This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



# Cybertwin-driven Federated Learning based Personalized Service Provision for 6G-V2X

Sahaya Beni Prathiba, *Student Member, IEEE*, Gunasekaran Raja, *Senior Member, IEEE*, Sudha Anbalagan, Sugeerthi Gurumoorthy, Neeraj Kumar, *Senior Member, IEEE*, Mohsen Guizani, *Fellow, IEEE*

***Abstract*—The rapid growth of Autonomous Vehicle (AV) technology and the integration of edge computing grasp new challenges along with the ever-increasing mobile internet traffic and services. Tackling such challenges through customized edge computing services is the critical research in 6G Vehicle-to-Everything (6G-V2X) communication. V2X contributes detailed information about the current navigation of vehicles, automatic payments for toll roads, parking fees and other services. With the countless, unique, and personalized service requirements of AVs over computation-intensive applications, exploring the edge resources for the excellent Quality of Service (QoS) provision is the greatest concern. This paper proposes a Federated Learning and edge Cache-assisted Cybertwin (FLCC) framework for personalized service provision in 6G-V2X. Integration of cybertwin in 6G enables the connectivity of the physical system to the digital realm, allowing for adequate instantaneous wireless access. The FLCC jointly considers the edge cooperation and optimizations through the proposed Federated Multi-agent Deep Reinforcement Learning based (FM-DRL) algorithm. The FM-DRL algorithm balances the FLCC's learning accuracy. It minimizes the time and cost by taking the factors such as cybertwin association, training data batch size, and bandwidth. Finally, caching is performed using the Federated Reinforcement Learning-based Edge Caching (FREC) algorithm to obtain the desired datasets required that train the model for providing personalized 6G-V2X services for the AVs. Numerical studies and simulation results reveal that the proposed system outperforms the baseline learning approaches by 17.6%.**

***Index Terms*—6G-V2X, Vehicular Edge computing, Cybertwin, Caching, Personalized Services, Federated Learning, Multi-Agent Deep Reinforcement Learning**

Manuscript received July 1, 2021; revised October 11, 2021; accepted November 28, 2021; Date of Publication xxxx xxx xxxxxx; date of current version xxxxx xxxxx xxxxx. The work of Sahaya Beni Prathiba and Gunasekaran Raja was supported by NGNLab, Department of Computer Technology, Anna University, Chennai 600025, India. The review of this article was coordinated by xxxxxxxxxxxx. Paper no. VTSI-2021-02308. *(Corresponding Author: Gunasekaran Raja).*

Copyright © 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Sahaya Beni Prathiba, Gunasekaran Raja, and Sugeerthi Gurumoorthy are with NGNLab, Department of Computer Technology, Anna University, Chennai 600025, India. (e-mail: sahayabeni@ieee.org; dr.r.gunasekaran@ieee.org; sugeerthiguru@gmail.com).

Sudha Anbalagan is with School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 632014, India. (email: sudha.a@vit.ac.in)

Neeraj Kumar is with the Thapar Institute of Engineering and Technology, Patiala 147004, India, with the Department of Computer Science and Information Engineering, Asia University, Taichung city, Taiwan 40704, and also with the School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, Uttarakhand, India. (e-mail: neeraj.kumar@thapar.edu)

Mohsen Guizani is with Machine Learning Department, MBZUAI, UAE 51133. (email: mguizani@ieee.org)

Digital Object Identifier xxxxxxxxxxxxxx

## I. Introduction

Advancements in Autonomous Vehicles (AVs) have sparked the interest of both research and development communities in recent years. AVs promise significantly improved road safety, fuel efficiency, and high performance. For such considerations, autonomous driving may embark on a new paradigm in the transportation of people and goods enhancement [1]–[4]. Vehicle-to-Everything (V2X) communication meets the ever-changing vehicular applications, connectivity, and customer expectations of automated vehicles while providing those applications with low latency, high bandwidth, good reliability, and better stability [5].

Vast quantities of environmental data from different sources, on the other hand, would severely over-saturate the onboard storage and become too difficult to meet the criteria using conventional computing methods and infrastructures [6], [7]. The combination of Artificial Intelligence (AI) with advanced edge computing approaches increases the processing capacities of edge servers [8], [9]. It helps to reach state-of-the-art efficiency for centralized pattern learning from a large dataset while still fulfilling the real-time needs of time-consuming procedures [10]–[12]. The optimistic vision is that V2X aided by Sixth Generation (6G) will be a critical component of future linked AVs [13], [14].

6G will function in tandem with Machine Learning (ML) to propose a slew of advanced features such as improved context-awareness, self-aggregation, integrated synchronization, and self-configuration [15]. On the other hand, classical enabled systems with a central processing server face crucial privacy and security problems such as a single point of failure, making it impossible to allow universal and stable AI for 6G [16], [17]. Furthermore, conventional centralized ML systems may not be appropriate for ubiquitous 6G systems because of the high overhead associated with centralized data aggregation and distribution.

Federated Learning (FL), a rising decentralized ML solution, interact with devices and train a shared model cohesively. The FL uses the local data and submits the model updates rather than raw data to centralized parameter servers [18]–[21]. However, the lack of reciprocal trust among autonomous cars is an essential factor that may prevent AVs from engaging in the training strategy to ensure edge intelligence via 6G. A significant study has been conducted on the cybertwin to build a safe shared learning process among untrustworthy persons to resolve this concern. The principal function of the cybertwin-based networking environment is the digital reflection of the

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



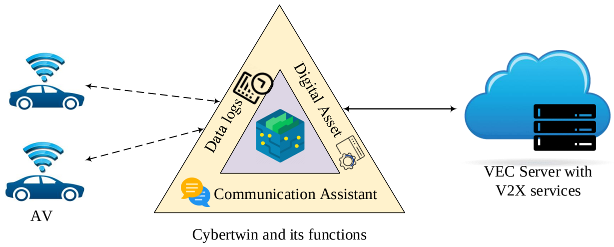


Fig. 1. Cybertwin based vehicular communications with three fundamental functions

end-user in the simulated cyberspace situated at the edge of the network.

To meet several current network architecture standards, the cybertwin has three main objective functions: communication assistance, network data recorder, platform for providing digital assets, as illustrated in Fig. 1. In favor of AVs, the network data recorder capability of the cybertwin, which acts as the digital representation of the edges, will collect and preserve all data of autonomous cars. The operational data from the AV will be converted into a digital asset for marketing via cybertwin, utilizing the digital asset function of the cybertwins. It will, however, hide independent AV identifiers from application service providers, safeguarding the AV's privacy.

Local caching at the network's edge can minimize network load, particularly traffic, using the most frequently requested items in local caches. It is a strategy of balancing data flow and data storage of specific information required [22]. Edge content caching is a critical technique for meeting high traffic demand with limited network resources while maintaining affordable costs and offering vital Quality of Service (QoS) [23]. Content caching is increasingly considered an essential network operation in developing network topologies such as content-centric networking [24], [25].

The primary contributions of the paper are listed below:

1) The Federated Learning and Edge Caching assisted Cybertwin (FLCC) framework provides unsubstantiated and adaptable communication between the AVs and edge servers. It offers personalized 6G-V2X services with low latency and high QoS in the AVs.
2) An optimization challenge is formulated to simultaneously evaluate edge cooperation, communication, and distribution of cybertwin.
3) The FM-DRL algorithm optimizes the performance of cybertwin-assisted edge servers by achieving edge cooperation to reduce the time-cost for determining the optimum solution.
4) An edge caching model through FREC algorithm retrieves the information efficiently for the provision of 6G-V2X personalized services.

The rest of this article is structured as follows. Section II represents the relevant works in recent days. The proposed working environment is presented in section III. In section IV, the edge intelligence and computing of cybertwin is delivered. The proposed edge caching model in vehicular networks is represented in section V. Section VI demonstrates the simulation results and discusses the performance achieved. Finally, Section VII concludes the article.

## II. Related Work

Edge computing is a novel platform that enables cognitive ability and information security [8]. Edge servers act as a pivotal part in reducing latency by acquiring the required processing capacity. Many research teams have investigated similar platforms in light of the exponential growth. Cloudlet is a novel idea, defined as a trustworthy computer or cluster of computers with many resources linked to the network and accessible to users and devices. Also, numerous practical uses are being investigated, particularly in driving. A traditional sector with a vital requirement for reduced latency [9] utilizes collaborative learning set up on the edge servers. The updated edge servers forecast the failures and enhance the mobility of the AVs. One of the criteria that uses edge computing infrastructure has been suggested, with the idea that the expense of driver focus in vehicular applications should be scheduled for improved interaction. However, the public is increasingly appreciating the importance of privacy protection. An offloading strategy based on the limited Markov decision approach for scheduling was given to address possible confidentiality and AV pattern security threats [1].

An autonomous and self-distributed computer can use an FL method to gather activity patterns to predict intrusion successfully [26]. Two approaches for ensuring data integrity for mobile edge computing have been proposed to light the risk of data malfeasance at the edges due to peripheral cyber-attacks or local system failure [5]. The distributed architecture and the improved data privacy assurance provided by FL have received a lot of attention. They have been thoroughly researched for traditional Mobile Edge Computing (MEC) applications like information caching and vehicular data sharing. FL is a distributed ML platform that constantly allows collaborating nodes to upgrade hyper-parameters while maintaining all learning data from the native systems.

FL's key difficulties in 6G communications include (i) increased network costs as a result of frequent model update and aggregate communication rounds, (ii) Protection problems resulting from a varied and different set of collaborating organisations, (iii) Issues with privacy induced by gradient leaks and membership inference attacks, and (iv) Issues with model training and inference reliability in large-scale 6G networks. The coordination mechanism, separated into two stages by the cybertwin-based communication paradigm, is overcoming these difficulties [3]. The cybertwin is the only one that has access to cloud-based apps. The IP address of a cybertwin is a point invisible to the outside network. It also acts as an object identifier that categorizes people, computers, and utilities alike data and software applications. Since this network infrastructure does not need to deal with end-to-end identification, the locator/identifier separation will be simple to enforce. Another impediment to locator split deployment is the lack of an identity verification system that allows the network to ensure that the identifier of the AV is not a forgery.

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



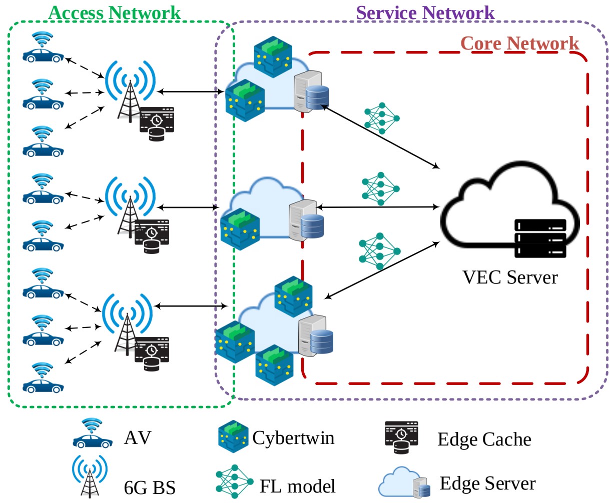


Fig. 2. Federated learning and edge cache-assisted cybertwin architecture for personalised 6G-V2X services

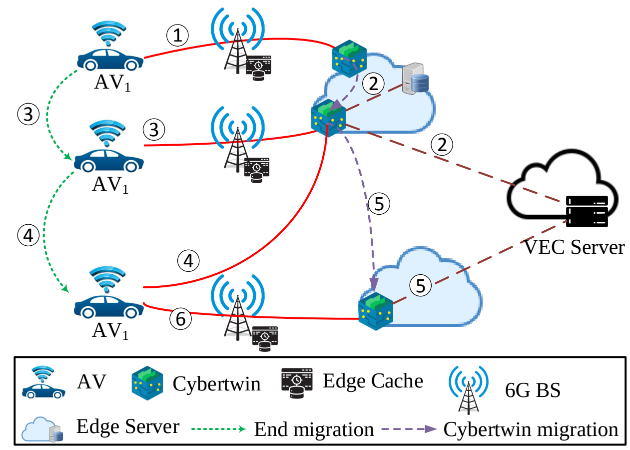


Fig. 3. The process of primary V2X communication and mobility management with cybertwin

The ends of the cybertwin-based contact model will acquire resources from the cybertwin, which will validate the ends using authentication techniques. In contrast, cybertwin collects all AV behavior data, determining the consistency of AV's behavior [27].

Local caching at the network's edge has emerged as a beneficial solution for improving the QoS of AV devices in the Radio Access Network (RAN) [28]. Aside from time, frequency, and space, the local Base Station (BS) cache capacity is a new form of resource. A large amount of required information is retrieved effectively, leading to the increased caching efficiency as the caching unit is placed near the network's edge [29].

The conventional methods do not provide an optimized solution for edge cooperation among cybertwins and a better retrieval of specific data from AVs. Hence, we have proposed a FLCC framework that uses FM-DRL algorithm and FREC algorithm for an efficient edge cooperation and edge caching of data to provide personalized services.

## III. System Model

This section describes the working of FLCC framework to provide the personalized services through cybertwin-assisted edge caching. The FLCC architecture is illustrated in Fig. 2, consisting of three network components: the access network, the service network, and the core network. The access network is responsible for establishing a connection between AVs and the edge servers. The core network establishes connection and communication between VEC server and edge servers. The service network is a logical network that offers V2X services to the AVs. Each AV in FLCC has a unique ID. On receiving the request, the VEC server provides the services to the corresponding AV ID. Interestingly, a BS can be linked to many cybertwins, but a cybertwin can be connected to only one BS. Edge cooperation is required to achieve the maximum utilization of cybertwins and resources. It is computed based on the AVs datasets, the computational capacity of BSs, and the rate of transmission.

The association of cybertwins to the servers is optimized as it is a critical component in determining successful edge cooperation. Moreover, the limited bandwidth available with the BSs is used appropriately to increase the information dissemination of connected edge servers. The proposed FM-DRL algorithm is used to achieve effective edge cooperation among cybertwins to provide personalized 6G-V2X services.

To obtain the personalized service, the cache at the edge server stores the AV information from the BS and provides it to the VEC server. This is achieved by the FREC algorithm that trains the data from the AVs in a distributed manner. It helps to get the specific data regarding the personalized request rather than all the data available. The obtained parameters are then aggregated to provide the personalized service to the AV. When the AV moves from one location to another, the mobility management of cybertwin handles the AV data efficiently and continues to personalized service provision. The diagrammatic illustration of cybertwin-assisted mobility management is illustrated in Fig. 3. The mobility management of cybertwin is discussed as follows. (1) When an AV requests a service, it first connects to its BS-hosted cybertwin. (2) Based on the scheduling of the cloud network management system, the cybertwin receives the partnered service from the edge network and core network. Then, it delivers the AV with essential service data. (3) When an AV relocates, it has to connect to the cybertwin via a new access point. (4) When the AV disconnects from the initial access point, the cybertwin can collect the service data and information from the cache and provide it to the AV when it reconnects to the cybertwin. (5) If the AV travels to a new location, it connects to the original cybertwin, resulting in a lengthier delay. (6) To receive the required service, the cybertwin migrates to a closer edge cloud and establishes a new connection. After the relocation, the AV will be linked to the new cybertwin. Thus, the FLCC framework guarantees high QoS in personalized 6G-V2X service provision.

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



## IV. FEDERATED LEARNING FOR EDGE INTELLIGENCE AND EDGE COMPUTING IN CYBERTWIN

The FLCC encompasses $M$ AVs, $K$ BSs, and a VEC. The AV generates running data and synchronises it with corresponding cybertwins that are present in the BSs. Here $d_m = \{(a_{m1}, b_{m1}), ..., (a_{ms_m}, b_{ms_m})\}$ represents the AV data $m$, where $s_m$ is the size of the data. AVs gather the data $a_m$, and $b_m$ is the label that represents the data $a_m$. In BSs, the cybertwin $m$ of AVs is denoted as $Ct_m$, with an action model $B_m$ and the current data in the static state $d_m$, and real time data in the dynamic state, $Ct_m = (B_m, d_m)$. The data required to operate cybertwin function-oriented applications are $d_m$ and $\gamma$. Synchronizing all raw data to cybertwins increases the connectivity burden and the chance of leakage of information. To train model $B$ from AV data, FLCC employs FL so that AVs in various applications can communicate with one another to transmit and share current data via Vehicle-to-Vehicle (V2V) communications. For edge intelligence, FL is used, which conducts the teaching and learning process collaboratively. The VEC server in the FL paradigm distributes the ML parameters of the model among the BSs for training in each iteration. Then, the BSs evaluate the model using input obtained from the cybertwins. After evaluation, it shares the updated parameter values back to the VEC.

### A. Edge Cooperation in Cybertwin

Each AV that belongs to the BS is associated with the corresponding cybertwin available at the edge server. The update of cybertwins necessitates a significant investment in computational and networking capabilities for synchronizing real-time data and creating a subsequent representation of models. Associating multiple AVs with separate BSs based on their computing capacities and contacting channel states is a great challenge. Thus, obtaining the information log for specific information required is a critical problem in cybertwin. The cybertwins are managed by the BSs, where the training data and computation activities are assigned to BS depending upon their relationship with the cybertwins.

Consider a cybertwin that holds $M$ AVs and $K$ BSs. For any random AV $u_m$, $m \in M$, the objective of the edge is to select the intended BS $n \in B$ to design the cybertwin $Ct_m$ of AV $m$. The association $(Ct_m, BS_n)$ is denoted as $\alpha\ (m,n)$. If $Ct_m$ is associated with BS $n$, then $\alpha(m,n) = d_m$ where $d_m$ is the data size, else $\alpha\ (m,n) = 0$.

A BS can be linked to many cybertwins, but a cybertwin can only be linked to one BS. Edge cooperation $\sum_{n=1}^{K} \alpha\ (m,n) = d_m$ is computed based on the datasets $d_m$ of AVs, the computational capacity of BSs $c_m$, and the rate of transmission $\varepsilon_{m,n}$ between $u_m$ and $\mathrm{BS}_n$, represented as,

$$\alpha\ (m,n) = f(d_m, c_n, \varepsilon_{m,n}) \tag{1}$$

Thus, the edge cooperation problem aims to maximize the resource utilization and the productivity of operating cybertwins.

### B. Federated Learning Model

The FL aims to build a global ML model $B$ by obtaining data from many cybertwins $Ct_m$ rather than sending the initial training sets. The global loss function $f(a,b)$ of the global model $B$ is,

$$\min_{\mu} \frac{1}{M} \sum_{m=1}^{M} \frac{1}{\mathrm{d}_n} \sum_{n=1}^{\mathrm{d}_n} f(\mu, a_{mn}, b_{mn}) \tag{2}$$

The client BSs are responsible for training their local models based on their experiences. Each AV optimizes the loss function using gradient descent and a predetermined learning rate on its local data. The VEC server receives the local model from the AVs and updates the global model by utilizing the FL algorithm. The global model aggregation is denoted as:

$$B = \frac{1}{M} \sum_{m=1}^{M} \mathrm{d}_m \mu_m \tag{3}$$

During the training phase, the BSs use the data acquired from the cybertwins $Ct_m$ to train the local models. A BS contains multiple cybertwins built from AVs, which are based on the coverage area provided by the edge server. Unlike traditional FL, the BSs in FLCC first aggregate their local models from several cybertwins before transmitting them to the VEC, potentially reducing the transmission burden. Since BS has $R_m$ cybertwins, aggregation at BS $m$ is possible.

$$A_i = \frac{1}{R_m} \sum_{m=1}^{M_m} S_{Ct_m} \mu_{Ct_m} \tag{4}$$

where $S_{Ct_m}$ is the data used for the training phase that is obtained from the data of cybertwin $Ct_m$ and $\mu_{Ct_m}$ is the model obtained after the phase of training of $Ct_m$. BS $i$ then transfers $A_m$ to the VEC, and global model updated by the VEC as,

$$B = \frac{1}{K} \sum_{i=1}^{K} A_i \tag{5}$$

Wireless connections are used to relay the local versions of cybertwins to the VEC. VEC gets parameters of the model from all the engaged BSs and updates the global model. As the wireless spectrum is constrained during this phase, message flow is critical for the training dataset to integrate effectively.

### C. Edge Cooperation in Cybertwin: Problem Formulation

While the wireless spectrum is constrained during this phase, message flow is critical for the training dataset to converge effectively. The delay efficiency of FLCC is examined, followed by optimization for edge cooperation in cybertwin. This helps in minimizing the aggregation of machine time-cost while maintaining its necessary learning accuracy. In the formulation of the problem in edge cooperation, considering that the gradient $\nabla f(\mu)$ of $f(\mu)$ is,

$$||\nabla f(\mu_{\tau+1}) - \nabla f(\mu_\tau)|| \leq C||\ (\mu_{\tau+1}) - \mu_\tau|| \tag{6}$$

where $C$ is a positive constant value and $||(\mu_{\tau+1}) - \mu_\tau||$ is the normal of $(\mu_{\tau+1}) - \mu_\tau$. The loss function is highly convex and it is formulated as,

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



$$f(\mu_{\tau+1}) \geq f(\mu_\tau) + (\nabla f(\mu_\tau), \mu_{\tau+1} - \mu_\tau) + 1/2||(\mu_{\tau+1}) \ - \mu_\tau||^2 \tag{7}$$

Major loss functions for FL, such as logic loss functions will fulfill the above assumptions. If (6) and (7) are met, the upper bound of the iterations performed globally is given as,

$$U(\Omega_l, \Omega_g) = \frac{O\left(log\left(1/\Omega_l\right)\ \right)}{1 - \Omega_g} \tag{8}$$

where $\Omega_l$ is the local accuracy $\frac{||\nabla \mathrm{f}(\mu_{\tau+1})||}{||\nabla \mathrm{f}(\mu_\tau)||} \leq \Omega_l, \Omega_g$ is the global accuracy, and $0 \leq \Omega_l, \Omega_g \leq 1$. Assume value of $\Omega_l$ is fixed and the upper bound $\mathrm{U}(\Omega_l, \Omega_g)$ is simplified to $\mathrm{U}(\Omega_g)$= $\frac{1}{1-\Omega_g}$. Here, $T$ is the time of one local testing iteration, $\log(1/\Omega)\mathrm{T}$ is the processing time of one global iteration, and $\mathrm{U}(\Omega_g)\ \mathrm{T}_g$ is the upper limit of cumulative learning time.

*1) Localized training phase in cybertwins:* The time-cost of local training in the $m$ BSs computes the processing efficiency and the scalability. Thus, the time-cost is formulated as,

$$\mathrm{T}_m = \frac{\sum_{n=1}^{R_m} \beta_n S_{Ct_m}}{c_m} c \tag{9}$$

*2) Model aggregation on BSs:* As per (4), BSs combine their local models from different cybertwins. The time required to compute for local aggregation is,

$$T_m^l = \frac{\sum_{n=1}^{R_m} |\mu_n|}{c_m} c_\beta \tag{10}$$

where $|\mu_n|$ is the local model's size and $c_\beta$ is the required CPU cycles for aggregating a unit of data. As all the clients are sharing the same global model, $|\mu_1| = |\mu_2| = ... = |\mu_n| = |\mu_g|$. Thus, the local aggregation produces a time-cost of,

$$T_m^l = \frac{R_m\,|\mu_g|}{c_m} c_\beta \tag{11}$$

*3) Model parameter transmission:* Following the global aggregation, the $m$ BS aggregates the local models, converts them into transactions, and transmit them to the remaining BS. BSs also aid in the transmission of transactions during the broadcast, the required time function is proportional to $log_2 K$, and $K$ is the scale of the BS in the AV network. Hence, the time-cost is,

$$T_m^p = \Theta log_2 K \frac{R_m\,|\mu_g|}{\varepsilon_m} \tag{12}$$

where $\Theta$ is the transmission time-cost factor derived from previous current transmission records. As a result, in comparison to other operations, the time for aggregation can be disregarded. The time-cost analysis of the single iteration considers the previous evaluations, and it is calculated as,

$$T = \max_m \frac{\sum_{n=1}^{R_m} \beta_n S_{Ct_m}}{c_m} c + \max_m \mathrm{O} log_2 K \frac{R_m|\mu_g|}{\varepsilon_m} \tag{13}$$

The increasing AV size in the 6G network, the desire for ultra-low latency connectivity, and the complex network state make it a significant problem to decrease the time required for the training stage in a variety of applications. As reliability and delay are two critical measures for assessing the decision-making abilities of FLCC's cybertwins, the edge cooperation dilemma is investigated to determine the exchange among precision of the training activities and its temporal length. Assigning the cybertwins of different AVs to the remaining BSs for training is a critical problem to resolve. The time-cost should be minimized by reducing the complex computation for the cybertwin edge cooperation. Additionally, maximizing the batch size of the training data $s_j$ of cybertwin $Ct_j$ will increase learning accuracy. It maximizes the cost of learning time to perform the further process. In addition to this, the FLCC framework should divide the bandwidth capacity to increase connectivity reliability. To decrease the overall time-cost of FLCC, edge collaboration policies should be designed vigilantly.

The optimization problem of FLCC is to diminish the time-cost and increase the time efficiency through FL. To formulate this optimization problem, the connection of cybertwins, the outcome of training batchsize, spectrum allotment are considered along with the processing capabilities $c_m$ and channel state $cs_{m,c}$.

The optimization problem is derived as,

$$\min_{R_m, s_n, U_{m,c}} \frac{1}{1 - \Omega_g} T \tag{14}$$

such that,

$$\Omega_g \geq \Omega_{th}, \Omega_g, \Omega_{th} \in (0, 1) \tag{15}$$

$$\sum_{m=1}^{K} R_m = S, R_m \in P, \tag{16}$$

$$\sum_{m=1}^{K} U_{m,c} \leq 1, c \in \sigma, \tag{17}$$

$$s^{\min} \leq S_n \leq S_n^{\min}, \forall j \in P \tag{18}$$

Constraint (16) guarantees that the total number of related cybertwins does not surpass the total dataset size. Constraint (17) ensures that each sub-channel can only be assigned to one BS. Constraint (18) guarantees that each cybertwin has a variety of training batch sizes. (14) is difficult to solve because the objective function has many variables, and the time expenditure of one BS is frequently influenced by the resource circumstances of the remaining BSs (14). To solve this, we use FM-DRL algorithm, which achieves efficient edge cooperation by minimizing the time cost. The working of FM-DRL algorithm for edge cooperation is depicted in Algorithm 1.

### *D. Edge Cooperative Federated Multi-Agent DRL*

The states of the system are determined by the present iteration's states and allocation policies. For obtaining the optimal solution, FLCC considers two things, one is formulating the Markov Decision Process (MDP) in terms of the optimization problem, the other is the consideration of multiple agents in the traditional DRL algorithm.

*1) Federated Multi-Agent DRL Algorithm:* In FM-DRL algorithm, the BSs acting as DRL agents such as AVs, cybertwins, a shared context, the system state $\gamma$, the action $\delta$, and the reward function $r$.

**State space:** The computing capacities contain the state of the environment $c_m$ of BSs, the count of cybertwins $R_m$ on every BS $m$, the size of training data of each cybertwin $S_j$, and the channel state $cs_{m,c}$. The multiple agent states are

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



**Algorithm 1** FM-DRL: Federated Multi-Agent Deep Reinforcement Learning Algorithm for Edge Cooperation

**Input:** Training sample ($\gamma$, $\delta_m$, $\mathrm{K}_m$, $\gamma'$)
**Output:** Action $\delta$ based on reward

```
for each BS K ≥ m do
    Set actors and network
end for
for BS K ≥ m do
    for event E ≥ h do
        Compute cybertwin
        for each time frame τ do
            Examine present state γ
            Compute action δ_m
            Compute reward r_m(τ)
            Transmit to the succeeding state γ_{τ+1} depending on
            actions (δ_1, δ_2, ..., δ_m)
            Save (γ_τ, δ_m(τ), r_m(τ), γ_{τ+1})
            Update critic component, intended network and principal
            actor in the model
        end for
    end for
end for
```

represented as $\gamma\ (\tau)\ =\ (c_m,\ R,\ S,\ cs)$, where for all the agent states, the dimension constitutes a vector state.

**Action space:** Allocation of cybertwins $R_m$ is considered as the action, where $s_m$ is the batch size of the data for training for its cybertwins followed by the bandwidth allocation $\chi_m$. Therefore, $\delta_m(\tau) = (R_m,\ s_m,\ \chi_m)$ is the action obtained. New action decisions $\delta_m(\tau)$ are made by the BS agents $m$ at iteration beginning $\tau$ considering the system state $\gamma\ (\tau)$. The action of the system is $\delta\ (\tau) = (\delta_1,\ ...,\ \delta_i\ ,\ ...,\ \delta_m)$.

**Reward:** Based on the time-cost $T_m$, the reward function of BS $m$ is given as,

$$r_m(\gamma(\tau), \delta_m(\tau)) = -T_m(\tau), \tag{19}$$

where r= $(r_1, ..., r_m)$ is the total reward vector obtained from agents. The time-cost $T$ is obtained from the agents having the maximum time-cost from $\max\{T_1,\ T_2,\ ...,\ T_i\}$.

In FLCC, every DRL agent has a similar reward function. BS agents adjust their behaviors during the training phase to optimize the reward function. In every iteration, the time cost is reduced. Further, the training strategy of BS $m$ is to identify the right policy matching the specific states to the particular actions $\delta_m = \pi_m(\gamma)$. BS $m$ takes the action $\delta_m$ for the state $\gamma$. The objective is to maximize the expected incentive, which is represented as,

$$r_\tau = \sum_m \kappa r_m(\gamma(\tau), \delta_m(\tau)) \tag{20}$$

where $\kappa$ is the rate of discount, $0 \leq \kappa \leq 1$. In the traditional DRL system, it isn't easy to obtain the individual states.

*2) Edge Cooperation using a Federated Multi-Agent DRL Algorithm:* The state and action spaces are extensively high since state variables of the network $cs$ and $cm$ are not discrete values. To solve the formulated MDP problem, FLCC uses the FM-DRL algorithm. This algorithm selects and evaluates behavior using policy-based Actor-Critic (AC) networks.

An identical reward function is found in all the agents of the FM-DRL algorithm. The policy network is then trained by the collaborative collection of the reward functions of all the agents, which further minimizes the time-cost of the framework. Each BS unit's training phase contributes a training set ($\gamma$, $\delta_m$, $\mathrm{K}_m$, $\gamma'$), where $\gamma'$ is cybertwin's state in the next time slot. For the training of actor component $\pi(\gamma_\tau \mid \Omega_{ß})$, and the critic component $\rho(\gamma_\tau,\delta_m|\theta)$ FLCC employs tuples that are collected directly from replay memory, with parameters $\Omega_\pi$ and $\Omega_\rho$. The AC components are then classified as a principal component and intended component. To generate the intended values for the principal component, we have to train the intended components constructed similar to the principal components.

$$\delta_m(\tau) = \pi_m(\gamma_\tau|\Omega_{\pi m}) + \varrho \tag{21}$$

where $\varrho$ is the random noise, $\Omega_{\pi m}$ is the explored policy of edge cooperation. The principal component actor Deep Neural Network (DNN) parameters are updated as

$$\Omega_\pi = \Omega_\pi + \Phi_\pi E[\nabla_{\delta m}\rho(\gamma_\tau, \delta_1, ..., \delta_m|\Omega_\rho)|_{\delta_m=\pi(\gamma_\tau|\Omega_\pi)}.\nabla_{\Omega\pi}\pi(\gamma_\tau)] \tag{22}$$

where $\Phi_\pi$ x represents the actor DNN's degree of learning. For training the major critic components, all the steps in the training process are computed on the contrary way of the gradient function. Thus, the loss function is mathematically formulated as,

$$\Omega_{\rho m} = \Omega_{\rho m} + \Phi_{\rho m}.E[2(b_\tau - \rho(\gamma_\tau, \delta_m|\Omega_{\rho m})).\nabla_\rho(\gamma_\tau, \delta_1, ..., \delta_m)] \tag{23}$$

where $\Phi_{\rho m}$ is the principal critic DNN's learning rate, the intended component's generated intended value is $\mathrm{b}_\tau$. The action is represented as $(\delta_1, ..., \delta_m)$ for the DRL agents.

In DNN, the parameters of the intended action $\Omega_\pi^T$ and the intended critic component $\Omega_\rho^T$ are trained differently. Intended action DNN $\Omega_\pi^T$ then updated by the agent and critic DNN $\Omega_\rho^T$ as

$${\Omega^T}_{\pi_m} = \omega\Omega_{\pi_m} + (1-\omega)\Omega_{\pi_m} \tag{24}$$

$$\Omega^T_{\rho_m} = \omega\Omega_{\rho_m} + (1-\omega)\Omega_{\rho_m} \tag{25}$$

where $\omega$ is the updation rate.

Initialization of the principal actor, critic actor, and the edge cooperation of the DNN is made by the agent. Based on the present state as defined by (21), principal actor DNN of the agent $m$ creates action $\delta_m$. The detected reward $\mathrm{r}_m(h)$ and successive state $\gamma_{(\tau+1)}$ are computed, and the training samples are stored as the tuples $(\gamma_\tau, \delta,\ \mathrm{r}_m\ (h)\ \ , \gamma_{(\tau+1)})$ in the replay memory. Based on (22) and (23), the principal actor and critic DNN are adjusted. (24) is used to update the intended components.

The DRL calculation cost training method is executed offline for many episodes under dynamic conditions. The learned DRL models are extended online to distribute the resources efficiently, lowering time-cost through FL while maintaining the training accuracy. The multi-agent DRL for edge cooperation activities comprises policies of cybertwin association $Q$, size of training data in batches $s$, and policies of bandwidth allocation $\chi$. FLCC uses the actor DNN to create actions due to the vast action and state space. The difficulty of

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



FM-DRL algorithm is found in the actor DNN training process $\Omega_\pi$ and critic DNN training process $\Omega_\rho$.

Assume DNN in FM-DRL consists of $H$ hidden layers, with mean of $H_{\text{avg}}$ neurons in every layers. Hence, the time taken for training a DNN is O($H^2_{avg}$ ). O((K · $H^2_{\text{avg}}$ )·E) ) is the overall time required for making the multi-agent DRL model learn with K agents. As parallel training is performed, the time taken is O($H2avg$.E), which is proportional to the DNN size and the number of episodes.

## V. Federated Learning empowered edge caching for personalized services in 6G-V2X

Huge computational power is necessary for a cybertwin with cognition to choose the best plan dynamically. Hence, the cybertwin agent must be implemented with caution. If the BS agent performs training, then each agent takes a long time, and the BS hinders the sensitive data, particularly in commercial and industrial settings. In addition to this, the distributed data also takes much time for retraining each cybertwin from scratch and convergence of cybertwins becomes challenging. To overcome these limitations, FLCC offers an FL framework based on earlier DDQN methods. FLCC is a decentralized model and the federated agents develop a shared predictive model. The training data in AVs are compared using ML with the existing data stored in the cloud. BS distributes the first input parameters to all the AVs generated by the pre-processing procedure. The AV subsequently transmits the near local variables for participating in the successive iteration of global training to the BS. The BS repeatedly collects the revised local data, and the updated model is constantly transmitted to agents. Each AV computes its local update $\nu_n^{\tau+1}$ after the initial parameter $\nu^\tau$ is distributed. The BS computes the total loss function globally using the parameters collected and acts as an edge aggregator. This process is repeated until BS reaches a point of convergence.

Consider a wireless system model with a BS and $\Upsilon$ AVs having local datasets $D = \{D1, D2, ..., Dn, ...Ds\}$. The learning problem observes the objective parameters $\nu$ and the loss function f($\nu$). $f_m(\nu) = \frac{1}{2} min\ _k \|a_m - \nu_k\|^2$ , where $\nu = [\nu_1, \nu_2\ \ldots, \nu_3]$ is an known loss function for K-means, $f_i(\nu) = \frac{1}{2}\left\|b_m - \nu^T a_m\right\|^2$, $b_m \in$ R for linear regression and $f_m(\nu) = \frac{con}{2}\ \|\nu\|^2 + \frac{1}{2}\max\left\{0,\ 1\ -\ b_m\nu^T a_m\right\}^2$ for Support Vector Machine (SVM). The loss function and the local training problem for the local dataset $D_n$ at AV $n$ is given by,

$$F_m(\nu) = \frac{1}{D_n}\sum_{N=1}^{\Upsilon_S} f_n(\nu) \tag{26}$$

$$\nu_n^\tau = \arg \min_{\nu_n \in R^d}\ F_n\left(\nu_n \mid \nu_n^{\tau-1}\right) \tag{27}$$

The global loss function is given as,

$$F(\nu) = \frac{\sum_{n\in D}\ f_n(\nu)}{D} = \frac{\sum_{n=1}^{N_s}\ D_n F_n(\nu)}{D} \tag{28}$$

where $D = \sum_{n=1}^{N_s}\ D_n$ and the learning problem is to find $\nu^* = F(\nu)$

Due to the difficulty of the local training problem, the general solution of (28) is impossible to obtain. A distributed algorithm is required for obtaining the general solution of (28). The working of FREC for the personalized service provision for AVs in 6G-V2X is given in Algorithm 2.

### A. Gradient Descent Algorithm

The training issues (28) are solved by using the gradient descent algorithm. For each AV $n$, $\nu_n^\tau$ is the local parameter with the iteration $\tau$, where $\tau = 1, 2, ...T$.

Moreover, initial values are assigned by the pre-trained parameters to the local parameters at $\tau = 0$. Every AV $j$ calculates the parameters for $\tau > 0$ as,

$$\nu_n^{\tau+1} = \nu_n^\tau - \varsigma\nabla F_n(\nu^\tau) \tag{29}$$

where the gradient step size is $\varsigma \geq$. $\nu^{\tau+1}$ is the global parameter that updates its value at BS after $T$ iterations

$$\nu^{\tau+1} = \frac{\sum_{n=1}^{\Upsilon_s}\ D_n\nu^{\tau+1}}{D} \tag{30}$$

For the next iteration at the local BS, the updated global parameter is used.

### B. Convergence Analysis

The convergence analysis problem of (28) is described in this section. The optimal solution is represented as $\nu_*$. Assume that for all $m$, $f_n(\nu)$ is convex;

$f_n(\nu)$ is L-smooth, i.e $f_n\left(\nu'\right) \leq f_m(\nu) + \nabla f_i(\nu) = \left(\nu' - \nu\right) + \frac{L}{2}\nu - \nu'^{\,2}$, for $\forall\ \nu$ and $\nu'$.

**Algorithm 2** FREC: Federated Reinforcement Learning based Edge Caching

**Input:** Pretrained parameters $\nu_n^\tau = \nu_n^\tau$
**Output:** Global model $\nu^{\tau+1}$

1: Initialization: $\nu_n^\tau = \nu_n^\tau$; Pre-trained parameters $\nu_n^\tau$ from cybertwin.
2: **for** $\tau = (1, 2, ...T)$ **do**
3: **for** an AV $n$ **do**
4: Calculate local model
5: Assign $\nu_n^{\tau+1} = \nu_n^\tau - \varsigma\nabla F_n(\nu^\tau)$
6: Estimate the convergence using (32)
7: **return** $\nu_n^{\tau+1}$
8: **end for**
9: Send $\nu_n^{\tau+1}$ to BS
10: **for** each BS **do**
11: Update $\nu^{\tau+1} = \frac{\sum_{n=1}^{\Upsilon_s}\ D_n\nu^{\tau+1}}{D}$
12: Store $\nu_m^{\tau+1} = \nu^{\tau+1}$
13: **end for**
14: Distribute $\nu_m^{\tau+1}$ to AVs from $n$ BS
15: **end for**

The linear regression feasibility is generated to update the FL rules. Assume $f(\nu)$ is L-smooth. Consider $\varsigma_t$ = 1/L and $\nu*$ = arg min($\nu$); thus,

$$\nu_\tau - \nu_* \leq \left(1 - \frac{\varpi}{L}\right)^\tau \nu_1 - \nu_* \tag{31}$$

where $O(\mu) = \frac{L}{\varpi}\log\left(\frac{\nu_1 - \nu_*}{\varpi}\right)$ is the gradient dispersion. The illustration of parameter distribution $\nu_m$ in each AV is represented by the gradient dispersion.

The convergence is represented as:

$$[f(\nu_\tau) - f(\nu_*)] \leq \mu^\tau\left[\Delta^\tau(f(\nu_*))\right] \tag{32}$$

Thus, $f(\nu)$ is bounded where $[\Delta^\tau(f(\nu_*))] = f(\nu_\tau) - f(\nu_*)$.

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



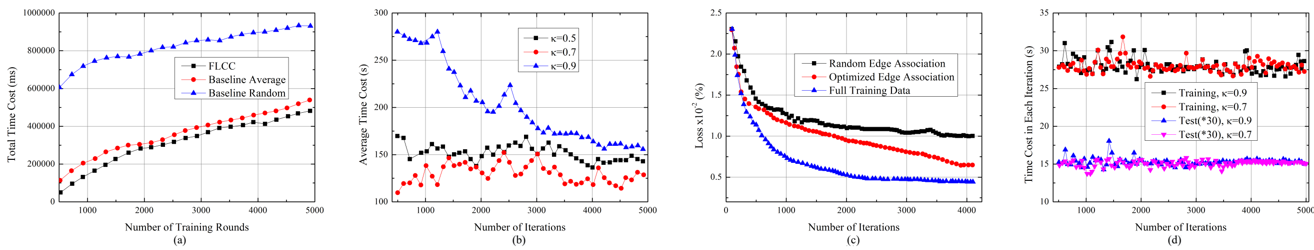

Fig. 4. Performance analysis of FLCC framework. (a) Analysis of total time-cost vs. training rounds. (b) Analysis of average time-cost. (c) Analysis of loss rate. (d) Analysis of time-cost in each iteration.

The local BS $n$ assign pre-trained parameters $\nu_n^\tau$, if the DRL learning process is initiated by the BS. The AV $n$ receives the parameters $\nu_n^\tau$ and the training process utilizes the local updates. Till convergence, this process calculates the updated parameters $\nu_n^{\tau+1}$. Then, the local BS $n$ receives the updated parameters for aggregation.

### C. Personalized V2X Service Provision to the AVs

The cache policy optimization problem of the FLCC to provide personalized V2X services is mathematically formulated as follows,

$$\max \Psi^{\Omega}\left(\gamma_i\left(t\right)\right), \forall t \tag{33}$$

Such that,

$$\sum_{j=1}^{J} \xi_j \zeta_{i,j}\left(t\right) \le \eta_{C_i} \forall e_i, \forall t$$

where, $\Psi^{\Omega}\left(\gamma_i\left(t\right)\right)$ is defined as the aggregated current state-action value function at a particular time $t$, $J$ is defined as the set of contents that are cached, $\xi_j$ is represented as size of the cached content, $\zeta_{i,j}\left(t\right)$ is denoted as the caching state action at that particular time $t$, $C_i$ is the total amount of cache that can be accommodated in the edge server, $\eta$ is the number of nearby edge servers, $e_i$ denotes the edge server. Thus, the objective of the caching mechanism incorporated is to maximize the expected long-term of the system to provide personalized V2X services.

The services of the edge server $e_i$ are represented as,

$$x_i = \frac{r^*_{e_i\dot{\varepsilon}} - \pi\left(r_{e\dot{\varepsilon},\widetilde{j}}\right)}{r_{e\dot{\varepsilon}}} \tag{34}$$

where $x_i$ denotes the services of edge server $e_i$ associated with the BS, $r_{e_i\dot{\varepsilon}}$ is the evaluated rewards for the current personalized V2X service $\dot{\varepsilon}$, $\widetilde{j}$ is the previous personalized V2X service, and $\pi\left(r_{e\dot{\varepsilon},\widetilde{j}}\right)$ is the reward pair function for the personalized service of edge server $e_i$ for evaluating the QoS.

Thus, the objective of providing personalized V2X services with high QoS can be represented as,

$$\max\left\{x_i\right\} = \max\left\{\frac{r^*_{e\dot{\varepsilon}} - \pi\left(r_{e\dot{\varepsilon},\widetilde{j}}\right)}{r_{e\dot{\varepsilon}}}\right\} \tag{35}$$

$$\max\left\{x_i\right\} = 1 - \frac{1}{r^*_{e\dot{\varepsilon}}}\min\left\{\pi\left(r_{e\dot{\varepsilon},\widetilde{j}}\right)\right\} \tag{36}$$

The BS distributes the global parameter to AV after aggregation of distributed parameters received from local AVs. The two main processes of FL are to train the system distributively at the AVs and aggregate the BS parameters.

## VI. Results and Discussions

The performance of the FLCC framework is analyzed based on the time-cost and loss rate. Edge caching performance for aiding the personalized V2X services using the proposed FREC algorithm is analyzed by comparing and contrasting the existing algorithms such as Centralized, Least Frequetly Used (LFU), Least Recently Used (LRU) and First In First Out (FIFO) algorithms [30]. Further, the performance analysis of the personalized service provision through cybertwin-assisted edge servers using FM-DRL algorithm is examined based on the deadline of services, requested resources, number of AVs, and computing resources required for each AV.

### A. Experimental Setup

The experiment is carried out using the real world dataset for evaluating the performance of the FLCC framework. The real edge computing cases like traffic flow monitoring is simulated with 5 BSs and their corresponding edge servers, 1 VEC, and 100 AVs. Within the coverage area of each BSs, the AVs are distributed randomly. For every AV, the CIFAR10 dataset is shuffled and assigned. Thus, training data is identically distributed to FL. The frequency of the 5 BSs are 3.5GHz, 2.7GHz, 4.5GHz, 3.4GHz, and 3.5GHz. The BSs and VEC have the transmission power of 34dBm and 42dBm, respectively.

### B. Performance analysis of FLCC framework

We compare the FLCC effectiveness with the existing algorithms such as Baseline Average and Baseline Random [31]. The Fig. 4(a) shows the latency analysis in the higher range training rounds. The baseline method uses heuristics, simple summary statistics, randomness, or ML to create a dataset prediction. These predictions provide the required metrics of comparison and used to measure the baseline's performance [32]. The lower latency of the FLCC proves that the total time-cost is minimal when compared to the other algorithms. Hence, the FLCC optimally allocates resources with minimal latency, and thus efficiency is improved.

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



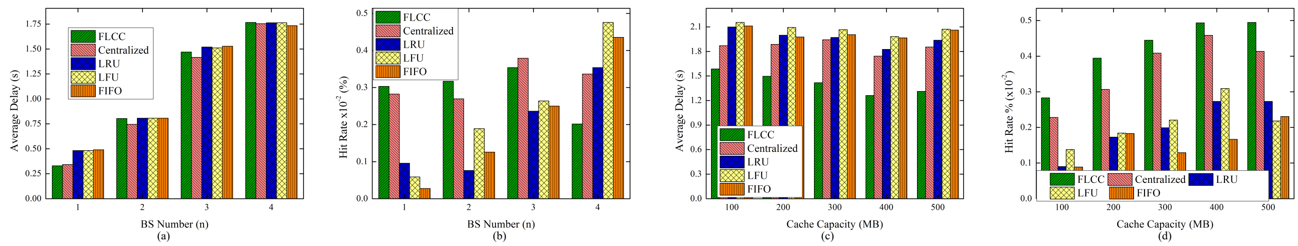


Fig. 5. Performance analysis of edge caching for aiding personalized 6G-V2X services. (a) Average delay vs. number of BS. (b) Hit rate vs. number of BS. (c) Average delay vs. cache capacity. (d) Hit rate vs. cache capacity.

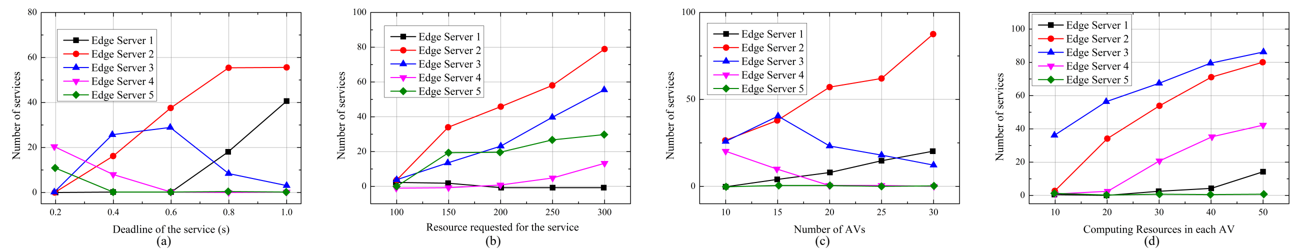


Fig. 6. Performance analysis of personalized service provision through cybertwin-assisted edge servers. Number of services served versus, (a) Service Time. (b) resources utilised. (c) number of AVs. (d) computing resources in each AVs.

In Fig.4(b), the average time-cost of FLCC to the number of iterations is plotted for different discount factors $\kappa$ such as 0.5, 0.7, and 0.9. When the number of iterations increases, the cost of $\kappa$ converges towards a single point; thereby, the cumulative record of the FM-DRL model is increased by the cumulative process. Also, it has minimized the FLCC's time-cost and when $\kappa$ =0.9, FLCC achieves best performance with lowered time-cost.

The learning performance loss is compared with optimal edge association and full training data learning algorithms in the Fig. 4(c). From the graph, it is observed that the well trained data has lower loss rate. Thus, the fully trained algorithm is more efficient when compared with the random edge association algorithm. The learning loss of FLCC is reduced in the fully trained data, and the adaptability towards 6G communication is enhanced.

The iterative exploration for training takes much time. Hence, when related to the testing, the training phase consumes more time for FLCC. In Fig. 4(d), time-cost increases with larger discount factor $\kappa$. Because, in policy training $\kappa$ consume more computation, leading to a larger time-cost.

### *C. Performance analysis of edge caching for providing personalized 6G-V2X services*

The network performance for edge caching under various BS numbers is represented in Fig. 5. The FLCC algorithm's network performance varies based on the BS number. In Fig. 5(a), When the BS number is one, optimal performance is achieved. However, the same diminishes with the increasing BSs since the volume of information that is exchanged between AV and BS increases, resulting in high communication costs. Fig. 5(b) shows the comparative analysis of hit rate, and it is observed that the FLCC algorithm outperforms the centralized approach by 30 percentage points when the BSs ranges 2. Other algorithms outperform the FLCC technique if the range of BSs is between 3 and 4. The models are unaware of the actual requesting actions of the AV, and perhaps BS signals improve performance as more relevant material is cached and more BS signals are involved.

In Fig. 5(c), the network's performance based on various cache potentials is investigated. The figure shows that the FLCC algorithm outperforms the other algorithms in terms of hit rate. Furthermore, due to the increased information exchange, the FLCC algorithm's average delay performance is low. In Fig. 5(d), the FLCC algorithm enhances the hit rate performance by 5, 27, 25, and 26 percentage points, compared with the centralized, LFU, LRU and FIFO, respectively.

### *D. Performance analysis of personalized service for cybertwin assisted edge server*

The performance of different edge serves with varying resources of computing that offers 6G-V2X personalized services is investigated in Fig. 6. The computing resources of the five edge servers are considered as 33, 42, 57, 71, and 99, respectively.

The evaluation of the number of services provided by each edge server with the variation in the service deadline is represented in Fig. 6(a). The variations in the services of edge servers observed are due to the different computing costs. Fig. 6(b) represents the number of services and resources each edge server serves for each resource demand. It is observed from the figure that edge servers 2 and 3 offers more resources. This is due to the availability of low and high resources of the edge servers. It is observed from the figure that edge servers 2 and 3 offer more resources. This is due to the availability of low and high resources of the edge servers. It is observed

0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3133291, IEEE Transactions on Vehicular Technology



from Fig. 6(c), when the number of AVs increases, the services offered are increased in edge server 2 and decreased in edge server 3. The variations in offering the services are due to the probability of successful service provision. Fig. 6(d) shows the comprehensive analysis from the number of services rendered by the edge servers when computing resources owned by each AV are changed. It is inferred that, with the increase in the AV's resources, the services offered by edge server 1 increase, and the services offered by edge server 5 decrease.

## VII. Conclusion

In this paper, a novel FL and caching-based cybertwin model called FLCC is introduced. The FLCC's operating efficiency is enhanced through the proposed FM-DRL and FREC algorithms for effective edge cooperation to provide personalized 6G-V2X services. The issue of connectivity involving cybertwins in the BSs and edges is framed, and a feasible strategy is discovered using the FM-DRL algorithm. Alongside, a cache-enabled FL system (FREC) is designed for retrieving the data to provide personalized services. Based on the numerical findings from a real-case dataset, the FLCC model effectively minimizes learning latency and offers good learning convergence.

## References


[1] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1272–1288, 2019.

[2] S. Anbalagan, A. K. Bashir, G. Raja, P. Dhanasekaran, G. Vijayaraghavan, U. Tariq, and M. Guizani, "Machine learning-based efficient and secure rsu placement mechanism for software defined-iov," *IEEE Internet of Things Journal*, 2021, doi: 10.1109/JIOT.2021.3069642.

[3] Q. Yu, J. Ren, Y. Fu, Y. Li, and W. Zhang, "Cybertwin: An Origin of Next Generation Network Architecture," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 111–117, 2019.

[4] S. B. Prathiba, G. Raja, A. K. Bashir, A. A. Alzubi, and B. Gupta, "SDN-assisted Safety Message Dissemination Framework for Vehicular Critical Energy Infrastructure," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[5] W. Tong, B. Jiang, F. Xu, Q. Li, and S. Zhong, "Privacy-Preserving Data Integrity Verification in Mobile Edge Computing," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 1007–1018.

[6] S. Anbalagan, D. Kumar, G. Raja, and A. Balaji, "SDN Assisted Stackelberg Game Model for LTE-WiFi Offloading in 5G Networks," *Digital Communications and Networks*, vol. 5, no. 4, pp. 268–275, 2019.

[7] S. B. Prathiba, G. Raja, S. Anbalagan, R. Narayanan, and K. B. Venkata Karthik, "SOSChain: Self Optimizing Streamchain for Last-Mile 6G UAV-Truck Networks," ser. 6G-ABS '21. New York, USA: Association for Computing Machinery, 2021, p. 19–24, doi: 10.1145/3477084.3484952.

[8] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 247–257, 2019.

[9] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative Learning of Communication Routes in Edge-Enabled Multi-Access Vehicular Environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1155–1165, 2020.

[10] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, 2020.

[11] S. B. Prathiba, G. Raja, K. Dev, N. Kumar, and M. Guizani, "A Hybrid Deep Reinforcement learning for Autonomous Vehicles Smart-Platooning," *IEEE Transactions on Vehicular Technology*, 2021, doi: 10.1109/TVT.2021.3122257.

[12] G. Raja, P. Dhanasekaran, S. Anbalagan, A. Ganapathisubramaniyan, and A. K. Bashir, "SDN-Enabled Traffic Alert System for IoV in Smart Cities," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*.

[13] J. He, K. Yang, and H.-H. Chen, "6G Cellular Networks and Connected Autonomous Vehicles," *IEEE Network*, pp. 1–7, 2020.

[14] C. D. Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021.

[15] J. R. Bhat and S. A. Alqahtani, "6G Ecosystem: Current Status and Future Perspective," *IEEE Access*, vol. 9, pp. 43 134–43 167, 2021.

[16] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-Intelligence-Enabled Intelligent 6g Networks," *IEEE Network*, vol. 34, no. 6, pp. 272–280, 2020.

[17] S. B. Prathiba, G. Raja, and N. Kumar, "Intelligent Cooperative Collision Avoidance at Overtaking and Lane Changing Maneuver in 6G-V2X Communications," *IEEE Transactions on Vehicular Technology*, 2021, doi: 10.1109/TVT.2021.3127219.

[18] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated Learning for Ultra-Reliable Low-Latency V2V Communications," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.

[19] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated Learning for 6G Communications: Challenges, Methods, and Future Directions," *China Communications*, vol. 17, no. 9, pp. 105–118, 2020.

[20] J. Posner, L. Tseng, M. Aloqaily, and Y. Jararweh, "Federated Learning in Vehicular Networks: Opportunities and Solutions," *IEEE Network*, vol. 35, no. 2, pp. 152–159, 2021.

[21] S. B. Prathiba, G. Raja, S. Anbalagan, K. Dev, S. Gurumoorthy, and A. P. Sankaran, "Federated Learning Empowered Computation Offloading and Resource Management in 6G-V2X," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.

[22] L. Wang, K.-K. Wong, S. Lambotharan, A. Nallanathan, and M. Elkashlan, "Edge Caching in Dense Heterogeneous Cellular Networks with Massive Mimo-Aided Self-Backhaul," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6360–6372, 2018.

[23] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, D. Niyato, and D. I. Kim, "Distributed deep learning at the edge: A novel proactive and cooperative caching framework for mobile edge networks," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1220–1223, 2019.

[24] W. Jiang, G. Feng, S. Qin, and Y. Liu, "Multi-agent reinforcement learning based cooperative content caching for mobile edge networks," *IEEE Access*, vol. 7, pp. 61 856–61 867, 2019.

[25] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, "Cooperative edge caching in user-centric clustered mobile networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1791–1805, 2017.

[26] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏot: A Federated Self-Learning Anomaly Detection System for Iot," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 756–767.

[27] Q. Yu, J. Ren, H. Zhou, and W. Zhang, "A Cybertwin Based Network Architecture for 6G," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, 2020, pp. 1–5.

[28] J. Cui, L. Wei, H. Zhong, J. Zhang, Y. Xu, and L. Liu, "Edge Computing in VANETs-An Efficient and Privacy-Preserving Cooperative Downloading Scheme," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1191–1204, 2020.

[29] 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.

[30] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep Reinforcement Learning-based Edge Caching in Wireless Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 48–61, 2020.

[31] T. Hisamitsu, Y. Niwa, and J. Tsujii, "A method of measuring term representativeness-baseline method using co-occurrence distribution," in *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*, 2000.

[32] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection–a new baseline," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6536–6545.


0018-9545 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIJING UNIVERSITY OF POST AND TELECOM. Downloaded on February 08,2022 at 03:03:45 UTC from IEEE Xplore. Restrictions apply.