Distributed Edge Caching for Zero Trust-Enabled Connected and Automated Vehicles: A Multi-Agent Reinforcement Learning Approach

Xiaolong Xu, Xuanhong Zhou, Xiaokang Zhou, Muhammad Bilal, Lianyong Qi, Xiaoyu Xia, and Wanchun Dou

Abstract

Zero Trust model enhances the security of wireless network environments, which is thought to be effectively applicable to Connected and automated vehicles (CAVs). Considering the abundance of real-time data in CAVs and the delay introduced by the data validation of the Zero Trust model, it may result in significant delay when processing real-time data. By caching popular content in advance on edge servers, edge caching can significantly reduce the response delay of real-time data in CAVs. However, achieving low-delay service responses requires ultra-dense deployments of edge servers, which increases the complexity of the wireless network. Therefore, it is challenging to achieve efficient cooperative caching between edge servers in Zero Trust-enabled CAVs. In this article, a Distributed Edge Caching method with Multi-Agent reinforcement learning for Zero Trust-enabled CAVs, named D-ECMA, is proposed. Specifically, a collaboration graph construction method is designed to obtain efficient collaborative relationships. Then a prediction method for the demand of services based on Spatial-Temporal Fusion Graph Neural Networks (STFGNN) is proposed to help edge servers adjust their caching policies. Following, a distributed edge caching method based on Multi-Agent Deep Deterministic Policy Gradient (MADDPG) for Zero Trust-enabled CAVs is designed. Finally, the effectiveness of D-ECMA is demonstrated through comparative experiments.

Introduction

With the development of artificial intelligence, communication networks, smart sensors and other technologies, vehicles are gradually transforming into connected and automated vehicles (CAVs). CAVs enable vehicles to do more than just drive, and provide various intelligent in-vehicle services such as accident detection and driver assistance to enhance traffic intelligence [1]. Thus, CAVs are gradually becoming the cornerstone of future intelligent transportation systems.

In the realm of connected and automated vehicles (CAVs), wireless networks play a pivotal role in facilitating intelligent vehicular services. However, while providing communication and data transmission for smart vehicles, these wireless networks also confront security threats and attacks from various angles. Risks such as malicious intrusions, data leaks, identity spoofing, and network interference loom large, potentially resulting in diminished vehicle system performance, passenger privacy breaches, and even traffic accidents [2]. To ensure the safety and reliability of CAVs, there is an urgent need to develop new models and technologies to effectively address these security challenges. Traditional network security methods are no longer sufficient, particularly considering the specific and real-time requirements of CAVs. Hence, novel security models and technologies are becoming increasingly crucial [3]

Zero Trust model has emerged as a pivotal strategy that challenges conventional assumptions of trust within networks. This model necessitates the continuous validation of users, devices, applications, and data, irrespective of their internal or external origins [4]. The strategy shifts the focus of network security from perimeter defenses to internal controls, in response to the escalating complexity of network threats. Zero Trust model is considered to be a comprehensive network security strategy for the CAVs, contributing to the preservation of the integrity, privacy, and availability of vehicular systems. By integrating the principles of the Zero Trust model, CAVs can effectively address multifaceted security challenges and ensure the security of wireless networks within intelligent vehicular applications. Due to the additional verification steps and permission controls introduced by the Zero Trust model, there is an

Xiaolong Xu is with the School of Software, Nanjing University of Information Science and Technology, China and Jiangsu Province Engineering Research Center of Advanced Computing and Intelligent Services and Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, China.; Xuanhong Zhou is with the School of Software, Nanjing University of Information Science and Technology, China; Xiaokang Zhou is with Shiga University, Japan, and the RIKEN Center for Advanced Intelligence Project, Japan; Muhammad Bilal is with Hankuk University of Foreign Studies, Korea; Lianyong Qi is with China University of Petroleum (East China), China, and with the State Key Laboratory for Novel Software Technology, Nanjing University, China; Xiaoyu Xia is with the with RMIT University, Australia; Wanchun Dou is with the State Key Laboratory for Novel Software Technology, Nanjing University, China.

Digital Object Identifier: 10.1109/MWC.001.2300414

extra delay in processing requests. Simultaneously, the abundance of real-time data in vehicular networks coupled with the data validation and authorization requirements of the Zero Trust model might result in additional latency while handling real-time data, consequently impacting the rapid response to requests. Therefore, Zero Trust-enabled CAVs (Z-CAVs) can effectively address security concerns, but rapid response to requests remains a challenge.

To address the delay in Z-CAVs, edge caching could be taken into consideration. By caching popular content in advance on edge servers (ESs), edge caching enables fast response to service requests and is therefore seen as a key technology to solve the delay problem in Z-CAVs. However, due to the limited storage resources of the ESs in Z-CAVs, how to determine an efficient caching strategy is an important issue. Achieving the coverage of caching services requires ultra-dense deployments, which inevitably entails significant costs and also increases the complexity of the network. As a result, it remains a challenge to achieve efficient cooperative caching across a limited number of ESs in Z-CAVs.

In order to solve the above problems, in this article, a distributed edge caching method with multi-agent reinforcement learning in Z-CAVs is proposed. Specifically, for reducing the complexity of the communication network, a collaboration graph construction method is designed, which extracts the relationships between nodes to obtain the best collaborative relationships. Considering that future demand of services helps adjust the current caching strategy, a demand prediction method based on Spatial-Temporal Fusion Graph Neural Networks (STFGNN) is then proposed to maximize long-term benefits. Finally, a distributed edge caching method based on multi-agent deep deterministic policy gradient (MADDPG) is designed to determine the optimal caching strategy. In particular, we arrange both the demand prediction network and the networks of MADDPG (i.e. actor networks and critic networks) to be trained in the cloud, then the cloud return the parameters of the actor networks to be executed by ESs, resulting in a collaborative edge-cloud framework with centralised training in the cloud and distributed execution at the edge. The main contributions are as follows.

- Design a collaboration graph construction method, which reduces the complexity of the communication network in Z-CAVs and achieves an efficient cooperation mechanism.
- Propose a prediction method for the demand of services based on STFGNN, which helps adjust the caching policy to maximize long-term returns.
- Design a distributed edge caching approach based on MADDPG in Z-CAVs, which minimizes the total system delay by cooperative caching between ESs.
- Verify the superiority of D-ECMA through comparative experiments.

The remaining parts of this article are organised as follows. We illustrate the related work. A framework for multi-agent edge caching in Z-CAVs is presented. We introduce the implementation details of D-ECMA. Comparative experiments are evaluated. We conduct the article.

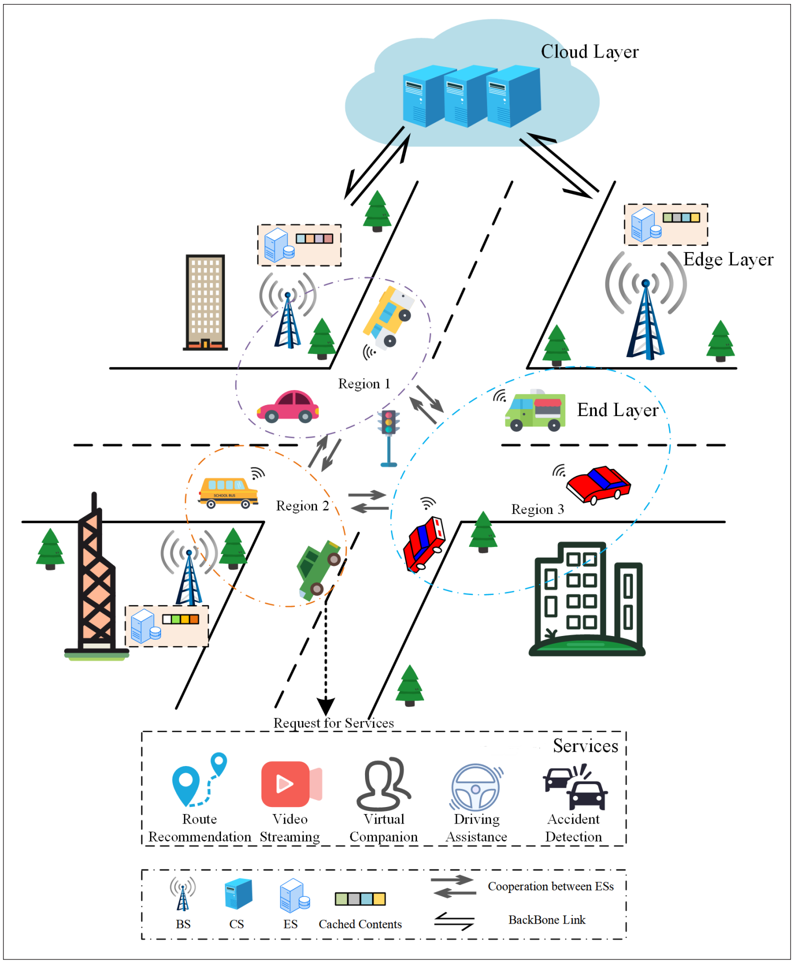

FIGURE 1. The architecture of multi-agent cooperative edge caching in Z-CAVs.

Related Work

The Zero Trust model introduces a novel approach to network security. By emphasizing distrust, continuous validation, and the principle of least privilege, it infuses fresh vitality into security defense strategies. The Zero Trust model posits that all users, devices, applications, and data should be regarded as untrusted, necessitating rigorous validation and authorization in every interaction. This data-centric, boundary-agnostic security philosophy positions the Zero Trust model as an ideal choice for addressing the complexities of modern network environments. Zayed *et al.* introduced a Zero Trust Architecture-based methodology for verifying vehicle owner identity through license plate recognition, enhancing security and trust in inter-vehicle communication within the Internet of Connected Vehicles [5]. Liu *et al.* presented a novel blockchain-enabled solution within a zero-trust framework for secure and trustworthy information sharing in IoT environments, addressing challenges of compromised devices, data privacy, and participant integrity [6].

Single-agent reinforcement learning only considers its own state and reward, hence it may not necessarily maximize the system's reward. In multi-agent reinforcement learning (MARL), each agent considers its own behaviour and that of other agents to maximize the total system reward. Compared to using reinforcement learning for each agent individually, MARL can learn the

Achieving service coverage requires a highly dense deployment of edge devices, which incurs significant costs and increases the complexity of network.

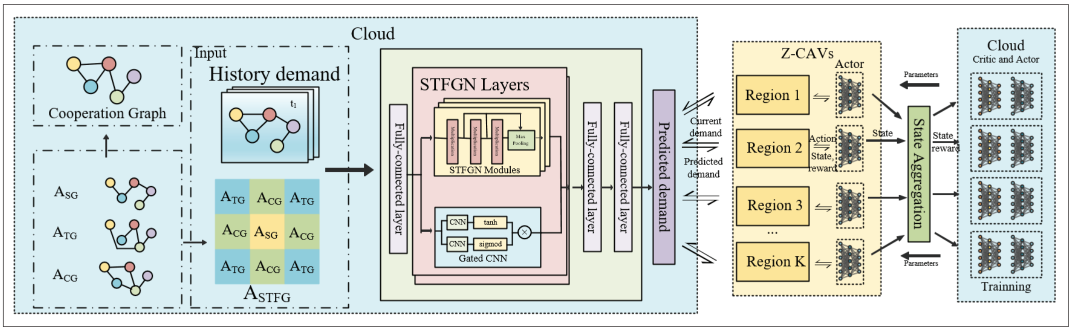

FIGURE 2. The framework of D-ECMA.

cooperative relationship between the agents and therefore has better performance. There is already a large body of research applying MARL to edge caching. Jiang *et al.* [7] first proposed a hierarchical edge caching architecture for CAVs, then extended the traditional reinforcement learning method Q-Learning to a multi-agent system and used a MARL-based algorithm to reduce system delay. Chen *et al.* [8] formulated the edge caching problem as a multi-agent decision problem based on a partially observable Markov decision process, and designed a multi-agent critic-actor framework in which a communication module is designed to aggregate the states of individual BSs. However, most studies learnt global information, resulting in a state dimension that is too high for reinforcement learning methods to converge.

Achieving service coverage requires a highly dense deployment of edge devices, which incurs significant costs and increases the complexity of network. As a result, efficient resource sharing is an important issue for edge caching in Z-CAVs. However, to our knowledge, few studies have considered the use of MARL to solve the edge caching problem in Z-CAVs. Since MARL makes optimal decisions based on the current state of the environment, it is suitable for Z-CAVs where the flow of traffic changes dynamically and user demands are random. Therefore, we propose a MADDPG-based collaborative multi-agent edge caching approach in Z-CAVs. A collaborative graph construction method is added in order to efficiently aggregate information from other edge nodes and not to introduce too high dimensional state spaces. In addition, considering the impact of future demand on caching performance in Z-CAVs, a demand prediction network is designed to optimise caching decision.

Multi-Agent Edge Caching Framework for Z-CAVs

The system framework of multi-agent edge caching in Z-CAVs is shown in Fig. 1, which consists three layers: Cloud layer, Edge layer and End layer. The Z-CAVs could offer services such as route recommendation, video streaming, virtual companion and so on.

- Cloud layer: The Cloud layer consists of a central cloud, assuming that the cloud server has sufficient storage space to cache all content. The Cloud layer and Edge layer are linked via backbone links.
- Edge layer: The Edge layer consists of BSs distributed in different areas of the Z-CAVs, each equipped with a ES. Considering that the storage space of ESs is limited, only some of the content can be cached. BSs are linked to each other via a wireless link and have a specific cooperation relationship with each other to maximize the sharing of caching resources.
- End layer: The End layer consists of different regions in IoV, each with a different content demand and content popularity. Vehicles in each region will send content requests over a wireless link.

Next, we will describe the components of delay in multi-agent cooperative edge caching in Z-CAVs from the following three ways: Local response, Content delivery and Cache replacement.

Local Response

When vehicle requests for a certain content, the request will be received by the BS in this region. The BS will first search the local ES to check if the content has been cached and, if so, send the content directly to the vehicle. Due to the proximity of the BS to the vehicle and the extremely fast transmission rate of wireless network, the delay of local response is usually ignored.

Content Delivery

If local ES does not cache the requested content, then it needs to request content from other ESs or the central cloud. The BS will first send a content request to its own collaborators based on the collaboration graph, and if the requested content is cached by any of the collaborators' ESs, it will be returned via wireless communication between the BSs. The delay incurred in this process is influenced by the state of the channel and the proximity of communication distance. If none of the collaborators' ESs cache the requested content, BS has to request the content from the central cloud via the backbone link, which must be able to fulfil BS's request as the central cloud has cached all the content. However, considering the distance of the central cloud from the BS, a large delay is incurred in the process, which is usually considered as a constant.

Cache Replacement

In addition to the delay of responding to requests, the system should also include the delay of cache replacement. At the beginning of each period, each ES develops a caching policy for the period based on content demand. For content that has been cached in the previous period but is not needed in the current period, the ES can simply discard the content, which does not incur delay.

For content that was not cached in the previous period but is needed in the current period, the BS needs to request them from the central cloud. It is assumed that all requests can be sent to the central cloud at the same time and the largest delay is taken as the delay for one cache replacement. The backbone link will receive requests from all regions at the same time, and this huge amount of data may cause congestion on the backbone link, so the cache replacement strategy should also be efficient.

In summary, we can calculate the delay for each period in each region. In this article, our goal is to minimise the total delay of the system, i.e. the sum of the delay of all periods in all regions.

Design of D-ECMA

In this section, the implementation of D-ECMA is described. Figure 2 shows the framework of D-ECMA. Firstly, we design a method for the construction of collaboration graph. Then, STFGNN is employed to predict the demand. Finally, D-ECMA for Z-CAVs is proposed.

Construction of Collaboration Graph

In order to make cooperative caching between edge nodes more efficient, we have devised a method for collaboration graph construction. Firstly, considering the effect of communication distance on delay, we take the inverse of the distance between any two edge nodes to obtain A_{SG}. Secondly, two nodes with similar demand variation may have a higher likelihood of cooperation, so we used FastDTW from [9] to calculate the temporal correlation of demand between two nodes to obtain A_{TG}. Then, since two nodes with similar request content are more likely to need cooperative caching, we use the calculation in [10] to calculate the content similarity between nodes to obtain A_{CG}. Finally, we average these three matrices and set a threshold. The edges that are smaller than the threshold are cropped and the remaining edges form the collaboration graph. In particular, in order to maintain a stable training environment, we will use the same collaboration graph for several adjacent periods, rather than updating the collaboration graph at the beginning of each period.

Demand Prediction based on STFGNN

Considering the impact of future demand on the current caching strategy, a prediction method for the demand of services based on STFGNN is proposed [11]. First, we combine A_{SG}, A_{TG} and A_{CG} into A_{STFG} to extract the spatio-temporal correlation of demand. Then, we introduce STFGN Layer, the main component of STFGNN. STFGN Layer consists of two modules: STFGN Modules and Gated GNN. STFGN Modules extract the implied spatio-temporal correlations through matrix multiplication of inputs and A_{STFG}, skip connect, maximum pooling and other operations. In particular, we stack multiple STFGN Modules to aggregate more complex spatio-temporal correlations. The STFGN Modules integrate spatio-temporal dependencies via A_{STFG}. However, the spatio-temporal correlations of the nodes themselves are also important, so we introduce the Gated GNN, which uses two independent dilated convolution operations and activates the convolution results

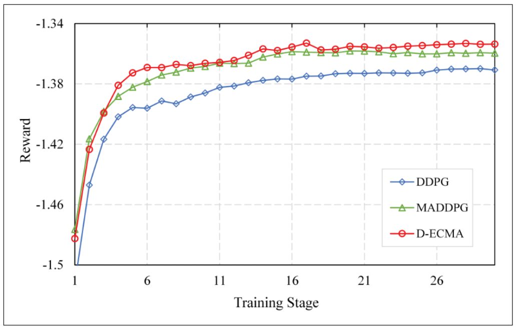

FIGURE 3. The convergence performance of reward under different methods.

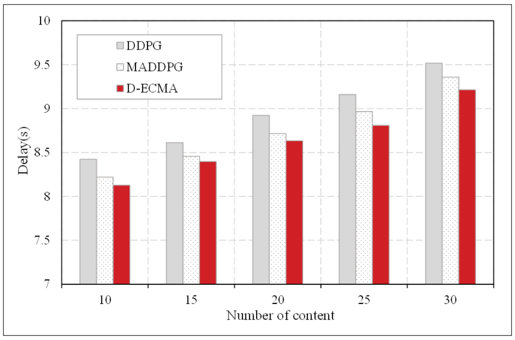

FIGURE 4. Comparison of delay under different numbers of content.

via tanh and sigmoid activation, then multiplies them together. Finally, we sum the outputs of the STFGN Modules and the Gated GNN as the input into the next STFGN Layer. After processing through multiple STFGN Layers, the computed results will be passed through two fully connected layers to obtain the final predicted demand.

Distributed Edge Caching Method with MADDPG

MARL is a machine learning method in which multiple agents continuously interact with the environment to obtain rewards and thus maximize the overall reward. In this part, we combine the MARL method MADDPG [12] with the previously proposed collaboration graph and demand prediction based on STFGNN to obtain D-ECMA. First, we introduce the Markov decision process model:

- State space. Unlike single-agent reinforcement learning which only considers its own state, multi-agent reinforcement learning also considers the state of other agents to maximize the overall reward. Therefore, based on the collaboration graph, we add the state of the collaborators to the state space as well. In addition, as future demand will have an impact on the caching policy, we also add the predicted demand to help the agent consider longer-term rewards. Thus, the state space is designed as: the content requests received by itself and collaborators, the caching policies of itself and collaborators, and the predicted demand.
- Action space. Since different ESs have different storage capacities, using binary encoding (i.e. 1 for caching this content and 0 vice versa) would result in inconsistent dimension of the action space per agent, which is not conducive to convergence. The action space is therefore designed to be the probability that each content will be cached. Suppose an ES can cache

MARL is a machine learning method in which multiple agents continuously interact with the environment to obtain rewards and thus maximize the overall reward.

MADDPG will train an actor network and a critic network independently for each edge node, where the actor network outputs a caching policy based on the local state and the critic network evaluates how good it is to adopt a caching policy in a certain state.

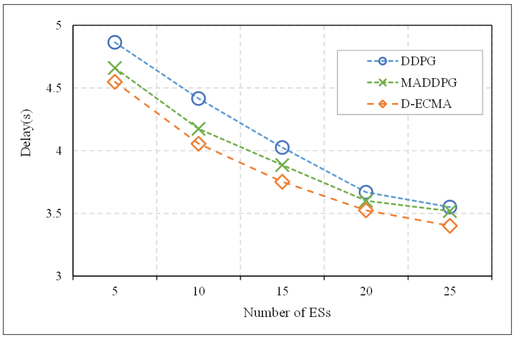

FIGURE 5. Comparison of delay under different numbers of ESs.

up to K content, and after it has obtained the caching probability of each content through the actor network, it selects the largest K content to cache.

- Reward. The goal of this article is to minimize the total system delay, so we set the reward to the opposite of the delay.

MADDPG will train an actor network and a critic network independently for each edge node, where the actor network outputs a caching policy based on the local state and the critic network evaluates how good it is to adopt a caching policy in a certain state. Noteworthy, the input of the actor network is the local state, whereas that of the critic network is the aggregated state. All networks are updated using deterministic policy gradients. In addition, the target network is added to improve the stability of the training and it will be updated using soft updates.

Next we describe the framework of D-ECMA in general terms. The collaboration graph construction and demand prediction network will be deployed in the cloud. Once the collaboration graph is constructed and the predicted demand is available, the central cloud will send this information over the backbone link to the BSs of the Z-CAVs. Then BSs will send the demand of the current moment to the central cloud for subsequent collaboration graph construction and prediction. Each ES will deploy a local actor network and since the input of the actor network is the local state, only the local BS needs to collect the state information and transmit it to the ES for decision making, instead of uploading it to the cloud for decision making, which saves a lot of time. After a fixed period of time, the BSs will send the history experience to the cloud, where the state information will be aggregated according to the collaboration graph. The central cloud will train all networks, and send the parameters of the trained actor networks to ESs for execution. Therefore, D-ECMA has the characteristics of centralized training and distributed execution, and is a edge-cloud collaboration framework in Z-CAVs.

Performance Evaluation

In this section, comparative experiments are carried out to verify the effectiveness of D-ECMA. The dataset used in this article is the in-vehicle user service demand information collected from Nanjing, China, which is used to simulate the services in Z-CAVs. To prove the superiority of our proposed method, Deep Deterministic Policy Gradient (DDPG) [13] and MADDPG, were used for comparison, and delay was chosen as the evaluation criterion. We compared the delay of the three methods over the course of a day, and in addition, comparative experiments were conducted on delay under different numbers of content and different numbers of ESs.

We first set the number of edge nodes to 15, the maximum caching capacity of each ES to 5, and the total number of content to 20. As shown in Fig. 3, we compare convergence performance of reward under different methods. All three methods eventually converged. Since the DDPG only considers its own state and cannot cache cooperatively with other agents, it eventually converges to a worst-case state. Both MADDPG and our D-ECMA take the states of other agents into account, and since D-ECMA aggregates only the agents most likely to cooperate, it eventually converges to the best performance.

We also compare the delay of the three methods for different numbers of content. As shown in Fig. 4, the delay increases with the number of content rises. It is due to the fact that ESs do not have enough storage space to cope with the added content and therefore have to request the service from other ESs or the central cloud, which introduces additional delay. When the amount of content is small, both MADDPG and D-ECMA have low delay through the cooperation of multiple ESs. When the number of content is 15, MADDPG and D-ECMA reduce delay by 1.8% and 2.5% respectively compared to DDPG. However, when the number of content is 30, the state space dimensions of MADDPG explode, so this method struggles to converge to an optimal solution, yielding results that differ from DDPG by only 1.6%. Our proposed D-ECMA not only maintains the best performance consistently, but also reduces the delay by 3.2%-3.8% in the face of a larger amount of content, better solving the problem of exploding state space dimensions.

In addition, we compare the delay under different numbers of ESs. As shown in Fig. 5, With the number of ESs on the rise, ESs can cooperate with more other ESs for edge caching, thus reducing the delay of the system. When the number of ESs is small, MADDPG can learn the cooperation between ESs very well and thus can reduce the delay significantly compared to DDPG. However, as the number of ESs grows, the advantage of MADDPG in reducing delay gradually decreases from 5.47% to 0.91%, which is obviously caused by the explosion of state space dimensions. Our proposed D-ECMA, based on efficient collaboration graph, can still maintain an effective cooperative caching in complex network relations, reducing the delay by 8.14%–3.96% and 2.13%–3.43% compared to DDPG and MADDPG respectively.

Conclusion

In this article, we proposed D-ECMA, a distributed edge caching approach with multi-agent reinforcement learning in Z-CAVs. Specifically, a collaboration graph construction method to obtain collaborative relationships was first proposed. Then, an STFGNN-based prediction method for the demand of services was designed to help ESs adjust their caching strategies to maximize long-term benefits. Following, we proposed an MADDPG-based distributed edge caching method for optimal caching policy. Finally, a collaborative edge-cloud framework with centralised training on the cloud and distributed execution at the edge was introduced. The superiority of D-ECMA was verified through comparative experiments on real datasets. However, the collaboration graph mechanism we proposed does not intelligently explore relationships among individual agents. To further enhance this study, we will continue exploring methods to extract relationships between agents, aiming to improve the effectiveness of multi-agent reinforcement learning.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant (No. 62372242 and 92267104), the Natural Science Foundation of Jiangsu Province of China under grant no. BK20211284.

References

[1] T. Danyang *et al.*, "Performance Measurement Evaluation Framework and Cobenefit/Tradeoff Analysis for Connected And Automated Vehicles (CAV) Applications: A Survey," *IEEE Intelligent Transportation Systems Mag.*, vol. 10, no. 3, 2018, pp. 110–22.

[2] P. M Jawandhiya *et al.*, "A Survey of Mobile Ad Hoc Network Attacks," *Int'l. J. Engineering Science and Technology*, vol. 2, no. 9, 2010, pp. 4063–71.

[3] SR Surya and G Adiline Magrica. A Survey on Wireless Networks Attacks," *2017 2nd Int'l. Conf. Computing and Communications Technologies (ICCCT)*, pp. 240–47, 2017.

[4] VA Stafford, Zero Trust Architecture, Nist Special Publication, 800:207, 2020.

[5] M. Zayed *et al.*, "Owner Identity Verification in the Internet of Connected Vehicles: Zero Trust Based Solution," *Cryptology ePrint Archive*, 2022.

[6] Y. Liu *et al.*, "A Blockchain-Based Decentralized, Fair and Authenticated Information Sharing Scheme in Zero Trust Internet-of-Things," *IEEE Trans. Computers*, vol. 72, no. 2, 2022, pp. 501–12.

[7] K. Jiang *et al.*, "Multi-Agent Reinforcement Learning for Cooperative Edge Caching in Internet of Vehicles," *2020 IEEE 17th Int'l. Conf. Mobile Ad Hoc and Sensor Systems (MASS)*, 2020, pp. 455–63.

[8] S. Chen *et al.*, "Multi-Agent Deep Reinforcement Learning-Based Cooperative Edge Caching for Ultra-Dense Next-Generation Networks," *IEEE Trans. Commun.*, vol. 69, no. 4, 2020, pp. 2441–56.

[9] S. Salvador and P. Chan, "Toward Accurate Dynamic Time Warping in Linear Time and Space," *Intelligent Data Analysis*, vol. 11, no. 5, 2007, pp. 561–80.

[10] F. Wang *et al.*, "Intelligent Video Caching at Network Edge: A Multi-Agent Deep Reinforcement Learning Approach," *IEEE INFOCOM 2020 — IEEE Conf. Computer Communications*, 2020, pp. 2499–2508.

[11] M. Li and Z. Zhu, "Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting," *Proc. AAAI Conf. Artificial Intelligence*, vol. 35, 2021, pp. 4189–96.

[12] R. Lowe *et al.*, "Multi-Agent Actor-Critic for Mixed Cooperativecompetitive Environments," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[13] T. P Lillicrap *et al.*, "Continuous Control with Deep Reinforcement Learning," arXiv preprint arXiv:1509.02971, 2015.

Biographies

Xiaolong Xu (xlxu@ieee.org) received the Ph.D. degree in computer science and technologyfrom Nanjing University, China, in 2016. He is currently a Professor with the School of Software, Nanjing University of Information Science and Technology. His research interests include edge computing, the Internet of Things (IoT), cloud computing, and big data.

Xuanhong Zhou (202083290423@nuist.edu.cn) is currently pursuing the B.S. degree in software engineering with the School of Software, Nanjing University of Information Science and Technology. His research interests include edge computing and the IoT.

Xiaokang Zhou (zhou@biwako.shiga-u.ac.jp) received the Ph.D. degree in human sciences from Waseda University, Tokyo, Japan, in 2014. He is currently an associate professor with the Faculty of Data Science, Shiga University, Japan. From 2012 to 2015, he was a research associate with the Faculty of Human Sciences, Waseda University, Japan. His research interests include ubiquitous computing, big data, machine learning, behavior and cognitive informatics, cyber-physical social systems, and cyber intelligence and security.

Muhammad Bilal [M'16, SM'20] (m.bilal@ieee.org) received the Ph.D. degree in information and communication network engineering from the School of Electronics and Telecommunications Research Institute(ETRI), Korea University of Science and Technology, South Korea. Since 2018, he has been an Associate Professor with the Department of Computer Engineering, Hankuk University of Foreign Studies, South Korea. His research interests include design and analysis of network protocols, cyber security, the IoT, named data networking, and future Internet.

Lianyong Qi (lianyongqi@gmail.com) received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China. He is currently a full professor with the College of Computer Science and Technology, China University of Petroleum (East China), China. His research interests include services computing, Big Data, and Internet of Things.

Xiaoyu Xia (xiaoyu.xia@rmit.edu.au) received his Ph.D. degree from Deakin University, Australia. Currently, he is a lecturer at RMIT University, Australia. His research interests include edge computing, service computing and system security. More details about his research can be found at https://sites.google.com/view/xiaoyuxia/.

Wanchun Dou (douwc@nju.edu.cn) received the Ph.D. degree in 2001. He is a Full Professor with the State Key Laboratory for Novel Software Technology, Nanjing University. To date, he has chaired four National Natural Science Foundation of China projects and published more than 100 articles in international journals and conferences. His research interests include big data, cloud computing, and service computing.