Minimizing the Age-of-Critical-Information: An Imitation Learning-based Scheduling Approach Under Partial Observations

Xiaojie Wang, Zhaolong Ning, Song Guo, *Fellow, IEEE*, Miaowen Wen, and H. Vincent Poor, *Life Fellow, IEEE*

Abstract—Recently, Age of Information (AoI) has become an important metric to evaluate the freshness of information, and studies on minimizing AoI in wireless networks have drawn extensive attention. In mobile edge networks, the change of critical levels for distinct information is important for users' decision making, especially when merely partial observations are available. However, existing researches have not addressed that issue yet. To tackle the above challenges, we first establish the system model, in which the information freshness is quantified by the changes of its critical levels. We formulate the Age-of-Critical-Information (AoCI) minimization issue as an optimization problem, with the purpose of minimizing the average relative AoCI of mobile clients to help them make timely decisions. Then, we propose an information-aware heuristic algorithm that can reach optimal performance with full obsevations in an offline manner. For online scheduling, an imitation learning-based scheduling approach is designed to decide update preferences for mobile clients under partial observations, where policies obtained by the above heuristic algorithm are utilized for expert policies. At last, we demonstrate the superiority of our designed algorithm from both theoretical and experimental perspectives.

Index Terms—Age of information, imitation learning, mobile edge networks, scheduling policy, critical levels.

1 Introduction

With the boom in new technologies and the emergence of multifarious applications, individuals have become heavily dependent on mobile terminals to obtain information, including news, advertisements, weather reports and notifications. For instance, a vehicle moving on the road can require the updated traffic information about some locations along its routes through Road Side Units (RSUs) to make driving plans. Another example is that sensors desire real-time updates of the channel state to monitor the environment and feed back information to servers through wireless communication systems. Therefore, the freshness of information has become a significant metric to qualify the experience of users in information centric systems.

Currently, *Age of Information* (AoI) has been utilized to measure the freshness of information. Generally, it is defined as the time elapsed from the last time when the information was updated to now. This concept was first introduced in [1] to capture the timeless requirements of applications that broadcast their information periodically. A vivid interest has been attracted for AoI since then, and it has been taken into consideration by variety of researches, such as queueing systems [2], mobile edge computing networks [3, 4] and wireless communication systems [5, 6]. Existing studies mainly focus on minimizing the average or peak AoI of clients under the constraints of wireless communication resources. Many scheduling policies have been proposed to achieve the above purpose based on request-response models, where multiple pairs of servers and clients coexist, and relay nodes can be leveraged for information transmission [7]. However, the importance of information is not explicitly expressed, and the fact that different information generally has different impacts on users' decisions is generally neglected [8].

1.1 Motivating Example

A representative application example of AoI minimization is sending traffic information of different roads to drivers and passengers in vehicular networks. To keep the freshness of local information, terminal users always require timely update. Due to the limited number of channels, not all users can update their information timely. The management server can design an information update preference for required terminals based on the AoI values of their information. However, on one hand, different users may be interested in diverse information, and distinct information can have different impacts on user decisions. On the other hand, users may not intend to expose their personal profiles to the server on account of individual privacy. Thus, merely partial observations of user status are available for the management server. Unfortunately, existing researchers have always neglected the above important factors for information update scheduling.

To illustrate the motivation of this paper, let's take an example as shown in Fig. 1. We consider that six road

- *X. Wang and S. Guo (Corresponding author) are with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China. E-mail: xiaojie.wang@polyu.edu.hk, song.guo@polyu.edu.hk.*
- *Z. Ning (Corresponding author) is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. E-mail: z.ning@ieee.org.*
- *M. Wen is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China. E-mail: eemwwen@scut.edu.cn.*
- *H. V. Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. E-mail: poor@princeton.edu.*

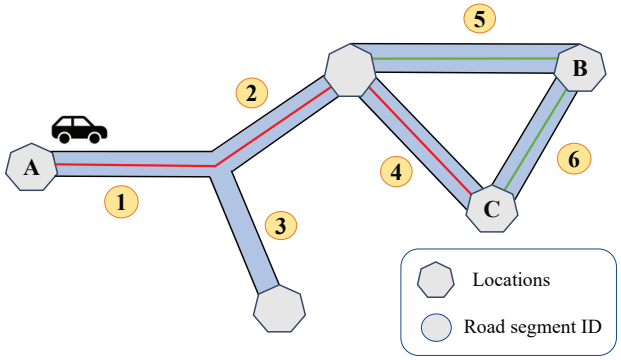


Fig. 1: An AoI related road network: For a vehicle from locations A to C, the traffic information of different road segments, especially 4, 5, and 6, affects the driver's decision for road selection.

segments have various traffic information, i.e., traffic status about road segments 1 to 6 is different. A vehicle prepares to go to locations C from A, and it concerns traffic information about road segments 1, 2, 4, 5 and 6. Consider that the traffic information of road segment 4 has two critical levels: i) a happened traffic jam; and ii) smooth traffic flow. Commonly, if the critical level of the information does not change, it has little impact on the vehicle's decision. That is, if the critical level of road segment 4 is ii in time slot t, the vehicle will choose the path through segments 1, 2 and 4 to destination. Nevertheless, if the corresponding level changes to i in time slot $t+1$, the vehicle will change its decision by selecting the path containing segments 1, 2, 5 and 6. Therefore, it is a priority to update the information when its level changes. The vehicle may concern the traffic condition of road segment 4 more than those of 5 and 6, since the length of the path through 1, 2 and 4 is shorter than that of 1, 2, 5 and 6.

1.2 Challenges

Based on the above considerations, the following challenges exist to design a feasible scheduling policy for information update:

- How to treat the information based on their different categories and critical levels is necessary to investigate, since it heavily impacts users' decisions. However, there is no existing work focusing on different impacts of various information, user interests and dynamic networks simultaneously, to the best of our knowledge.
- Due to the privacy concern, users may not reveal all their personal information to the server, and only expose what they need to update [9]. For example, their interest ratios for different information are not revealed. Thus, existing scheduling policies based on full observations are not applicable for such partial observations of the system state.
- Dealing with the dynamics of mobile terminals is necessary for realistic network scenarios, given the fact that they cannot always stay in specific locations. Meanwhile, novel scheduling algorithms based on both wireless resource constraints and information diversities are challenging. Capturing the change of critical level for users via limited wireless spectrum should be elaborately designed to meet individual information update demands and support users to make timely decisions.

Overall, information diversity, partial observations, user dynamics, and limited wireless communication bandwidths make the design of online scheduling algorithm rather challenging.

1.3 Contributions

The purpose of this paper is to minimize the average Age of Critical Information (AoCI) of mobile clients by designing feasible scheduling algorithms. We define AoCI as the utility of critical information related to the factors that have direct impacts on user decisions, including critical levels, user interests, information categories and AoI. We propose an imitation learning-based scheduling algorithm, named LISA, that allows the learning agent to imitate the behaviors of experts. The expert data can be collected based on conducting the information-aware heuristic algorithm offline, where the learning agent can imitate to find efficient policies with further possible state. It is similar to supervised learning, but more intelligent since it can guide agents to tackle situations never met before. To the best of our knowledge, we are the first to consider minimizing the average AoCI of mobile clients under partial observations based on imitation learning. Our contributions are summarized as follows:

- We first establish the system model based on request-response communications, and formulate the information update scheduling issue as an optimization problem. We define the concept of AoCI to evaluate the importance of information.
- We propose an offline scheduling algorithm, i.e., an information-aware heuristic algorithm based on dynamic programming, which can obtain the optimal scheduling solution for mobile clients based on the full knowledge of personal profiles. It is suitable to act as the expert policy for online learning.
- For online learning under partial observations, we design an imitation learning-based algorithm that allows the learning agent to mimic the behaviors of the expert, and can obtain a near-optimal scheduling solution under the guideline of the expert. Specifically, technologies of Variational Auto Encoder (VAE) and Multi-Layer Perceptron (MLP) are leveraged in the training process.
- We demonstrate the superiority of our designed algorithm from both theoretical analysis and experimental scenarios. Compared with representative studies, experimental results show that our algorithm has advantages over the average AoCI under various network parameters and has a short convergence time.

The rest of this paper is structured as follows: in Section 2, we review the related work; we illustrate the system model and formulate the studied problem in Section 3; in Section 4, we design an information-aware heuristic algorithm that can be conducted offline, followed by presenting

the designed imitation learning-based scheduling algorithm in Section 5; then, we introduce the experiment setting and discuss the experimental results in Section 6; finally, we conclude this paper in Section 7.

2 Related Work

In this section, we review the recent studies about AoI and imitation learning.

2.1 AoI

AoI is defined as a metric to evaluate the freshness of information [10]. Existing studies focus on AoI minimization under the constraint of wireless communication resources, and its applications in real-time information networks. The authors in [11] focus on a system consisting of a central station and several terminals. A mobile agent exists in the former to help update information of these terminals, while the latter designs distributed scheduling policies for traffic flows in wireless networks. The above considered model is similar with ours, but their scheduling policies are based on the full knowledge of the system state. Similar to our work, researches [13] and [14] consider the bandwidth constraints in wireless networks, but neglecting the weighted impact of information for Mobile Clients (MCs). A system including multiple terminals collecting data to a base station is considered in [4], while multiple end users upload their video frames to the edge servers for processing in [15]. Information update between a server and a terminal by several intermittent relay nodes has been studied in [18].

Although there exists some scheduling policies related to AoI, the category, critical level of information and the interest ratio of users are not focused, which have great impacts on personal decisions. The authors in [19] propose a metric named Age of Incorrect Information (AoII), but it along with the corresponding scheduling method cannot be applied in our system, because: 1) it focuses on network status updates instead of information critical levels and the interest ratio of users, whereas one critical level of one single information category may refer to a set of network status with different interest ratios to distinct MCs; 2) it is not suitable for highly-dynamic application scenarios in this paper, since static network situations and merely one pair server-client model are considered; 3) we concentrate on user-oriented scheduling with partial observable information, while the authors in [19] design a server-oriented method with full network observation.

2.2 Imitation Learning

Imitation learning aims to mimic expert behaviors in a given task. A mapping between states and actions can be learned by training a learning agent based on expert demonstrations. Then, the agent can perform tasks based on the learned model [20]. The advantages of imitation learning can be summarized as: (1) it can utilize few expert demonstrations to teach complex tasks; (2) it does not need to deliberately design a reward function related to the task; and (3) compared with reinforcement learning, it has good performance from the beginning based on the expert supervision.

DAGGER [31] is a typical imitation learning method, iteratively training a deterministic policy based on Markov Decision Processes (MDPs). At first, it utilizes the expert policy to obtain trajectories D. Then, at each iteration, it trains policy $\hat{\pi}$ to best mimic the expert, and collects trajectories for D. Its objective is to find a policy $\hat{\pi}$ that can minimize the surrogate loss under the distribution of real states, i.e.,

$$\hat{\pi} = arg \min_{\pi \in \Pi} E_{s \sim d_\pi}[l(s, \pi)], \tag{1}$$

where Π is the class of policies the agent should consider, and $l(\cdot)$ is the loss function to be minimized when training policy $\hat{\pi}$. Function $E[\cdot]$ calculates the expected value of its input.

Currently, imitation learning has a wide spread of applications, such as robotic motion planning [22], autonomous driving [23] and information gathering [25]. This algorithm allows the agent to imitate the expert that can compute the best information sensing locations based on its full knowledge of the world maps. However, the application of imitation learning in wireless networks is still in its initial stage. An imitation learning enabled resource allocation algorithm is proposed in [26] to accelerate the solving speed of optimization problems. For device-to-device communications, imitation learning is utilized to find suitable resource allocation strategy in wireless networks, where the learning speed can be accelerated based on expert trajectories [27].

Different from existing researches, we investigate imitation learning with AoI-related information update scheduling. Our objective is to find feasible solutions for the scheduling issue under the consideration of information diversities, user preferences, dynamic networks and partial observations. To the best of our knowledge, we are the first to design an imitation learning-based scheduling approach for the AoI-related issue under partial observations.

3 System Model And Problem Formulation

In this section, we present the considered system model and formulate the optimization problem.

3.1 System Model

As illustrated in Fig. 2, a request-response model is investigated, where a server can provide services for MCs with maximum number K due to its bandwidth constraint. This model is general in the real world, and can be applied in many scenarios. For instance, the server can be an RSU in vehicular networks to serve multiple passing-by vehicles [28, 29], or a base station in cellular networks providing services for mobile users [30]. Without loss of generality, it is reasonable to consider the AoCI minimization issue in this general network framework. For simplicity, we refer to different terminals served by the server as MCs, which can randomly enter in or leave the wireless coverage of the server. We consider that the information update scheduling is determined in each time slot, and three periods are included.

At the beginning of time slot t, the updated information (various real-time information from sensors or pushing services from a remote center) randomly arrives at the local

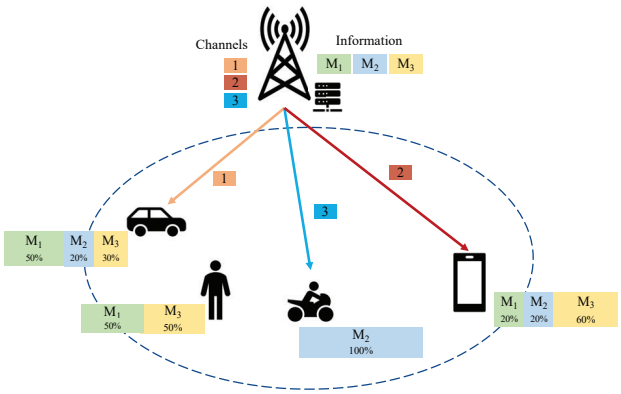


Fig. 2: Illustrative system model.

server, and $N(t)$ MCs send requests to the local server for information update in time slot t. Then, the server determines a scheduling preference to update the information of MCs. Note that not all clients have the update opportunity in each time slot, and not all information belonging to one client can be updated at the same time. There are $\min\{N(t), K\}$ MCs can initialize the information update process. At the end of each time slot, the clients receive the required information. In this work, we mainly focus on the update scheduling issue for MCs. First, several formal definitions are given as follows:

Definition 1 (Information category). *One information category is related to a specific event that MCs focus on. In our system, the information can be classified into $|M|$ categories, denoted by $M = \{1, ..., |M|\}$, and each client can be interested in several information categories.*

Definition 2 (Critical level). *Each information category can have several critical levels that reflect different statuses of one specific event changing with time. In our work, each kind of information has $|L|$ critical levels, denoted by $L = \{1, 2, ..., |L|\}$.*

For one information category f_j that MC u_i focuses, its critical level may be different from that on the server, when u_i does not get update. Variables $v(t, f_j)$ and $v(t, u_i, f_j)$ represent the critical level of category f_j on the server and MC u_i in time slot t, respectively. In the following, we provide the definition of level difference:

Definition 3 (Level difference). *The level difference for one information category on an MC can be defined as the difference between its local critical level and that on the server, i.e., $| v(t, f_j) - v(t, u_i, f_j) |$.*

For each MC, it can be interested in several information categories, and prepare to receive their update information.

3.2 AoCI

To model the impact of information with different categories and critical levels, we introduce the concept of AoCI, defined as the relative age of critical information that has significant impacts on user decisions. It also implies that capturing the change of critical levels as early as possible is significant to avoid missing the chance of making appropriate decisions. For information category f_j that MC u_i focuses on, we first provide the expression of its AoI in time slot t by:

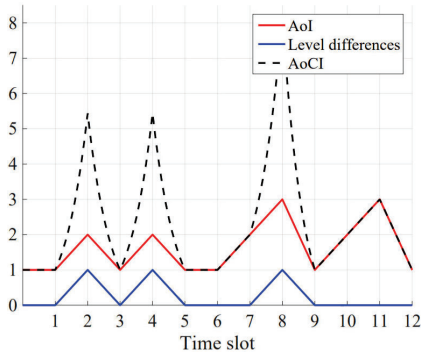


Fig. 3: Example change in AoI, level difference and AoCI.

$$h(t, u_i, f_j) = \begin{cases} 1, & \text{if } \xi_i(t) = 1 \text{ and} \\ & \beta_{ij}(t) = 1, \\ h(t-1, u_i, f_j) + 1, & \text{otherwise.} \end{cases} \quad (2)$$

We utilize $\xi_i(t) \in \{0, 1\}$ to denote the server decision on MC u_i in time slot t. If u_i is selected for update, $\xi_i(t) = 1$. Otherwise, $\xi_i(t) = 0$. Variable $\beta_{ij}(t)$ is an indicator, representing whether information category f_j of MC u_i is updated or not.

Then, we derive the AoCI for each information category. It begins with an example in Fig. 3, which shows the update trends of one information category related to an MC. The horizontal axis represents the change of time slots, and the vertical axis is the value of three-related metrics, i.e., AoI, level differences and AoCI. The server updates its local information in each time slot, and the MC requests update for its focused information category (e.g., category A). We assume that the critical level of A on the server changes in time slots 2, 4 and 8, while the MC obtains update in time slots 1, 3, 5, 6, 9 and 12. After MC updates, the AoI of its information category drops to 1 and the level difference becomes 0, illustrated by the red and blue lines, respectively. We intend to describe the AoCI curve that can reflect the variation trends of both AoI and level differences as the black dotted line shows, and provide the following definition:

Definition 4 (AoCI). *The AoCI of one information category reflects the utility of critical information that has significant relationships with the AoI and level difference, i.e.,*

$$I(t, u_i, f_j) = h(t, u_i, f_j)\, b^{|v(t,f_j) - v(t,u_i,f_j)|}, \quad (3)$$

where b is a constant value and above one.

The specified explanation of Definition 4 can be found in Appendix A of Supplementary File. If $\beta_{ij}(t) = 1$, the final value of AoCI in time slot t is $I(t, u_i, f_j) = 1$.

It is worth noticing that the scheduling policy of AoCI is quite different from that of AoI. Since we are the first to investigate impacts on users' decisions caused by different information categories and critical levels, a toy example is provided to show the differences between these two kinds of scheduling policies, as well as the advantages by considering AoCI instead of AoI. As shown in Fig. 4, a

system consists of three MCs and one server that can update one information in each time slot. Each MC is simplified to have interests in only one information category. For example, MC 1, MC 2 and MC 3 are 100% interested in information categories A, B and C, respectively. A greedy scheduling can be leveraged here for the sample scenario by giving a priority to the bigger AoI or AoCI. We notice that the schedule for AoI is in a round-robin manner, while that for AoCI mainly depends on the level change of the server. The result of the scheduling policy for AoCI is better than that of AoI, since it can schedule the information with level differences as early as possible. Thus, MCs can timely update the information that may affect their decisions.

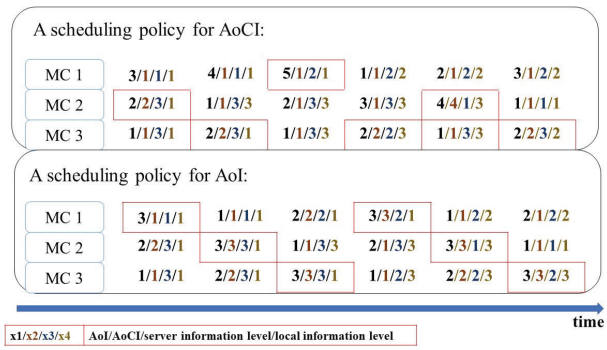


Fig. 4: Scheduling policies for AoI and AoCI.

3.3 Problem Formulation

The purpose of this paper is to timely capture the impact of AoCI for MCs. In other words, we intend to minimize the average relative AoCI for MCs. Then, the information with a bigger AoCI and a greater level difference will have a higher priority to update. The relative AoCI of MC u_i is:

$$I\left(t,u_i\right)=\sum_{j=1}^{|M|}\alpha_{ij}I\left(t,u_i,f_j\right). \tag{4}$$

We consider that each MC has different interests for distinct information categories. Symbol α_{ij} denotes the interest ratio of information category f_j for MC u_i, and $\sum_{j=1}^{|M|}\alpha_{ij}=1$. The problem of minimizing the average relative AoCI of MCs is formulated as follows:

$$\begin{aligned}
\text{P1:}\quad & \min_{\xi_i(t),\beta_{ij}(t)}\sum_{t=1}^{T}\sum_{i=1}^{N(t)}\frac{1}{TN(t)}I\left(t,u_i\right)\\
&=\sum_{t=1}^{T}\sum_{i=1}^{N(t)}\sum_{j=1}^{|M|}\frac{1}{TN(t)}I\left(t,u_i,f_j\right)\alpha_{ij}, && (5)\\
\text{s.t.}\quad & N\left(t\right)\le K, && (6)\\
& \sum_{j=1}^{|M|}\frac{s_j\beta_{ij}\left(t\right)}{r_n}\le\hat{t}, && (7)\\
& \sum_{j=1}^{|M|}\alpha_{ij}=1, && (8)\\
& 0\le v(t,f_j)\le|L|,0\le t\le T, && (9)\\
& 0\le|v(t,f_j)-v(t-1,f_j)|\le|L|, && (10)
\end{aligned}$$

where equation (5) is our objective, aiming to minimize the average relative AoCI of MCs. We assume that the number of MCs during one time slot remains unchanged, while increasing or decreasing at the beginning of each time slot. It is reasonable and realistic, because we can always find a schedule that guarantees one time slot short enough to keep the number of MCs invariant. Constraint (6) makes sure that the number of selected MCs by the server cannot exceed the maximum number K. Constraint (7) guarantees that the total information transmission delay of MC u_i cannot exceed time span $\hat{t}$ of one time slot, where s_j is the current information size of category f_j, and r_n is the transmission rate. The total interest ratios for all categories of one MC should be 1 as defined in equation (8). Constraints (9) and (10) ensure that the critical level in each time slot is below $|L|$, and the absolute value of level changes should not exceed $|L|$.

To solve the formulated optimization problem, we consider two situations: 1) all the MCs' personal knowledge is known by the server; 2) partial knowledge is known. For the above two situations, different scheduling policies are designed, i.e., an information-aware heuristic algorithm and an imitation learning-based scheduling algorithm. The former one can find the optimal scheduling policy based on all the personal knowledge. The latter one leverages imitation learning in the scheduling processes. A learning model can be trained via offline based on the state-action pairs obtained by the heuristic algorithm. Then, the learning agent can make proper online scheduling decisions merely based on partial known information. The above two algorithms will be specified in the following two sections, and the main notations can be found in Table 1.

4 An Information-aware Heuristic Algorithm

In this section, we specify the designed information-aware heuristic algorithm. We first transfer the original problem into subproblems, and then design a heuristic algorithm based on the MCs' personal profiles.

4.1 Subproblem Transformation

Since the purpose of Problem P1 is to minimize the average related AoCI for MCs, and the change of AoCI can be captured in each time slot, we intend to minimize the related AoCI in each time slot by the following subproblem:

$$\begin{aligned}
\text{P2:}\quad & \min_{\xi_i(t),\beta_{ij}(t)}\sum_{i=1}^{N(t)}\frac{1}{N(t)}I\left(t,u_i\right),\\
\text{s.t.}\quad & \text{Equations (6)-(10).}
\end{aligned}$$

For this subproblem, the intuitive solution is to find K MCs out of all MCs, so that if their corresponding information can be updated, the average relative AoCI can be minimized. Consequently, two steps are necessary: first, for each MC, we intend to find its information category that should be updated to minimize the personal AoCI based on its size and the channel capacity; second, we queue MCs in an ascendant order based on their computed minimum relative AoCI. The first K MCs can be selected.

TABLE 1: Main notations

Notation	Description
u_i	Mobile client with identity i
f_j	Information category with identity j
$v(t, f_j)$	The critical level of information category f_j on the server in time slot t
$v(t, u_i, f_j)$	The critical level of information category f_j on MC u_i in time slot t
$h\left(t, u_i, f_j\right)$	The AoI of category f_j on MC u_i at time slot t
$\xi_i\left(t\right)$	It denotes whether MC u_i is selected for update in time slot t
$I\left(t, u_i, f_j\right)$	The AoCI of category f_j on MC u_i at time slot t
$\beta_{ij}\left(t\right)$	It represents whether category f_j on MC u_i is selected for update in time slot t
$N(t)$	Total number of MCs in time slot t
F_i^m	The selected information category group for update by the server
s_m	The size of information m
M	Total number of information categories in the network
S	The maximum size of overall information can be transmitted through one channel
$\hat{S}$	Remaining channel volume for information transmission
s_τ	Real system state in time slot τ
o_τ	Observation in time slot τ
$p(s_{\tau+1}\|s_\tau, a_\tau)$	State transition possibility from state s_τ to $s_{\tau+1}$ by taking action a_τ
$r(s_\tau, a_\tau)$	The received reward by taking action a_τ at state s_τ
$o_{\leq\tau}$	History observations
$a_{<\tau}$	History actions
$p(s_\tau\|o_{\leq\tau}, a_{<\tau})$	The belief state
$b_\tau := \phi(\Psi_\tau)$	The history record pairs of observations and actions
$l(s, \pi)$, $l(b, \pi)$	The loss functions of imitation learning
$\hat{\pi}, \pi^*(s)$	A learned policy
$\pi^*(s)$	An expert policy
$p_{\theta^*}\left(s\right)$	Priority distribution to generate states
$p_{\theta^*}\left(o\|s\right)$	The likelihood to generate observations
$q_\varphi(s\|o, a, r)$	A generation model prepared to train
$p_\theta(s\|o, a, r)$	A real distribution of states
$L(o, a, r, q)$	The loss function of the training model for belief representation
$L(b, \varepsilon)$	The loss of the policy model
$w_\sigma(b_{\tau-1}, a_{\tau-1}, o_{<\tau})$	The output of the representation update model
$\pi_\varepsilon(b)$	The output of the policy model

The first step of the subproblem is to minimize the personal AoCI, i.e.,

$$\text{P3:} \quad I\left(t, u_i\right) = \min_{\beta_{ij}(t)} \sum_{j=1}^{|M|} I\left(t, u_i, f_j\right) \alpha_{ij},$$
$$\text{s.t.} \quad \text{Equations (6)-(10).}$$

The problem in **P3** exhibits the following properties:

Proposition 1 (Optimal substructure of P3)**.** *Let $\beta_i^*\left(t\right) = \left\{\beta_{i1}^*\left(t\right), \beta_{i2}^*\left(t\right), ..., \beta_{iM}^*\left(t\right)\right\}$ be the optimal solution of Problem P3, and $F_i = \left\{f_{i1}\beta_{i1}^*, f_{i2}\beta_{i2}^*, ...f_{iM}\beta_{iM}^*\right\} \setminus \{0\}$ be the selected information categories of MC u_i to update. Without one randomly selected information category $f_{id}, 1 \leq d \leq M$, we define $f_i^{'} = F_i \setminus \{f_{id}\}$, and the minimization problem for $f_i^{'}$ becomes P4:*

$$\text{P4:} \quad I^{'}\left(t, u_i\right) = \min_{\beta_{ij}(t)} \sum_{j=1}^{|M|} I\left(t, u_i, f_j\right) \alpha_{ij}, j \neq d,$$
$$\text{s.t.} \quad \text{Equations (6)-(10).}$$

Then $\beta_i^{'}\left(t\right) = \beta_i^*\left(t\right) \setminus \left\{\beta_{id}^*\left(t\right)\right\}$ is the optimal solution for Problem P4.

The proof can be found in Appendix B of Supplementary File. Based on the proof, we can get the following proposition:

Proposition 2 (Overlapping subproblem of P3)**.** *Problem P3 can be solved recursively by increasing the size of F_i from $F_i^1 = \{f_{i1}\}$, $F_i^2 = \{f_{i1}, f_{i2}\}$..., to $F_i^M = \{f_{i1}, ..., f_{iM}\}$. In each step, a subproblem similar to Problem P4 can be formed, and the optimal solution for each subproblem can be obtained.*

We define the maximum size of overall information can be transmitted through one channel is S, and $I\left(t, u_i, F_i^m|\hat{S}\right), 1 \leq m \leq M, 1 \leq \hat{S} \leq S$, is the relative AoCI based on selected information category F_i^m, when the total update information can be transmitted through one channel in time slot t is $\hat{S}$. Then, the optimal solution for Problem P3 is illustrated in Proposition 3.

Proposition 3 (Optimal solution of P3)**.** *The optimal solution of Problem P3 can be deduced based on the recursive computation:*

$$\min_{\varphi} I\left(t, u_i, F_i^m|\hat{S}\right)$$
$$= \min\left\{I\left(t, u_i, F_i^{m-1}|\hat{S}\right), I\left(t, u_i, F_i^m|\left(\hat{S} - s_m\right)\right)\right\},$$

where $1 \leq m \leq M$.

For information category f_{im}, there are two choices: one is to update, and the other is not to update. For the former, the relative AoCI based on F_i^m (information category f_{im} in this update group) is $I\left(t, u_i, F_i^m|\left(\hat{S} - s_m\right)\right)$, where s_m is the size of update information m, and cannot be divided into smaller ones. For the latter, the relative AoCI based on F_i^m is $I\left(t, u_i, F_i^{m-1}|\hat{S}\right)$. The minimization of AoCI determines whether information category f_{im} should be updated in the current subproblem or not.

For the second step, how to select K MCs to update is based on the results of the first step. We greedily select an MC that can make the AoCI reach the minimum value in each iteration. There are K iterations totally. For each iteration, MC $u_i\left(i \in N(t) \setminus C_{i-1}\right)$ is selected to minimize the total average AoCI, where C_{i-1} is the group of selected MCs during the former $i-1$ iterations. Therefore, the optimal solution of Problem P2 can be obtained as demonstrated in Proposition 4.

Proposition 4 (Optimal solution of P2)**.** *For time slot t, the information selection decision of each MC by the information-aware heuristic algorithm leads to the optimal solution of Problem P2.*

The proof can be found in Appendix C of Supplementary File.

Algorithm 1 Pseudo-code of the information-aware heuristic algorithm

Require: $clients$, $server$, $info$
Ensure: Scheduling results based on **a** and **v**
1: **for** $i < timeslots.size()$ **do**
2: Update the information status on the server
3: Initialize $preUpdateInfo$
4: **for** $j < clients.size()$ **do**
5: Initialize $inter_v$ and $inter_{flag}$ to record the result of each subproblem
6: **for** $l < client[i].interestedInfo.size()$ **do**
7: **for** $h < allowedInfoSize$ **do**
8: **if** $h < client[i].interestedInfo[l]$ **then**
9: $inter_v[l][h] = inter_v[l-1][h]$
10: **else**
11: $u = client[i].Compute(inter_{flag})$
12: **if** $inter_v[l-1][h] > u$ **then**
13: $inter_v[l][h] = u$
14: **end if**
15: $inter_{flag}.update()$
16: **end if**
17: **end for**
18: **end for**
19: $preUpdateInfo[j] = inter_{flag}[l][h]$
20: **end for**
21: **if** $clients.size() < K$ **then**
22: Compute the average AoCI in time slot i based on $preUpdateInfo$
23: $\mathbf{v}.update(preUpdateInfo)$
24: **else**
25: $\mathbf{v} = FindMinKValues(inter_v)$
26: Compute the average AoCI in time slot i based on $preUpdateInfo$
27: **end if**
28: **end for**
29: Compute the average AoCI for all time slots

4.2 Overall Steps

Algorithm 1 presents the processes of the proposed information-aware heuristic algorithm. In each time slot, the server updates its own information, including information sizes, critical levels and contents. Upon the update requirements of MCs, the server computes the AoCI of each MC based on the attributes of its interested information categories, which are revealed to the server when the MC sends an update requirement. A heuristic algorithm is conducted to obtain the personal AoCI based on the channel capacity, as shown from lines 5 to 19. After obtaining all the AoCI of MCs, a greedy algorithm is carried out to select K MCs for information update from lines 21 to 25.

Theorem 1. *The time complexity of the proposed information-aware heuristic algorithm is* $O(N(t)(MS+k))$.

The proof can be found in Appendix D of Supplementary File.

5 An Imitation Learning-based Scheduling Algorithm

Last section presents an information-aware heuristic algorithm to schedule the update requirements of MCs. However, the corresponding implementation is based on the assumption that full knowledge of personal profiles can be acquired. Actually, individuals may be unwilling to reveal their private information to others. When they send update requirements to the server, they may only expose identities of their interested information categories, while keeping local information critical levels and interested ratios unrevealed. This is possible and easy to understand, just like the case that we buy fruits in the store and do not tell the sellers how much we like the fruit and how many are left at home. Though the server can learn the updated result of MCs after their update, the required MCs in the next slot may be not the same with the current slot due to their mobility, resulting in the fact that the server cannot always know the full network state. In this situation, the information-aware heuristic algorithm is not effective. Therefore, novel scheduling algorithms are necessary to meet the requirements of MCs based on partial known user profiles.

In this section, we propose an imitation learning-based scheduling algorithm, named LISA, which is robust to handle personal update requirements under uncertainty environments. We first map the original scheduling problem to a Partially Observable Markov Decision Process (POMDP). Then, we take the advantage of imitation learning to solve the problem based on POMDP. At last, we specify the whole process and provide comprehensive analysis of the designed learning algorithm.

5.1 Problem Transformation

To map the original scheduling algorithm to a POMDP setting, this subsection begins with a brief overview of POMDP, and shows its relationship with our problem. After that, we define a mapping from the scheduling issue to a POMDP.

5.1.1 *POMDP*

A tuple $(\mathbb{S}, \mathbb{A}, \mathbb{R}, \mathbb{O}, P, Q, \mathbb{T})$ can be utilized to represent a discrete-time finite horizon POMDP, where $\mathbb{S}$ is the state space, $\mathbb{A}$ the action space, $\mathbb{R}$ the reward function, $\mathbb{O}$ observations, P the state transition functions, Q conditional observation probabilities and $\mathbb{T}$ the time horizon.

Since the environment cannot be observed directly in a POMDP, real state $s_\tau \in \mathbb{S}$ is hidden at time τ, and only observation $o_\tau \in \mathbb{O}$ can be received. When the agent takes action a_τ, the environment transfers from states s_τ to $s_{\tau+1}$ based on state transition probability $p(s_{\tau+1}|s_\tau, a_\tau)$, and gets observation $o_{\tau+1} \in \mathbb{O}$ based on probability $q(o_{\tau+1}|s_{\tau+1}, a_\tau)$ along with reward $r(s_\tau, a_\tau)$.

Because o_τ cannot reflect the real state of the environment, it is necessary to infer a distribution of the real states based on history observations $o_{\leq\tau}$ and actions $a_{<\tau}$. This inferred state is formally called *belief state*, defined by distribution $p(s_\tau|o_{\leq\tau}, a_{<\tau})$. Let $\Psi_\tau := (o_{\leq\tau}, a_{<\tau})$ denote the history record pairs of observations and actions, and $b_\tau := \phi(\Psi_\tau)$ be a function of Ψ_τ. If we can learn b_τ such that sufficient statistics of posterior distribution over real

states can be estimated, i.e., $p(s_\tau|o_{\leq\tau}, a_{<\tau}) \approx p(s_\tau|b_\tau)$, we can utilize b_τ as a representation of the *belief state* and train the leaning algorithm based on it.

5.1.2 Mapping the original scheduling problem to POMDP

At each time τ, state s_τ should include all knowledge related to the server and MCs' information, while action a_τ should be the selection of MCs and their prepared update information categories. This makes state and action spaces extremely large and consumes much training time, which is unacceptable for online scheduling problems. Fortunately, we can transfer the original Problem P1 into Problems P2 and P3 as described in Section 4.1. In this POMDP setting, Problem P3 is the key component to obtain an efficient solution. Thus, we mainly focus on it and first find the AoCI for each MC based on its best prepared update information categories with blurry observations. Then, based on the result of Problem P3, its solution can be derived as described in Section 4.1.

For Problem P3, let state s_τ be the attributes of its local information, that is $s_\tau = \{\boldsymbol{\alpha}_i, \mathbf{v}(\tau, u_i), \mathbf{v}(\tau), S, \hat{\tau}\}$, where $\boldsymbol{\alpha}_i = \{\alpha_{ij}\}$, $\mathbf{v}(\tau, u_i) = \{v(\tau, u_i, f_j)\}$, and $\mathbf{v}(\tau) = v(\tau, f_j)$, $j \in \{1, M\}$. Action $a_\tau = \left\{\beta_{ij}^p(t)\right\}, j \in \{1, M\}$, denotes the possibilities to select the interested information categories. Since our objective is to minimize the AoCI, the reward function $r(s, a)$ is defined as $r(s_\tau, a_\tau) = -I(\tau, u_i)$. Then, observation o_τ can be parts of state s_τ, e.g., $o_\tau = \{\mathbf{v}(\tau), S, \hat{\tau}\}$, where the interested ratios and local critical levels are unrevealed.

5.2 Information Update Scheduling via Imitation

Imitation learning allows to train policies by imitating expert demonstrations. They are efficient for past operations, but cannot be directly utilized to solve the formulated problem due to long-term costs and complicated implementation. Even so, it is a useful approach for problems with good expert policies. In this subsection, we present how to make the agent imitate expert policies. First, we map s, a, r to b based on the expert demonstration, i.e., learning the belief representation to find their intrinsic relationships. Then, a history about (b_t, a_t) can be formed. After that, we train the learning model offline.

5.2.1 Oracle policy acquisition

To provide excellent demonstrations for the formulated scheduling problem based on POMDP settings, we first collect data based on the information-aware heuristic algorithm as described in Section 4. In other words, we can recruit volunteers or testers that agree to expose their personal interests to servers, and let them involve in the data collection process during a specific time period. This is feasible in reality and easy for implementation. Our expert policy is defined as follows:

Definition 5 (Oracle policy). *An expert policy $\pi^*(s)$ can map state s to action a by solving the optimization problem defined in Subsection 3.3 based on the information-aware heuristic algorithm in Section 5, with the purpose of maximizing cumulative reward R in the MDP setting, i.e.,* $(\mathbb{S}, \mathbb{A}, \mathbb{R}, P, \mathbb{T})$.

Based on the expert policies, we can collect dataset $X = \{s_\tau, a_\tau, o_\tau, r_\tau\}_{\tau=1}^G$, where G is the number of data collection iterations. The items in X can be utilized for training the belief and policy models. However, we cannot directly employ imitation learning for our POMDP-based problem, since true state s cannot be fully observed. Given the distribution of history records, we can define loss function $L(b_\tau, \pi)$ to capture the imitation ability of policy π. Therefore, we intend to find policy $\hat{\pi}$ in each iteration to minimize the expected loss by:

$$\hat{\pi} = arg\min_{\pi\in\Pi} E_{\substack{\Psi\sim p(\Psi|\pi),\\ b=\phi(\Psi)}}[l(b, \pi)]. \tag{11}$$

It is obvious that the agent cannot directly imitate the expert demonstration, since there is a mismatch between s and b, resulting in a big realizability error. Thus, we should first establish a correct relationship between s and b.

5.2.2 Belief representation

State s_τ is generated by priority distribution $p_{\theta^*}(s)$, and observation o_τ is generated by likelihood $p_{\theta^*}(o|s)$, which come from parametric families of distributions $p_\theta(s)$ and $p_\theta(o|s)$, respectively. Unfortunately, parameter θ^* is hidden from our view. Based on dataset X, we train a generation model $q_\varphi(s|o, a, r)$ to approach true posterior density $p_\theta(s|o, a, r)$. Thus, the purpose of the training model is to minimize the gaps between distributions of $q_\varphi(s|o, a, r)$ and $p_\theta(s|o, a, r)$, i.e.,

$$\min_\varphi D_{KL}\left(q_\varphi(s|o, a, r)||p_\theta(s|o, a, r)\right), \tag{12}$$

where:

$$\begin{aligned} &D_{KL}\left(q_\varphi(s|o, a, r)||p_\theta(s|o, a, r)\right) \\ &= -\sum_{\tau=1}^G q_\varphi(s|o, a, r)\ln\frac{p_\theta(s|o, a, r)}{q_\varphi(s|o, a, r)} \\ &= \ln p_\theta(o, a, r) + \sum_{\tau=1}^G q_\varphi(s|o, a, r)\ln q_\varphi(s|o, a, r) \\ &- \sum_{\tau=1}^G q_\varphi(s|o, a, r)\ln p_\theta(s, o, a, r). \end{aligned} \tag{13}$$

We define:

$$\begin{aligned} L(q) = &-\sum_{\tau=1}^G q_\varphi(s|o, a, r)\ln q_\varphi(s|o, a, r) \\ &+\sum_{\tau=1}^G q_\varphi(s|o, a, r)\ln p_\theta(s, o, a, r). \end{aligned} \tag{14}$$

Then, we can obtain:

$$\ln p_\theta(o, a, r) = D_{KL}\left(q_\varphi(s|o, a, r)||p_\theta(s|o, a, r)\right) + L(q). \tag{15}$$

Since $D_{KL}\left(q_\varphi(s|o, a, r)||p_\theta(s|o, a, r)\right)$ is always above 0, $\ln p_\theta(o, a, r) \geq L(q)$ holds, where $L(q)$ is regarded as the evidence lower bound [32]. Minimizing the gaps between $q_\varphi(s|o, a, r)$ and $p_\theta(s|o, a, r)$ equals to maximize $L(q)$. As a result, we define the loss function of the training model for belief representation as:

$$L(o, a, r, q) = E_{q_\varphi(s|o,a,r)}\left(\ln\frac{q_\varphi(s|o, a, r)}{p_\theta(o, a, r|s)p_\theta(s)}\right). \tag{16}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3053136, IEEE Transactions on Mobile Computing

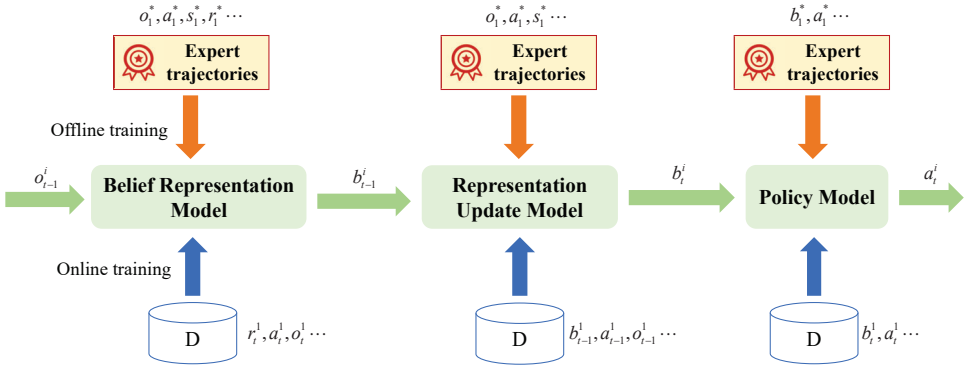


Fig. 5: Overview of the designed learning architecture.

We intend to train the model by picking mini-batches based on $\{o_\tau\}_{\tau=1}^{G}$, $\{a_\tau\}_{\tau=1}^{G}$, $\{r_\tau\}_{\tau=1}^{G}$ and $\{s_\tau\}_{\tau=1}^{G}$. Monte Carlo estimation is utilized to obtain training results [33].

5.2.3 *Representation update*

For online learning, we merely need to input o_τ, a_τ and r_τ to predict belief state b_τ. Regrettably, a_τ and r_τ are not available beforehand, and can be obtained through the policy module. The available information is $b_{<\tau}$, $a_{<\tau}$ and $o_{<\tau}$. Similar to [32], we can train a representation update model that can update b_τ based on $b_{\tau-1}$, $a_{\tau-1}$ and $o_{\tau-1}$. Since training data can be extracted from expert demonstrations, we can also train the representation update model offline beforehand based on equation $b_\tau = w_\sigma(b_{\tau-1}, a_{\tau-1}, o_{<\tau})$. To learn σ, we set the loss function as:

$$L(b, a, o, w) = E(\| w_\sigma(b, a, o) - b \|^2), \tag{17}$$

where $E(\cdot)$ represents the expected average value of its inputs.

5.2.4 *Oracle policy imitation*

After training the model for belief representation, we can train the policy model offline. Based on belief representation model q_φ and belief update model w_σ, we can get belief state b. Each time, one pair of o_τ and a_τ is inputted into the model, and the output is $b_{\tau+1}$. Gradually, we can get $\{b_{\tau^*}\}_{\tau=1}^{G}$ for expert policies. Then, we train the policy model based on the mini-batches from $\{b_\tau\}_{\tau=1}^{G}$ and $\{a_\tau\}_{\tau=1}^{G}$. The loss of the policy model is:

$$L(b, \varepsilon) = E(\| \pi_\varepsilon(b) - a \|^2). \tag{18}$$

For online learning, in time slot t, we treat each MC separately. States and actions for one MC are independent from others. For each MC, we provide the following definition:

Definition 6 (Single POMDP). *For MC u_i in time slot t, its true state is s_t^i and observation is o_t^i. Then, we can form a single POMDP problem based on tuple $(\mathbb{S}^i, \mathbb{A}^i, \mathbb{R}^i, \mathbb{O}^i, P^i, Q^i, \mathbb{T})$.*

To solve this single POMDP, we input the belief model o_{t-1}^i, a_{t-1}^i and r_{t-1}^i, and its output is $b_{t-1}^i = q_\varphi(o_{t-1}^i, a_{t-1}^i, r_{t-1}^i)$. Then, we predict b_t^i by representation update model $b_t^i = w_\sigma(b_{t-1}^i, a_{t-1}^i, o_t^i)$. Until now, the POMDP problem has been transformed into an MDP problem. We can utilize the trained policy model to get action $a_t^i = \pi_\varepsilon(b_t^i)$. The action here refers to the update possibilities of interested information categories of MC i.

5.3 Imitation Learning-based Scheduling

As shown in Fig. 5, there are mainly three modules in our learning algorithm, i.e., belief representation module q_φ, representation update module w_σ and policy module π_ε. Since expert demonstrations can be obtained based on the information-aware heuristic algorithm described in Section 4, these three modules can be treated by minimizing the following optimization problems:

$$\begin{aligned} q = arg\min_{\varphi} E_{\substack{s \sim p_\theta(s), \\ o \sim p_\theta(o|s,a,r)}} (\ln p_\theta(s|o,a,r)) \\ - D_{KL}(q_\varphi(o|s,a,r)||p_\theta(s)). \end{aligned} \tag{19}$$

$$w = arg\min_{\sigma} E_{\substack{o \sim p_\theta(o|s,a,r), \\ b \sim d_{q_\varphi(b|o,a,r)}}} (\| w_\sigma(b,a,o) - b \|^2). \tag{20}$$

$$\pi = arg\min_{\varepsilon} E_{b \sim d_{w_\sigma(q_\varphi(b|o,a,r))}} (\| \pi_\varepsilon(b) - a \|^2). \tag{21}$$

In each time slot t, there are total $N(t)$ POMDP problems, each of which is related to one MC with the purpose of finding the scheduling sequences of local interested information categories. To solve each POMDP problem, we transfer it into MDP based on belief representation module q_φ and representation update module w_σ to get $b_t = w_\sigma(q_\varphi(o_{t-1}, a_{t-1}, r_{t-1}), a_{t-1}, o_t)$. Then policy model π_ε can be utilized to get prediction a_t, and the corresponding pseudo-codes can be found from lines 8 to 18 in Algorithm 2.

Following that, we have the update possibilities for each interested information categories of MC u_i. We can greedily select the information categories with big update possibilities from the residential groups as candidate update information categories. However, their total sizes cannot exceed the channel capacity. The top K MCs with the minimum AoCI based on candidate update information categories can be obtained. Thus, we can get the update list. The whole process is illustrated in Algorithm 2.

5.4 Theoretical Analysis

In this subsection, we provide a comprehensive theoretical analysis for the designed imitation learning-based scheduling algorithm. The overall loss of our learning algorithm is:

$$L = \lambda_1 L(o, a, r, q) + \lambda_2 L(b, a, o, w) + L(b, \pi), \tag{22}$$

where parameters λ_1 and λ_2 are utilized to control the weighting factors among the three losses.

Algorithm 2 Pseudo-code of the imitation learning-based scheduling algorithm

Require: $clients$, $server$, $info$
Ensure: The updated scheduling result
$X \leftarrow GetExpertTrajectries()$ /*Utilize Algorithm 1
Train models offline
for $i < timeslots.size()$ **do**
  if $i\%\Delta == 0$ **then**
    Train models online/*Utilize training methods specified in Section 5.4
  end if
  Update the information status on the server
  Initialize $preUpdateInfo$
  for $j < clients.size()$ **do**
    Interact with client j to get o_t^j
    if $client.idFirstEnter$ **then**
      Random take an action a_t^j
    else
      $b_{t-1}^j = GetEstimation(o_{t-1}, a_{t-1}, r_{t-1})$
      $b_t^j = GetReEstimation(b_{t-1}, a_{t-1}, r_{t-1})$
      $a_t^t = getPredicted(b_t^j)$
      $preUpdateInfo.Add(a_t^j)$
    end if
  end for
  if $clients.size() < k$ **then**
    Compute average AoCI in time slot i based on $preUpdateInfo$
    $\mathbf{v}.update(preUpdateInfo)$
  else
    $\mathbf{v} = FindMinkValues(inter_v)$
    Compute average AoCI in time slot i based on $preUpdateInfo$
  end if
end for
Compute average AoCI for all time slots

For imitation learning, two kinds of algorithms can be applied here. One is the supervised approach [34], and the other is DAGGER [31]. For the supervised approach, it first collects expert trajectories in an offline manner and then trains the learning model. Since it cannot obtain the expert's online direct guidance, online expert trajectories cannot be obtained, and we can only utilize the agent trajectories to continue training the online learning model. Different from the supervised approach, DAGGER allows an existing expert to direct the behaviors of the agent. For online learning, the expert behavior trajectories can be collected by a few rounds at the beginning. Based on the whole collected data set, the model can be trained further, narrowing down the gaps between the behaviors of the expert and the agent. This is possible in reality, since after the agent making decisions by state transformation, the corresponding states can be recorded and the expert can select actions for each collected state triggered by the agent. The pseudo-codes of the supervised approach and DAGGER are listed in Appendix G of Supplementary File.

For simplicity and without loss of generality, we utilize policy η to represent trained policy $\pi_\varepsilon(w_\sigma(q_\varphi(\cdot),\cdot),\cdot)$. Formally, we can regard that the overall loss defined in equation (22) is a $0-1$ loss. For the supervised approach, we train these three models every Δ time slots. The expected average AoCI $J(\eta)$ by conducting policy η can be bounded by the following theorem.

Theorem 2 (Upper bound of the supervised approach)**.** *Let η be the policy carried by the learning agent for T steps, and $J(\eta) < J(\eta^*) + T^2 e/\Delta^2$ holds, where $e < 1.2e_1 + 2e_2 + e_3$. Variables e_1, e_2 and e_3 are the probabilities that models π_ε, w_σ and q_φ make one mistake under state distribution d_η, respectively.*

The proof can befound in Appendix E of Supplementary File.

For DAGGER, it adopts expert policy η^* with possibility γ_i in each step i by enabling $\eta_i = \gamma_i\eta^* + (1-\gamma_i)\hat{\eta}_i$. Let l_σ^{max} and l_ε^{max} be the upper bounds of losses, i.e., $L(b,\pi) \le l_\varepsilon^{max}$ and $L(b,a,o,w) \le l_\sigma^{max}$, respectively. Then, let $\epsilon_\varepsilon = \min_\pi \Delta/T \sum_{i=1}^{T/\Delta} E_{\substack{s\sim d_{\eta_i},\\ b\sim \mathfrak{d}_{\varphi_i,\sigma_i}}} [L(b,\pi)]$, and $\epsilon_\sigma = \min_w \Delta/T \sum_{i=1}^{T/\Delta} E_{\substack{s\sim d_{\eta_i},\\ b\sim \mathfrak{d}_{\varphi_i,\sigma_i},\\ o\sim \Pi_{\eta_i}}} [L(b,a,o,w)]$ be the training losses of the best policy, respectively. We can obtain the following lemma for the representation update and policy modules:

Lemma 1. *For the representation update module, the average loss $L(\hat{w})$ should satisfy:*

$$\begin{aligned} L(\hat{w}) &= E_{\substack{s\sim d_{\hat{\eta}_i},\\ b\sim \mathfrak{d}_{\hat{\varphi}_i,\hat{\sigma}_i},\\ o\sim \Pi_{\hat{\eta}_i}}} [L(b,a,o,\hat{w})] \\ &\le E_{\substack{s\sim d_{\eta_i},\\ b\sim \mathfrak{d}_{\varphi_i,\sigma_i},\\ o\sim \Pi_{\eta_i}}} [L(b,a,o,\hat{w})] + 2l_\sigma^{max}\min\left(1,\Delta\gamma_i\right), \end{aligned} \tag{23}$$

and the average loss $L(\hat{\pi})$ for the policy module should satisfy:

$$\begin{aligned} L(\hat{\pi}) &= E_{\substack{s\sim d_{\hat{\eta}_i},\\ b\sim \mathfrak{d}_{\hat{\varphi}_i,\hat{\sigma}_i}}} [L(b,\hat{\pi})] \\ &\le E_{\substack{s\sim d_{\eta_i},\\ b\sim \mathfrak{d}_{\varphi_i,\sigma_i}}} [L(b,\hat{\pi})] + 2l_\varepsilon^{max}\min\left(1,\Delta\gamma_i\right). \end{aligned} \tag{24}$$

The proof can be found in lemma 4.1 in reference [31]. For the belief representation module, we can obtain:

Lemma 2. *The upper bounds of the loss for belief representation model can be roughly obtained by:*

$$l_\varphi^{max} < 1 - \ln p_\theta(o,a,r), \tag{25}$$

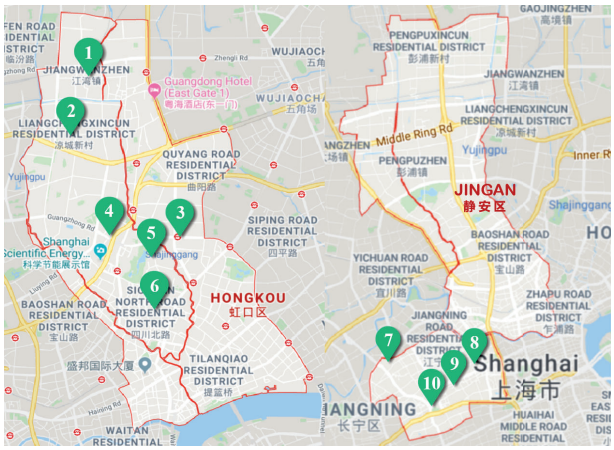
and the lower bound is:

$$\begin{aligned} \epsilon_\varphi &= \min_q \Delta/T \sum_{i=1}^{T/\Delta} E_{\substack{s\sim d_{\eta_i},\\ o\sim \Pi_{\eta_i}}} [L(o,a,r,q)] \\ &\ge l_\varphi^{min}, \end{aligned} \tag{26}$$

where l_φ^{min} can be obtained by the explanation in Section 3 in [33]. The average loss $L(\hat{q})$ should satisfy:

$$\begin{aligned} L(\hat{q}) &= E_{\substack{s\sim d_{\hat{\eta}_i},\\ o\sim \Pi_{\hat{\eta}_i}}} [L(o,a,r,\hat{q})] \\ &\le E_{\substack{s\sim d_{\eta_i},\\ o\sim \Pi_{\eta_i}}} [L(o,a,r,\hat{q})] + 2l_\varphi^{max}\min\left(1,\Delta\gamma_i\right). \end{aligned} \tag{27}$$

Based on the above lemma, we can obtain the following theorem:



Hongkou district		Jingan district	
ID	Locations	ID	Locations
1	31.304725,121.473037	7	31.228587,121.455745
2	31.294622,121.470535	8	31.230982,121.456374
3	31.276524,121.491138	9	31.234170,121.431700
4	31.276615,121.477732	10	31.223078,121.446153
5	31.273372,121.487506		
6	31.263758,121.486227		

Fig. 6: Selected GPS locations in Shanghai.

Theorem 3 (Upper bound of DAGGER). *For DAGGER in our learning model, with probability at least* $1-\delta$*, policy* $\hat{\pi} \in \hat{\pi}_{1:T/\Delta}$ *exists, and should satisfy:*

$$E_{\substack{s\sim d_{\eta_i},\\ o\sim \Pi_{\eta_i}}}[L] < \rho_{T/\Delta} + \epsilon_\varepsilon + \epsilon_\sigma + l_\varphi^{min} + 2\left(\lambda_1 + \lambda_2 l_\sigma^{max} + l_\varepsilon^{max}\right)\Delta\gamma_i + \left(\lambda_1 + \lambda_2 l_\sigma^{max} + l_\varepsilon^{max}\right)\sqrt{\frac{2\Delta\log(1/\delta)}{T\mathbb{K}}}, \tag{28}$$

where $\rho_{T/\Delta}$ *is the average regret of* $\eta_{1:T/\Delta}$.

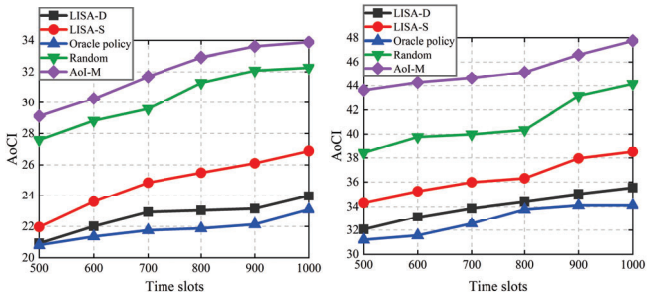
The proof can be found in Appendix F of Supplementary File.

6 Performance Evaluation

In order to validate the performance of our proposed algorithm LISA, we conduct extensive simulations based on real-world taxi traces in Shanghai (China) with the support of tensorflow.

6.1 Simulation Setup

A data set of real-world taxi traces in Shanghai collected from April 1, 2015 to April 30, 2015, including the recorded information of more than 1000 taxies, is leveraged to demonstrate the feasibility of our solution. According to administrative divisions [35], Shanghai can be divided into seven regions. We select two districts as examples, i.e., Hongkou and Jingan, and deploy servers in several locations as illustrated in Fig. 6. We set the wireless communication range of a server by 200 m, and schedule the requirements of passing-by MCs to compute their average AoCI in the communication range of servers. We set the number of information categories between 1 and 10, and the total number of critical levels is 5. For different MCs, we randomly set their interest



(a) Jingan district (b) Hongkou district

Fig. 7: Performance of average AoCI with different number of time slots.

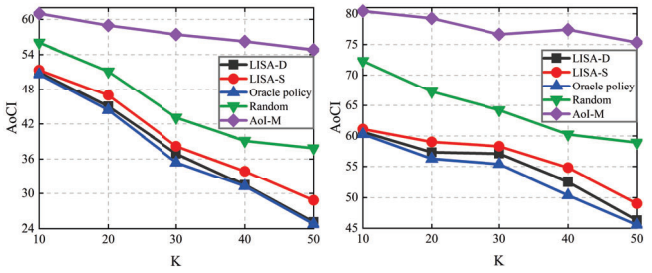
information categories when they move in the coverage of the server. For the information managed by the server, its level changes are also set randomly at the beginning of each time slot. The information update size is randomly distributed between 1 and 5 MB, and the transmission power between MCs and servers is 10 dBm with noise power 172 dBm [36]. For expert policies, we collect data in 200 time slots to train offline models, leveraging multi-layer perceptrons. For the belief representation module, we define it has 4 convolutional layers and 2 fully connected layers with ReLU non-linearities. The belief update module and the policy module both have 4 fully connected layers. We use Adam optimizer [37] to train the three modules.

Five representative algorithms are compared:

- LISA-S: We utilize supervised learning for LISA, i.e., models are trained based on the data collected by expert policies offline. For online training, the data can only be collected by the agent policy.
- LISA-D: We leverage DAGGER algorithm [31] to train models online. The offline training process is the same with that of LISA-S. For online training, each 200 time slots, we collect data based on expert policies within the first 50 time slots. Then, models based on expert trajectories can be trained further to guide the agent behaviors.
- Oracle policy: It refers to the expert policy, i.e., the utilized information-aware heuristic algorithm designed in Section 4.
- Random scheme: Similar to the expert policy, it selects K MCs with the minimum AoCI. However, for each MC, it randomly selects the updated information categories satisfying the channel capacity.
- AoI-M [4]: It is a traditional information update scheduling algorithm, aiming at minimizing the average AoI of users on base stations by considering varying sample sizes, multiple data transmission units, as well as general and heterogeneous sampling behaviors among source nodes.

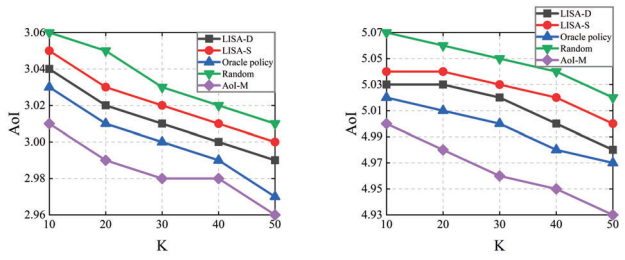
6.2 Simulation Results

Fig. 7 illustrates the performance of average AoCI with different number of time slots. From Fig. 7(a), we can observe that the performance of the expert policy is the best, while that of LISA-D is the closest to it. This is because



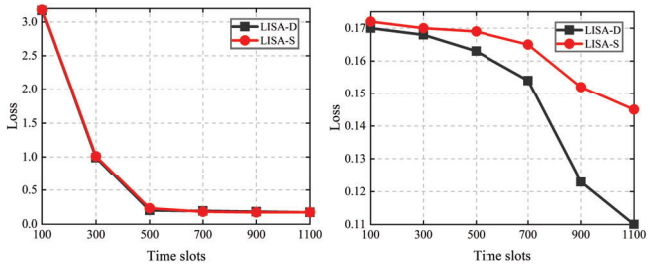
(a) Jingan district (b) Hongkou district

Fig. 8: Performance of average AoCI with different values of K.



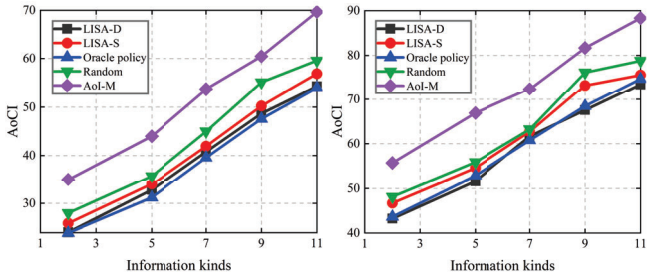
(a) Jingan district (b) Hongkou district

Fig. 9: Performance of average AoI with different K.



(a) Offline training (b) Online training

Fig. 10: Performance of losses for offline training and online training.



(a) Jingan district (b) Hongkou district

Fig. 11: Performance of average AoCI with different number of information categories.

the expert policy can obtain the optimal solution based on the awareness of MC profiles. Besides offline training, LISA-D also collects expert trajectories online to train the learning model, which can revise the potential errors from a long-term perspective. However, LISA-S cannot collect online expert trajectories but reduces imitation learning as supervise learning. If a new state encountered by the agent that the offline expert never met, a decision error may occur and affect the further decision. For random and AoI-M algorithms, they cannot find optimal solutions but randomly select the updated information categories with the purpose of minimizing the AoCI and AoI, respectively. Similar results can be found in Fig. 7(b). Since there are more MCs in the dataset of Hongkou district, their information cannot get updated simultaneously in one time slot. Thus, the average AoCI of Hongkou district is bigger than that of Jingan district.

The performance of average AoCI with different values of K is illustrated in Fig. 8, where K is the maximum number of MCs that are allowed to update information categories simultaneously in one time slot. When the value of K becomes big, more MCs can communicate with the server simultaneously. Thereby, more information categories can be updated. The average AoCIs become small for all the five algorithms when the value of K increases. The performance of AoI-M is the worst, since it only focuses on minimizing the average AoI while neglecting the average AoCI. Similar results can be found in Fig. 8(b) for Hongkou district.

Besides the performance of average AoCI, we also measure the performance of average AoI for the five algorithms as shown in Fig. 9. From Fig. 9(a), we can observe that the average AoI achieved by the designed LISA-D and LISA-S algorithms are not much worse than that of AoI-M, and their values are all around 3. This is because, though our designed algorithm does not directly minimize the average AoI, we can achieve that purpose through information update scheduling by minimizing the average AoCI, which has a tight relationship with AoI. In addition, as the value of K becomes big, the AoI performance trends of the five algorithms also drop. The reason is similar with that of Fig. 8. In Fig. 9(b), AoI values of the five algorithms in Hongkou district are bigger since more MCs require information update.

The loss trends of our learning models based on the dataset of Jingan district are illustrated in Fig. 10. All the three modules in our designed algorithm can be trained together both online and offline, whose total loss is defined in equation (22). For offline training loss in Fig. 10(a), we can notice that the loss trend of LISA-D overlaps with that of LISA-S, because the two offline training processes are the same, i.e., training the model by expert behavior trajectories collected by the information-aware heuristic algorithm. However, their online training processes are different, resulting in the loss gaps between LISA-D and LISA-S. LISA-D collects online expert behavior trajectories to further train the models. However, LISA-S cannot collect the expert behavior trajectories and only the agent's trajectories are available, leading to imperfect training results by introducing errors from expert behaviors.

The influence of the number of information categories on the average AoCI is shown in Fig. 11. When there are more kinds of information, MCs are likely to have more interested information categories. In other words, one MC can have more local information categories. Then, the number of

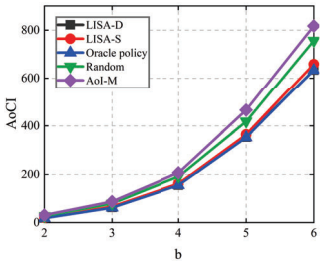


Fig. 12: Performance of average AoCI with different values of b.

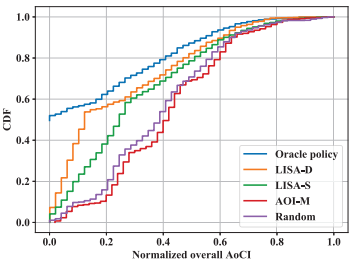


Fig. 13: CDF curves.

information categories that need to update becomes big, resulting in bigger average AoCI. The performance of LISA-D is also the closest to that of the expert policy, while that of LISA-S is better than those of random and AoI-M algorithms, since imitation learning is leveraged in LISA-S and LISA-D to imitate the expert behaviors.

Fig. 12 shows the impacts of different values of b on the average AoCIs with the dataset of Jingan district, where b reflects the importance of critical levels for the computation of AoCI. As the value of b increases, the average AoCI becomes high. This is because b affects the computation of personal AoCI and becomes more important when its value grows. In addition, the gaps of the AoCI of LISA-D, LISA-S, expert policy, random scheme and AoI-M become big when the value of b increases. This is because the level differences between local MCs and the server become more important for the computation of AoCI. LISA-D, LISA-S and expert policy can always update information by considering level changes. As a result, these three algorithms have relative lower AoCI than the other two algorithms.

The Cumulative Distribution Function (CDF) curves of the five algorithms are shown in Fig. 13. The horizontal axis is based on the normalized AoCI of MCs. We observe that the curve of expert policy is the highest, since the AoCI of MCs are mainly between 0 and 0.6. However, the values of AoI-M and random scheme are mainly centralized between 0.4 and 0.8. The performance of LISA-D and LISA-S is better than that of AoI-M and random scheme while worse than that of the expert policy. This is because the expert policy can always find the best scheduling method, and schedule MCs that can make the average AoCI have a small value first. AoI-M only considers AoI, while Random scheme makes scheduling decision randomly based on the value of personal AoCI. LISA-D and LISA-S can imitate the policy of experts effectively.

7 Conclusion

In this paper, we have investigated imitation learning with information update scheduling to minimize the average AoCI, by considering the importance of personal information under partial observations. Specifically, we first established the system model and formulated the scheduling issue as an optimization problem. Then, we proposed an offline scheduling algorithm, i.e., an information-aware heuristic algorithm that can obtain the optimal scheduling result. For online scheduling based on partial observations, we designed an imitation learning-based scheduling algorithm to guide the learning agent to mimic the behaviors of experts. We first transferred the scheduling problem in the POMDP setting to an MDP by belief representation and representation update modules, and then selected actions based on the policy module. We provided theoretical analysis for the designed learning algorithm to derive its upper bound. Experimental results showed that our designed algorithm has advantages on average AoCI and CDF based on different network parameters compared with other representative algorithms.

References

[1] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *IEEE International Conference on Sensing, Communication and Networking*, pp. 350–358, 2011.

[2] Q. He, D. Yuan, and A. Ephremides, "Optimal link scheduling for age minimization in wireless systems," *IEEE Transactions on Information Theory*, vol. 64, no. 7, pp. 5381–5394, 2017.

[3] L. Corneo, C. Rohner, and P. Gunningberg, "Age of information-aware scheduling for timely and scalable Internet of things applications," in *IEEE Conference on Computer Communications*, pp. 2476–2484, 2019.

[4] C. Li, S. Li, and Y. T. Hou, "A general model for minimizing age of information at network edge," in *IEEE Conference on Computer Communications*, pp. 118–126, 2019.

[5] T.-W. Kuo, "Minimum age TDMA scheduling," in *IEEE Conference on Computer Communications*, pp. 2296–2304, 2019.

[6] I. Kadota, A. Sinha, and E. Modiano, "Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1359–1372, 2019.

[7] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 193–204, 2017.

[8] C. Yi and J. Cai, "Transmission management of delay-sensitive medical packets in beyond wireless body area networks: A queueing game approach," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2209–2222, 2018.

[9] X. Wang, Z. Ning, M. Zhou, X. Hu, L. Wang, Y. Zhang, F. R. Yu, and B. Hu, "Privacy-preserving content dissemination for vehicular social networks: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1314–1345, 2018.

[10] Y. Sun, I. Kadota, R. Talak, and E. Modiano, "Age of information: A new metric for information freshness," *Synthesis Lectures on Communication Networks*, vol. 12, no. 2, pp. 1–224, 2019.

[11] R. Talak, S. Karaman, and E. Modiano, "Distributed scheduling algorithms for optimizing information freshness in wireless networks," in *IEEE Workshop on Signal Processing Advances in Wireless Communications*, pp. 1–5, 2018.

[12] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *Annual Allerton Conference on Communication, Control, and Computing*, pp. 486–493, 2017.

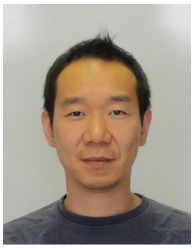
[13] R. Talak, S. Karaman, and E. Modiano, "Optimizing information freshness in wireless networks under general interference constraints," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 15–28, 2019.

[14] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of

information in broadcast wireless networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2637–2650, 2018.
[15] J. Zhong, W. Zhang, R. D. Yates, A. Garnaev, and Y. Zhang, "Age-aware scheduling for asynchronous arriving jobs in edge applications," in *IEEE Conference on Computer Communications Workshops*, pp. 674–679, 2019.
[16] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *IEEE Conference on Computer Communications*, pp. 2731–2735, 2012.
[17] R. Yates, Z. Jing, and Z. Wuyang, "Updates with multiple service classes," in *IEEE International Symposium on Information Theory*, pp. 2731–2735, 2012.
[18] R. D. Yates, "Age of information in a network of preemptive servers," in *IEEE Conference on Computer Communications Workshops*, pp. 118–123, 2018.
[19] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2215–2228, 2020.
[20] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys*, vol. 50, no. 2, p. 21, 2017.
[21] Z. Ning, P. Dong, X. Wang, L. Guo, J. J. Rodrigues, X. Kong, J. Huang, and R. Y. Kwok, "Deep reinforcement learning for intelligent Internet of vehicles: An energy-efficient computational offloading scheme," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1060–1072, 2019.
[22] N. T. Fitter, R. Funke, J. C. Pulido, L. E. Eisenman, W. Deng, M. R. Rosales, N. S. Bradley, B. Sargent, B. A. Smith, and M. J. Mataric, "Using a socially assistive humanoid robot to encourage infant leg motion training," *IEEE Robotics and Automation Magazine*, vol. 26, no. 2, pp. 12–23, 2019.
[23] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *IEEE International Conference on Robotics and Automation*, pp. 1–9, 2018.
[24] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2020.3012509*, 2020.
[25] S. Choudhury, A. Kapoor, G. Ranade, S. Scherer, and D. Dey, "Adaptive information gathering via imitation learning," in *Robotics: Science and Systems Conference*, pp. 41–50, 2017.
[26] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "LORM: Learning to optimize for resource allocation in wireless networks with few training samples," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 665–679, 2020.
[27] M. Lee, G. Yu, and G. Y. Li, "Learning to branch: Accelerating resource allocation in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 958–970, 2020.
[28] X. Wang, Z. Ning, and L. Wang, "Offloading in Internet of vehicles: A fog-enabled real-time traffic management system," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4568–4578, 2018.
[29] Z. Ning, P. Dong, X. Wang, X. Hu, J. Liu, L. Guo, B. Hu, R. Kwok, and V. C. Leung, "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Transactions on Mobile Computing, DOI: 10.1109/TMC.2020.3025116*, 2020.
[30] J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, B. Hu, and V. C. Leung, "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4283–4294, 2018.
[31] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International conference on artificial intelligence and statistics*, pp. 627–635, 2011.
[32] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson, "Deep variational reinforcement learning for POMDPs," in *International Conference on Machine Learning*, pp. 1–17, 2018.
[33] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *stat*, vol. 1050, p. 1, 2014.
[34] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *International conference on artificial intelligence and statistics*, pp. 661–668, 2010.
[35] "Administrative divisions of shanghai." https://en.wikipedia.org/wiki/List_of_administrative_divisions_of_Shanghai.
[36] L. Chen and J. Xu, "Task replication for vehicular cloud: Contextual combinatorial bandit with delayed feedback," in *IEEE Conference on Computer Communications*, pp. 748–756, 2019.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, pp. 1–11, 2015.

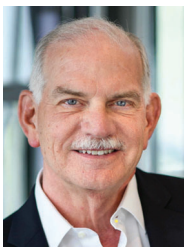
Xiaojie Wang (M'19) received the M.S. degree from Northeastern University, China, in 2011. From 2011 to 2015, she was a software engineer in NeuSoft Corporation, China. She received the PhD degree from Dalian University of Technology, Dalian, China, in 2019. Currently, she is a postdoctor in the Hong Kong Polytechnic University. Her research interests are wireless networks, mobile edge computing and machine learning.

Zhaolong Ning (M'14-SM'18) received the Ph. D. degree from Northeastern University, China in 2014. He was a Research Fellow at Kyushu University from 2013 to 2014, Japan. Currently, he is an associate professor in Dalian University of Technology and a research fellow in The University of Hong Kong. His research interests include Internet of things, mobile edge computing, deep learning, and resource management. He has published over 120 scientific papers in international journals and conferences. Dr. Ning serves as an associate editor or guest editor of several journals, such as IEEE Transactions on Industrial Informatics, IEEE Transactions on Social Computational Systems, The Computer Journal and so on. He is elected to be the Young Elite Scientists Sponsorship Program by CAST.



Song Guo (M'02-SM'11-F'20) received the PhD degree in computer science from the University of Ottawa and was a professor with the University of Aizu. He is a full professor with the Department of Computing, The Hong Kong Polytechnic University. His research interests include big data, cloud computing and networking, and distributed systems with more than 400 papers published in major conferences and journals. His work was recognized by the 2016 Annual Best of Computing: Notable Books and Articles in Computing in ACM Computing Reviews. He is the recipient of the 2017 IEEE Systems Journal Annual Best Paper Award and other five Best Paper Awards from IEEE/ACM conferences. He was an associate editor of the IEEE Transactions on Parallel and Distributed Systems and an IEEE ComSoc distinguished lecturer. He is now on the editorial board of the IEEE Transactions on Emerging Topics in Computing, the IEEE Transactions on Sustainable Computing, the IEEE Transactions on Green Communications and Networking, and the IEEE Communications. He also served as general, TPC and symposium chair for numerous IEEE conferences. He currently serves as an officer for several IEEE ComSoc Technical Committees and a director in the ComSoc Board of Governors. He is a fellow of the IEEE.

Miaowen Wen (SM'18) received the Ph.D. degree from Peking University, Beijing, China, in 2014. From 2012 to 2013, he was a Visiting Student Research Collaborator with Princeton University, Princeton, NJ, USA. He is currently an Associate Professor with South China University of Technology, Guangzhou, China, and a Hong Kong Scholar with The University of Hong Kong, Hong Kong. He has published a Springer book entitled Index Modulation for 5G Wireless Communications and more than 80 journal papers. His research interests include a variety of topics in the areas of wireless and molecular communications. Dr. Wen was the recipient of four Best Paper Awards. Currently, he is serving as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE COMMUNICATIONS LETTERS, and the Physical Communication (Elsevier), and a Guest Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING.



H. Vincent Poor (M'77-SM'82-F'87) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at UrbanaChampaign. Since 1990 he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor of Electrical Engineering. During 2006 to 2016, he served as Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, signal processing and machine learning, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the recent book Multiple Access Techniques for 5G Wireless Networks and Beyond. (Springer, 2019).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. honoris causa from the University of Waterloo awarded in 2019.