

Research Article

MEC-Based Cooperative Multimedia Caching Mechanism for the Internet of Vehicles

Longzhe Han ¹, Sheng Li ¹, Chenchen Ao ¹, Yan Liu ², Guangming Liu ¹,
Yiyang Zhang ³, and Jia Zhao ¹

¹School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China

²College of Computer and Software Engineering, East China Normal University, Shanghai 200062, China

³College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin 300222, China

Correspondence should be addressed to Jia Zhao; zhaojia925@163.com

Received 2 May 2022; Accepted 20 August 2022; Published 10 September 2022

Academic Editor: Yuanlong Cao

Copyright © 2022 Longzhe Han et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multimedia applications are expected to widely deploy over vehicular networks. In order to meet the low-latency and high-speed transmission requirements of multimedia applications, edge caching is introduced to reduce the network traffic and the transmission delay. Due to the limited storage of the edge cache server, an efficient approach for the content management plays a decisive role for the edge cache performance. This paper proposes a vehicle-to-infrastructure-based cooperative caching mechanism for Internet of Vehicles to improve the edge cache utilization. The system model is established with the goal of maximizing the cooperative caching hit rate. To jointly consider the collaborations between macrobase stations (MBS) and multiple roadside units (RSU), we propose a reinforcement learning algorithm to adaptively control the cache management. According to the content popularity and the network status, the proposed algorithm can dynamically adjust cached content across relevant MBSs and RSUs. The simulation results show that the proposed cooperative caching mechanism significantly improve the cache utilization and the quality of services.

1. Introduction

With the rapid development of the Internet of Vehicles (IoV) and 5G communication technology, a large number of multimedia applications, such as traffic video processing, in-vehicle infotainment, and transportation environment monitoring, are emerged to enrich the intelligent transport system [1–3]. To provide high quality of services for multimedia applications in IoV, mobile edge computing (MEC) has attracted the attention as an emerging technology to improve system performance, resource utilization, and reduced transmission delay [4–6]. By introducing computation and storage capabilities to network edge nodes, such as roadside units (RSU) and base stations, the transmission pressure on the core network can be effectively relieved, and at the same time, the content transmission delay can be reduced [7–10]. However, the limited cache space of edge nodes, the time-varying content popularity, the high speed of vehicles, and the constant change

of the IoV topology are challenging for the edge cache performance. It is necessary to design an efficient management strategy to efficiently manage the edge cache [11–13].

Currently, there are numerous studies conducted on the subject of edge cache management for IoV. Huang et al. [14] proposed a cache location selection mechanism based on the vehicle trajectory, which can effectively reduce the system load and cache energy consumption. Shi et al. [15] proposed a deep learning communication model based on multimodal compression, which exploited the redundancy between deep learning models in different scenarios to accelerate content transmission in edge networks. In [16], a mixed integer nonlinear programming method was proposed to minimize the cooperative delay between edge servers, and the Lyapunov optimization method was used to optimize the delay problem. In [17], the authors comprehensively considered the mobility of vehicles and proposed an edge caching scheme with perceptible mobility probability. By dividing the data

into data blocks of different sizes and buffering these data blocks in the edge server close to the vehicle, the overhead and transmission delay of backhaul traffic were reduced. Meng et al. [18] studied the cache service strategy of offline networking in the edge computing environment and proposed a cache storage algorithm on node core. In [19], a new information-centric heterogeneous network framework was designed, using a distributed algorithm with alternating-direction multipliers to solve the problem of cache resource allocation. In order to further reduce the transmission delay and improve the response rate, the authors [20] proposed a cooperative cache allocation and calculation offload scheme, and the MEC servers were cooperated to perform calculation tasks and data caching. With the rapid development of artificial intelligence, the deep reinforcement learning [21] has been widely used in edge caching and resource allocation of vehicle networks with its unique perception and decision-making capabilities [22, 23].

The main contributions of this paper are as follows: to take full advantage of edge cache resources, a hierarchical cooperative architecture, including MBSs, RSUs, and vehicles, are introduced. We establish a Markov decision model based on the proposed architecture to describe the cooperative caching process. We propose a reinforcement learning cache management algorithm, which follows the Deep Deterministic Policy Gradient (DDPG) scheme. The proposed algorithm has fast convergence rate and can self-adapt to the complex network environment.

The rest of this paper is organized as follows. Section 2 presents the system model for cooperative edge caching. Section 3 discusses the proposed cooperative caching mechanism. The experiment settings and result analysis are presented in Section 4. Finally, in the Section 5, we discuss concluding remarks and our future work.

2. System Model

2.1. Cooperative Edge Caching Model. In order to make full use of the storage of MBSs, RSUs, and vehicles, we construct a three-layer cooperative cache architecture, as shown in Figure 1. The core layer includes MBSs and the backhaul network, and the MBSs are connected to the RSUs through wired links. For the cooperative RSU layer, it consists of RSUs distributed in different areas, and the RSUs communicate through wireless links. The vehicle layer includes vehicles running in different areas. MBSs, RSUs, and vehicles have storage to temporarily buffer certain amount of content. Initially, the vehicle sends a content request. If the vehicle itself has the content, it will obtain directly from its cache. If not, it will send the request to the local RSU. If the local RSU does not store the content, the local RSU queries the cooperative RSUs. If neither the local RSU nor the cooperative RSUs have the content, the request is sent to the core layer.

MBS is responsible for collecting system status information, controlling global resource management, and content caching decisions. Compared with obtaining content from a remote server, the cooperative caching model can effectively reduce the transmission delay and transmission cost. The set of RSUs can be expressed as $R = \{1, 2, 3, \dots, R\}$. $V = \{v_1, v_2, \dots, v_N\}$ represents the set of vehicles under

the coverage of the RSU. The RSU is responsible for collecting relevant information of vehicles under its own coverage area and uploading to the MBS.

2.2. Content Delivery Model. In the multilevel cooperative edge caching model, the vehicle v_i can send content requests to the RSU or adjacent vehicles. Vehicles within the coverage area of one RSU use the same frequency band to communicate, and it is leading to interference between vehicles. Therefore, the transmission rate from RSU r to vehicle v can be obtained from Shannon's formula as

$$R_{r,v} = b_{r,v} B_R \log \left(1 + \frac{p_r h_{r,v}}{\sigma^2 + \sum_{v'=1, v' \neq v}^V p_r h_{r,v'}} \right), \quad (1)$$

where $b_{r,v}$ represents the channel bandwidth allocated by the RSU r to the vehicle v , B_R represents the channel bandwidth of the RSU r , and p_r is the transmission power of the RSU r . $h_{r,v}$ is the channel gain between the RSU r and the vehicle v , and σ^2 represents noise power. $\sum_{v'=1, v' \neq v}^V p_r h_{r,v'}$ is the V2I communication downlink interference [24].

Orthogonal frequency division multiple access (OFDMA) is used between MBSs and vehicles. Vehicles associated with the MBS are assigned an orthogonal subcarrier, and the transmission rate from the MBS to vehicle v_i is

$$R_{m,v} = \frac{B_m}{W} \log \left(1 + \frac{p_m h_{m,v}}{\sigma^2} \right), \quad (2)$$

where B_m is the channel bandwidth of the vehicle, and p_m represents the transmission power of the vehicles. $h_{m,v}$ is the channel gain between the vehicle v and the MBS, and σ^2 represents the noise power [25].

2.3. Content Popularity Model. Assuming that there are K content requests, then the request probability of these K contents are $P_1, P_2, P_3, \dots, P_K$, and the probability obeys the Zipf distribution [26]. The relationship between the content request probability and the content popularity level can be expressed as [27]

$$P(s) = \frac{\Phi}{s^\theta} s \in \{1, 2, 3 \dots K\}, \quad (3)$$

$$\Phi = \sum_{i=1}^K \frac{1}{i^{-\theta}}, \quad (4)$$

where s represents the content popularity level, and θ is the Zipf impact factor, also known as the popularity slope. If θ is getting larger, the distribution of Zipf is steeper, and the popularity tends to be concentrated [28, 29]. The value of the Zipf factor depends on the users' behavior. The relationship between the request probability and popularity level can be further expressed as

$$P(s) = \frac{s^{-\theta}}{\sum_{i=1}^K i^{-\theta}}. \quad (5)$$

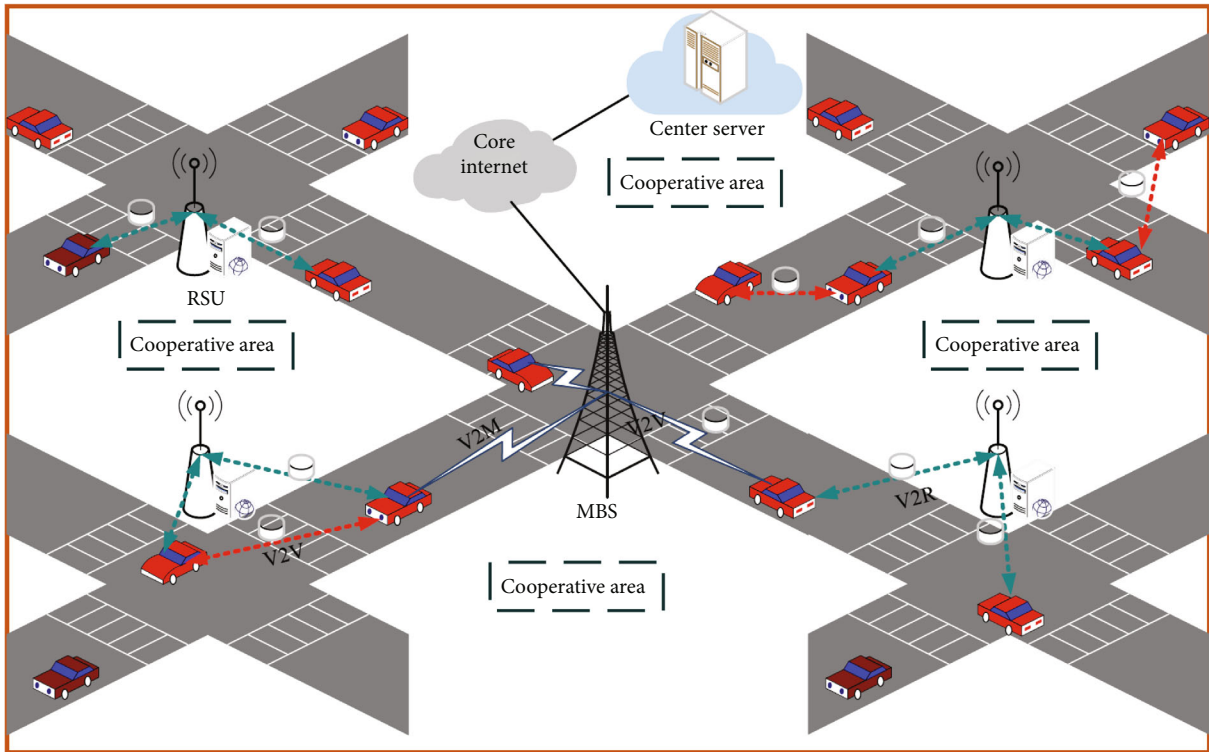


FIGURE 1: Cooperative edge cache model of IoV.

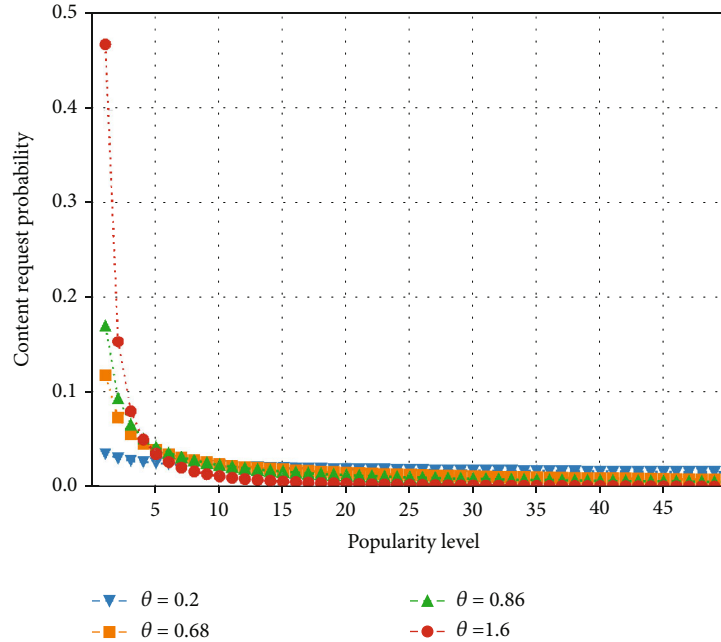


FIGURE 2: Relationship between content popularity level and request probability.

Figure 2 shows the relationship between the popularity level and the request probability. It can be seen that the influence of the popularity inclination on the request probability distribution. The content with high request probability only accounts for a small part of all content [30].

3. Cooperative Caching Mechanism

3.1. Problem Model. The cooperative caching is able to theoretically achieve a high cache hit rate than the noncooperative caching. We use a binary variable $c_{i,k} \in \{0, 1\}$, $i \in R, k \in K$ to

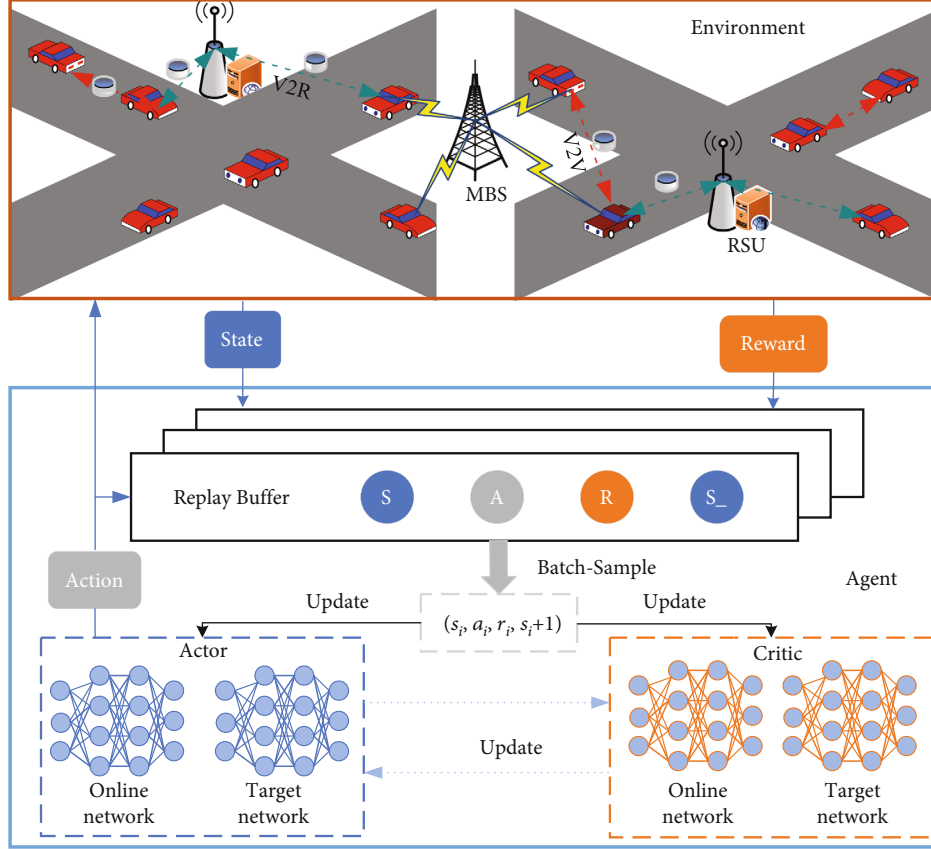


FIGURE 3: Schematic of the cooperative caching algorithm.

represent the cache state of the content k of RSU_i . $c_{i,k} = 1$ means that the content k is cached in the RSU_i , and $c_{i,k} = 0$ means that the RSU_i does not cache the content k . The cooperative edge cache hit rate of the RSU_i can be expressed as

$$h_i = \sum_{k \in K} p_k \left(c_{i,k} + \sum_{n \in R} c_{n,k} \delta_{n,i} \right), \quad (6)$$

where the binary variable $\delta_{n,i} \in \{0, 1\}$ indicates whether the RSU_n is cooperated with the RSU_i . Therefore, the average cooperative cache hit ratio of the system can be expressed as

$$\bar{h} = \frac{1}{R} \sum_{i \in R} h_i = \frac{1}{R} \sum_{i \in R} \sum_{k \in K} p_k h_i = \frac{1}{R} \sum_{i \in R} \sum_{k \in K} p_k \left(c_{i,k} + \sum_{n \in R} c_{n,k} \delta_{n,i} \right). \quad (7)$$

For RSU_p , the size of the cache space is S^{RSU_p} , and then the optimization problem of maximum average cooperative cache hit rate can be expressed as the following:

$$\begin{aligned} \max_{\{c_{i,k}\}} \quad & \bar{h} \\ \text{s.t.} \quad & c_{i,k} \in \{0, 1\}, i \in R, k \in K \\ & \sum_{k \in K} c_{i,k} \leq S^{RSU_i}, i \in R. \end{aligned} \quad (8)$$

Regarding to the vehicle cache, the cache hit rate of the vehicle j can be expressed as

$$h_j = \sum_{k \in K} p_k c_{j,k}. \quad (9)$$

Therefore, the average cache hit rate for all vehicles is expressed as

$$\bar{h}_j = \frac{1}{V} \sum_{j \in V} h_j = \frac{1}{V} \sum_{j \in V} \sum_{k \in K} p_k h_j = \frac{1}{V} \sum_{j \in V} \sum_{k \in K} p_k c_{j,k}. \quad (10)$$

The size of the cache space of the vehicle j is S^j . Under the limitation of the cache space, the problem of the maximum average cache hit rate of vehicles can be expressed as the following:

$$\begin{aligned} \max_{\{c_{j,k}\}} \quad & \bar{h}_j \\ \text{s.t.} \quad & c_{j,k} \in \{0, 1\}, j \in V, k \in K \\ & \sum_{k \in K} c_{j,k} \leq S^j, j \in V. \end{aligned} \quad (11)$$

Maximizing the cache hit rate of the system is to maximize the average cache hit rate of the cooperative caches and vehicles, as the following form:

```

1: Initialize Actor online network parameters  $\theta^Q$ , Critic online network parameters  $\theta^\mu$ , experience replay memory  $M$ 
2: Initialize Actor target network parameters  $\theta^{Q'}$ , Critic target network parameters  $\theta^{\mu'}$ 
3: Initialize caching state of RSUs, MBS and Vehicles, content popularity
4: for episode=1,  $M$  do
5:   Environment state space initialization, initialization system cache hit rate
6:   Randomly choose action  $N$  as action exploration
7:   for  $t=1,2,3,\dots,T$  do
8:     Select action  $a_t = \mu(s_t, \theta^\mu) + N_t$  according to observed state  $s_t$  and
       current strategy
9:     Calculate reward  $R(t)$  based on current selection action  $a_t$  and state  $s_t$ ,
       update state  $s_t \rightarrow s_{t+1}$ 
10:    Update the reward  $R(t) = CP \sum \sum \nu (\bar{h}(t) + \bar{h}_j(t))$  and store
        $(s_t, a_t, R(t), s_{t+1})$  in  $M$ 
11:    Randomly sample  $N$  samples from the experience replay memory  $M$ ,
        $(s_i, a_i, R(t^i), s_{i+1})$ 
12:    Evaluate  $y_i = R(t^i) + \gamma Q'(s_{i+1}, \mu'(s_{i+1} | \theta^{\mu'})) | \theta^{Q'}$ 
13:    Update Critic Network Parameters  $\theta^\mu$  by Minimizing Loss
14:     $L(\theta) = 1/N \sum_i (y_i - Q(s_i, a_i | \theta^Q))^2$ 
15:    Update Actor Network Parameters  $\theta^Q$  via Policy Gradients
16:     $\nabla_{\theta^\mu} J \approx 1/N \sum_i \nabla_a Q(s, a | \theta^Q) |_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s, \theta^\mu) |_{s_i}$ 
17:    Update target network parameters
18:     $\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$ 
19:     $\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$ 
20:   end for
21: end for

```

ALGORITHM 1: Cooperative edge caching algorithm for Internet of Vehicles based on DDPG.

$$\begin{aligned}
& \max_{\{c_{i,j,k}\}} && H = (\bar{h} + \bar{h}_j), \\
& \text{s.t.} && C1 : c_{i,k} \in \{0, 1\}, i \in R, k \in K, \\
& && C2 : c_{j,k} \in \{0, 1\}, j \in V, k \in K, \\
& && C3 : \sum_{k \in K} c_{i,k} \leq S^{RSU^i}, i \in R, \\
& && C4 : \sum_{k \in K} c_{j,k} \leq S^{V^j}, j \in V.
\end{aligned} \tag{12}$$

3.2. Cooperative Caching Algorithm Based on DDPG. To solve the optimization problem in the previous section, it is necessary to build the Markov decision process for the cooperative edge caching scenario. The Markov decision process is a tuple including state, action, and reward. The components are defined as follows: the system state at each time t is defined as $s_t = [S^{\text{MBS}}, R * S^{\text{RSU}}, V * S^v, q_t]$, that is, at time t , the cooperative cache space, cache state information, vehicle cache information, and vehicle content request. The action space at each time t is defined as $a_t = [a_0, a_1, a_2, a_3]$, where a_0 represents the content cached in the MBS, a_1 means the content cached in the RSU, a_2 represents the content cached in the vehicle itself, and a_3 is for the content cached in the randomly.

The system joint reward function $R(t)$ is expressed as $R(t) = CP \sum \sum \nu (\bar{h}(t) + \bar{h}_j(t))$, where C is the characteristic constant, and $\bar{h}(t) + \bar{h}_j(t)$ is the average cache hit rate of

TABLE 1: Simulation parameters.

System parameter	Value/description
RSU and MBS cache capacity	10 TB, 15 TB
Vehicle cache capacity	5 TB
RSU and MBS transmission power	35 dBm, 38 dBm
Number of contents	7000, 9000, 10000
Zipf impact factor	0.68
Noise power	-95 dBm
Wireless bandwidth	10 MHz
Wired bandwidth	20 MHz
Number of vehicles	25, 30, 35, 40, 45
Number of neural network layers	2
Number of neurons	[300, 400]
Learning rate	0.00025
Batch sampling size	64
Replay buffer size	7500

the cooperative cache and the average cache hit rate of the vehicle j at time t , respectively. P represents the penalty coefficient given by the vehicle.

The system block diagram of the cooperative edge caching algorithm is shown in Figure 3. The environment consists of an actor network, a critical critic, and an experience replay memory. The actor and critic network are both composed of two different deep neural networks. The online network is used

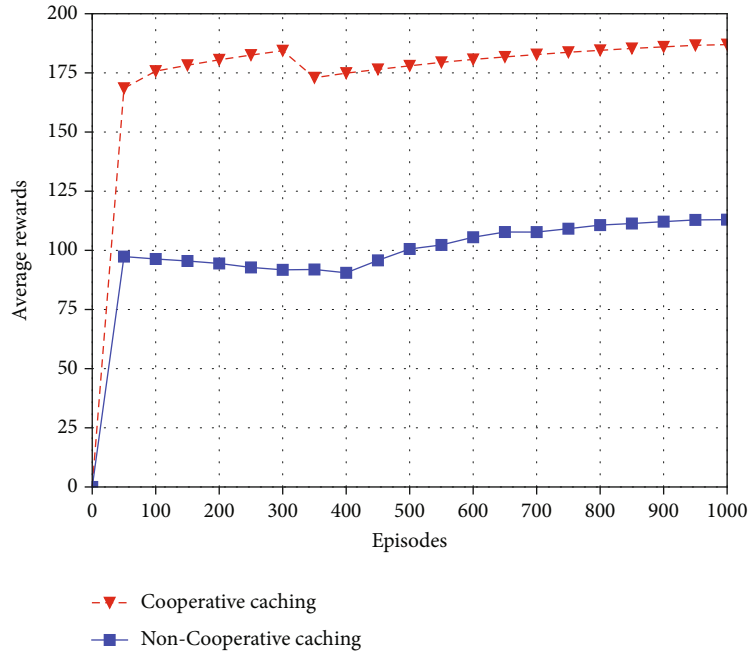


FIGURE 4: Comparison of cumulative average rewards of different caching schemes.

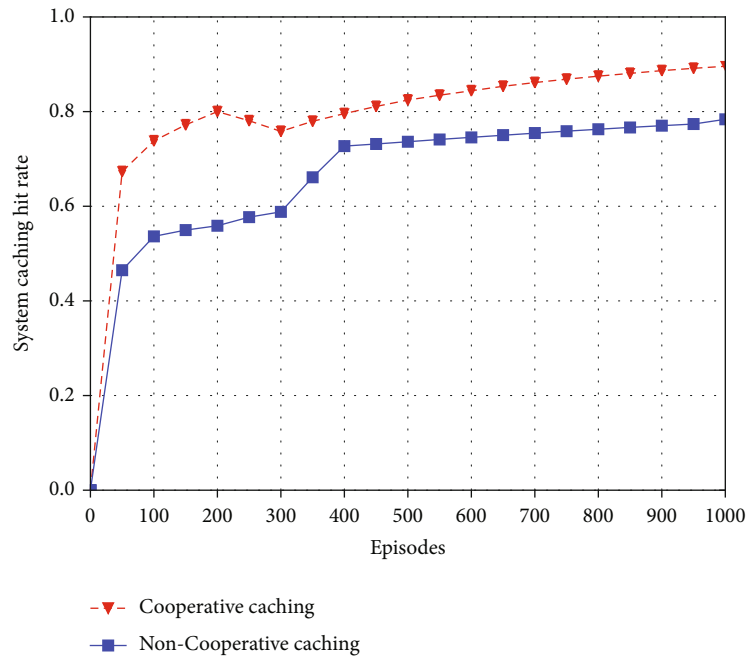


FIGURE 5: The impact of different caching strategies on the system cache hit rate.

for actions, and the target network is used for the evaluation of actions. The agent receives the environmental state information and executes the corresponding action. Algorithm 1 shows the flow of the cooperative edge caching algorithm [16]. First, it initializes the network parameters of the actor network, the critic network, and the experience replay memory. After the parameter initialization is completed, the agent obtains the environmental state information and makes a decision for the content caching. The agent receives

immediate reward feedback from the system, and the system enters the next new state. The agent store the current status information into the experience replay memory for future training.

4. Experiment Results and Analysis

In the simulation environment, the cache capacity of the MBS is 15 TB, and the coverage radius is 2 km. The RSU

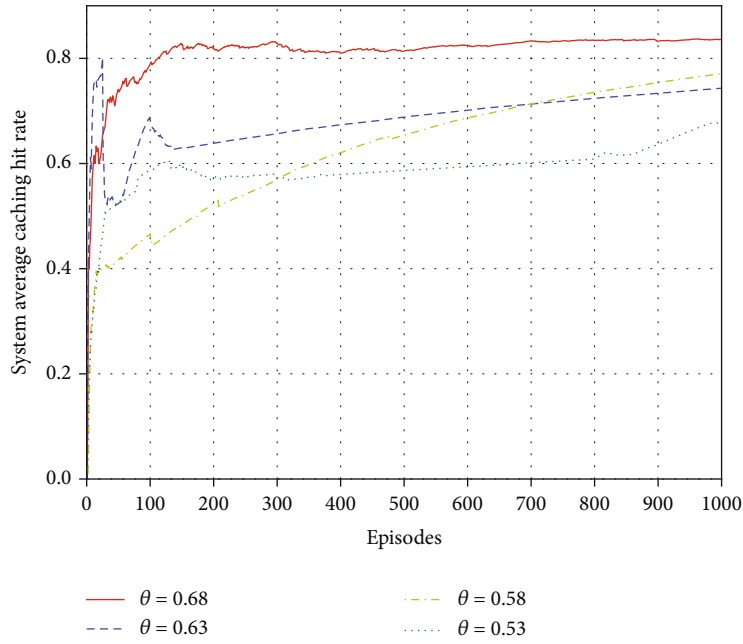


FIGURE 6: Comparison of the average hit rate under different Zipf distributions.

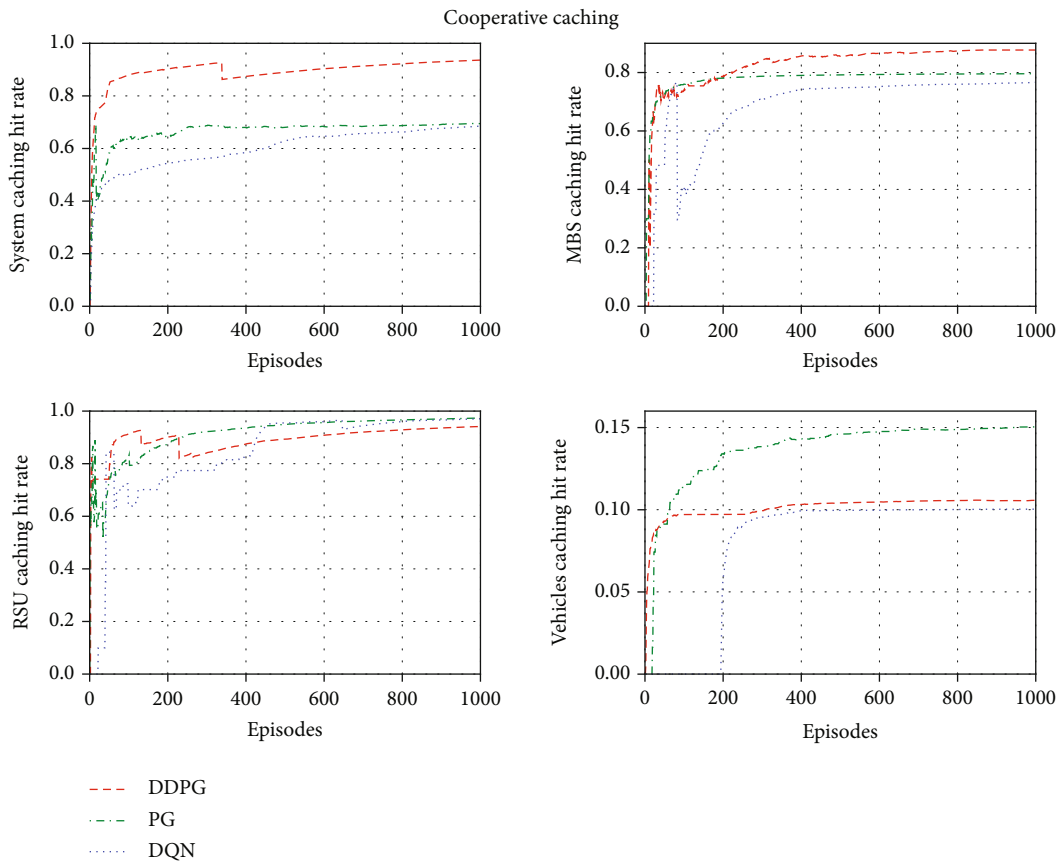


FIGURE 7: Comparison of the cache hit ratio under the cooperative cache scheme.

cache capacity is 10 TB, and the coverage radius is 200 m. The range of the number of vehicles is from 25 to 45, and the size of the vehicle cache is 5 TB. The transmission power

of RSU and MBS content is 35 dBm and 38 dBm, respectively. The Zipf impact factor is 0.68. The neural network parameters are set as two hidden layers, and the activation

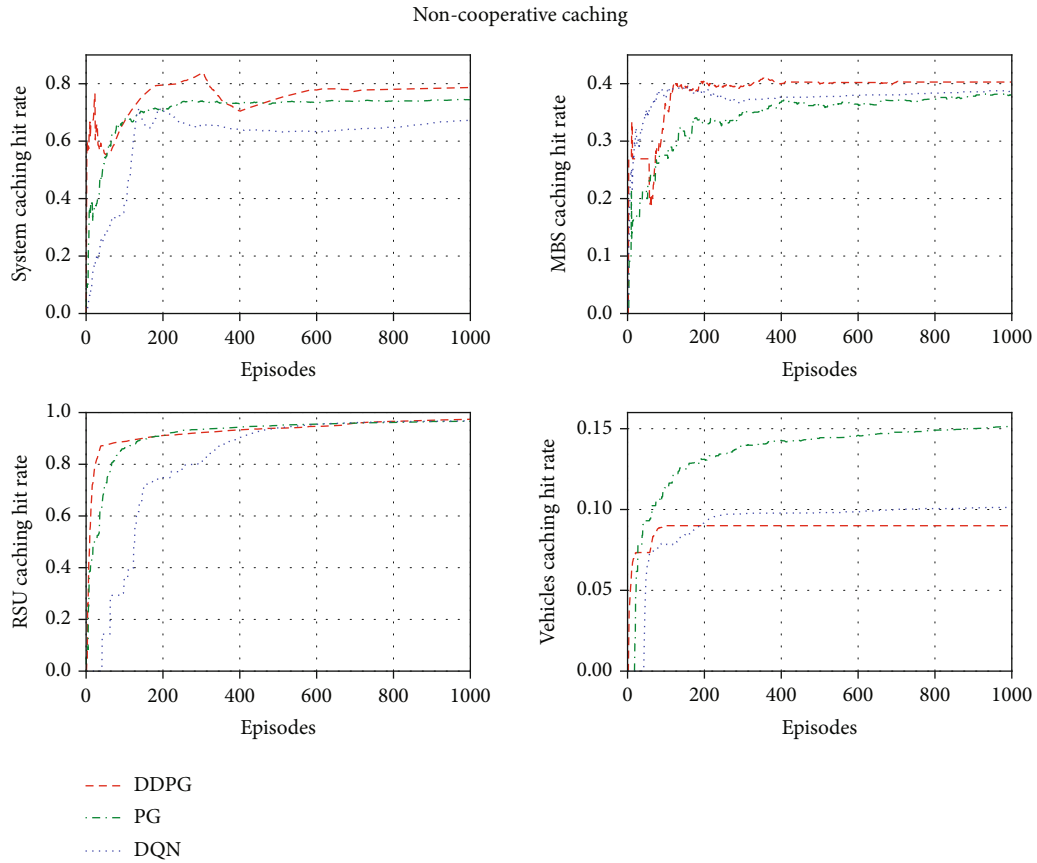


FIGURE 8: Comparison of cache hit ratios under the noncooperative cache scheme.

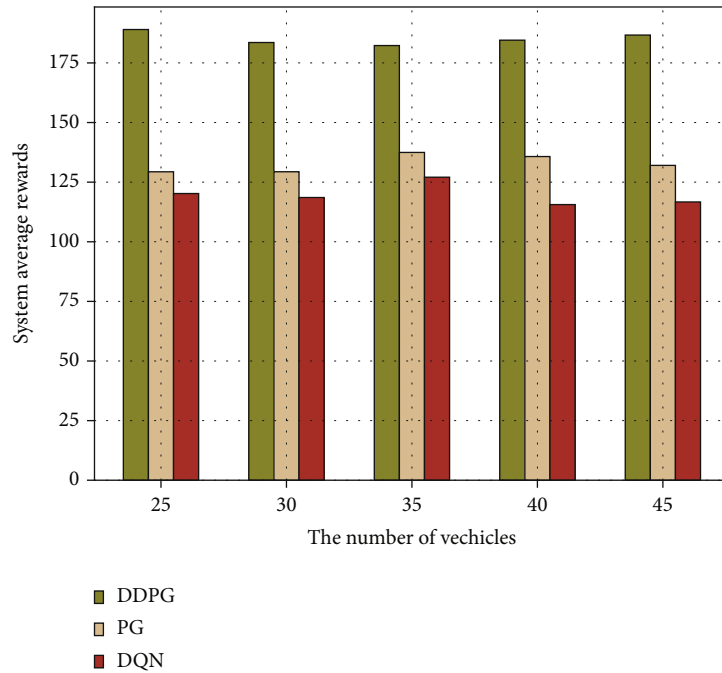


FIGURE 9: The effect of the number of vehicles under different algorithms.

function is ReLU [15]. The detailed settings of parameters are shown in Table 1.

To verify the performance of the cooperative edge caching strategy, we compare the cooperative caching scheme with noncooperative caching scheme under the parameters setting of Table 1. The learning process of cooperative and noncooperative caching strategies is shown in Figure 4. The average reward of cooperative caching rapidly increases to 170 after 50 episodes and then gradually stabilizes. For noncooperative caching, with the increase of training episodes, the average reward has stabilized around 100. The cooperative caching strategy not only make the most use of the cache space but also effectively improve the system performance. The advantage of noncooperative caching is that it does not need to consider the content caching status of other edge servers, and the system complexity is low.

Figure 5 shows the comparison of the cache hit rate for different caching strategies. With the continuous increase of training times, the cooperative cache hit rate obtained by the system is stable above 85%. For the noncooperative caching, the system cache hit rate is roughly 10% lower than the cooperative caching. When the training reaches 400 rounds, the hit rate of the noncooperative cache gradually tends to 75%. The reason for the gap is that the noncooperative caching cannot make full use of the cache space, which causes a waste of storage and a low system caching hit rate.

The relationships of the system caching hit rate under different Zipf distributions is shown in Figure 6. When the Zipf distribution parameter is large, it indicates that the vehicle users have more requests for the content with high popularity. Caching the high popularity content is beneficial to the improvement of the system cache hit rate. When the content requests increase, the noncooperative caching is difficult to meet the vehicle requests. The proposed cooperative cache strategy fully considers the cooperation between RSU and MBS, and the system cache hit rate increases significantly.

Figures 7 and 8 show the comparison of the average cache hit rate of different algorithms under the different schemes. With the increase of training times, the average caching hit rate of MBS gradually tends to a stable value above 80%. For the (deep Q network) DQN algorithm, the caching hit rate fluctuates greatly in the first 250 episodes, because DQN is difficult to deal with the complex state information. For the average cache hit rate of RSU, the DDPG algorithm has better performance in the first 250 episodes. After 250 training episodes, the effect is slightly lower than that of the (policy gradient) PG and the DQN algorithm.

Figure 9 presents the cooperative cache performance under the different numbers of vehicles. Compared with PG and DQN based algorithms, the DDPG-based algorithm can bring better benefits to the system and tend to be stable when dealing with the complex environment. At the same time, it also verifies that the DDPG-based algorithm has unique advantage for improving the overall average hit rate.

5. Conclusions

This paper focus on the improving of cache performance in the IoV environment and proposes a V2I-based cooperative caching

strategy. We propose MBS-RSU-vehicle three layer architecture and model the problem as maximizing the average cooperative cache hit rate. The objective function is solved by using the reinforcement learning algorithm based on DDPG. In order to verify the performance of the proposed cache strategy, the effects of cooperative caching and noncooperative are compared under different system parameters. In future work, we will further consider the content transmission delay and use game theory to solve the problem of resource competition between cooperative cache servers.

Data Availability

The data used to support the study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is jointly supported by the National Natural Science Foundation of China (No. 61962036, No. 52069014, No. 61961026), by the Science and Technology Research Project of Jiangxi Provincial Department of Education (No. GJJ190957), and by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

References

- [1] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 80–87, 2018.
- [2] X. Xu, H. Li, W. Xu, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in Internet of Vehicles: a survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.
- [3] B. Jedari, G. Premsankar, G. Illahi, M. D. Francesco, A. Mehrabi, and A. Ylä-Jääski, "Video caching, analytics, and delivery at the wireless edge: a survey and future directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 431–471, 2021.
- [4] Z. Wu and D. Yan, "Deep reinforcement learning-based computation offloading for 5G vehicle-aware multi-access edge computing network," *China Communications*, vol. 18, no. 11, pp. 26–41, 2021.
- [5] H. Wu, J. Zhang, Z. Cai, F. Liu, Y. Li, and A. Liu, "Toward energy-aware caching for intelligent connected vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8157–8166, 2020.
- [6] C. Song, W. Xu, T. Wu, S. Yu, P. Zeng, and N. Zhang, "QoE-driven edge caching in vehicle networks based on deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5286–5295, 2021.
- [7] I. A. Elgendy, W. Z. Zhang, H. He, B. B. Gupta, A. El-Latif, and A. Ahmed, "Joint computation offloading and task caching for multi-user and multi-task MEC systems: reinforcement learning-based algorithms," *Wireless Networks*, vol. 27, no. 3, pp. 2023–2038, 2021.

- [8] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen, "Content popularity prediction towards location-aware mobile edge caching," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 915–929, 2019.
- [9] B. Mao, F. Tang, Z. M. Fadlullah et al., "A novel non-supervised deep-learning-based network traffic control method for software defined wireless networks," *IEEE Wireless Communications*, vol. 25, no. 4, pp. 74–81, 2018.
- [10] M. Chen, Y. Hao, L. Hu, K. Huang, and V. K. N. Lau, "Green and mobility-aware caching in 5G networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8347–8361, 2017.
- [11] J. Luo, F. R. Yu, Q. Chen, and L. Tang, "Adaptive video streaming with edge caching and video transcoding over software-defined mobile networks: a deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1577–1592, 2020.
- [12] R. Xie, J. Wu, R. Wang, and T. Huang, "A game theoretic approach for hierarchical caching resource sharing in 5G networks with virtualization," *China Communications*, vol. 16, no. 7, pp. 32–48, 2019.
- [13] T. Zhang, X. Xu, L. Zhou, X. Jiang, and J. Loo, "Cache space efficient caching scheme for content-centric mobile ad hoc networks," *IEEE Systems Journal*, vol. 13, no. 1, pp. 530–541, 2019.
- [14] X. Huang, K. Xu, Q. Chen, and J. Zhang, "Delay-aware caching in internet-of-vehicles networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10911–10921, 2021.
- [15] P. Shi, F. Gao, S. Liang, and S. Yu, "Multi-model inference acceleration on embedded multi-core processors," in *Proceedings of 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, pp. 400–403, Sanya, China, 2020.
- [16] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: a deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10190–10203, 2018.
- [17] Z. Zhang, J. Dai, M. Zeng, D. Liu, and S. Mao, "Scalable video caching for information centric wireless networks," *IEEE Access*, vol. 8, pp. 77272–77284, 2020.
- [18] Z. Meng, Z. Weicheng, L. Yan, and Z. Lidong, "The information center Internet of Vehicles (IoV) cache service strategy based on mobile edge computing," in *Proceedings of 2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1443–1448, Changsha, China, 2020.
- [19] G. Manogaran, V. Saravanan, and C.-H. Hsu, "Information-centric content management framework for software defined Internet of Vehicles towards application specific services," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4541–4549, 2021.
- [20] X. Peng and H. Garg, "Intuitionistic fuzzy soft decision making method based on CoCoSo and CRITIC for CCN cache placement strategy selection," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 1567–1604, 2022.
- [21] S. Xu, X. Liu, S. Guo, X. Qiu, and L. Meng, "MECC: a mobile edge collaborative caching framework empowered by deep reinforcement learning," *IEEE Network*, vol. 35, no. 4, pp. 176–183, 2021.
- [22] L. Ale, N. Zhang, H. Wu, D. Chen, and T. Han, "Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5520–5530, 2019.
- [23] L. Wang, H. Wu, Z. Han, P. Zhang, and H. V. Poor, "Multi-hop cooperative caching in social IoT using matching theory," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2127–2145, 2018.
- [24] X. Zhang, H. Li, J. Wang et al., "Data-driven caching with users' content preference privacy in information-centric networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 5744–5753, 2021.
- [25] L. Yao, Y. Wang, X. Wang, and W. U. Guowei, "Cooperative caching in vehicular content centric network based on social attributes and mobility," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 391–402, 2021.
- [26] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: machine learning for network edge caching in the big data era," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 28–35, 2018.
- [27] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.
- [28] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the Internet of Vehicles," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 246–261, 2020.
- [29] Y. Qian, Y. Jiang, L. Hu, M. S. Hossain, M. Alrashoud, and M. Al-Hammadi, "Blockchain-based privacy-aware content caching in cognitive Internet of Vehicles," *IEEE Network*, vol. 34, no. 2, pp. 46–51, 2020.
- [30] R. Wang, Z. Kan, Y. Cui, D. Wu, and Y. Zhen, "Cooperative caching strategy with content request prediction in Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8964–8975, 2021.