2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS) | 978-1-6654-6667-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/IUCC-CIT-DSCI-SMARTCNS55181.2021.00029

A Reinforcement Learning-based Task Classification Mechanism for Privacy-Enhanced Mobile Crowdsensing Strategy

Mengyao Peng
College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian, China, Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, Fuzhou, Fujian, China,
e-mail: 18875857995@163.com

Hui Lin
College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian, China, Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, Fuzhou, Fujian, China,
e-mail: linhui@fjnu.edu.cn

Xiaoding Wang
College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian, China, Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, Fuzhou, Fujian, China,
e-mail: wangdin1982@fjnu.edu.cn

Abstract—The emergence of the Internet of Things enables efficient connections between things through the Internet, providing a professional platform for information collection, transmission, and sharing. Nowadays, as an important computing model in the Internet of Things, Mobile Crowdsensing(MCS) has received more and more attention. It provides strong technical support for the collection and interaction of information between individuals or devices from different regions. However, while realizing data sharing, it also inevitably brings about the privacy leakage of related data. In order to solve this problem, many privacy protection strategies based on different technologies have been proposed to ensure the privacy of crowdsensing tasks and crowdsensing data. They include the strategies for classifying and grading crowdsensing tasks and workers. In response to this strategy, this paper proposes a algorithm to calculate the number of classifications of the crowdsensing tasks and workers to improve classification efficiency, that is, a reinforcement learning-based task classification mechanism(RTCM). This mechanism uses the Q learning algorithm in reinforcement learning. Through continuous learning iterations, it is possible to select strategies with higher privacy protection degree and task completion quality from different classification strategies. In this way, the system can implement a more efficient privacy protection function according to the optimal strategy. Experiments show that this mechanism can improve the efficiency of crowdsensing tasks and workers classification. And, in different areas, according to different demand standards, the appropriate classification strategy can be quickly selected.

Index Terms—Mobile Crowdsensing, Privacy Protection, Reinforcement Learning, Task Classification

I. Introduction

With the vigorous development of the Internet of Things, mobile crowdsensing technology has also received more and more attention. The typical system architecture of mobile crowdsensing mainly includes three parts: server group, data users(task releasers) and data providers(task receivers). Specifically, there will be sensors in the corresponding equipment of the task receiver in the perception layer for task perception and data perception. The data user is the data collection center in different IoT applications, such as the medical treatment center in the medical IoT, etc. Server group mainly refers to a set of servers with different functions. In this system, on the one hand, after the data users sending crowdsensing tasks to the server, the server will process the relevant task data and release it to the data providers. On the other hand, after selecting a task, the data providers will upload the corresponding crowdsensing data to the server as required. Finally, the server is responsible for verifing the uploaded data before sending it to the data users. Then, the data users can use these data, which are collected from the data provider and verified by the server, to complete the application requirements [1], including patient health data monitoring, road traffic status monitoring, bus arrival status monitoring and so on [2].

However, while mobile crowdsensing technology brings us convenience, there are also some unavoidable problems. Including the leakage of sensitive information of crowdsensing tasks and task receivers in the process of data transmission [3]. What's more, the requirements of task completion degree and the privacy protection degree of the same kind of tasks in different areas may be different. In our previous work [4], a strategy for classifying and grading crowdsensing tasks has been proposed to protect the privacy of tasks. So, how many types of tasks should you specifically divide? Therefore, based on the previous work, this paper proposes an algorithm to calculate the

This work is supported by National Natural Science Foundation of China under Grant No. U1905211 and 61702103, Natural Science Foundation of Fujian Province under Grant No. 2020J01167 and 2020J01169.

978-1-6654-6667-7/21/$31.00 ©2021 IEEE
DOI 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00029

optimal classification method under different needs. And this algorithm was not given in the previous work.

Reinforcement learning, as an important field in machine learning that often used to solve efficiency optimization problems [5] [6], emphasizes how to choose actions that can maximize benefits based on the environment [8]. Reinforcement learning [7] can continuously optimize choices in the corresponding state through the rewards or punishments given by the environment, and finally learn the optimal strategy. Therefore, this paper uses the *Q*- learning algorithm in reinforcement learning to complete the optimal strategy selection mechanism for the classification of crowdsensing tasks [9]. In this way, the degree of protection of sensitive information in tasks is improved according to different actual needs. In this paper, in order to enhance the privacy protection of the mobile crowdsensing strategy, a Reinforcement Learning-based Task Classification Mechanism, named RTCM, is proposed for MCS. The main contributions of this paper are listed as follows:

1) In order to protect the sensitive information in the crowdsensing tasks from being leaked, this paper proposes to classify the tasks and task receivers and to add corresponding levels to different categories.Since only the tasks of the same level can receive the tasks of the corresponding level, the purpose of protecting the privacy of the tasks is achieved.

2) On the basis of the above work, in order to cater to the different needs of different applications for the degree of privacy protection and the quality of task completion, this paper uses the Q learning algorithm in reinforcement learning to train the optimal classification strategy in different situations to improve the efficiency of the system, so as to better realize the privacy protection of the crowdsensing tasks.

3) The experimental results show that the strategy proposed in this paper can well adapt to the privacy needs of different applications, and at the same time, achieve a balance between the degree of privacy protection and the quality of task completion.

The rest of the paper is organized as follows. We summarized some related work in Section II. We introduced the system model of this strategy in Section III. In Section IV, we introduced the strategy RTCM proposed in this paper in detail. In Section V, we analyze the relevant experimental data and results. Finally, it is summarized in Section VI.

II. Reated Work

At present, as there is not much work related to the protection of the privacy of tasks in MCS, we will also collect other privacy protection strategies in the MCS. In our previous work [10], the authors proposed a blockchain-based data aggregation strategy, in which the authors designed a new blockchain header structure and block generation method. And it is used to store the classified crowdsensing tasks, so as to prevent the direct or indirect leakage of task privacy. In [11], the authors are the first to discuss the privacy protection of task locations and propose a codebook-based task allocation mechanism to protect it. In addition, the selected allocation codebook (SAC) method is introduced to solve the problem of high computational resource consumption in the task allocation process and protect the task location privacy to some extent. In [12], the authors regard each trajectory as a vector in the high dimension space and design a trajectory protection algorithm to perturb the true trajectory before submission. They use the differential privacy (DP) as the privacy model so they can estimate the amount of noise given a privacy level. And their mechanism not only guarantees privacy protection, but also preserves trajectories' utility. In [13], the authors construct a differential game model to solve the trade-off problem between the data utility and privacy preserving in mobile crowdsensing system, and solve the feedback Nash equilibrium solutions based on the dynamic programming in the MCS system. In [14], the authors propose a location privacy protection scheme (ELPPS) for a mobile crowd-sensing network in the edge environment, to protect the position correlation weight between sensing users through differential privacy. In [15], the authors carefully design a scalable grouping based privacypreserving participant selection scheme, where participants are grouped into multiple participant groups and then auctions are organized within groups via secure group bidding. By leveraging Lagrange polynomial interpolation to perturb participants' bids within groups, participants' bid privacy is preserved. In [16], the authors propose SFAC, a secure federated learning framework for UAV-assisted MCS. Specifically, by applying local differential privacy, they design a privacy-preserving algorithm to protect UAVs' privacy of updated local models with desirable learning accuracy. In [17], the authors design a location-based symmetric key generator, which enables two parties to self-generate a symmetric key without depending on fully trusted authorities. By utilizing this key generator and Proxy Re-encryption, they propose a privacy preserving protocol to protect location information in task release and task allocation.

III. System Model

The system model of RTCM proposed in this paper is shown in Fig. 1. The system mainly shows the calculation process of mobile crowdsensing technology in different areas and different types of IOT application environments. The figure mainly contains three different IoT applications, including IOMT [18], IIOT [19] and the IOV [20].In scenario 1, that is, the application environment of the medical Internet of Things. The working process of MCS in this scenario is mainly divided into four steps. In the first step, task releasers(including doctors, private health doctors, etc.) use the reinforcement learning algorithm [21] proposed in this paper to select the best classification and

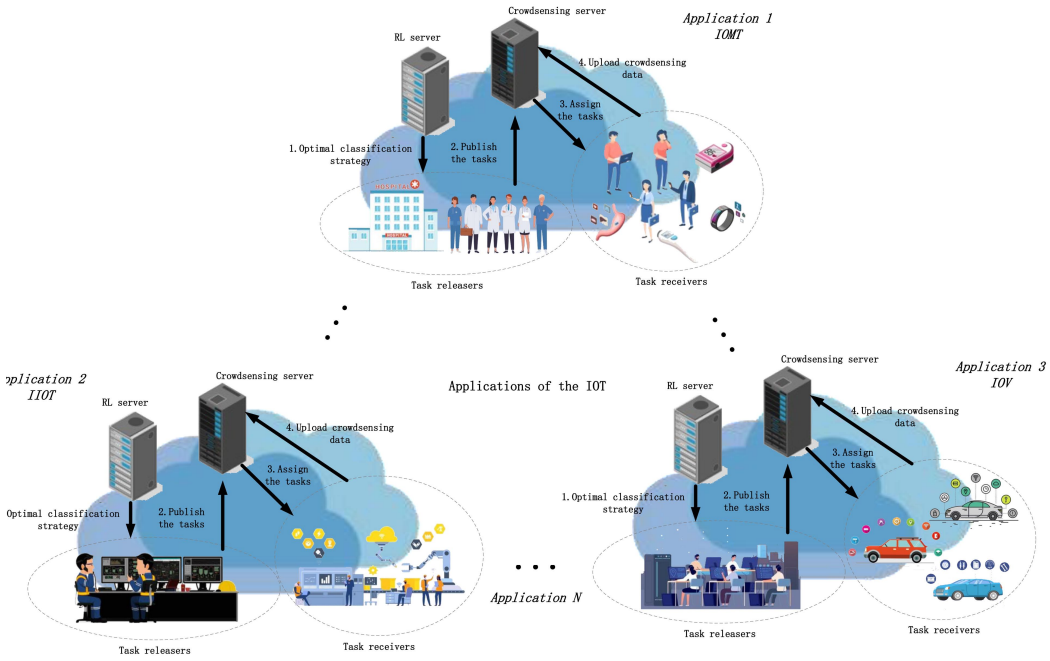


Fig. 1. The system model of RTCM

grading strategy according to the actual needs. In the second step, the task releasers classify the crowdsensing tasks according to the results provided by the algorithm and uploads the tasks to the crowdsensing server. In the third step, after the server processes the relevant data, the tasks will be assigned to task receivers(including patients, health care workers, etc.) for completion. Finally, the task receivers upload the completed crowdsensing data to the crowdsensing server for processing. At this point, the process of MCS in application 1 has been completed. The task completion process of application 2 and application 3 in the system model diagram is basically the same as application 1, except that the task releasers and task receivers in different applications are different. In addition, in the other actual application process, the task releasers and task receivers will continue to change with demand.

In order to realize the privacy protection of sensitive information in the crowdsensing tasks, we propose to classify the tasks and classify the task receivers into the same number of level according to the calculation results of the proposed algorithm. This classification idea has been proposed in previous work [4]. So, in RTCM, we focus on how to choose a better classification strategy. In this paper, the Q-learning algorithm in reinforcement learning is used to achieve more efficient privacy protection functions according to different actual needs [22].

IV. The Implementation Details of the RTCM

Reinforcement learning is an important field in machine learning. It emphasizes how agents take actions based on the environment, so as to maximize the benefits obtained. A common model used in reinforcement learning is the standard Markov Decision Process (MDP) [23]. Therefore, this paper uses the Q-learning algorithm in reinforcement learning to realize the optimal strategy selection of crowdsensing task classification and grading. Q-learning is a value-based algorithm in the reinforcement learning algorithm group. The main idea is to construct a Q-table to store the Q value of each pair of state and action, and then select the action that can obtain the greatest benefit based on the Q value. Next, it mainly introduces the algorithmic decision-making process proposed in this paper.

A. The RTCM-MDP of the Strategy

The standard Markov Decision Process(as shown in Fig. 2) is generally expressed as a five-tuple $< S, A, P, R, \gamma >$. Among them, S represents the states, A represents the actions that can be taken, P represents the state transition function, R represents the reward that can be obtained by taking a certain action in a certain state, and γ represents the discount factor. In short, our goal is to be able to find a strategy that can maximize returns.

Next, we will mainly introduce the five elements of MDP of the RTCM in this paper which is different from

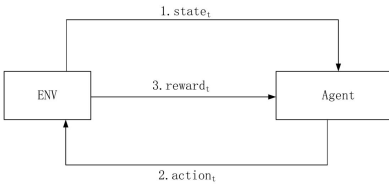


Fig. 2. The standard Markov Decision Process

traditional MDP. State S mainly shows the changes in the status of crowdsensing workers. In this paper, one indicator is mainly used as the basis for status changes; namely, changes in the proportion of malicious users in the crowdsensing workers.

The action set A mainly includes a series of actions that the agent can take in each state S_t. In RTCM, A contains different task classification strategies that can be adopted by the task releasers. However, it is still necessary for the algorithm to learn and iterate the strategies contained in A to obtain the optimal strategy, and then the task releasers divide the tasks according to the results. For example, $A = \{'3', '4', '5'\}$ means that the task releasers can divide the tasks into 3 categories($a_1 =' 3'$), 4 categories($a_2 =' 4'$) or 5 categories($a_3 =' 5'$).

At time t, the system state S_t becomes state S_{t+1} after taking action a_t, so the state S_{t+1} at the next moment is determined by the state S_t and the action a_t at the previous moment, that is, $P(S_{t+1}|S_t, a_t)$.

As the name implies, the reward $R(S_t, a_t, S_{t+1})$ represents the return value that will be generated by taking the action a_t in the state S_t. In this paper, the formula for calculating the return value R can be expressed as:

$$R = i \cdot \prod_{k=1}^{n} (1 - P_k)^{\lambda \cdot \frac{N}{n}} + j \cdot \frac{\sum_{k=1}^{n} (1 - P_k)}{n}. \quad (1)$$

Where i and j respectively represent the proportion of privacy protection degree and task completion degree in actual demand($i + j = 1$). n represents the number of categories of task classification. P_k represents the possibility of malicious actions by users in category k, which is generated by a random function. λ represents the proportion of malicious users, and N is the total number of workers.

In this paper, Q-learning-based task classification strategy, in which the algorithm for calculating the R value is shown in Algorithm 1.

Finally, γ in the five-tuple refers to the discount factor, usually between 0 and 1 ($\gamma \in [0, 1]$), which is used to express the influence of current interests and long-term interests. When γ is closer to 0, it means that the strategy pays more attention to the influence of current benefits. When γ is closer to 1, it means that the strategy pays more attention to the influence of subsequent benefits.

Algorithm 1 R Value Calculation

Input: i, j, N, λ
Output: R
1: Choose a_t from S_t using ε-greedy, seeing Algorithm 3
2: Let $n = a_t$
3: Use random function to calculate P_k
4: Use equation (1) to calculate R
5: return R

Since both the state space and the action space are limited, the crowdsensing task classification strategy problem is a limited Markov Decision Process. After the strategy is transformed into MDP through the above process, the task classification strategy problem can be transformed into optimizing the return value. That is, by looking up the Q-table, we can quickly catch the return that each action can bring in each state.

B. Q-Learning-based Task Classification

The flow of Q-learning algorithm is shown in Fig. 3. First, a Q-table needs to be created and initialized. The rows in the table indicate different states and the columns indicate the actions that can be taken. And at the very beginning, each state-action value in the table is initialized to 0. Then, under the initial state, the agent selects an action from the action space to perform, and then calculates the reward value according to equation (1), which is used to calculate the Q value. The next step is to update the Q value just obtained to the original Q-table, and update the state to the next state. Finally, repeat the previous steps to optimize the Q-table through continuous update iterations.

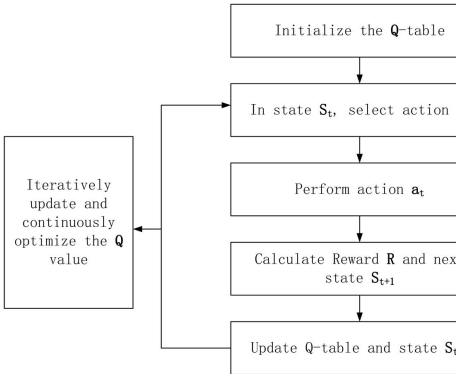


Fig. 3. The Flow of Q-Learning Algorithm

In the update iteration process, we use the time difference method (TD) to achieve the update, and the update formula can be defined as:

$$Q(S_t, a_t) \longleftarrow Q(S_t, a_t) + \alpha[R + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}) - Q(S_t, a_t)]. \quad (2)$$

Where,

$$loss = [R + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1})] - Q(S_t, a_t). \quad (3)$$

In the above equation (2), α indicates the learning rate($\alpha \in [0, 1]$). If α is smaller, it means more previous training results are retained; otherwise, less previous results are retained. γ represents the discount factor, $\gamma = 0$ means that only the current benefit influence is considered, and if $\gamma = 1$, it means that the long-term benefit influence is more considered. R represents the actual return value obtained in the previous state S_t. And $\max_{a_{t+1}} Q(S_{t+1}, a_{t+1})$ represents the estimated value of the maximum benefit in the next state S_{t+1}. $Q(S_t, a_t)$ represents the estimated value of benefit in state S_t. Therefore, through training, the gap between reality and estimation can be continuously narrowed, and the Q value can be continuously optimized.

Then, the algorithm of Q-learning-based crowdsensing task classification strategy is shown in Algorithm 2:

Algorithm 2 Q-learning Algorithm

Input: $S, A, episode, h$
Output: $Q(S, A)$
1: Initialize $Q(S, A)$ arbitrarily
2: for $i = 1$ to $episode$ do
3: Initialize S
4: for $j = 0$ to h do
5: Choose a_t from S_t using policy derived from Q(e.g., ε-greedy), seeing Algorithm 3
6: Take action a_t to get R seeing Algorithm 1
7: Update $Q(S_t, a_t)$ according to equation (2)
8: $S \longleftarrow S_{t+1}$
9: end for
10: end for
11: return $Q(S, A)$

The input of Algorithm 2 is the state space S, the action space A, and $episode$, h parameter values, where $episode$ indicates the number of training rounds and h indicates the number of states in the set S. Then, the output is Q table. In Algorithm 2, step 1 is to initialize $Q(S, A)$, that is, the initial assignment is all 0. Steps 2 to 10 are to train the Q table $episode$ times. Step 3 enter the initial state S, and then, step 4 to 9 are also a loop, but this loop is to train the Q value in each state S_t. Step 5 is to select one of several actions a_t to execute, and the greedy algorithm is used here. As shown in Algorithm 3, in order to prevent local optimization, a certain action a_t is randomly selected in the state S_t with the possibility of ε; and the action a_t with the largest Q value is selected under the possibility of $1 - \varepsilon$. Step 6 and step 7 mean that after selecting the corresponding action a_t, observe the obtained reward value R and the next state value S_{t+1}. Finally, return the result of the $episode$ round iteration.

Algorithm 3 ε-greedy Algorithm

Input: The result of random function
Output: a_t
1: if $rand() < \varepsilon$ then
2: Choose a_t in A at random
3: else
4: $a_t = argmax_a Q(S_t, a_t)$
5: end if
6: return a_t

The above is the process of selecting crowdsensing task classification strategy based on Q-learning proposed in this paper. The return value under different states and different actions can be observed intuitively through the Q table, and the task releasers can choose different strategies for task classification according to actual needs. In this way, the protection of sensitive information in the sensing task is realized indirectly and efficiently.

V. Simulation Results

A. Experiment Setup and Parameter Setting

- Experiment Setup
  The simulation of RTCM is implemented in Python on a computer equipped with Intel Core i7 processor, 16G running memory, CPU frequency 1.50GHz 64-bit win10 system.
- Parameter Setting
  Table I gives the parameters of this simulation.

TABLE I
Experimental parameter setting

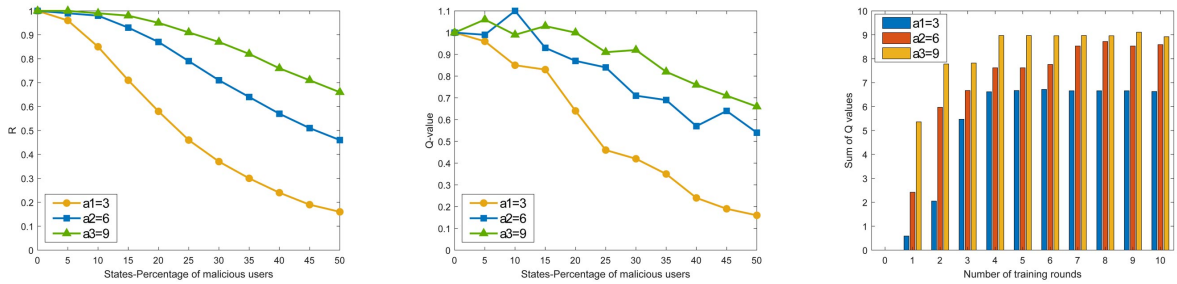
Parameter Name	Value or Range
The number of crowdsensing workers N	1000
Number of states h	11
Set of actions that can be taken a_t	3/6/9
The percentage of the importance of privacy protection i	[0,1]
The percentage of the importance of task completion j	[0,1]
Learning rate α	0.1
Discount factor γ	0.6
ε of ε-greedy algorithm	0.1

B. Experimental Results

The strategy proposed in this article is to adapt to different needs in different applications. We divide the experimental part into three situations. The first case is that when the application pays more attention to the degree of privacy protection of the collected data than the degree of data completion, we set the corresponding parameter values (ie, i=1, j=0) to conduct the experiment. The second case is that when the application pays more attention to the completion of the posted task than the privacy protection of the collected data, we set the corresponding experimental parameters (ie, i=0, j=1) for experimental analysis. The third situation is that when

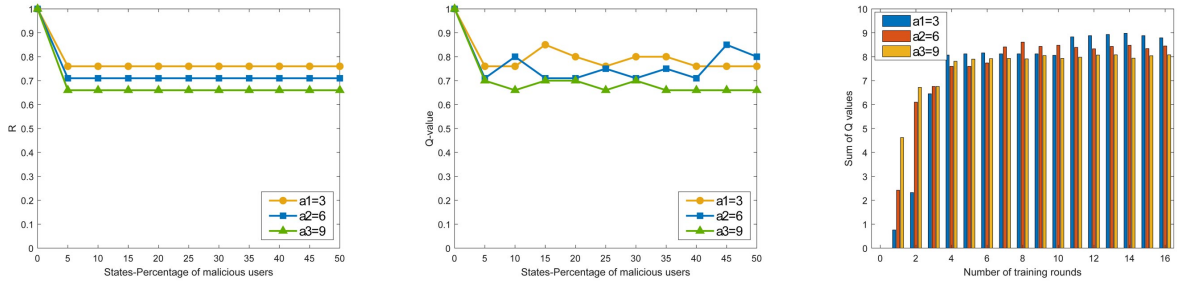
the application has high requirements for the privacy protection of the collected data and the completion of the task, we set the experimental parameters (ie i=0.5, j=0.5) to complete the experiment. Of course, in the actual application process, we can set the parameters differently according to different needs.

- More emphasis on privacy protection($i = 1, j = 0$)
  In some regional applications, task releasers pay more attention to privacy and security issues in tasks than task completion. Therefore, in order to present the experimental results in a clearer way, we set the parameters $i = 1$ and $j = 0$; that is, only the privacy and security issues of the task are considered. The experimental results are shown in the Fig. 4, which mainly contains 3 subgraphs.
  Fig. 4 (a) shows the theoretically obtainable return value R under different conditions. As shown in the figure, each state represents a different proportion of malicious users, and the task releasers want to divide the task into 3, 6 or 9 categories. Moreover, the return value R of any classification strategy will decrease with the continuous increase of malicious users. At the same time, comparing the three classification strategies, the degree of privacy protection when the task is divided into 9 categories is significantly higher than the other two. For example, when $S = 30\%$, $R(a_1) = 0.37$, $R(a_2) = 0.71$, and $R(a_3) = 0.87$. Next, Fig. 4 (b) shows the actual Q value obtained under different conditions, although the curve in the figure is not as smooth as that in Fig. 4 (a) (for example, when $s = 15\%$, $R(a_1) = 0.71$, while $Q(a_1) = 0.83$), but the trend of the curve is still similar; that is, the more malicious users, the smaller the Q value. And in most states, the Q value that divides tasks into 9 categories is significantly higher than other strategies. For example, when $s = 30\%$, $Q(a_1) = 0.42$, $Q(a_2) = 0.71$, and $Q(a_3) = 0.92$. Finally, Fig. 4 (c) is the basis for our final selection strategy. This figure shows the sum of Q values after 10 iterations of the update. The Q value in this figure begins to stabilize after about 6 iterations. From this figure, we can clearly see that the Q value of the strategy that divides the task into 9 categories is significantly higher than the other two strategies. Therefore, when the task publisher pays more attention to the privacy and security of the task, the strategy divided into 9 categories is the optimal choicet. For example, from the 7th to the 9th round, the Q value is stable at 6.66 when $a_1 = 3$, stable at 8.53 when $a_2 = 6$, and stable at 8.97 when $a_3 = 9$.
- More emphasis on task completion($i = 0, j = 1$)
  In some regional applications, task releasers pay more attention to task completion than task privacy and security. Therefore, in order to present the experimental results in a clearer way, we set the parameters $i = 0$ and $j = 1$; that is, only the degree of completion of the task is considered. The experimental results are shown in the Fig. 5, which mainly contains 3 subgraphs.
  Fig. 5 (a) shows the theoretically obtainable return value R under different conditions. As shown in the figure, each state represents a different proportion of malicious users, and the task releasers want to divide the task into 3, 6 or 9 categories. Moreover, the return value R of any classification strategy will remain stable as malicious users continue to increase. At the same time, comparing the three classification strategies, the task completion degree when the tasks are divided into 3 categories is significantly higher than the other two. For example, when $s = 40\%$, $R(a_1) = 0.76$, $R(a_2) = 0.71$, and $R(a_3) = 0.66$. Next, Fig. 5 (b) shows the actual Q value obtained under different conditions, although the curve in this figure is not as smooth as that in Fig. 5 (a) (for example, when $s = 20\%$, $R(a_1) = 0.76$, while $Q(a_1) = 0.80$), but the trend of the curve is still similar; that is, the more malicious users, the Q value fluctuates slightly, but it is basically stable. And in most states, the Q value that divides the task into 3 categories is significantly higher than other strategies. For example, when $s = 15\%$, $Q(a_1) = 0.85$, $Q(a_2) = 0.71$, and $Q(a_3) = 0.70$. Finally, Fig. 5 (c) is the basis for our final selection strategy. This figure shows the sum of Q values after 16 iterations of the update. The sum of Q value in the figure begins to stabilize after about 10 iterations. From the figure, we can clearly see that for the strategies that divide the task into 3 categories, the sum of the Q values is significantly higher than the other two strategies and is stable around 8.90. Therefore, when the task releasers pay more attention to the task completion, the 3-category strategy is the best choice. For example, from the 11th to the 16th round, the sum of the$Q$ value is stable at 8.90 when $a_1 = 3$, stable at 8.40 when $a_2 = 6$, stable at 8.01 when $a_3 = 9$.
- More emphasis on the balance of privacy protection and task completion($i = 0.5, j = 0.5$)
  In some regional applications, the task releasers require both a certain degree of privacy protection and a proper degree of task completion for the crowdsensing tasks. Therefore, in order to present the experimental results in a more intuitive way, we set the parameters $i = 0.5$ and $j = 0.5$; that is, the degree of privacy protection of the crowdsensing tasks is as important as the degree of completion. The experimental results are shown in the Fig. 6, which mainly contains 3 subgraphs.
  Fig. 6 (a) shows the theoretically obtainable return value R under different conditions. As shown in the figure, each state represents a different proportion of malicious users, and it is still assumed that the task releasers want to divide the task into 3, 6 or



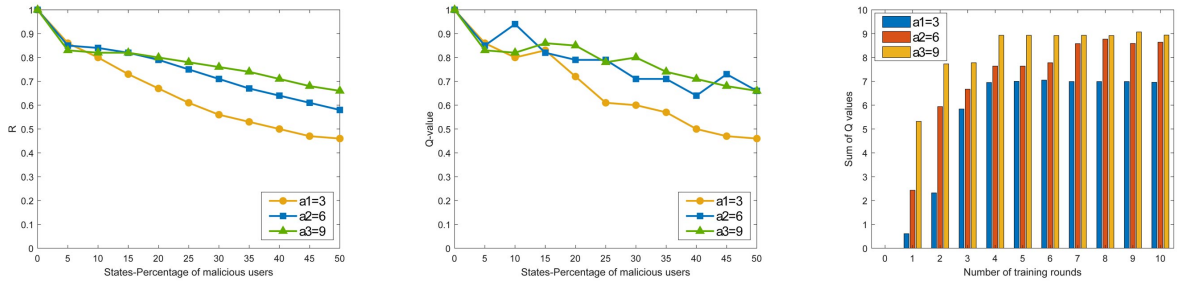
(a) R value (b) Q value (c) The sum of Q value

Fig. 4. Experimental comparison chart of Q table when $i = 1$, $j = 0$



(a) R value (b) Q value (c) The sum of Q value

Fig. 5. Experimental comparison chart of Q table when $i = 0$, $j = 1$



(a) R value (b) Q value (c) The sum of Q value

Fig. 6. Experimental comparison chart of Q table when $i = 0.5$, $j = 0.5$

9 categories. Moreover, the return value R of any classification strategy will decrease with the continuous increase of malicious users. At the same time, comparing the three classification strategies, the task completion degree when the tasks are divided into 3 categories is obviously lower than that of the other two; and when $a_2 = 6$ and $a_3 = 9$, the R values under the two strategies are relatively close. For example, when $s = 20\%$, $R(a_1) = 0.67$, $R(a_2) = 0.79$, and $R(a_3) = 0.80$. Next, Fig. 6 (b) shows the actual Q value obtained under different conditions, which is the same as the above, although the curve in the figure is not as smooth as that in the Fig. 6 (a) (for example, when $s = 20\%$, $R(a_1) = 0.67$, and $Q(a_1) = 0.72$), but the trend of the curve is still similar, especially when $a_2 = 6$ and $a_3 = 9$, the Q value of the two strategies is closer to the R value. For example, when $s = 25\%$, $Q(a_2) = 0.79$ and $Q(a_3) = 0.78$. And in most states, the Q value that divides the task into three categories is significantly lower than the other two strategies. For example, when $s = 35\%$, $Q(a_1) = 0.57$, $Q(a_2) = 0.71$, and $Q(a_3) = 0.74$. Finally, Fig. 6 (c) is the basis for our final selection strategy. This figure shows the sum of Q values after 10 iterations of the update in this case. The sum of the Q value in the figure began to stabilize after about 6 iterations. From

the figure, we can clearly see that in the strategies that divide tasks into 6 categories and 9 categories, the sum of Q values is significantly higher than the other strategy. Therefore, when task releasers pay more attention to the balance between task privacy protection and completion, the 9-category strategy is the relatively optimal choice, but according to the experimental results, the 9-category and 6-category strategies are not too much difference. For example, from the 11th to the 16th round, the sum of the Q value is stable at 6.66 when $a_1 = 3$, stable at 8.60 when $a_2 = 6$, and stable at 8.90 when $a_3 = 9$.

In summary, the task releasers can reasonably set the values of i and j $(i+j=1)$ according to different requirements in different scenarios. Therefore, a more appropriate classification and grading strategy for crowdsensing tasks can be selected to indirectly and efficiently improve the privacy protection ability of sensitive information in the tasks in the MCS technology.

VI. Conclusion

In order to more efficiently improve the ability of mobile crowdsensing technology to protect sensitive information in the crowdsensing tasks in different application scenarios, this paper proposes a task classification mechanism (RTCM) based on reinforcement learning. Specifically, the task releasers can use the algorithm in advance to select the optimal classification strategy in different situations according to different requirements for privacy protection and task completion. Furthermore, we can clearly see from the experimental results that the algorithm can effectively calculate task classification strategies that meet actual needs based on different regions. So as to achieve the purpose of improving the privacy protection function of the tasks in the mobile crowdsensing technology.

References

[1] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, and P. Bouvry, "A Survey on Mobile Crowdsensing Systems: Challenges, Solutions and Opportunities," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2419-2465, 2019.

[2] L. Zhao, X. Li, B. Gu et al., "Vehicular Communications: Standardization and Open Issues," IEEE Communications Standards Magazine, 2018, doi: 10.1109/MCOMSTD.2018.1800027.

[3] T. Li, T. Jung, Z. Qiu, H. Li, L. Cao and Y. Wang, "Scalable Privacy-Preserving Participant Selection for Mobile Crowdsensing Systems: Participant Grouping and Secure Group Bidding," IEEE Transactions on Network Science and Engineering, vol. 7, no. 2, pp. 855-868, 2020.

[4] H. Lin, S. Garg, J. Hu, X. Wang, M. J. Piran and M. S. Hossain, "Privacy-enhanced Data Fusion for COVID-19 Applications in Intelligent Internet of Medical Things," IEEE Internet of Things Journal, 2020, doi: 10.1109/JIOT.2020.3033129.

[5] J. Mills, J. Hu, G. Min, "Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 5986 - 5994, 2020.

[6] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, N. Georgalas, "Computation Offloading in Multi-Access Edge Computing Using a Deep Sequential Model Based on Reinforcement Learning," IEEE Communications Magazine, vol. 57, no. 5, pp. 64-69, 2019.

[7] J. Wang, J. Hu, G. Min, A. Zomaya, N. Georgalas, "Fast Adaptive Task Offloading in Edge Computing Based on Meta Reinforcement Learning," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 1, pp. 242-253, 2021.

[8] K. Mason, S. Grijalva, "A Review of Reinforcement Learning for Autonomous Building Energy Management," Computers & Electrical Engineering, vol. 78, pp. 300-312, 2019.

[9] B. Luo, Y. Yang and D. Liu, "Adaptive Q -Learning for Data-Based Optimal Output Regulation With Experience Replay," IEEE Transactions on Cybernetics, vol. 48, no. 12, pp. 3337-3348, 2018.

[10] 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 Transactions on Industrial Informatics, vol. 17, no. 10, pp. 7204-7212, 2021.

[11] X. Dong, W. Zhang, Y. Zhang, Z. You, S. Gao, Y. Shen, C. Wang, "Optimizing Task Location Privacy in Mobile Crowdsensing Systems," IEEE Transactions on Industrial Informatics, 2021, doi: 10.1109/TII.2021.3109437.

[12] H. Huang, X. Niu, C. Chen and C. Hu, "A Differential Private Mechanism to Protect Trajectory Privacy in Mobile Crowd-Sensing," 2019 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6, 2019, doi: 10.1109/WCNC.2019.8885628.

[13] H. Gao, H. Xu, L. Zhang and X. Zhou, "A Differential Game Model for Data Utility and Privacy-Preserving in Mobile Crowdsensing," IEEE Access, vol. 7, pp. 128526-128533, 2019.

[14] M. Li, Y. Li and L. Fang, "ELPPS: An Enhanced Location Privacy Preserving Scheme in Mobile Crowd-Sensing Network Based on Edge Computing," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 475-482, 2020, doi: 10.1109/TrustCom50675.2020.00071.

[15] T. Li, T. Jung, Z. Qiu, H. Li, L. Cao and Y. Wang, "Scalable Privacy-Preserving Participant Selection for Mobile Crowdsensing Systems: Participant Grouping and Secure Group Bidding," IEEE Transactions on Network Science and Engineering, vol. 7, no. 2, pp. 855-868, 2020.

[16] Y. Wang, Z. Su, N. Zhang and A. Benslimane, "Learning in the Air: Secure Federated Learning for UAV-Assisted Crowdsensing," IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 1055-1069, 2021.

[17] Y. Jiang, K. Zhang, Y. Qian and L. Zhou, "P2AE: Preserving Privacy, Accuracy, and Efficiency in Location-dependent Mobile Crowdsensing," IEEE Transactions on Mobile Computing, 2021, doi: 10.1109/TMC.2021.3112394.

[18] S. Vishnu, S. R. J. Ramson and R. Jegan, "Internet of Medical Things (IoMT) - An overview," 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), pp. 101-104, 2020, doi: 10.1109/ICDCS48716.2020.243558.

[19] A. C. Panchal, V. M. Khadse and P. N. Mahalle, "Security Issues in IIoT: A Comprehensive Survey of Attacks on IIoT and Its Countermeasures," 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN), pp. 124-130, 2018, doi: 10.1109/GCWCN.2018.8668630.

[20] L. Zhao, W. Zhao, A. Hawbani, A. Al-Dubai, G. Min, A. Y. Zomaya, C. Gong, "Novel Online Sequential Learning-based Adaptive Routing for Edge Software-Defined Vehicular Networks," IEEE Transactions on Wireless Communications, 2020, doi:10.1109/TWC.2020.3046275.

[21] J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, N. Georgalas,"Computation Offloading in Multi-Access Edge Computing Using a Deep Sequential Model Based on Reinforcement Learning," IEEE Communications Magazine, vol. 57, no. 5, pp. 64-69, 2019.

[22] Z. Wang, Y. Liu, Z. Ma, X. Liu and J. Ma, "LiPSG: Lightweight Privacy-Preserving Q-Learning-Based Energy Management for the IoT-Enabled Smart Grid," IEEE Internet of Things Journal, vol. 7, no. 5, pp. 3935-3947, 2020.

[23] S. Muralidharan, A. Roy and N. Saxena, "MDP-Based Model for Interest Scheduling in IoT-NDN Environment," IEEE Communications Letters, vol. 22, no. 2, pp. 232-235, 2018.