Y. Li, X. Tong, L. Qi, and N. Jiang, “Privacy protection based on stream cipher for spatiotemporal data in IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7928–7940, 2020.

[46] J. Cao, D. Mao, Q. Cai, H. S. Li, and J. P. du, “A review of object representation based on local features,” *Journal of Zhejiang University SCIENCE C*, vol. 14, no. 7, pp. 495–504, 2013.

[47] Z. Liang, J. Du, and C. Li, “Abstractive social media text summarization using selective reinforced Seq2Seq attention model,” *Neurocomputing*, vol. 410, pp. 432–440, 2020.

[48] F. F. Kou, J. P. Du, C. X. Yang et al., “Hashtag recommendation based on multi-features of microblogs,” *Journal of Computer Science and Technology*, vol. 33, no. 4, pp. 711–726, 2018.