Secure Mobile Crowdsensing Game

Liang Xiao*, Jinliang Liu*, Qiangda Li*, H. Vincent Poor†

* Dept. of Communication Engineering, Xiamen Univ., 361005 China. Email: lxiao@xmu.edu.cn

† Dept. of EE, Princeton University, Princeton, NJ. Email: poor@princeton.edu

***Abstract*—By recruiting sensor-equipped smartphone users to report sensing data, mobile crowdsensing (MCS) provides location-based services such as environmental monitoring. However, due to the distributed and potentially selfish nature of smartphone users, mobile crowdsensing applications are vulnerable to faked sensing attacks by users who bid a low price in an MCS auction and provide faked sensing reports to save sensing costs and avoid privacy leakage. In this paper, the interactions among an MCS server and smartphone users are formulated as a mobile crowdsensing game, in which each smartphone user chooses its sensing strategy such as its sensing time and energy to maximize its expected utility while the MCS server classifies the received sensing reports and determines the payment strategy accordingly to stimulate users to provide accurate sensing reports. Nash equilibrium (NE) of a static MCS game is evaluated and a closed-form expression for the NE in a special case is presented. Moreover, a dynamic mobile crowdsensing game is investigated, in which the sensing parameters of a smartphone are unknown by the server and the other users. A Q-learning discriminated pricing strategy is developed for the server to determine the payment to each user. Simulation results show that the proposed pricing mechanism stimulates users to provide high-quality sensing services and suppress faked sensing attacks.**

I. Introduction

With the development of smartphones equipped with sensors including accelerometers, magnetometers and global positioning systems, mobile crowdsensing (MCS) provides location-based services such as environmental monitoring (e.g., weather and noise mapping), network monitoring (e.g., cellular and WiFi coverage mapping), and traffic monitoring [1]–[4]. In an MCS system that offers crowdsensing applications, a platform or server recruits smartphone users at locations of interest to report sensing data.

Due to the potentially selfish and distributed nature of smartphone users that aim to maximize their individual utilities, mobile crowdsensing applications have to ensure that users are stimulated to contribute accurate sensing reports and suppress faked sensing attacks by selfish users. Otherwise, if knowing that cheating incurs no punishment, a user has incentive to save its sensing efforts and avoid the possible leak of privacy by sending a faked or a low-quality sensing report. In addition, both under-sensing and over-sensing have to be avoided and the minimum payment is desirable for an MCS server to establish sensing applications in a given coverage area.

The work of L. Xiao was supported in part by National Science Foundation of China under Grants 61271242 and 61001072, and the work of H. V. Poor was supported in part by the U. S. National Science Foundation under Grant CMMI-1435778.

Game theory has shown its strength as a means to study the interactions among smartphones and MCS servers. Auctions, pricing and reputations can motivate user collaborations [5]–[9]. For instance, an auction strategy was proposed to motivate user contributions to crowdsensing applications, in which the server pays the user with the lowest sensing price [5]. However, the utility that a server receives from a sensing report depends on the smartphone's location, sensing time, sensors' accuracies and transmission power levels. More specifically, a high-quality sensing report deserves a higher payment to stimulate efficient user cooperation in mobile crowdsensing. Consequently, a secure and efficient MCS server evaluates the received sensing reports and pays the users according to their sensing accuracies.

In this work, the interactions among selfish smartphone users and a server in a location sensitive MCS application are formulated as a secure mobile crowdsensing game. In this game, smartphone users determine their sensing strategies to maximize their individual utilities, while the server evaluates the sensing accuracy of each report and pays users accordingly. Faked sensing attackers are punished with zero payment to suppress their attack incentives. The Nash equilibrium (NE) of a static MCS game is evaluated. A smartphone user determines its participation according to its sensing costs, reward records of the server, and the risk that its sensing report is falsely classified by the server.

In addition, the case in which there are repeated MCS interactions is formulated as a dynamic MCS game. A higher price stimulates more smartphone users to contribute mobile crowdsensing applications, while it decreases the utility of the server due to a high payment and sometimes leads to network congestion in the area; and vice verse. As smartphone users have no knowledge of the system parameters such as their sensing and transmission costs, they learn their optimal pricing strategies via trials. As a primitive form of model-free reinforcement learning as well as an asynchronous dynamic programming method, Q-learning can converge to the optimal action-values with probability one if all actions are repeatedly sampled in all states [10]. Therefore, a Q-learning strategy is designed for an MCS server to design its pricing strategy based on the observations of the sensing results of various pricing strategies. The proposed learning strategy can motivate users to

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

send higher-quality sensing reports and significantly decrease the number of faked sensing reports.

The remainder of the paper is organized as follows. We review related work in Section II and present the system model in Section III. We describe a static mobile crowdsensing game and discuss the NE of the game in Section IV. We propose a Q-learning pricing strategy for MCS servers in a repeated MCS game in Section V. We provide simulation results in Section VI and conclude in Section VII.

II. Related Work

Auctioning has been used to stimulate smartphone users to participate in MCS applications. For example, an auction strategy was proposed for a user-centric model to motivate smartphone users to participate in mobile sensing in [5], where a Stackelberg equilibrium of a crowd sensing game was provided. A location-aware auction strategy has been proposed to motivate users to submit bids that reflect the actual costs of performing sensing tasks and stimulate smartphone users to join mobile crowdsourcing applications in [6]. However, it is challenging for a user to calculate the exact costs in the game.

Reputation systems provide another mechanism to motivate user cooperation in crowd sensing. An anonymous reputation and trust strategy was proposed to stimulate mobile users to undertake mobile sensing tasks in [7]. The participation levels of users were formulated as a Bayesian game in [8]. A social norm system was designed to integrate payment and reputation mechanisms for crowdsoucing websites in [9]. In addition, a sensing task allocation strategy was proposed in [11] to achieve user fairness and energy efficiency among smartphones for participatory sensing systems.

III. System model

In a mobile crowdsensing system, an MCS server located in a cloud recruits M smartphone users located in an area of interest to gather sensing data. As shown in Fig. 1, the MCS server first broadcasts a price range of the sensing duty to motivate users to participate in the sensing service. Upon receiving the MCS recruiting information, each smartphone user decides on its sensing strategy, e.g., whether to undertake the MCS task and to expend resources on the sensing effort such as the sensing time and power. Each MCS report is evaluated by the server and the corresponding smartphone user is paid accordingly. This process repeats if the MCS application is performed over time.

The MCS server is assumed to classify all the sensing reports into L types, where Class 0 represents fake data, Class L corresponds to a report of the highest quality, and the class number indicates the sensing accuracy. The type of an MCS report, denoted as x with $x \in A$, represents the sensing effort of the corresponding smartphone, where $A = \{-1, 0, \cdots, L\}$ is the action set. A smartphone spends the largest sensing effort such as sensing time and energy to gather the sensing data if its sensing report has $x = L$; it takes less sensing efforts if $1 \leq x < L$, and sends a fake report to save the sensing cost

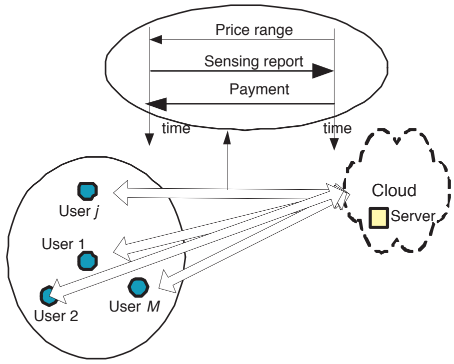


Fig. 1. System model of a mobile crowdsensing game.

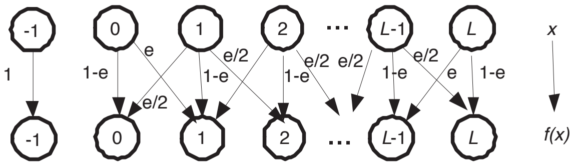


Fig. 2. Sensing report identification model, in which the probability of an MCS report of Class x with $-1 \leq x \leq L$ to be classified by the server as Class $\hat{x} = f(x)$ is given by (1).

if $x = 0$; finally if $x = -1$ it does not participate in the MCS application.

The server pays each user based on the report classification results denoted as $\hat{x} = f(x)$, where $f(\cdot)$ is the classification function. The report classification error rate denoted as e is defined as the probability that a report is classified by the MCS server as another type, i.e., $e = Pr(\hat{x} \neq x)$. As a concrete example, we consider a classification model as shown in Fig. 2, in which the server knows whether a user sends an MCS report; a fake report is sometimes regarded as a good report while a report of highest quality is sometimes viewed as a Type $L-1$ report. The classification result in the server follows the following probability mass function:

$$Pr(\hat{x}|x) = \begin{cases} 1-e, & \hat{x} = x \geq 0 \\ 1, & (x, \hat{x}) = (-1, -1) \\ e, & (x, \hat{x}) = (0, 1), \text{ or } (L, L-1) \\ 0.5e, & \hat{x} = x \pm 1, 0 < x < L \\ 0, & \text{o.w.} \end{cases} \quad (1)$$

IV. Static Mobile Crowdsensing Game

A mobile crowdsensing process as illustrated in Fig. 1 is formulated as a game consisting of $M+1$ players, i.e., a server and M smartphone users. The M users are assumed to be rational and selfish and choose their own sensing strategies from the action set A. The action of smartphone user j is denoted as $x_j \in A$ indicating its sensing effort. For simplicity, the M users' actions are denoted as $\mathbf{x} = [x_j]_{1 \leq j \leq M}$ and we define $\mathbf{x}_{-j} = [x_i]_{1 \leq i \neq j \leq M}$.

The server takes an action denoted as $\mathbf{y} = [y_i]_{-1\leq i\leq L}$, in which $y_i \in \mathbf{P}$ is the payment to a user whose report is classified as Type i, and $\mathbf{P}$ is the discrete feasible payment set. A user who is found to send zero or a faked report receives zero payment with $y_{-1} = y_0 = 0$. Otherwise, user j is paid by the server according to its report classification result $\hat{x}_j$ and the payment is $\mathbf{y}(\hat{x}_j) = \mathbf{y}(f(x_j))$.

In the MCS game, let $C(i, j)$ be the nonnegative cost of action i to user j, and $G(i)$ denote the benefit that the server obtains from an MCS report of Class i. The server is assumed to be able to extract more useful information from a more accurate report, i.e., $G(0) < 0 < G(i) < G(j)$ for $0 < i < j$. In addition, users that do not respond to the recruiting effort do not provide any utility, and thus $G(-1) = C(-1, j) = 0$, $\forall j$.

The immediate utilities of user j choosing action x and the server taking action $\mathbf{y}$ are denoted as u_j and u_s, respectively. By definition, we have

$$u_j(x, \mathbf{y}) = \mathbf{y}(f(x)) - C(x, j), \quad 1 \leq j \leq M \qquad (2)$$
$$u_s(x, \mathbf{y}) = G(x) - \mathbf{y}(f(x)). \qquad (3)$$

In the presence of the evaluation error in terms of the MCS reports, let U_j (or U_s) denote the expected utility of user j (or the server) over all the report evaluation results of the server, which are given by

$$U_j(x, \mathbf{y}) = \sum_{l=-1}^{L} \mathbf{y}(l)Pr(l|x) - C(x, j), \quad 1 \leq j \leq M \qquad (4)$$

$$U_s(x, \mathbf{y}) = G(x) - \sum_{l=-1}^{L} \mathbf{y}(l)Pr(l|x), \qquad (5)$$

where the report identification error rate $Pr(l|x)$ is given in (1). By (5), the average total utility of the server taking action $\mathbf{y}$ from M users taking action $\mathbf{x}$, again denoted by U_s, is given by

$$\begin{aligned} U_s(\mathbf{x}, \mathbf{y}) &= \sum_{i=1}^{M} U_s(x_i, \mathbf{y}) \\ &= \sum_{i=1}^{M} \left(G(x_i) - \sum_{l=-1}^{L} \mathbf{y}(l)Pr(l|x_i) \right). \end{aligned} \qquad (6)$$

In summary, a one-shot mobile crowdsensing game is given by $\mathbb{G} = \langle \{s, 1, 2 \cdots, M\}, \{\mathbf{x}, \mathbf{y}\}, \{U_s, U_{1\leq j\leq M}\} \rangle$, in which M smartphone users and the server s take pure strategies over the action sets $\mathbf{x} \in A^M$ and $\mathbf{y} \in \mathbf{P}^{L+1}$ to maximize their individual utilities, U_s and U_j, respectively, with $1 \leq j \leq M$. For ease of reference, our commonly used notation is summarized in Table I.

A. Nash Equilibrium of a Static MCS Game

A Nash equilibrium (NE) strategy of the static mobile crowdsensing game $\mathbb{G}$ denoted as ($\mathbf{x}^* = [x_j^*], \mathbf{y}^*$) is the best response of the game. If all the system parameters such as the sensing costs and gains are known by all the users and the server, the NE strategies can be written as

M	Number of users in the area
L	Highest level of classification types of reports
x_j	Action of user j
$A = \{-1, 0, \cdots, L\}$	Action set
$\mathbf{y}$	Pricing strategy of the server
$C(x, j)$	Cost of action x to user j
$G(x)$	Gain of action x to the server
u_j (or u_s)	Utility of user j (or the server)
N_i ($\hat{N}_i$)	Number of reports of (detected as) type i
U_j	Expected utility of user j
α	Learning rate
$\mathbf{s}$	System state
δ	Discount factor in learning
$V(\mathbf{s})$	Value of state $\mathbf{s}$
$Q(\mathbf{S}, b)$	Q function
e	Classification error of server

TABLE I

SUMMARY OF SYMBOLS AND NOTATION.

$$\begin{aligned} x_j^* &= \arg\max_{x\in A} U_j(x, \mathbf{y}^*) \\ &= \arg\max_{x\in A} \sum_{l=-1}^{L} \mathbf{y}(l)Pr(l|x) - C(x, j), \quad 1 \leq j \leq M \end{aligned} \qquad (7)$$

$$\begin{aligned} \mathbf{y}^* &= \arg\max_{\mathbf{y}} U_s(\mathbf{x}^*, \mathbf{y}) \\ &= \arg\max_{\mathbf{y}\geqslant 0} \sum_{i=1}^{M} \left(G(x_i^*) - \sum_{l=-1}^{L} \mathbf{y}(l)Pr(l|x_i^*) \right). \end{aligned} \qquad (8)$$

As a concrete example, we evaluate a special case with $L = 1$ and $M = 1$, in which only one smartphone user exists in the area, which sends a good sensing report with $x = 1$, a faked report with $x = 0$ and no response with $x = -1$.

Theorem IV.1. *The static MCS game $\mathbb{G}$ with $L = 1$ and $M = 1$ has a unique NE that is given by*

$$(x^*, y^*) = \left(1, \max\left\{ \frac{C(1) - C(0)}{1 - 2e}, \frac{C(1)}{1 - e} \right\} \right), \qquad (9)$$

if the gain of an accurate sensing report

$$G(1) > \max\left\{ C(1), \frac{1 - e}{1 - 2e}(C(1) - C(0)) \right\}. \qquad (10)$$

Proof: As $L = 1$ and $M = 1$, the server classification model in (1) can be simplified as

$$Pr(\hat{x}|x) = \begin{cases} 1 - e, & (x, \hat{x}) = (0, 0), \text{ or } (1, 1) \\ e, & (x, \hat{x}) = (0, 1), \text{ or } (1, 0) \\ 1, & (x, \hat{x}) = (-1, -1) \\ 0, & \text{o.w.} \end{cases}. \qquad (11)$$

The expected utilities of the user and the server become

$$U_1(x, y) = \sum_{l=0}^{1} y(l) Pr(l|x) - C(x), \tag{12}$$

$$U_s(x, y) = G(x) - \sum_{l=0}^{1} y(l) Pr(l|x). \tag{13}$$

According to (7), (8), and (11)-(13), $U_1(1, y^*) > 0$ yields $y^* > \frac{C(1)}{1-e}$, and $U_1(1, y^*) > U_1(0, y^*)$ yields $y^* > \frac{C(1)-C(0)}{1-2e}$, respectively. On the other hand, $U_s(1, y) > 0$ yields $y < \frac{G(1)}{1-e}$. Moreover, $U_s(1, y^*) \geq U_s(1, y)$ indicates that y^* is the smallest value to ensure a valid report with $x^* = 1$. Therefore, we have (9) after simplification. ■

V. Learning in A Dynamic MCS Game

Interactions among a server and smartphone users repeat for a given MCS application over time, yielding a dynamic MCS game. In practice, the sensing parameters of a smartphone such as the actual sensing costs $C(\cdot, \cdot)$ are usually unknown by the server and the other smartphone users. Therefore, reinforcement learning methods such as Q-learning can be applied by the MCS server to determine its payment strategy in a dynamic MCS game without a known sensing model.

In the learning-based dynamic MCS game, the server decides on the payment strategy according to the utility history for the previous trials: a high price for a sensing report decreases the server's immediate utility while stimulating more smartphone users to apply more sensing effort in the future. In addition, a very high report payment decreases the server's long-term utility by encouraging over-sensing in the area of interest.

The server applies a Q-learning process to obtain an optimal pricing policy, in which the system state observed by the server at time k, denoted by $\mathbf{s}^{(k)} \in \mathbf{S}$, indicates the quality of the received sensing reports. The learning of the pricing strategy is formulated as a Markov decision process with $|\mathbf{S}|$ states. The system state $\mathbf{s}^{(k)} = [\hat{N}_i]_{0 \leq i \leq L}$ incorporates the number of reports of each type, where N_i and $\hat{N}_i$ are the actual and estimated number of reports of class i. The parameter N_i can be estimated during the information extraction process of the received sensing reports. With $I(\cdot)$ denoting an indicator function, we have

$$N_i = \sum_{j=1}^{M} I(x_j = i), \tag{14}$$

$$\hat{N}_i = \sum_{j=1}^{M} I(\hat{x}_j = i), \quad 0 \leq i \leq L. \tag{15}$$

We assume that the smartphone users apply an ϵ'-greedy policy to choose their sensing strategies, in which the action with the highest expected reward is chosen most of the time while an action is selected at random with a small probability ϵ'. In particular, each user uses a mixed strategy: the best action corresponding to the maximum utility is chosen with probability $1 - \epsilon'$ and the other strategies are taken with equal probabilities. Thus the action of user j is given by

$$Pr(x_j = x) = \begin{cases} 1 - \epsilon', & x = \arg\max_{x \in A} U_j(x, \mathbf{y}) \\ \frac{\epsilon'}{L+1}, & \text{o.w.} \end{cases}. \tag{16}$$

According to Eq. (6), as $u_s(-1, y) = 0$, the estimated utility of the server in a time slot is given by

$$u_s(\mathbf{s}, \mathbf{y}) = \sum_{i=0}^{L} N_i G(i) - \hat{N}_i \mathbf{y}(i). \tag{17}$$

Let $Q(\mathbf{s}, \mathbf{y})$ denote the Q function of state $\mathbf{s}$ and pricing strategy $\mathbf{y}$. The server updates its Q function according to the following method:

$$Q\left(\mathbf{s}^{(k)}, \mathbf{y}^{(k)}\right) = (1 - \alpha) Q\left(\mathbf{s}^{(k)}, \mathbf{y}^{(k)}\right) + \alpha\left(u_s\left(\mathbf{s}^{(k)}, \mathbf{y}^{(k)}\right) + \delta V\left(\mathbf{s}^{(k+1)}\right)\right), \tag{18}$$

$$V\left(\mathbf{s}^{(k)}\right) = \max_{\mathbf{y}} Q\left(\mathbf{s}^{(k)}, \mathbf{y}\right), \tag{19}$$

where $\alpha \in (0, 1]$ is the learning rate, $\delta \in [0, 1]$ is the discount factor indicating the myopic nature of the server, $V(\cdot)$ is the highest value of the state, and u_s is given by Eq. (17).

The server is assumed to apply the ϵ-greedy policy to determine its pricing strategy $\mathbf{y}$. Thus the action of user j at time k, denoted by $\mathbf{y}^{(k)}$, is given by

$$Pr\left(\mathbf{y}^{(k)} = \mathbf{p}\right) = \begin{cases} 1 - \epsilon, & \mathbf{p} = \arg\max_{\mathbf{y} \in \mathbf{P}} Q\left(\mathbf{s}^{(k+1)}, \mathbf{y}\right) \\ \frac{\epsilon}{|\mathbf{P}|}, & \text{o.w.} \end{cases}. \tag{20}$$

The learning process described above is summarized in Algorithm 1.

Algorithm 1. Q-learning pricing strategy in a dynamic MCS game.

```
Set δ = 0.8, α = 0.1.
Initialize Q(s, y) = 0, V(s) = 0, ∀s, y.
Repeat (for each episode)
  Observe the initial system states s^(1);
  For k = 1, 2, 3, ...
    Select and perform an action y^(k) ∈ P via (20);
    Observe the subsequent state s^(k+1) and immediate payoff u_s;
    Update Q(s^(k), y^(k)) via (18);
    Update V(s^(k)) via (19);
  End for
End for
```

VI. Simulation Results

In the simulation, we evaluate the performance of the MCS game by assuming $L = 1$, $M = 1$, $\mathbf{C} = \{0, 0.01, 1\}$ and $\mathbf{G} = \{0, -0.01, 8\}$, if not specified otherwise. First, the utilities of players at the NE in the static MCS game are shown in Fig. 3, indicating that the utility of the smartphone user increases with the classification error rate while the utility of the server decreases with it. In addition, the utility of the server significantly increases with the gain from a good sensing report, $G(1)$, which does not impact the performance of the user because the user participates in the sensing application

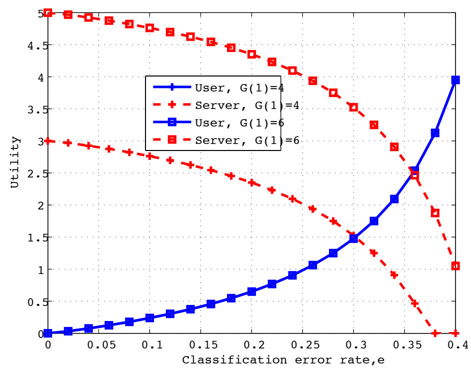


Fig. 3. The players' utilities in the NE of a static MCS game versus the report classification error rate of server.

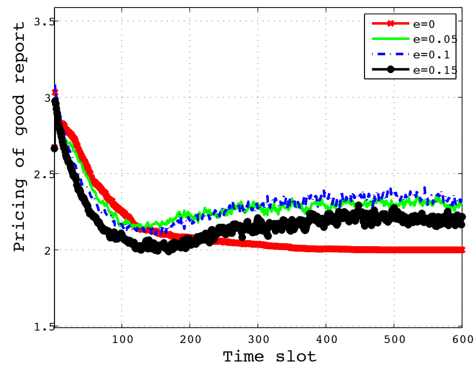
only if the payment is greater than its individual average sensing cost.

Figure 4 presents the performance of the Q-learning pricing strategy in the dynamic MCS game in which the system parameters such as **C** and **G** are unknown to the players. As shown in Fig. 4 (a), the price of a good sensing report offered by the server quickly converges to a value that increases with the report classification error rate, e. Moreover, it is indicated in Fig. 4 (b) that a server quickly learns a pricing strategy to achieve a high utility and in Fig. 4 (c) that it suppresses the population with faked reports. Finally, Fig. 4 (d) shows that a smartphone user is motivated to provide accurate sensing reports, e.g., the probability that a user will send a good sensing report is greater than 97% after 300 time slots for $e = 0$.

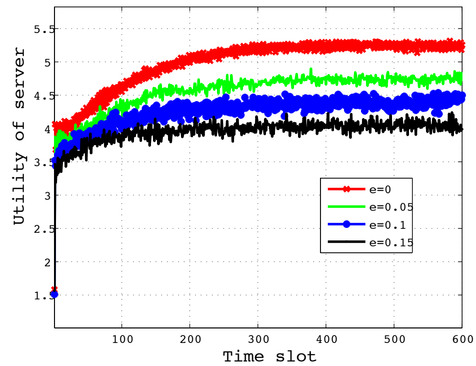
VII. Conclusion

We have formulated the interactions among an MCS server and smartphone users for mobile crowdsensing applications as a mobile crowdsensing game, in which the server pays users according to the qualities of their sensing reports in order to minimize faked sensing attacks. The NE of a static MCS game for a special case has been provided. In addition, a dynamic MCS game was investigated in which the sensing parameter of a user is unknown by the other users as well as the server. A Q-learning pricing strategy has been proposed for a server in this game to perform secure mobile crowdsensing. Simulation results for a special case have preliminarily shown the efficacy of the proposed pricing strategy.

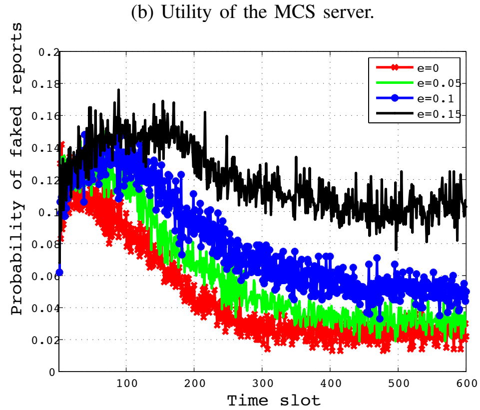
In our future work, we plan to carry out more extensive theoretical analysis and simulations under various sensing scenarios to evaluate the performance of the mobile crowdsensing game. Another interesting topic for further study is the use of multiagent reinforcement learning techniques to design secure mobile crowdsensing strategies.



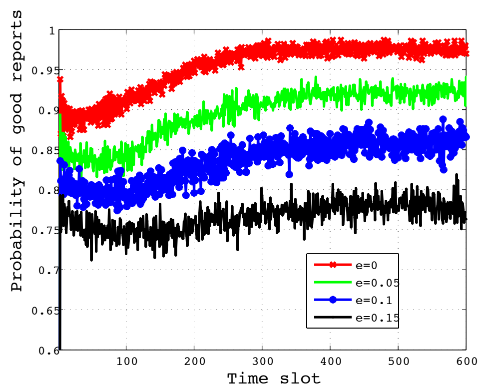
(a) Payment for high quality MCS reports.



(b) Utility of the MCS server.



(c) Probability of faked reports.



(d) Probability of accurate reports.

Fig. 4. Performance of the pricing strategy of a server with Q-learning in a dynamic MCS system with $\mathbf{C} = \{0, 0.01, 1\}$ and $\mathbf{G} = \{0, -0.01, 8\}$.

REFERENCES

[1] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, 2011.

[2] A. C. Weaver, J. P. Boyle, and L. I. Besaleva, "Applications and trust issues when crowdsourcing a crisis," in *Proc. IEEE Int'l Conf. Computer Commun. and Networks (ICCCN)*, 2012, pp. 1–5.

[3] X. Hu, X. Li, E. C. H. Ngai, V. C. M. Leung, and P. Kruchten, "Multidimensional context-aware social network architecture for mobile crowdsensing," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 78–87, 2014.

[4] W. Guo and S. Wang, "Mobile crowd-sensing wireless activity with measured interference power," *IEEE Wirel. Commun. Lett.*, vol. 2, no. 5, pp. 539–542, 2013.

[5] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. ACM Int'l Conf. Mobile Computing and Networking (MobiCom)*, 2012, pp. 173–183.

[6] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proc. IEEE Int'l Conf. Computer Commun. (INFOCOM)*, 2014, pp. 1231–1239.

[7] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Artsense: Anonymous reputation and trust in participatory sensing," in *Proc. IEEE Int'l Conf. Computer Commun.(INFOCOM)*, 2013, pp. 2517–2525.

[8] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *Proc. IEEE Int'l Conf. Computer Commun. (INFOCOM)*, 2013, pp. 1402–1410.

[9] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proc. IEEE Int'l Conf. Computer Commun. (INFOCOM)*, 2012, pp. 2140–2148.

[10] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, August 1992.

[11] Q. Zhao, Y. Zhu, H. Zhu, J. Cao, G. Xue, and B. Li, "Fair energy-efficient sensing task allocation in participatory sensing with smartphones," in *Proc. IEEE Int'l Conf. Computer Commun. (INFOCOM)*, 2014, pp. 1366–1374.