Intelligent and Converged Networks
2020, 1(3): 234–242

ISSN 2708-6240
DOI: 10.23919/ICN.2020.0015

Deep reinforcement learning based worker selection for distributed machine learning enhanced edge intelligence in internet of vehicles

Junyu Dong, Wenjun Wu*, Yang Gao, Xiaoxi Wang, and Pengbo Si

Abstract: Nowadays, Edge Information System (EIS) has received a lot of attentions. In EIS, Distributed Machine Learning (DML), which requires fewer computing resources, can implement many artificial intelligent applications efficiently. However, due to the dynamical network topology and the fluctuating transmission quality at the edge, work node selection affects the performance of DML a lot. In this paper, we focus on the Internet of Vehicles (IoV), one of the typical scenarios of EIS, and consider the DML-based High Definition (HD) mapping and intelligent driving decision model as the example. The worker selection problem is modeled as a Markov Decision Process (MDP), maximizing the DML model aggregate performance related to the timeliness of the local model, the transmission quality of model parameters uploading, and the effective sensing area of the worker. A Deep Reinforcement Learning (DRL) based solution is proposed, called the Worker Selection based on Policy Gradient (PG-WS) algorithm. The policy mapping from the system state to the worker selection action is represented by a deep neural network. The episodic simulations are built and the REINFORCE algorithm with baseline is used to train the policy network. Results show that the proposed PG-WS algorithm outperforms other comparation methods.

Key words: edge information system; internet of vehicles; distributed machine learning; deep reinforcement learning; worker selection

1 Introduction

In recent years, the Edge Information System (EIS) has received a lot of attentions for its powerful capabilities that integrate edge caching, edge computing, and edge Artificial Intelligence (AI)[1]. As a typical scenario of edge intelligent services, the Internet of Vehicles (IoV) can provide reliable internet services through Vehicle-to-Everything (V2X) communication in the global network of vehicles[2]. With the assistance of EIS, intelligent vehicles will have more sensitive perceptions and lower latency communications to complete vehicular tasks and improve traffic efficiency than traditional IoV.

As a foundation of the autonomous driving, the High Definition (HD) mapping is essential for the intelligent IoV. The HD mapping models the surface of the roads, which integrates the important elements of the roads, such as the slope, curvature, lane-making types as well as dynamic obstacles. And the process of generating HD map involves three phases: data acquisition, data accumulation, and data confirmation. Great efforts have been made for HD mapping in the industry[3,4]. However, there are still some difficulties in the implementation of this application. For example, most of the driving decisions need to be made in time[4], and HD map is dynamic which needs to be updated in real time[5]. Due to bandwidth, storage, and privacy issues in IoV, it is often impractical to transmit all the data to a centralized facility to process.

Distributed Machine Learning (DML) which has

• Junyu Dong, Wenjun Wu, Yang Gao, Xiaoxi Wang, and Pengbo Si are with the Faculty of Information Technology, Beijing University of Technology, Beijing 100022, China. E-mail: wenjunwu@bjut.edu.cn.

∗ To whom correspondence should be addressed.

Manuscript received: 2020-07-31; revised: 2020-09-28; accepted: 2020-11-11

received a lot of research attentions recently is a promising solution for the HD mapping. With the support of the DML framework, the massive original data do not need to be transmitted to a centralized location[6] (such as the Cloud), and the bandwidth is saved. The raw map data can be processed, the HD mapping and intelligent driving model can be trained on multiple local servers (such as the intelligent vehicles), and the update of the model can be done timely. The existing research of DML includes aggregation method[7], the trade-off between local update and global parameter aggregation[8], etc.

In fact, there are still some issues related to HD mapping and intelligent driving decision model leveraging DML. Due to the various latencies caused by the fluctuation of the wireless communication environment, the timeliness of the local model and parameters of each intelligent vehicle might be different. And the asynchronous stochastic gradient descent based aggregation method is more practical than the synchronous Stochastic Gradient Descent (SGD) based method. Although asynchronous SGD-based method can converge faster, the final convergence performance may be poor as some of the outdated gradients are collected[7]. Therefore, how to select suitable vehicles for parameter aggregation in HD mapping becomes a primary consideration.

Generally, the above-mentioned vehicles scheduling problem can be considered as a resource allocation problem in EIS, which has aroused a lot of attentions in academia. Using the Deep Reinforcement Learning (DRL) method, the smart contract execution nodes selection in blockchain scenario is addressed[9], the task scheduling problem maximizing the long-term revenue of the edge server in the mobile blockchain for internet of things is solved[10], and the joint optimization framework about caching, computation, and security for delay-tolerant data is considered[11]. But most of the existing researches did not pay attention to the characteristics of DML-based HD mapping, such as the timeliness of the local model and parameters and the effective sensing area of each vehicle.

In this paper, the intelligent IoV with the support of DML is studied, and the DML-based HD mapping and intelligent driving decision model training are considered as the example. A Road Side Unit (RSU) with edge intelligent functions is deployed as the aggregator to interact with intelligent vehicles in its coverage. All the intelligent vehicles are workers within the DML framework. The worker selection problem is modeled as a Markov Decision Process (MDP), in which the aggregation performance related factors are used to design the reward. A DRL-based solution is proposed, called the Worker Selection based on Policy Gradient (PG-WS) algorithm. The episodic simulations are built and the REINFORCE algorithm with baseline is used to train the policy network. Testing results confirm the effectiveness of the proposed PG-WS algorithm. The main contributions of this paper are summarized as follows:

- We focus on a typical scenario of edge intelligent services, the intelligent IoV, and design a DML framework based on the interaction between RSU and the intelligent vehicles to enable the applications, such as HD mapping and intelligent driving decision model training.
- The characteristics and requirements of the typical applications which include HD mapping and intelligent driving decision model training with the support of DML in IoV are fully considered. The timeliness of the local model, the transmission quality of model parameters uploading, and the effective sensing area of the worker, which are closely related to the aggregation performance, are taken into account in the worker selection problem.

The rest of this paper is organized as follows. The system model is presented in Section 2. The MDP of the worker selection problem is modeled in Section 3. Then the DRL-based solution method and the episodic simulation are given in Section 4. Important results of the experiment are presented in Section 5. Finally, Section 6 concludes this paper.

2 System model

2.1 System architecture

The intelligent IoV scenario is considered in this paper, and the system architecture is given in Fig. 1. To realize the typical applications, such as HD mapping and intelligent driving decision model training with

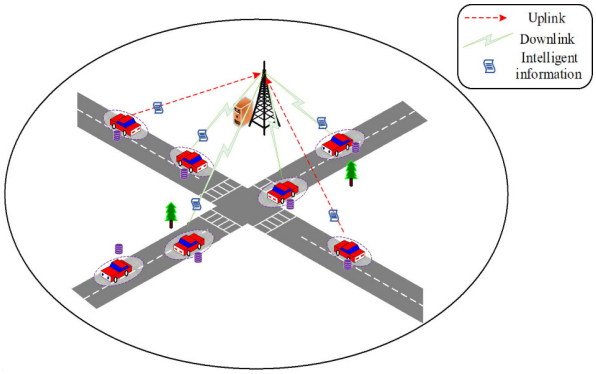


Fig. 1 System architecture.

the assistance of DML, an RSU with edge computing capabilities is deployed in the scenario. It performs as the aggregator node (denoted as aggregator) within the DML framework. N vehicles are active in the coverage area of the RSU, which act as worker nodes (denoted as workers). They are responsible for collecting road information related to the roadway and traffic to train the local model. A series of indexes of all the workers are denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$, $|\mathcal{N}| = N$.

The RSU communicates with vehicles within its coverage in multiple rounds, and each round of interaction is divided into two parts. During the uplink, the worker uploads the intelligent model parameters, then the RSU performs the aggregation of global parameters. As for downlink, the RSU broadcasts the latest parameters, and the workers which successfully received the new parameters will update their own local model parameters.

According to the investigation, the important factors that affect the aggregation performance of the DML-based HD mapping and intelligent driving decision model training include the timeliness factors and regional factors. The former ones reflect the time of the onboard model parameters from last update and the delay of uploading the local model parameters to the RSU. The regional factors represent the spatial distribution of all vehicles and the sensing range of each onboard sensor, which is defined as the effective sensing area in this paper. The definitions of these metrics are presented below.

2.2 Global parameter broadcast

The timeliness of the onboard intelligent driving decision model is related to the downlink transmission quality, which measures the interval of the onboard model parameters from last global update. The interval of the current local model of the i-th worker from the last global parameter updating is denoted by L_i, and its initial value is 1, indicating that the local model is in the latest state. The size of the intelligent model parameters is the same, and the outage possibility of each worker can be obtained according to the downlink transmission quality[12]. Once a worker's link outage is greater than the threshold, the downlink global parameter broadcast information of this round can not be successfully received and L_i is increased by 1. Otherwise, L_i is reset to the initial value to represent that it has updated the intelligent model. Based on L_i, the timeliness of the local model can be defined as

$$F_i = \max\left(1 - \frac{1}{25}L_i^2, -1\right) \tag{1}$$

where the range of the values of F_i is $[-1, 1)$, and F_i is negative when L_i is greater than 5, which reflects that the local parameters are outdated. The larger the F_i is, the "fresher" the local parameters are.

2.3 Local parameter uploading

The delay corresponding to the process of uploading model parameters by the worker is defined considering the uplink transmission quality. When multiple workers perform uploading process at the same time, the aggregator needs to arrange a schedule and allocate the channel resources for these workers. Obviously, the delay of the worker is highly depending on its order in the schedule and the uplink channel quality. D_i represents the upload delay of the i-th worker. Obviously, the larger the D_i is, the larger the upload delay is.

2.4 Effective sensing area

Considering that the intelligent driving decision model and HD mapping are based on HD live road information, the effective sensing area related to the spatial distribution of workers and the range of onboard sensors is defined. The scene is abstracted as a convex area, and the whole workers are randomly distributed in it. Each worker has a sensing area, and E_i represents the sensing area of the i-th worker. In reality, the sensing area

between vehicles may overlap. To study the impact of different geographical information, we used the Voronoi diagram to calculate the area of each worker. The larger the sensing area E_i is, the more HD live information it can obtain.

3 Problem formulation

3.1 MDP model

Considering the long-term system performance, the worker selection of the aggregator can be modeled as an MDP. The necessary elements are defined as quadruple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state space and the action space, respectively. $\mathcal{P}(S'|S, A)$ represents the transaction probability from state $S \in \mathcal{S}$ to $S' \in \mathcal{S}$ with action $A \in \mathcal{A}$, and $\mathcal{R}$ denotes the space of reward. In the following of this section, the state, action, reward, and policy of each time step are described in detail.

(1) System state

The system state includes the available channel resources of the RSU, the channel resources requirements of the workers, the reward of model timeliness, and effective sensing area of the workers. To compromise between accuracy and complexity, we only observe the system from time step t to $t + T - 1$.

At each time step t of the MDP, the available channel resources states of the RSU are denoted as a matrix $\boldsymbol{C}(t) = [C_{1,mn}]_{T\times B_1}$, of which the m-th row represents the number of the available channel resources at time step $t + m$ by a binary number. Similarly, the channel resource demand of the i-th worker is denoted as $\boldsymbol{C}_{S,i}(t) = [C_{i2,mn}]_{T\times B_2}$ and $\boldsymbol{C}_{L,i}(t) = [C_{i3,mn}]_{T\times B_3}$, of which each row is a binary number. The m-th row of $\boldsymbol{C}_{S,i}(t)$ and $\boldsymbol{C}_{L,i}(t)$ is the required number of channel resources and the time duration of the parameter uploading according to the estimated wireless channel quality of the i-th worker at time step $t + m$. The limitation of system resources is represented by the number of digits B_y, $y \in \{1, 2, 3\}$ of binary numbers, and the parameter upload delay D_i is implied in these matrixes.

The normalized value of timeliness of the local model $F_{i,\text{norm}}$ and the effective sensing area $E_{i,\text{norm}}$ are also contained in the system state, denoted by $\boldsymbol{G}_i(t) = [G_{i,m1}, G_{i,m2}]_{T\times 2}$, where $G_{i,mn} \in [0, 1]$, $i \in \{1, 2, \ldots, N\}$. $G_{i,m1}$ and $G_{i,m2}$ represent the estimated value of $F_{i,\text{norm}}$ and $E_{i,\text{norm}}$ at time step $t + m$, respectively.

Combining the states defined above, the state of the whole system can be obtained as

$$S(t) = \{\boldsymbol{C}(t), \boldsymbol{C}_{H,1}(t), \ldots, \boldsymbol{C}_{H,i}(t), \ldots, \boldsymbol{C}_{H,N}(t)\} \tag{2}$$

where $\boldsymbol{C}_{H,i}(t) = [\boldsymbol{C}_{S,i}(t), \boldsymbol{C}_{L,i}(t), \boldsymbol{G}_i(t)]$ if the i-th worker has not been selected at current time step t, Otherwise, $\boldsymbol{C}_{H,i}(t)$ is set to be $\mathbf{0}$ matrix as an indicator of a selected worker.

(2) Action space

The action at time step t is the worker selection decision. Note that multiple workers can be selected at the same time step. To simplify the problem, we decompose the worker selection decision at each time step into a sequential decision:

$$A_t = \left[a_{t,1}, \ldots, a_{t,k}, \ldots, a_{t,K_t}\right] \tag{3}$$

where $A_t \subseteq \mathcal{N}$ is a subset of $\mathcal{N}$, $a_{t,k} \in \mathcal{N}$ represents the index of worker selected at the k-th decision of time step t, and $K_t \leqslant N$ is the number of all the selected users at time step t. Actually, the system state and reward are also decomposed according to the sequential decision. To ensure the potential resource reservation in the MDP, a null action, which means no worker is selected, is also included in the action space. Thus, the total number of the actions is $N + 1$. When the null action is selected or the available channel resources can not satisfy the requirement of worker to be selected, the system will move on to the next time step.

(3) Reward

According to the system model in Section 2, the factors that affect the performance of the intelligent driving decision model and HD mapping include the timeliness of the local model, the delay of uploading parameters, and the effective sensing area. As the ranges of the values of F_i, D_i, and E_i can be quite different in the system, the normalized values $E_{i,\text{norm}}$, $F_{i,\text{norm}}$, and $D_{i,\text{norm}}$ are used to design the reward to ensure the fairness of different factors. The reward of each time step t is expressed as

$$R_{t+1}(A_t) = \sum_{i \in J_1(t)} [r_1 E_{i,\text{norm}}(A_t) + r_2 F_{i,\text{norm}}(A_t)] - \sum_{i \in J_2(t)} r_3 D_{i,\text{norm}}(A_t) \quad (4)$$

where r_1, r_2, and r_3 are the weights, $J_1(t)$ is the set of workers that have been selected and scheduled successfully at time step t, and $J_2(t)$ is the set of workers waiting in the queue and the workers who are uploading parameters at time step t.

Further, the long-term optimization function is the cumulated value function v of the initial state S_0, expressed as

$$f(U) = v(S_0) = \sum_{t=1}^{T_\text{M}} \lambda^{t-1} R_t \quad (5)$$

where $U = [A_1, A_2, \ldots, A_t, \ldots]$ is the worker selection scheme of the whole MDP, λ is the discount factor, and T_M is the maximum number of time steps of the MDP.

(4) Policy

The policy π is the mapping from the states to the probability distribution of the actions, $\pi: \mathcal{S} \to \mathcal{A}$, which indicates the probability that each worker may be selected. At the k-th decision of time step t, the vector of probabilities can be calculated by the policy $V_{t,k} = \pi(S_{t,k})$, and $a_{t,k}$ can be decided according to $V_{t,k}$ using the roulette method.

3.2 Complexity analysis

According to what we designed in Section 3.1, the dimension of the state space is $n_\text{s} = 2^{T(B_1+B_2N+B_3N)} \times 1001^{2TN}$ since each element of $\boldsymbol{G}_i(t)$ keeps three decimal places between 0 and 1. According to Ref. [13], it is inconvenient to adopt Dynamic Programming (DP) methods such as value iteration and policy iteration because the complexity is $O(n_\text{s}^2)$. The complexity will decrease if the DRL-based method is used. A neural network is used to represent the policy, of which the input is the system state and the output is a vector of probabilities. Therefore, the complexity is mainly depending on the number of neurons in the input layer n_i. When n_i equals the number of elements in each state, the corresponding complexity is $n_\text{i} = T(B_1 + B_2N + B_3N) + 2TN$. Besides, the greedy algorithm is a potential method because it ranks the aggregator's return corresponding to all the actions and selects the action with the maximum return in each decision directly. The computational complexity of the greedy algorithm is $n_\text{g} = (B_2 + B_3 + 2T)N$, and it is reasonable to use it as one of the comparison algorithms.

4 DRL-based solution

The DRL-based solution framework is given in Fig. 2, and the ultimate target is to seek an ideal policy to maximize the cumulative reward. The policy network π_θ takes the state $S_{t,k}$ as input and outputs the probability distribution of the actions at the k-th decision of the time step t. The agent chooses the appropriate action $a_{t,k}$ and the system state transits to $S_{t,k+1}$. When the decisions of the current time step are finished, the corresponding reward R_{t+1} is calculated. Therefore, a trajectory of the system behavior is obtained and denoted by $\tau = \{S_t, A_t, R_{t+1}\}_{t \in [0, T_\text{M}-1]} \sim \pi_\theta$, where $S_t = [S_{t,1}, \ldots, S_{t,k}, \ldots, S_{t,K_t}]$.

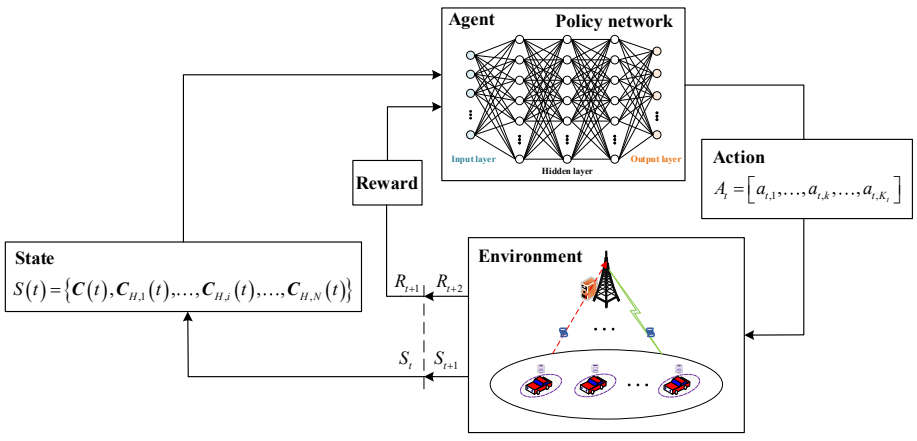


Fig. 2 DRL-based solution framework.

In this paper, the policy gradient method is used, and the Monte Carlo REINFORCE algorithm with baseline is used to train the policy network. Thus, the proposed method is named as the PG-WS algorithm. The parameter θ of the policy network is initialized firstly. It consists of the weight W and the bias b of the neural network. The Gaussian distribution with a standard deviation of 0.01 and a mathematical expectation of 0 is used to initialize the weight W, and the bias b uses the constant value 0 as the initial value. Several snapshots of the scenario are imported into the simulation platform for episodic simulations. For each training iteration, Q episodes for each snapshot are simulated to obtain corresponding trajectories $\tau = \{S_t^q, A_t^q, R_{t+1}^q\}_{t\in[0,T_{\mathrm{M}}-1]} \sim \pi_\theta$, $q \in [1, Q]$, and each trajectory is used as a training sample. The process of one episodic simulation is presented in Algorithm 1.

In the training process, we use the average of the state's value function as the baseline $b = Q^{-1}\sum_q v^q\left(S_0^q\right)$. The update rule[14] of parameters can be denoted as

$$\Delta\theta = \sum_{q=1}^{Q}(v^q(S_0^q) - b)\nabla_\theta \ln \pi_\theta(A\,|S) \qquad (6)$$

Algorithm 1 Process of one episodic simulation in HD mapping

1. input: a differentiable policy parameterization $\pi(a\,|s,\theta)$;
2. initialize policy parameter θ: $W \sim N(0, 0.01)$, $b = 0$;
3. run episodes $q = 1, 2, \ldots, Q$;
4. $t = 0$, $k = 1$;
5. **while** time step $t < T_{\mathrm{M}}$ **do**
6. **if** time step t is for global parameter broadcast **then**
7. **for** each worker **do**
8. **If** the link outage of the global parameter broadcast is greater than threshold, **then** $L_i \leftarrow L_i + 1$;
9. **else** $L_i = 1$;
10. **elseif** time step t is for local parameter uploading
11. **for** each worker **do**
12. $L_i \leftarrow L_i + 1$;
13. **end if**
14. get an action $a_{t,k}^q$ based on π_θ;
15. **if** $a_{t,k}^q$ is null or the available resources are insufficient **then**
16. compute R_{t+1}^q according to Eq. (4);
17. update S_t^q;
18. record $\{S_t^q, A_t^q, R_{t+1}^q\}$;
19. $t \leftarrow t + 1$, $k = 1$;
20. **else**
21. the aggregator selects the worker $a_{t,k}^q$;
22. update $\boldsymbol{C}(t)$;
23. $\boldsymbol{C}_{H,a_{t,k}^q}(t)$ is set to be **0** matrix;
24. $k \leftarrow k + 1$;
25. **end if**
26. **end while**
27. calculate $v(S_0^q) = \sum_{t=1}^{T_{\mathrm{M}}} \lambda^{t-1} R_t$.

5 Performance evaluation

5.1 Training phase

The main parameters adopted in the simulation are shown in Table 1. The system has one observed RSU with other 18 interference RSUs loated around, the bandwidth is set to 10 MHz, and the node transmission power is 23 dBm. The relevant parameter settings refer to the 3GPP technical specifications for V2X scenario[15].

Table 1 Simulation parameters.

Parameter	Setting
Inter sites distance (m)	100
Number of vehicle nodes N	30
Noise density (dBm/Hz)	−174
Size of intelligent information (bits)	12 000
Number of channel resource units	48
Moving speed of vehicle nodes (km/h)	[45, 90]
Time of tasks in uplink transmission	{1, 2, 4, 8}
Threshold of link outage	0.05
Time length of observed states T	20
Discount factor λ	0.9
Initial weights r_1, r_2, and r_3	1

The policy network is a neural network with a fully connected hidden layer of 50 neurons. In order to compromise between training time and model accuracy, we sampled 50 snapshots of the scenario given in Section 2, each of which contains 30 vehicle nodes with continuous distributed model parameter uploading requests. For each snapshot, $I = 5$ episodes are simulated to visit more system states in the training process, and each episode lasts for $T_{\mathrm{M}} = 150$ time steps.

Greedy and random algorithms are evaluated as comparison methods. In our experiments, the aim of greedy algorithm is to select the worker with the maximum reward at each time step.

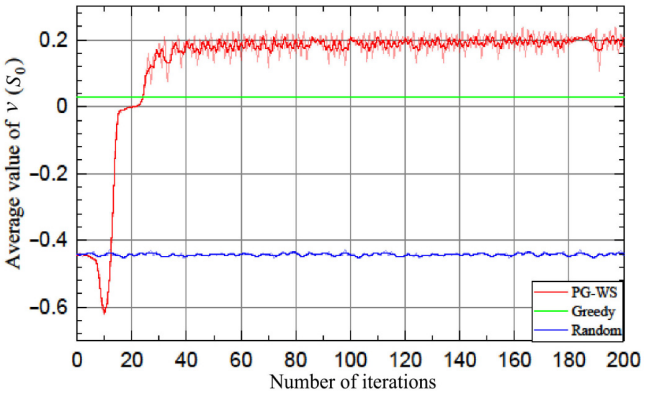


Fig. 3 Performance of average value in the training process.

Figure 3 shows the average value corresponding to three different methods in the training phase, and the learning rate of PG-WS is 0.0005. We used a sliding window to average the value of different iterations in Fig. 3 to reduce accidental factors. As the policy network is initialized with random parameters, the average value of the PG-WS algorithm begins with that of random algorithm. Then the policy network tries to explore more possible probability distributions of the actions, and the average value of the PG-WS algorithm decreases at the first 10 iterations. However, through the guide of the reinforce signal, the policy network becomes wiser after more training iterations. Then the value rises to a level slightly lower than greedy algorithm at 10–20 iterations. After 20 iterations, the performance of our method exceeds greedy algorithm and the average value converges to around 0.19. It is reasonable that the total reward of random algorithm is negative which means the delay in uplink is a large proportion of the total reward. Since the three aggregate rewards have been normalized in simulation, the difference instead of percentage of two methods is used to reflect how much the performance has improved. The average reward of the proposed method is 0.165 more than greedy algorithm.

The effect of different learning rates (lr) on the training performances is shown in Fig. 4. All the training processes experience the performance decrease during the exploration period at the beginning, and there is a plateau when the average value is around 0. It can be seen that the most appropriate learning rate is about 0.0005. When the learning rate is greater than 0.0005,

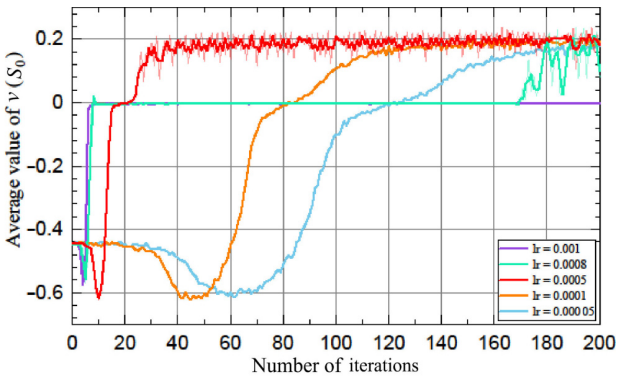


Fig. 4 Training performance with different learning rates.

the policy network is trapped into the plateau. Since the learning rate is large, the policy network might miss the right direction to the optimal point. When the learning rate is less than 0.0005, the convergence speed of model decreases, but the average reward can still reach about 0.19.

5.2 Testing phase

The performance of the proposed method is tested in the environments with different parameter settings. 250 snapshots are sampled for each test set, and 5-episode simulation is performed for each snapshot.

In Fig. 5, the performance comparison is given when the weight of model timeliness and effective area is varied from 1.0 to 1.5. The average reward of PG-WS is 0.154, 0.078, and 0.116 higher than that of the greedy algorithm with different ratio values. From the problem formulation and the simulation results, the advantages of PG-WS compared to other methods mainly reflect that DRL is "far-sighted" and can make appropriate reservations of resources for the future.

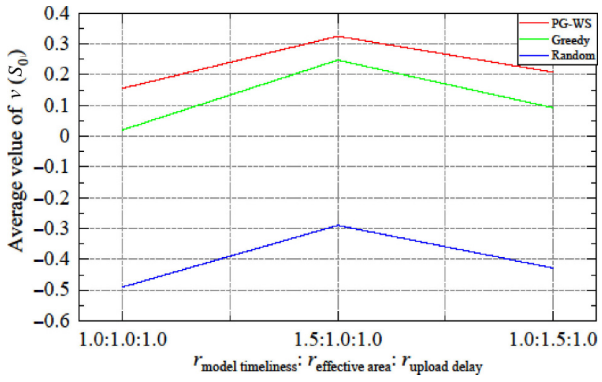


Fig. 5 Performance comparison with different reward weights.

6 Conclusion

In this paper, we focus on the intelligent IoV, and design a DML framework based on the interaction between RSU and the intelligent vehicles for HD mapping and intelligent driving decision model training. The timeliness of the local model, the transmission quality of model parameters uploading, and the effective sensing area related to these two applications are taken into account in the worker selection problem. To optimize the long-term return of RSU, MDP is modeled and a DRL-based worker selection method called PG-WS is proposed as the solution. Simulation results show that the proposed PG-WS algorithm outperforms other comparation methods. In the future research, the proposed PG-WS algorithm will be used to implement the worker selection for a real DML-based intelligent application in the IoV scenario and the performance of the real DML-based intelligent application will be evaluated to validate the effectiveness of the proposed node selection algorithm in practice.

Acknowledgment

This work was supported by the Science and Technology Foundation of Beijing Municipal Commission of Education (No. KM201810005027), the National Natural Science Foundation of China (No. U1633115), and the Beijing Natural Science Foundation (No. L192002).

References

[1] J. Zhang and K. B. Letaief, Mobile edge intelligence and computing for the internet of vehicles, *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, 2020.

[2] W. C. Xu, H. B. Zhou, N. Cheng, F. Lv, W. S. Shi, J. Y. Chen, and X. M. Shen, Internet of vehicles in big data era, *IEEE/CAA J. Autom. Sin.*, vol. 5, no. 1, pp. 19–35, 2018.

[3] TomTom HD map for autonomous driving extends to Japan, https://corporate.tomtom.com/news-releases/news-release-details/tomtom-hd-map-autonomous-driving-extends-japan?releaseid=1045730, 2017.

[4] HERE introduces HD live map to show the path to highly automated driving, https://360.here.com/2016/01/05/here-introduces-hd-live-map-to-show-the-path-to-highly-automated-driving/, 2016.

[5] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, Street-view change detection with deconvolutional networks, *Auto. Robots*, vol. 42, no. 7, pp. 1301–1322, 2018.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, Communication-efficient learning of deep networks from decentralized data, arXiv preprint arXiv: 1602.05629, 2017.

[7] J. M. Chen, X. H. Pan, R. Monga, S. Bengio, and R. Jozefowicz, Revisiting distributed synchronous SGD, arXiv preprint arXiv: 1604.00981, 2016.

[8] S. Q. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, When edge meets learning: Adaptive control for resource-constrained distributed machine learning, presented at IEEE INFOCOM 2018-IEEE Conf. Computer Communications, Honolulu, HI, USA, 2018, pp. 63–71.

[9] R. Zhang, F. R. Yu, J. Liu, T. Huang, and Y. J. Liu, Deep reinforcement learning (DRL)-based Device-to-Device (D2D) caching with blockchain and mobile edge computing, *IEEE Trans. Wireless Comm.*, vol. 19, no. 10, pp. 6469–6485, 2020.

[10] Y. Gao, W. J. Wu, H. X. Nan, Y. Sun, and P. B. Si, Deep reinforcement learning based task scheduling in mobile Blockchain for IoT applications, presented at ICC 2020-2020 IEEE Int. Conf. Communications (ICC), Dublin, Ireland, 2020, pp. 1–7.

[11] M. Li, F. R. Yu, P. B. Si, W. J. Wu, and Y. H. Zhang, Resource optimization for delay-tolerant data in blockchain-enabled iot with edge computing: A deep reinforcement learning approach, *IEEE Int. Things J.*, vol. 7, no. 10, pp. 9399–9412, 2020.

[12] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications, *IEEE Trans. Wirel. Comm.*, vol. 16, no. 11, pp. 7574–7589, 2017.

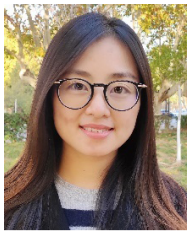
[13] H. Liu, S. W. Liu, and K. Zheng, A reinforcement learning-based resource allocation scheme for cloud robotics, *IEEE Access*, vol. 6, pp. 17 215–17 222, 2018.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning*: *An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

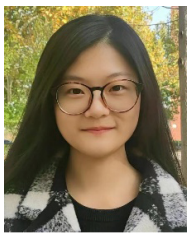
[15] *Enhancement of 3GPP Support for V2X Scenarios*, 3GPP TS 22.186, 2019.

Junyu Dong received the BS degree in communication engineering from North China University of Technology in 2019. He is currently pursuing the MS degree at Faculty of Information Technology, Beijing University of Technology. His research interests include internet of vehicle and mobile edge computing.

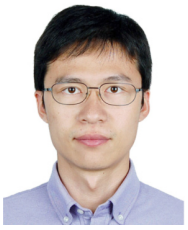
Xiaoxi Wang received the BS degree in communication engineering from Hebei University of Engineering in 2019. She is currently pursuing the MS degree at Faculty of Information Technology, Beijing University of Technology. Her current research interests include distributed machine learning and wireless networks.



Wenjun Wu received the BS and PhD degrees from Beijing University of Posts and Telecommunications, Beijing, China in 2007 and 2012, respectively. From 2012 to 2015, she was a post-doctoral researcher at the School of Electronic and Information Engineering, Beihang University, Beijing, China. She is now working as an associate professor at the Faculty of Information Technology, Beijing University of Technology, Beijing, China. Her research interests are in the fields of mobile edge computing, blockchain, and deep reinforcement learning.



Yang Gao received the BS degree in communication engineering from Beijing University of Technology, Beijing, China in 2018. She is currently pursuing the PhD degree in electronic science and technology at Beijing University of Technology, Beijing, China. Her current research interests include mobile edge computing, blockchain, deep reinforcement learning, and wireless resources management.



Pengbo Si received the BE and PhD degrees from Beijing University of Posts and Telecommunications, Beijing, China in 2004 and 2009, respectively. He joined Beijing University of Technology, Beijing, China in 2009, where he is currently a full professor. During November 2007 and November 2008, he was a visiting student at Carleton University, Ottawa, Canada. During November 2014 and November 2015, he was a visiting scholar at the University of Florida, Gainesville, FL, USA.