Journal of Scientific & Industrial Research
Vol. 81, January 2022, pp. 94-102

Self-Adaptive Learning and Cellular Automata based Mobile Crowdsensing

Saurabh Anand[1]*, Anant Ram[1] and Manas Kumar Mishra[2]
[1]GLA University, Mathura,Uttar Pradesh, India
[2]VIT Bhopal, Bhopal, Madhya Pradesh, India

Received 29 July 2021; revised 01 December 2021; accepted 28 December 2021

Mobile Crowdsensing (MCS) is frequently utilized for computation assignments, but it is particularly useful for sensing complicated environments. Previously, the MCS platform spent a lot of time and effort establishing incentive mechanisms and task assignment algorithms to encourage mobile users to participate. In actuality, because of their sensing environment and other participants' methodologies, MCS participants face numerous uncertainties, and it is unknown how they interact with one another and make sensing decisions. This study uses the perspectives of MCS participants to develop a web detection arrangement that will maximize their payoffs through MCS participation. Self-adaptive cellular automata-based Markov decision process exhibits interactions among mobile clients and detecting contexts. With the help of Self-Adaptive Support Learning (SASL) and Cellular Automata (CA), we developed a novel method that uses the ideal detecting technique for each client to improve the predicted payoff against random detecting scenarios in a stochastic multi-agent environment. With distinct dynamic sensing, the SASL and CA based smart Crowdsensing enhances user's payoff, as shown in the simulation.

Keywords: Markov decision process, Optimization, Reinforcement learning, Smart crowdsensing

Introduction

Mobile crowdsensing (MCS) has been widely used to collect data from smart devices all around the world. Mobile devices are frequently equipped with a variety of sensors that can be used for a variety of social and commercial functions, including activity tracking, environmental monitoring, social intelligence, and e-commerce.[1]

A standard MCS framework outsources minor detecting tasks to many gadget clients to focus on sensing data and control of mobile devices. Members of mobile crowdsensing face significant vulnerabilities because of the detecting environment and how it interacts with the MCS benefit supplier and other members.[2]

One of the key challenges in MCS is how to imitate diverse clients' interest in detecting programs, because detecting chores use assets and increase members' charges. As a result, extraordinary efforts have been made to plan motivating force components for MCS selection. Using the Stackelberg diversion method, a motivating component was planned. A switch sells off advertising greater flexibility in picking astutely experienced members has been used with three online motivating force instruments.[3] Another fundamental problem is determining how to assign errands to members based on detecting differences.

The current assessment has been focused on the MCS benefit provider assignment. Different types of MCS tasks, as well as delay-tolerant and time-sensitive assignment have been used to model a worker choice system.[5]

The researchers created a multi-objective optimization issue to address a multi-task determination issue.[6] The issue of detecting assignment task has moreover been considered in portable social systems and a web task assignment calculation was outlined employing a greedy technique.[7]

Detecting errands is typically associated with distinct places and MCS members under time constraints which makes the optimum job allocation problem NP-hard. Using approximation calculations, a pleasant errand allotment arrangement with a demonstrated surmised percentage was revealed.[8]

Members of the MCS face a variety of instabilities that influence their decisions due to stochastic detecting conditions, for example, members are contemplating similar detecting strategies that can be used to distinguish detected data. The financial outcomes rewarded by the benefit provider are based not just on the benefit provider's claim efforts, but

*Author for Correspondence
E-mail: sauranand@gmail.com

also on the choices of other members. In any event, most of the prior study[2–8] relied on the platform, and members' vulnerabilities and rapidly changing situations were not considered.

Neither a poor decision approach will not deliver sufficient payoffs to members, nor the benefit provider will be able to obtain high-quality sensed MCS data. Furthermore, considering participants' views and making excellent decisions could be an analytical assignment for using MCS in a certain process. This phenomenon is not limited to participant learning, user inclinations, or alternative options,[9] but it also considers how to establish a series of optimum detecting choices from the perspective of the participants, especially in vulnerable situations.

Reinforcement learning (RL) is a subset of machine learning algorithms that is linked to several types of real-time control scenarios.[10] The Deep RL approach is used to make motion planning in a random environment in an automated manner.[11] A single-agent RL is studied planned incentives for mobile crowd sensing participants,[12] and the researchers simulated support machine learning calculations that maximizes the use of a battery.[13] However, RL is not very useful in a multi-agent system scenario, which is common in many real-time problems, such as mobile crowd sensing, because individual members have distinct natures and preferences.[14]

In addition to the preceding strategies focusing on the information preparation stage, it is also possible to split categories based on the applications listed in Table 1.

Table 1 — List of Crowdsensing and Sensing technologies, together with collected data

Ref.	Technology	Data Type	Application
17	Redundancy in Software n/w	Image	Disaster management
18	Scalability analysis	Mobile data	Property modelling
19	Redundancy reduction	Mobile data	Humidity, Pressure
20	Comparison of meta data	Image, Video	Disaster management
21	Optimization algorithm	Air Pollution data	Cost effective data
22	Kinetic energy (KE) algo	KE	Energy Allocation
23	Redundancy reduction	Text	Paraphrasing
24	Data compression	Questionnaire	Dimension reduction
25	Spatial-Temporal	GPS data	City mobility
26	Pattern Recognition	Mobile data	Sensor stream
27	Comparison of meta-data	Image	Image crowdsensing

Contributions

We use self-adapted Markov decision processes and cellular automata to model the behaviour of mobile crowd sensing users in this study. Each client cannot observe the decisions of others but must pick its level of exertion in detecting activities based on surrounding data, such as its own record of detected signal quality. In that setting, agents make decisions based on the environment's dynamical behaviour or uncertainty, which are accompanied with payoffs. Using SASL and CA, a novel algorithm for "smart crowdsensing" has been presented. We focused on the most important works in specific fields:

1. We consider the perspectives of participants to design an optimum system for determining the endeavours that will maximize participants' payoffs.
2. We are working on a web based MCS computation, specifically self-adaptive Crowd sensing. Deep RL was used to memorize the method of mobile crowd sensing assistance.
3. We presented algorithms in various stochastic and random scenarios, which affects how mobile crowd sensing members are compensated for a set of crowdsensing assignments.

Crowdsensing Architecture

With the help of sensing the dynamical environment and users' decision payoffs, we define the Crowdsensing architecture analytically.

Analytical Formulation of MCS

For the analytical formulation, the authors considered N different users as well as a service provider. Users and service providers can communicate in two ways (as appeared in Fig. 1). Every user has the capacity to acquire or detect data in a specific range or zone for a specific time window.

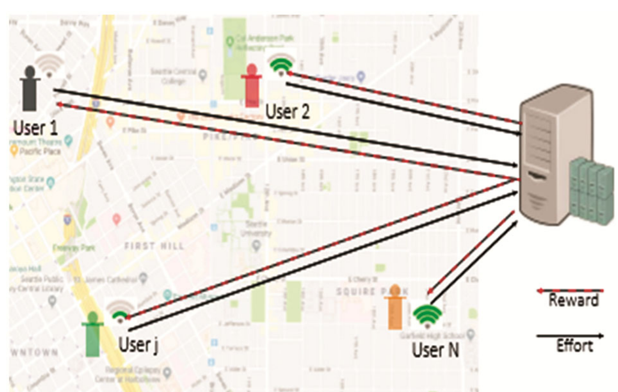

Fig. 1 — Schematic diagram of Mobile crowd sensing Architecture to optimize Users' payoff

In this study, an MCS architecture was used to handle a discrete and definite time t = 1......T, as well as a total number of users N, who are equipped with various sensors and are responsible for sensing information for the service provider.

The service provider recognizes MCS information and provides Rt (Time-Dependent) rewards to each user based on the cost of the user's effort.

Add up all the interested customers' efforts to collect data and allocate a portion of Mt based on commitment to the entire MCS data, which will be analysed later in this section.

Analytical formulation of User behaviour

Each client (i) considers an exertion effort zi, and the detected information provided to the service provider in a specific time range (0 to t) to participate in an MCS assignment. This performance can be measured in terms of user expenses. Every client will also receive an award determined by the benefit supplier based on the quality of the identified data.

We also get the concepts zt = [z1...zN] and m_t = [m1……mN], which represent the total effort and collected Reward at each time instant, whereas Zi is the set of possible effort for operator i.

Following the introduction of the degree of esteem of data (EoD), we discuss the idea of quality of data (QoD) Si(t), which is a real valued quantity that indicates how much quality information has been discovered by users at a specific time instant t. S_t = [s_1(t)……s_N(t)] represents the QoD for all users, with Oi representing the feasible effort of user i.

The mobility of users is a major concern in MCS architecture; because of this mobility, there is variance in the information collected, hence Si(t) may be random or stochastic in character for the time slot and regularly unpredictable. Simultaneously, we may calculate the user effort, which is equal to Si(t) Z_i(t). According to the calculations, the user is unable to determine the impact of other users' efforts on themselves, and each user considers his or her own work autonomous.

Sensing or Payoff Cost

In addition, discovering fetched cost, the remuneration from the benefit supplier determines user i′s payback. The reward is determined using the following formula in Eq. (1):

$$m_i(t) = \frac{z_i(t).s_i(t)}{\Sigma z_j(t).s_j(t)} M_i(t) \quad \ldots (1)$$

where j refers to the number of users from 1 to N.

Because this reward is proportionate to the EoD at each time step, the participation cost or user i's work should be subtracted from the overall reward while calculating the payment cost. Participation is required. The user w_i (t) cost or effort is computed as follows in Eq. (2):

$$w_i(t) = K_i\, Z_i(t) \quad \ldots (2)$$

where Ki >=0 and Ki is determined by the amount of effort exerted by distinct users.

The Payoff cost of the user will be shown in Eqs (3) & (4):

$$\xi_i(t)[(Z_i(t), S_i(t)] = m_i(t) - w_i(t) \quad \ldots (3)$$

$$\xi_i(t)[(Z_i(t), S_i = \frac{z_i(t).s_i(t)}{\Sigma z_j(t).s_j(t)} M_i(t) - K_i\, Z_i(t) \quad \ldots (4)$$

Optimization of Payoff Cost

The payback cost was calculated with a user-centric perspective in mind. Our goal in this study is to find a sequential choice of Zi (t) that includes the discounted payoff, resulting in a total discounted payoff cost $\Theta_i\ (t)$by Eq. (5):

$$\Theta_i\ (t) = \Sigma \mu_i(t)\, \xi_i(t) \quad \ldots (5)$$

where μ_i = [0, 1] and is labelled as user i's discount factor

There are two key flaws with the prior method for determining z_i(t). Because there is no knowledge of payoff functions throughout the learning and training process, it is difficult to make a crowd sensing decision in an unknown or random setting. The first issue is that users are arranged in a random fashion in an unknown or stochastic environment, and there is a large space state for the users' effort and detecting environment, limiting the effectiveness of present algorithms that predict accuracy for future detecting environments. The second challenge is that modelling user behavior in a random environment is challenging. For example, if one user puts out effort z_i(t), this will have an impact on the payoff of other users and play a role in determining future decisions. As a result, we have adopted machine learning techniques, which will self-learn to consider the selected effort and different QoD in a stochastic environment.

Methodology

In this section, we will show how to solve the problem of multi-agent self-adapting support learning, which can regulate a hybrid stochastic environment (both static and discrete time). The SASL is used to solve the single agent crowd sensing problem. The main issue, however, is multi-agent crowd sensing in a stochastic environment with mobile users. Every user is unaware of the efforts of others. SASL is amplified using cellular automata calculations, and we are working on a new Smart Crowd-sensing technique. The innovative approach may simultaneously discover detecting endeavours for each member in a variety of dynamical detecting settings.

In traditional Markov decision models, we are just as interested in the stochastic choice that considers previous states and activities. Normal expansion of Markov decision models to the multi-agent situation with self-adapting learning could be used to determine the payoffs of Mobile Crowdsensing users.

Analytical Formulation

We are looking at a collection of activities $Y_1 ... Y_n$ as well as a set of detected QoD levels $N_1 ...... N_n$ for the whole system.

Each step state is equal to N_1 x N_2 .... x…N_n.

We are going to define QoD and Sensing Effort functions, Transition function for sensing the environment in Eq. (6):

$$f: N_1 \times N_1 \dots\dots N_n \times Y_1 \times \dots Y_N \rightarrow N_1 \dots N_n \qquad \dots (6)$$

Reward for Each Participant is represented by Eq. (7):

$$R_i: N_1 \times N_1 \dots\dots N_n \times Y_i \rightarrow R \qquad \dots (7)$$

Self-adapting learning tries to autonomously tune the parameters of policy Θ_i (t) in a multi-agent Markov decision model. The loss function for optimization is used to provide self-adapting learning in Eq. (8).

$$J(\Theta_i) = (Y - Q(S, Z))^2 \qquad \dots (8)$$

and Y can be defined by Eq. (9).

$$Y = \xi + \mu(MAX)(Q(S_{t+1}; Z_{i+1}) \qquad \dots (9)$$

The following constraints were considered while converting a single agent framework to a multi-agent framework:

1. Each member can monitor both collective effort activities zt and collective reward profiles m_t in the training section.
2. During testing, each team member can use their own set of data, such as QoD observations for specialists.
3. No special communication computations between members about their detection tactics are expected.
4. Each participant may be exposed to random and stochastic QoD elements (shown in Fig. 2).

To detect isolated Q for each specialist using available s_i, z_i and expanding the SASL and defining the settings of multi agent crowd sensing, Because the nature of the environment will be dynamic from the perspective of each MCS user, the pay out of one user will influence the payoff of another. Any user would be unaware of this modification. As a result, RL (Q-Learning)[15,16] is not the best method for memorizing dynamic conditions, and the Q function can upgrade freely with self-adapting learning for each member.

Although it is possible to include all agent selections [$z_1 ...... z_N$] as SASL table/network inputs during the preparation handle to help update the Q function with self-adaption, it is not recommended. Because, just one agent will be working during the testing phase, Q learning will be unable to perform on continuous activities such as the effort levels shown in Fig. 3.

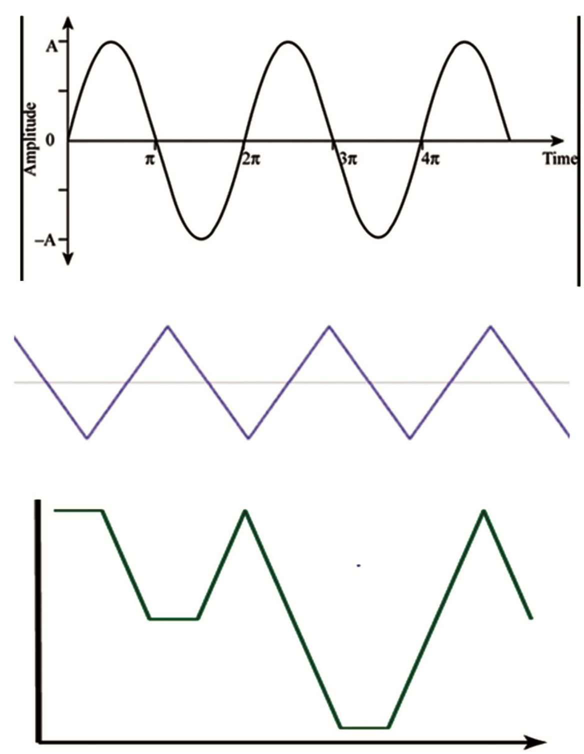

Fig. 2 — QoD dynamics for different Dynamic signals for simulation: (a) Sinusoidal, (b) Linear, (c) Markov chain

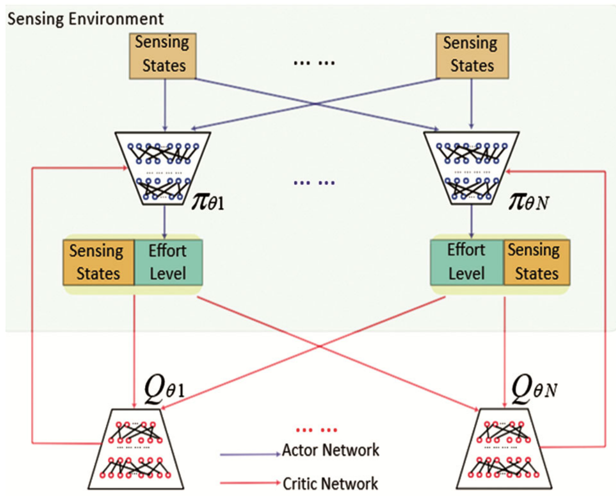

Fig. 3 — Architecture of Multi-agent Self adopting Learning

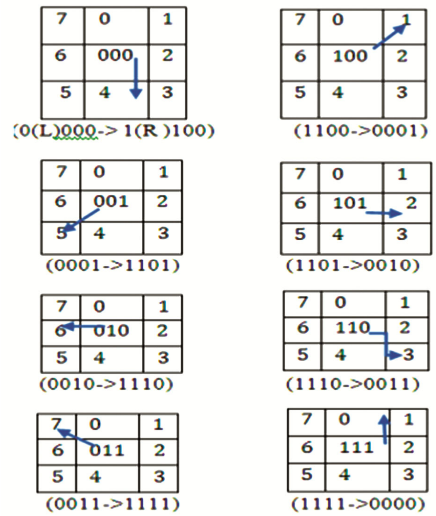

Fig. 4 — Transition behaviour of Users' effort in dynamic environment based on Cellular Automata

Self-adaptive Learning with Cellular Automata: Smart Crowdsensing

We employ the actor-critic network instead of the Q learning technique in this part since the Q learning technique fails to adapt to the multi-agent stochastic environment. As a result, the two neural networks have been used in the actor-critic network model. The $\pi_\Theta(i)$ has learned actor policies for every MCS user, and $Q_\Theta(i)$ has learned critic policies.

In this paradigm, the critic network serves as a Q network in a deep Q network, and the actor network serves as an inference network for the deep Q network, which may be directly mapped from input states to maximize effort. We use the output of critic network. for delivering a better effort level actor network uses the feedback loss of the critic network and adjusts the weights of actor neural network based on that feedback during the training phase.

Once the actor neural network has been trained, the actor network will provide the actions directly at the time of testing; no critic network inputs are required; only actor network inputs are required. We introduced user interaction in the critic part, and the feedback output feeds to the actor network, resolving the issue of user interaction.

Cellular Automata

In the MCS architectural network, the cellular automata (CA) model has been employed to aid the network in forming new arrangements depending on the present and prior states. Users' payoffs were used as states in the MCS architectural network, which alluded to the user's sensory environment.

Because the framework is a mix of static and dynamic environments, CA was employed to model it. As a result, predicting crowd sensing behaviour in terms of cellular automata is more appropriate. All crowdsensing metrics were predicted to be in a binary state of development (z, m, QoD). The parameters might be either static or dynamic. Therefore, the twofold arrangement of binary no (0,1) has been considered for all users. Consider 0 for the static component and 1 for the dynamic component.

We are looking at three components, each of which has two states; thus, the total number of states is $2^3 = 8$. CAs demonstrate how consumers' behaviour changes over time. the state expectations is given by Eq.10.

$$P = P_1 \dots \dots \dots P_8 \qquad \dots (10)$$

S(t) – Current State

S — All discrete states that can be imagined.

From the previous two states, Eq. 11 was applied to predict the future state.

P: A discrete variable with a finite set of states.

$$P(t+1) = P(t).P(t-1) \qquad \dots (11)$$

Consider the states' S= 1 to 8, which means we should expect 8 distinct states in this case. We treated eight distinct states as eight neighbours in our study. In the case of user effort, Eq. 11 is the state prediction. In Fig. 4 a representation of rules is given. The 8 states are illustrated with the self-adaptive algorithm's interchangeable behaviour. The SASL's adaptive nature is enhanced by the cellular Automata. Different scenarios have taken care of changing the states of the hyperparameters, so optimising the reward payoffs with higher adaptability will be beneficial.

After that, policy gradient return computations were utilized to update the Network represented by Eqs (12) & (13):

$$\Delta_{\Theta} J(\Theta) = \Delta_{\Theta}\pi_{\Theta}(i)(S_{t-k} \dots\dots S_t)\Delta_{\Theta} Q_{\Theta}(i)(S^i, Z_1 \dots\dots Z_N) \quad \dots (12)$$

$$Z_i = \pi_{\Theta}(i)(S_{t-k} \dots\dots S_t) \quad \dots (13)$$

Smart Crowdsensing is represented by Algorithm1 which employs self-adaptive learning and cellular automata.

Results and Discussion

To simulate the Smart Crowd Sensing Algorithm, different sensing environments and characteristics were used. To learn the detecting policy over effort level, four groups of actor-critic networks were built for four different MCS users. For the actor-critic neural network, a two-layer fully associated neural network was used. The window side (K) has been considered for training and testing purposes, and it can vary depending on the experimental setup.

K is also used to indicate how much genuine QoD data ($s_t \dots\dots s_t(K)$) has been incorporated into the actor-critic network. Normal batch normalization and weight update procedures were utilized during the training period. The training has been completed up to T=50 steps, and each user's payoffs have been calculated up to T=50 steps.

Algorithm 1: Self adaptive learning and Cellular Automata Based Smart Crowdsensing Algorithm

```
Input: No. of Crowdsensing users → N, discount
       factor → ϵ, Window Length → k, Batch
       → P
Output: State → Z,S, QoD
Data: Weights of the SASL (θi), (ϕi)
initial state z1
while θi,t = Σ ϵi,t × ζi,t do
    for t= 1 to T do
        for i= 1 to N do
            Zi(t) = Π θi[S(t−k)....St]
            Calculate payoff θi,t, calculate
            next-step environment S(t+1)
            (st, zi, θi, t, s(t+1))
            s(t+1) → st
if users behavior is dynamic then
    Adopt the Rules of CAs
    to transition of state
    P(t + 1) = P(t) × P(t − 1)
    if user is static then
        Adopt SASL
for i= 1 to N do
    J(ϕ) = (y − Qϕ(s, z))^2
    y = ζ + ϵmax × Qϕ(s(t+1), z(t+1))
    Updation of Actor-Network
    Δθi J(θi) = Δθi Πθi(s(t−k).....st) Δzi ϕi(s^j, z1.....zn)
    Zi = Πθi(S(t−k).....St)
```

The simulation results for four participants with sinusoidal QoD flow are illustrated in Figs 5 & 6. The graph depicts the average payoffs as a function of time (episodes). The mean values were discovered after 10 simulation runs. The four users that used the window length K displayed the same learning behaviors. All users require knowledge of the QoD aspects as well as other users' preferred designs.

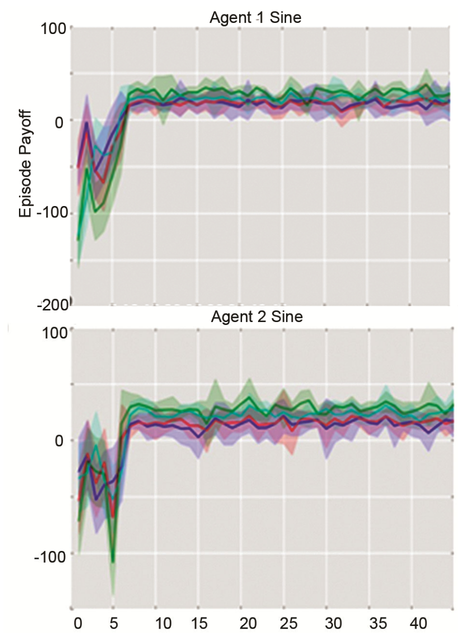

Fig. 5 — Sinusoidal Training Dynamics for all users (agent 1, 2)

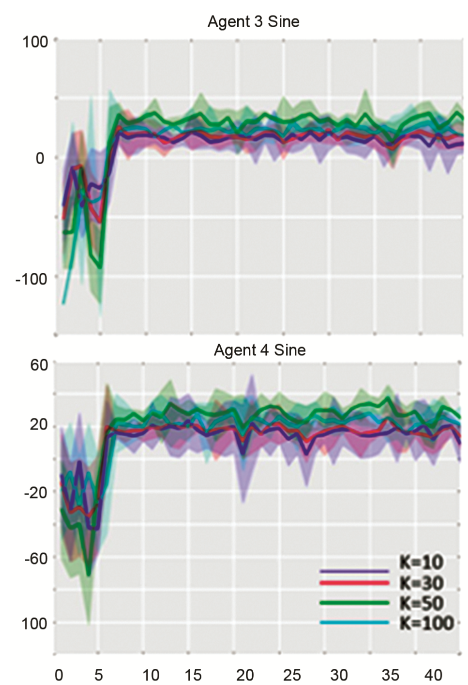

Fig. 6 — Sinusoidal Training Dynamics for all users (agent 3, 4)

That is why the plots show negative payoffs at the start of the training, implying that the service provider did not deliver sufficient rewards to the users at the start of the training and did not begin to adopt a suitable approach on z_i. These payoffs may stabilize as more training happens, indicating that the weights of the neural network are stabilizing.

When K varies from 10 to 50 in the simulation part, every user receives bigger payoffs, so we can deduce that the smart crowdsensing method will operate better when it finds more accurate past data. When we raise the value of K= 50 to 100, the previous data become less useful in determining optimal sensing policies. Increasing the value of K makes training the neural network for these windows more computationally intensive.

The normal obtained awards for four distinct types of users after four different forms of Dynamic QoD were applied in certain time steps (episodes) following the training (Table 2). The simulation of a blend of QoD dynamics is shown in Figs 7 & 8. We found that at K=50, we received the greatest optimal values of crowd sensing payoffs for our experimental setup.

Table 2 — Testing Rewards for Four users varying with value K

Memory Length	10	30	50	100
Linear	47.37	49.86	51.02	49.65
Sinusoidal	29.67	26.78	39.12	29.06
Markov	36.11	39.71	43.26	41.98
Mixed	21.71	36.12	39.10	26.74

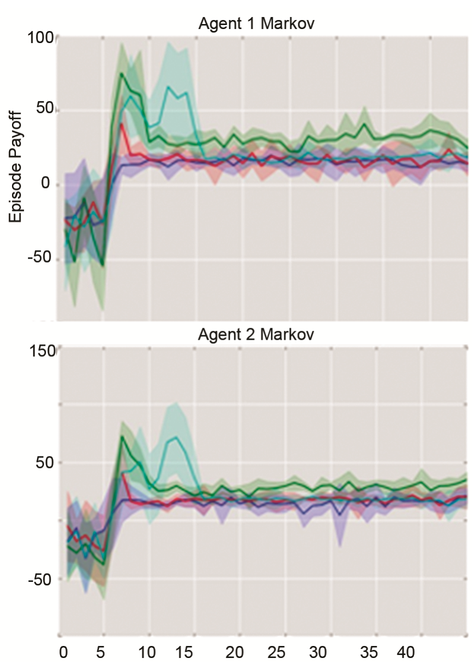

Fig. 7 — Mixture of linear, Sinusoidal, Markov Training Dynamics for all users (agent 1, 2)

We also compare our method to the usual model-predictive control (MPC)[28–30] which is employed for resource allocation and optimal control. To get the expected optimal sensing actions at each timestep, we fit each agent a local model for the reward dynamics and solve the MPC with fixed T. The average outcomes for SASL, where each agent trains an independent actor-critic learner to learn its own sensing judgments, are also treated as a baseline learner. In this design, there is no consideration for agent communication or coordination.

In all situations, the MPC delivers solutions with substantially smaller rewards than the SASL algorithms, as demonstrated in Table 3. We discovered that under mixed sensing dynamics, the policy discovered using MPC has a negative payoff. This is achievable because MPC uses each agent's single observation to guide each agent to attain smaller rewards with high sensing costs.

Our proposed approach is also used to implement a larger-scale MCS challenge.[29] As illustrated in Fig. 8, as more training episodes are applied, the average reward for the 4-agent and 8-agent cases both converge. It requires fewer samples to train an effective RL decision maker with a reduced number of agents.

However, because a multi-agent actor-critic SASL has been trained, the local learning-based agent can use the knowledge provided by the critic to make

Table 3 — Comparison with MPC and SASL

Memory Length	MPC	SASL
Linear	1.62	23.65
Sinusoidal	0.63	25.98
Markov	1.14	12.65
Mixed	−40.43	10.76

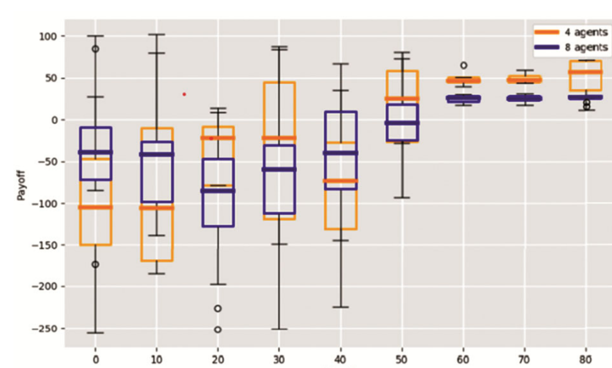

Fig. 8 — Participants' rewards for training episodes in the MCS (Boxplot)

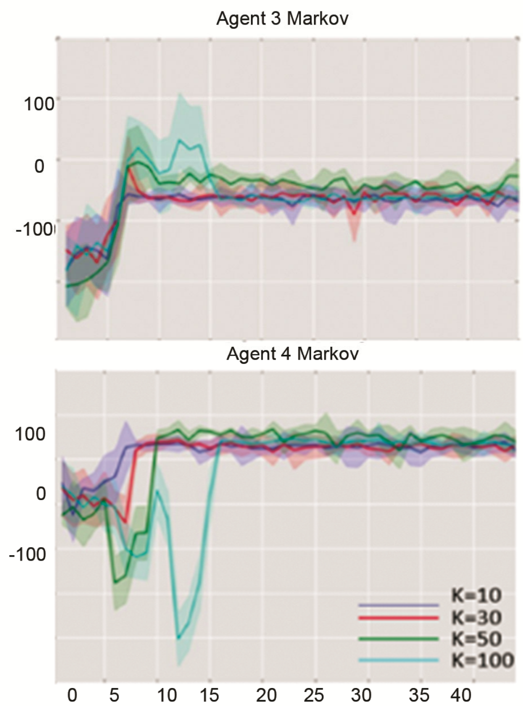

Fig. 9 — Mixture of linear, Sinusoidal, Markov Training Dynamics for all users (agent 3, 4)

cooperative decisions. More intriguingly, the MPC agent performs the poorest on the mixture of QoI dynamics, presumably because the linear model is unable to discover a good representation of system dynamics, while the local agents are not cooperating effectively.

Because of the dynamic environment, users' mobility and sensing capabilities changing over time, three forms of heterogeneous QoD dynamics have been used. In this case, QoD temporal evolution has been modelled using sine dynamics, linear dynamics, and Markov dynamics (shown in Fig. 9). By assessing parameters amplitudes (A), frequencies (F), and transition frameworks for diverse MCS users, framework aspects have been made more challenging for a novel algorithm of smart crowdsensing. We also allowed for the possibility of a negative signal to avoid erroneous and false information.

Conclusions

This study has explored the engagement of MCS participants and the difficulty of selecting the sensory effort to maximize the payout for each user. First, because participants are confronted with random detecting scenarios, complex communications, as well as MCS members, the issues of simulating and decision-making have been solved. Then, for smart Crowdsensing, we propose a self-adaptive learning with cellular automata computation that can employ the control of a deep SASL with a Markov decision model to effectively discover the best detecting choice for each member in real time. Simulations in various obscure detecting conditions validate our innovative smart Crowdsensing technique. We will also investigate benefit providers component plan and MCS members' decision-making in future study, and test our ideas using real-world crowdsensing data.

For this investigation, some constraints were considered.

1. The authors' tests are limited to fourQoD signals.
2. The MCS's social contact was not considered in this study.
3. The experimental work did not consider the relationship between the service provider and the MCS worker.
4. Third-party attacks are not considered when calculating the balance, profit, and so on.

Real-time MCS data, as well as the integration of various types of QoD, will be studied in the future. In the future, the relationship mechanism between the service provider and the MCS worker could be examined to improve the algorithm's effectiveness. The cellular automata in this study only investigated three states; however, more states could be considered in the future to attain more effective results.

References

1 Ganti R K, Ye F & Lei H, Mobile crowdsensing: current state and future challenges, *IEEE Commun Mag*, **49/11**(2011).

2 Yang D, Xue G, Fang X & Tang J, Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing, in *Proc 18th Ann Int Conf Mobile Comput Netw*, ACM, (2012), 173184.

3 Zhang X, Yang Z, Zhou Z, Cai H, Chen L & Li X, Free market of crowdsourcing: Incentive mechanism design for mobile sensing, *IEEE Trans Parallel Distribut Syst*, **25/12** (2014) 3190–3200.

4 Zhan Y, Xia Y & Zhang J, Incentive mechanism in platform-centric mobile crowdsensing: A one-to- many bargaining approach, *Comput Netw*, **132** (2018) 40–52,

5 Guo B, Liu Y, Wu W, Yu Z & Han Q, Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems, *IEEE Trans Hum Mach Syst*, **47/3**(2017) 392–403.

6 Liu Y, Guo B, Wang Y, Wu W, Yu Z & Zhang D, Taskme: multi-task allocation in mobile crowd sensing, *Proc 2016 ACM Int Joint Conf Pervas Ubiquit Comput, ACM* (2016) 403–414.

7 Xiao M, Wu J, Huang L, Wang Y & Liu C, Multi-task assignment for crowdsensing in mobile social networks,

2015 IEEE Conference on. Comput Commun (INFOCOM), (2015) 2227–2235.

8 He S, Shin D-H, Zhang J & Chen J, Toward optimal allocation of location dependent tasks in crowdsensing, IEEE INFOCOM 2014 — IEEE Conf Comput Commun, 2014 IEEE INFOCOM 2014, 745–753, doi: 10.1109/INFOCOM.2014.6848001.

9 Karaliopoulos M, Koutsopoulos I & Titsias M, First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling, *Proc 17th ACM Int Symp Mobile Ad Hoc Netw Comput*, ACM (2016) 271–280.

10 Sutton R S, Barto A G, Reinforcement learning: An introduction, MIT press, (1998).

11 Schulman , Levine S, Abbeel P, Jordan M & Moritz P, Trust region policy optimization, *Int Conf Mach Learn*, (2015) 1889–1897.

12 Wang H & Zhang B, Energy storage arbitrage in real-time markets via reinforcement learning, *arXiv preprint* arXiv:1711.03127, (2017).

13 Xiao L, Li Y, Han G, Dai H & Poor H V, A secure mobile crowdsensing game with deep reinforcement learning, *IEEE Trans Inf Forens Secur*, (2017).

14 Lowe R, Wu Y, Tamar A, Harb J, Abbeel O P & Mordatch I, Multi-agent actor-critic for mixed cooperative-competitive environments, *Adv Neural Inf Process Syst*, (2017) 6379–6390.

15 Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, Graves A, Riedmiller M, A. Fidjeland K, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S & Hassabis D, Human-level control through deep reinforcement learning, *Nature*, **518(7540)** (2015) 529.

16 Mnih V, Badia A P, Mirza M, Graves A, Lillicrap T, Harley T, Silver D & Kavukcuoglu K, Asynchronous methods for deep reinforcement learning, in *Int Conf Mach Learn*, (2016) 1928–1937.

17 Hua Y, He W, Liu X and Feng D, SmartEye: Real-time and efficient cloud image sharing for disaster environments, *Proc INFOCOM*, 2015.

18 Dao T, Roy-Chowdhury A K, Madhyastha H V, Krishnamurthy S V & Porta T L, Managing redundant content in bandwidth constrained wireless networks, *Proc CoNEXT*, 2014, 349–361.

19 Whitmore A, Agarwal A & Xu L D, The Internet of Things survey of topics and trends, *Inf Syst Front*, **17(2)** (2015) 261–274.

20 Gorlatova M, Sarik J, Grebla G, Cong M, Kymissis I & Zussman G, Movers and shakers: Kinetic energy harvesting for the internet of things, *Proc SIGMETRICS*, Austin, 2014, 407–419.

21 Yan T, Kumar V & Ganesan D, CrowdSearch: Exploiting crowds for accurate real-time image search on mobile phones, *Proc MobiSys*, June 15–18, 2010, San Francisco, California, USA, 2010.

22 Aggarwal C C, Xie Y & Yu P S, On dynamic data-driven selection of sensor streams, *KDD* (2011).

23 Willett W, Ginosar S, Steinitz A, Hartmann B & Agrawala M, Identifying redundancy and exposing provenance in crowdsourced data analysis, *TVCG* **19(12)** (2013) 2198–2206.

24 Zhang D, Xiong H, Wang L & Chen G, CrowdRecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint, *Proc ACM UbiComp*, (2014), 703–714,

25 Zhu Z, Zhang L & Wakikawa R, Supporting mobility for internet cars, *IEEE Commun Mag*, **49(5)** (2011) 180–186.

26 Chen C, Zhang D, Li N & Zhou Z, 2014, B-Planner: Planning bidirectional night bus routes using large scale taxi GPS traces, *IEEE Trans Intell Transport Syst*, **15(4)** (2014) 1451–1465.

27 Chon Y, Lane N, Kim Y, Zhao F & Cha H, Understanding the coverage and scalability of place- centric crowdsensing, *Proc UbiComp*, 2013.

28 Yang W & Ai T, Refueling stop activity detection and gas station extraction using crowdsourcing vehicle trajectory, *Acta Geod Cartogr Sin*, **46** (2017) 918–927.

29 Pryss R, Mobile crowdsensing in healthcare scenarios: taxonomy, conceptual pillars, smart mobile crowdsensing services, in *Digital Phenotyping and Mobile Sensing*, edited by H Baumeister & C Montag (Springer) 2019, 221–234.

30 Tokosi T O & Scholtz B M, A classification framework of mobile health crowdsensing research: a scoping review, *Proc South African Inst Comput Sci Inf Technol*, (2019).