Neural Network-Based Game Theory for Scalable Offloading in Vehicular Edge Computing: A Transfer Learning Approach

Juan Zhang, *Member, IEEE*, Yulei Wu, *Senior Member, IEEE*, Geyong Min, *Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*— With the unprecedented scalability issues rising in vehicular edge computing (VEC), we argue in this paper that the scalability, along with the remarkable growth of demands for offloading, should be integrated into the modelling for effective offloading decision-making strategies requested by a large number of vehicles. A two-stage game-theory model can depict offloading decision-making strategies by considering both the revenue of network operators and the cost of VEC users. However, heuristic processes of solving such models show significant limitations in terms of high computational complexity and energy consumption due to the changing VEC environment. Therefore, our objective in this study is to solve the game-theory model efficiently and achieve scalable offloading for the changing VEC environment. We first develop a two-stage game-theory model for the offloading decision-making strategy for VEC, by which an operator's revenue, energy consumption and latency are considered. Then a neural network (NN) model is designed to learn the predicted behaviours of the established game-theory model for offloading decisions in a more efficient manner. After that, a feature-based transfer learning algorithm is proposed for scalable offloading optimization under unseen VEC environments. Experimental results show that the proposed NN can significantly improve the efficiency of solving the game theory model, and the developed transfer learning approach can effectively achieve the scalability of offloading decisions in a changing VEC environment. The results demonstrate that the accuracy of the proposed transfer learning approach is 37% higher than that of several state-of-the-art algorithms, and the runtime halves.**

***Index Terms*— Game theory, mobile edge computing, neural networks, offloading, scalable optimization.**

Manuscript received 12 February 2023; revised 1 September 2023 and 5 December 2023; accepted 16 December 2023. Date of publication 16 January 2024; date of current version 2 July 2024. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/R030863/1. The Associate Editor for this article was S. Wan. *(Corresponding authors: Yulei Wu; Geyong Min.)*

Juan Zhang was with the Department of Computer Science, University of Exeter, EX4 4QF Exeter, U.K. She is now with the Department of Computer and Information Sciences, Northumbria University, NE1 8ST Newcastle upon Tyne, U.K. (e-mail: juan.zhang@northumbria.ac.uk).

Yulei Wu was with the Department of Computer Science, University of Exeter, EX4 4QF Exeter, U.K. He is now with the Faculty of Engineering and the Bristol Digital Futures Institute, University of Bristol, BS8 1UB Bristol, U.K. (e-mail: y.l.wu@bristol.ac.uk).

Geyong Min is with the Department of Computer Science, University of Exeter, EX4 4QF Exeter, U.K. (e-mail: G.Min@exeter.ac.uk).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TITS.2023.3348074

I. Introduction

With the rapid development of autonomous vehicles (AVs), vehicular networks and mobile edge computing (MEC), vehicular edge computing (VEC) has attracted significant attention for its service provisions in driving safety, convenience, and efficiency. Offloading in VEC plays an important role because of the significant influence on the energy consumption of AVs [1], the performance of VEC applications, the revenue of network operators, and the cost of AVs [2], etc. Many offloading problems in VEC have been studied, but challenges on scalability still exist due to the considerable increase in the number of AVs and time-varying changes in VEC environments [3].

Game theory has been widely adopted to tackle the optimization problems of offloading in VEC. Solving game theory models usually requires a large number of iterations, resulting in high computational complexity and energy consumption. Many studies that resorted to heuristic knowledge to reduce the computational complexity in the solving process of game theory models have been successful in given VEC scenarios [4]. However, if the scenario changes, game theory models will need to be solved again. For example, the time-varying number of AVs leads to changes in VEC environments, and thus the latency, energy consumption, and cost of VEC systems may not meet the requirements of supporting AV services such as object tracking through the already-solved game theory models. Existing solutions are therefore not scalable for the complex and changing VEC environments, calling for scalable solutions for game theoretical offloading models in VEC. The neural network (NN) is a potential candidate to overcome the high computational complexity and energy consumption deficiencies of traditional methods of solving game theoretical models because of the nature of approximating any functions with a black box [5]. By using NN, the solving process of a game theory model can be captured and the probability distribution of the process datasets for offloading decisions can be fitted and further inferred [6].

To further enhance the scalability of NN-based solutions for game theoretical offloading models in new and unseen VEC environments, the well-known transfer learning paradigm can be explored for scalable offloading optimization in VEC [7]. The limitation of data acquisition of AVs in the changing VEC environment makes it difficult to obtain sufficient data for the development of learning-based methods, leading to a dilemma of the deep learning methods for scalable optimization of offloading [8]. Realizing the transfer learning in VEC with NN-based solutions for scalable offloading optimization is therefore challenging. Existing studies only focus on the learning of abstract feature representations by deep NNs [9]. The effective transfer learning method should be integrally developed with the features that have contributed to the offloading strategies inherited from the proposed NN-based model.

In this paper, we propose a transfer learning approach for the scalable optimization of game theoretical offloading decisions in VEC. The main contributions can be summarized as:

- A two-stage game theory model of offloading decisions in VEC is proposed based on the Stackelberg game framework. The revenue maximization for network operators, as the objective of the Leader in the model, leads the actions of the Follower in the model aiming at the cost minimization of the AV in the offloading strategy optimization.
- An NN-based model is developed with the consideration of characteristics of the two-stage game model to improve the efficiency and effectiveness of the game-playing process. Specifically, an advanced learning algorithm based on the probabilistic moderation/Bayesian approach for opponent strategies, instead of simply using the empirical average, is developed to solve the game process more efficiently.
- The transfer learning is applied for scalable optimization of game theoretical offloading in VEC. The transferable representations are thoroughly explored with the objective of scalable optimization for offloading in VEC. Feature selection and transfer matching are proposed jointly for dimensionality reduction in the proposed transfer learning algorithm. Parallelism for batches of data is utilized, and the synchronization at the end of each training iteration to update the NN parameters is ensured.
- Extensive simulation experimental results show that the proposed NN can efficiently improve the game-solving process, and the accuracy through the devised transfer learning algorithm can be increased up to 37% by comparing with the related algorithms. Besides, the runtime is half of the time required by these algorithms, which satisfies the scalability requirement for offloading in VEC.

The remainder of this paper is organized as follows. Related work and overview of scalable optimization of VEC are presented in Section II. The proposed two-stage game model is elaborated in Section III, followed by the NN-based solution in Section IV. Section V elaborates on the transfer learning for the scalable optimization of game theoretical offloading in VEC. Section VI presents the experimental results and analysis. Finally, Section VIII concludes the paper.

II. Related Work

A. Game Theory

In communication networks, game theory has been broadly applied to develop efficient and effective solutions for resource allocation and routing problems [10]. Three critical elements including *Player*, *Strategy*, and *Utility* are normally involved in a game model. The playing process can be described that players playing with each other based on the opponent's strategy with the objective of maximizing the payoff of players and the playing system. Once the objective is reached, the trade-off of the game model should be at a point named Nash equilibrium where no players have potential motivations to deviate from their strategies by considering the opponent's choice. As one of the optimization methods, game theory shows a great advantage in improving decision-making strategies through competition between players and is able to integrate with learning methods in decision-making processes [11].

1) Game Theoretical Models for Offloading: A multiuser noncooperative offloading game model was proposed to adjust the offloading probability of vehicles [12]. To overcome the challenges of security and insufficient information in cooperative offloading between vehicles, an offloading decision game model was proposed in the scenario where blockchain was employed to facilitate data sharing between vehicles [13]. The further cooperative computation offloading and secure handover in VEC were introduced in [14]. The balance of energy efficiency and reputation gain in UAV scheduling in MEC was studied through the game theoretical approach [15]. Energy-efficient offloading problems in UAV-assisted computing were solved by the PDDQNLP algorithm [16], and system dynamics and complexity issues in space/aerial-assisted offloading were addressed in [17]. Besides, dynamic task offloading decisions in vehicles were also solved by the proposed end-edge-cloud architecture and the asynchronous advantage actor-critic (A3C) based offloading algorithm [18]. However, the limitation caused by the slow convergence rate in solving traditional game models should be overcome.

2) Neural Networks for Solving Game Theoretical Models: As a good candidate, NNs have been studied for addressing the limitations of slow convergence issues of game theoretical models. Rezek et al. [19] introduced the similarities of inference in game theory and machine learning through the analogies between best responses in fictitious play and Bayesian inference approaches. Shiri et al. [20] proposed a method of an NN-based mean-field game theory to address the control problem of massive autonomous UAVs' path planning. Even though initial attempts of NN for solving game theoretical models have been used on scenarios as aforementioned, the combination of the two fields, i.e., game theory and machine learning, is still not well explored and needs to be comprehensively studied for the development of solutions in extensive applications.

B. Scalability

Scalability refers to the ability of a system, a network, or a process in terms of dealing with the growing amount of work, which is achievable through computational offloading in mobile edge computing [21]. Although the scalability problem has been explored in various scenarios due to limitations of the labeled data in the new data sets, most of the studies are exclusive to applications and not well explored in edge computing scenarios.

1) Scalable Optimization: The expectation of conserving energy and computational resource motivates the scalability requirements of traffic systems when the number of AVs changes in VEC. A modified transfer learning technique was utilized to learn people's behaviour and predict human-vehicle interactions so that citywide traffic flow can be effectively improved [22]. Wang et al. [23] proposed vehicular platoon control strategies for connected vehicle platoons under abnormal communications and solved the scalability problem by increasing road capacity. In [24], a novel transfer evolutionary optimization framework was devised for the joint evolution of the scalability problem. A multilane spatiotemporal trajectory optimization method was presented to ease urban congestion and increase road capacity, where the full potentials of connected vehicles with the consideration of vehicular safety, traffic capacity, and fuel efficiency were explored [25]. However, there still lacks an effective method considering computation elements simultaneously, such as energy consumption, latency, and the revenue of network operators in VEC to improve the scalability performance in offloading strategies. New optimization methods should be explored to expand the scalability solutions.

2) Scalable Optimization for Offloading Decisions: To achieve the scalability of a MEC system in offloading decisions, a task offloading and service replication scheme was proposed to minimize the response time of users while satisfying the latency requirements of user groups by deploying the scheme locally and on remote MEC servers [26]. Baresi et al. [27] presented a serverless edge computing architecture and declared that the scalability of the proposed architecture could be extended by the offloading strategy as proposed in [28]. A survey [29] stated that the computation offloading modelling with game theory should be scalable for Nash equilibrium searching in dynamic applications, which is in line with the inherent distributed characteristic of game theory methods.

Most of the prior works on scalable optimizations for VEC and MEC are mainly concentrated on architecture design, edge computing deployment, and admission control for tasks. However, the scalability problems of offloading decision-making strategies by solving with machine learning techniques have not been addressed comprehensively. Therefore, we propose a transfer learning approach to explore the solution of scalable optimization based on game theoretical offloading in VEC.

III. The System Model

The system framework is depicted in Fig. 1, which consists of a set of AVs in the source domain D_s, a series of AVs in the target domain D_T and a shared base station. The AVs in D_s and D_T are in the same network coverage area. We denote vehicles in the D_s as V_{s_1}, V_{s_2}, V_{s_3}, and V_{s_4}, etc. Accordingly, the vehicles in the D_T are represented by V_{T_1}, V_{T_2}, V_{T_3}, and V_{T_4}, etc. Any vehicle in the source domain D_s and target domain D_T can be treated as V_s, and V_T, respectively. In the practical driving environment, an AV in the D_s may run on the ramp or meet the roadblocks, which requires the AV to react in real-time for driving safety and reliability. Therefore, the study on offloading strategies can deliver greater safety benefits of the AV such as overtaking and stopping in real-time driving. Besides, the offloading mechanism also has a meaningful effect on the revenue of the corresponding network operator. Meanwhile, the offloading decision strategies of V_s can be referred by the AVs in D_T for scalability improvement in offloading in VEC.

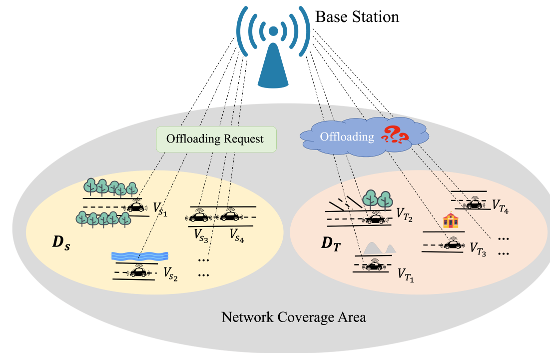


Fig. 1. The framework of scalability requirement in offloading decisions of AVs. The AVs in the source domain D_s and target domain D_T share a base station and they are involved in the same network converge area. The AVs in the D_s first achieve offloading strategies, and then features that affect offloading strategies such as consumed energy, latency, and cost, etc. are transferred to AVs in D_T for the scalable optimization of offloading in VEC.

A. The Two-Stage Game Theoretical Model for VEC Offloading

The vehicle V_s in D_s generally obeys the binary offloading decisions, which can be expressed as

$$f_{V_s} = \begin{cases} 1, & \textit{offloading} \\ 0, & \textit{not offloading} \end{cases} \tag{1}$$

where $f_{V_s} = 1$ means that the V_s performs the data offloading, $f_{V_s} = 0$ means that data are not offloaded to an edge server.

It is assumed that V_s needs to process a task denoted by $\{x, c_{V_s}, T_{max}\}$, where $x \in [0, \bar{x}]$ is the data size to be determined whether to offload to VEC. $\bar{x}$ is the upper bound of the data size. c_{V_s} denotes the computing requirements to process the task x, e.g., the required number of CPU cycles. T_{max} denotes the upper bound of latency requirements.

1) Local Computing: The latency of local computing of V_s can be represented as

$$t_{V_s}^L = \frac{c_{V_s}}{\psi_{V_s}^L} \tag{2}$$

where $\psi_{V_s}^L$ is the computing power (CPU frequency) of the V_s. The energy consumption of local computing can be denoted as

$$E_{V_s}^L = \kappa^L (\psi_{V_s}^L)^2 c_{V_s} \tag{3}$$

where κ^L is the effective switched capacitance on the chip structure of V_s [30]. The cost of executing the task locally is

$$C_{V_s}^L = \alpha E_{V_s}^L + (1-\alpha) t_{V_s}^L, \alpha \in [0, 1] \tag{4}$$

where α is the weight that measures the level of sensitivity of latency and energy consumption to the cost. If $\alpha < 0.5$, more attention is paid to latency, and vice versa. $\alpha = 0.5$ means that the V_s weights the latency the same as the energy consumption.

2) Vehicular Edge Computing: We denote an edge server located at a base station as m, to which the task is offloaded from V_s. The latency spent on the results transmission through downlink to V_s can be ignored due to a small size. The channel transmission rate of uplink can be expressed as

$$R_{(V_s,m)} = \lambda w \log_2 \left(1 + \frac{p_{(V_s,m)} h_{(V_s,m)}}{\rho^2 + I_{(V_s,m)}}\right) \tag{5}$$

where $w = B/K$ denotes the assigned bandwidth by separating bandwidth B into K parts. K is the number of AVs connected to an edge server. $p_{(V_s,m)}$ and $h_{(V_s,m)}$ denote the power of transmission and the channel gain between the edge AV V_s and the edge server m. ρ^2 denotes the noise power, and $I_{(V_s,V_n)}$ is the interference of V_s suffering from the communication of other vehicles, which can be denoted as

$$I_{(V_s,V_n)} = \sum_{n=1,n\neq s}^{N} p_{(V_n)} h_{(V_s,V_n)} \tag{6}$$

where $h_{(V_s,V_n)}$ is the channel gain of the vehicle V_s to the vehicle V_n on the channel [31]. $\lambda \in \{0, 1\}$, that is, if $\lambda = 1$, the channel is assigned to vehicle V_s for task offloading, otherwise, $\lambda = 0$. Accordingly, V_s has interference from other vehicles or the interference of $I_{(V_s,V_n)}$ is 0.

The latency of transmission and computation of task x at VEC m can be expressed as

$$t_{V_s}^{EC} = \left(\frac{x}{R_{(V_s,m)}} + \frac{c_{V_s}}{\psi_{V_s}^{EC}}\right) \tag{7}$$

The energy consumption on transmission and the computation of x at the VEC server can be formulated as

$$E_{V_s}^{EC} = p_{(V_s,m)} \frac{x}{R_{(V_s,m)}} + \kappa^{EC} (\psi_{V_s}^{EC})^2 c_{V_s} \tag{8}$$

where κ^{EC} is the energy coefficient.

Therefore, the total cost of the transmission and computation on VEC considering latency and energy consumption is

$$C_{V_s}^{EC} = \beta E_{V_s}^{EC} + (1-\beta) t_{V_s}^{EC}, \beta \in [0, 1] \tag{9}$$

where the meaning of β is similar to that of α in Eq. (4).

The total cost spent on the task x can be concluded as

$$C_{tc} = f_{V_s} C_{V_s}^{EC} + (1 - f_{V_s}) C_{V_s}^{L} \tag{10}$$

3) The Two-Stage Game Theoretical Model: The problem of the offloading can be modelled with a two-stage game theoretical model, as described in Fig. 2. The **Leader** in Stage 1 aims to achieve the optimal price that maximizes the network operator's revenue. The **Follower** in Stage 2 intends to determine the offloading data size with the minimum cost of the edge AV. These two games interact by backward induction with each other and solve the optimal results to reach a balance that any changes in the outcome of one game will break the optimal outcome of the other game. The vehicle is the one that first requests the edge service from a network operator in the practical scenario and may cancel the computing service request if the price provided by a network operator exceeds the expectation of the vehicle. Therefore, there is an occasion

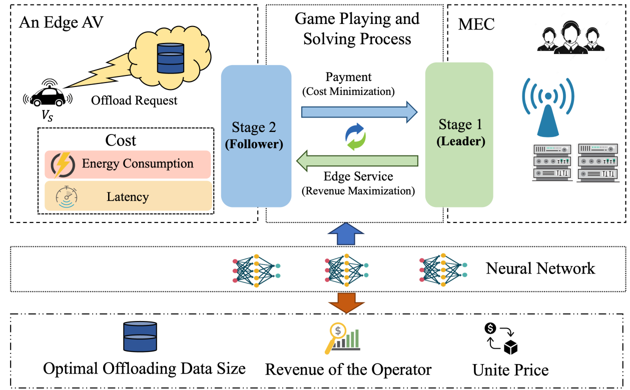


Fig. 2. The game theoretical offloading decision-making process solved by neural Networks.

when the price of the computing service provided by a network operator significantly exceeds the expectation of the vehicle. It is possible for the vehicle to cancel the computing service request, thereby the VEC service is highly unlikely to occur. Based on this situation, the optimization of Stage 2 is first introduced in the following analysis and the two-stage game theoretical model is depicted in the upper part of Fig. 2.

a) Cost minimization and offloading strategy: The energy consumption and latency constraints in data processing locally or by VEC can be expressed as

$$E = (1 - f_{V_s}) E_{V_s}^{L} + f_{V_s} E_{V_s}^{EC} \leq E_{max} \tag{11}$$

$$T = (1 - f_{V_s}) t_{V_s}^{L} + f_{V_s} t_{V_s}^{EC} \leq T_{max} \tag{12}$$

where E and T represent the total energy consumption and latency, respectively. E_{max} and T_{max} denote the maximum tolerances of energy consumption and latency, respectively.

b) Stage 2 sub-game (Follower): The total cost, shown by Eq. (10), is expected to be reduced as much as possible. Thus, the cost-minimization problem can be formulated as

$$\begin{aligned} \min \ & C_{tc} = f_{V_s} C_{V_s}^{EC} + (1 - f_{V_s}) C_{V_s}^{L} \\ s.t. \ & C1 : (1 - f_{V_s}) E_{V_s}^{L} + f_{V_s} E_{V_s}^{EC} \leq E_{max} \\ & C2 : (1 - f_{V_s}) t_{V_S} + f_{V_s} t_{V_s}^{EC} \leq T_{max} \\ & C3 : 0 \leq \sum_{k=1}^{K} \lambda_k p_{(V_s,m)} \leq P_{max} \\ & C4 : \{f_{V_s}, \lambda\} \in \{0, 1\}, \{\alpha, \beta\} \in [0, 1] \end{aligned} \tag{13}$$

We suppose that the price charged for unit energy spent on communication and computation is p_e, which is consistent among network operators [2]. Therefore, the total payment for the transmission and commutation can be expressed as

$$C_{p_e}^{EC} = p_e \cdot \left(f_{V_s} E_{V_s}^{EC} + (1 - f_{V_s}) E_{V_s}^{L}\right) \tag{14}$$

c) Pricing and revenue maximization: The price is unicast by a network operator as a response for the offloading request of the V_s in the D_s. After receiving the quoted price, the V_s will respond with its acceptable price π to the VEC operator. Then the operator will update its quoted price in the next round of offers. By continuous iterations, an optimal price that promotes the maximization of an operator's revenue is expected to be reached while satisfying the cost minimization of the V_s.

d) Stage 1 sub-game (Leader): The revenue maximization of an operator through VEC service provision can be achieved by minimizing the energy cost and maximizing the price as

$$\max\ \Phi = \pi x - C_{p_e}^{EC} = \pi x - p_e E_{V_s}^{EC} \tag{15}$$

where π is the quoted price, x is the size of data for offloading.

B. Equilibrium Analysis

We suppose that there exists a Nash equilibrium of the optimal price π^* satisfying the revenue maximization of a network operator with the optimal data size x^*, i.e.,

$$\pi^*, x^* = \arg\max\ \Phi(\pi, x, C_{tc}) \tag{16}$$

where the optimal data size x^* contributes the minimum cost of the vehicle V_s for offloading

$$x^* = \text{argmin}\ \ C_{tc}(x, E, T) \tag{17}$$

1) Leader's Game – Stage 1: In this stage, we formulate the pricing strategy, energy cost and latency into the revenue of network operators, which will be considered as the input information of the Follower in the next stage of the game.

Given the data size x in Stage 2, an operator (OP) playing with the communicated AV to maximize the revenue is a non-cooperative game. The pricing process is performed as Pricing Game (PG) $\Omega = \{K, \{x_{s_k}\}_{k\in K}, \{\Phi_{s_k}\}_{k\in K}\}$, where $k \in K = \{k = \{1, \ldots, K\}, OP\}$, and K is the set of players. $\{x_{s_k}\}_{k\in K}$ represents the strategy set of task offloading. The $\{\Phi_{s_k}\}_{k\in K}$ is the revenue of the OP when the strategy is x_{s_k}. Therefore, the revenue maximization can be described as $\{\Phi_{s_k}\}_{k\in K}$. Once the offloading data size x and price π are determined, the energy consumption and latency are resolved in Stage 2, then the revenue $\{\Phi_{s_k}\}_{k\in K}$ in Stage 1 can be maximized.

Theorem 1: Suppose that the Nash equilibrium exists in the offloading decisions with optimal x^* and π^* which stand for the optimal data size and the optimal price that is offered by an operator, respectively. The (x^*, π^*) should be at the equilibrium point of this game if the following conditions hold

$$\Phi(\pi^*, x^*, C_{tc}^*) \geq \Phi(\pi, x^*, C_{tc}^*), \quad \forall \pi \geq 0, \tag{18}$$

$$\Phi(\pi^*, x^*, C_{tc}^*) \geq \Phi(\pi^*, x, C_{tc}^*), \quad \forall x \geq 0 \tag{19}$$

Definition 2: The E^*, T^* and x^* are the optimal energy cost, latency and offloading data size of the game $\mathbb{G}$, if players in this stage meet $C(E^*, T^*, x^*) \leq C(E, T, x)$, where $C(E^*, T^*, x^*)$ is the optimal and minimum cost in the convex function and satisfies the agreed offloading data size, energy cost and latency.

Theorem 3: A Nash equilibrium exists in the game $\mathbb{G} = \{\{V_S, operator\}, \{f_{V_S}\}, \Phi\}$.

Proof: The strategy space of $\mathbb{G}$ is f_{V_S}, which is not empty and determines the offloading decision-making and the cost of the offloading. For the π is the function of x, to learn about the second derivative of Eq. (15), with Eq. (8), we have

$$\frac{\partial \Phi}{\partial \pi} = x + \pi \frac{\mathrm{d}x}{\mathrm{d}\pi} - p_e p_{(V_s,m)} \frac{1}{R_{(V_s,m)}} \frac{\mathrm{d}x}{\mathrm{d}\pi} - P_e \kappa^{EC} (\psi_{V_s}^{EC})^2 \frac{\partial c_{V_s}}{\partial \pi} \tag{20}$$

$$\frac{\partial^2 \Phi}{\partial \pi^2} = \left(\pi - p_e p_{(V_s,m)} \frac{1}{R_{(V_s,m)}}\right) \frac{\mathrm{d}^2 x}{\mathrm{d}\pi^2} + 2\frac{\mathrm{d}x}{\mathrm{d}\pi} - P_e \kappa^{EC} (\psi_{V_s}^{EC})^2 \frac{\partial^2 c_{V_s}}{\partial \pi^2} \tag{21}$$

Let $\frac{\partial^2 \Phi}{\partial \pi^2} = 0$, then

$$\left(\pi - p_e p_{(V_s,m)} \frac{1}{R_{(V_s,m)}}\right) \frac{\mathrm{d}^2 x}{\mathrm{d}\pi^2} + 2\frac{\mathrm{d}x}{\mathrm{d}\pi} = P_e \kappa^{EC} (\psi_{V_s}^{EC})^2 \frac{\partial^2 c_{V_s}}{\partial \pi^2} \tag{22}$$

The curves of the quadratic equation shown in Eq. (21) should be studied.

The right term of the Eq. (22) is the quadratic equation with one variable, whose roots should be equal to the roots of the left term. There, we have one root 0, and another root

$$\frac{-2R_{(V_s,m)}}{\pi R_{(V_s,m)} - P_e P_{(V_s,m)} - P_e \kappa^{EC} (\psi_{V_s}^{EC})^2 R_{(V_s,m)}}.$$

As the non-zero root should be less than 0 on $(-\infty, 0]$, when

$$\pi \in \left(\frac{P_e P_{(V_s,m)}}{R_{(V_s,m)}} + P_e \kappa^{EC} (\psi_{V_s}^{EC})^2, +\infty\right) \tag{23}$$

the $\frac{\partial^2 \Phi}{\partial \pi^2} < 0$. The Φ is concave, which means that the optimal price exists for revenue maximization. Therefore, the existence of the Nash equilibrium of the game can be reached. ■

Theorem 4: The uniqueness of the Nash equilibrium in the game $\mathbb{G}$ is reached.

Proof: Under the proof of Theorem 3, a Nash equilibrium is proved to exist in the game. Then we suppose that π^* is the Nash equilibrium of the game, which is the best response agreed by the network operator and the V_s for the offloading strategy, where $\pi^* \in \{\pi_1, \pi_2, \ldots, \pi_n\}$. Let the first derivative of Eq. (20) be 0 for the response of the game playing. The right function of Eq. (22) is monotony on $(-\infty, 0]$, and the corresponding result of the function is positive with the non-zero root. Only when $\frac{\mathrm{d}x}{\mathrm{d}\pi} \in \left(\frac{-2_{(V_s,m)}}{\pi R_{(V_s,m)} - P_e P_{(V_s,m)} - P_e \kappa^{EC} (\psi_{V_s}^{EC})^2 R_{(V_s,m)}},\ 0\right)$, then the left function of Eq. (22) has an extremum satisfying $\frac{\partial^2 \Phi}{\partial \pi^2} < 0$. The price is

$$\pi = \frac{P_e P_{(V_s,m)}}{R_{(V_s,m)}} + P_e \kappa^{EC} (\psi_{V_s}^{EC})^2 \tag{24}$$

Therefore, the uniqueness of the Nash equilibrium in the game can be reached by showing the response of the network operator and V_s, and Theorem 4 is concluded. ■

2) Follower's Game – Stage 2: In this stage, the offloading data size will be agreed upon by the game-playing process and the revenue maximization of the network operator is expected to be achieved. The cost of the Follower is described as

$$C^* = \min\ \ p_e \cdot E_{V_S}^{EC}$$
$$s.t.\ E^* = \min\{E_{V_S}^{EC}\}, \quad p_e^* = \min\{p_e\} \tag{25}$$

where C^* is the minimum cost for communication and computation services. E^* and p_e^* stand for the optimal energy consumption and the agreed price, respectively. The offloading data size is implied in the function.

Definition 5: The agreed price π^* is optimal while satisfying the utility $\Phi(\pi^*, E^*, t^*) \geq \Phi(\pi, E^*, t^*)$.

Theorem 6: The network operator can obtain revenue optimization through cost minimization measuring in Stage 2.

Proof: The offloading data size $x^* \in \{x_1^*, x_2^*, \ldots, x_n^*\}$ is the Nash equilibrium of the cooperative game for C^* if the network operator plays with an AV with satisfying $\Phi(\pi^*, x^*, C_{tc}^*) \geq \Phi(\pi^*, x, C_{tc})$, $\forall x \geq 0$. According to Eq. (14), we have

$$C_{p_e}^{EC} = p_e \left(p_{(V_s,m)} \frac{x}{R_{(V_s,m)}} + \kappa^{EC} (\psi_{V_s}^{EC})^2 c_{V_s} \right) \quad (26)$$

$$\frac{\partial C_{p_e}^{EC}}{\partial x} = p_e \left(p_{(V_s,m)} \frac{1}{R_{(V_s,m)}} + \kappa^{EC} (\psi_{V_s}^{EC})^2 \frac{\mathrm{d} c_{V_s}}{\mathrm{d}x} \right) \quad (27)$$

$$\frac{\partial^2 C_{p_e}^{EC}}{\partial x^2} = p_e \kappa^{EC} (\psi_{V_s}^{EC})^2 \frac{\mathrm{d}^2 c_{V_s}}{\mathrm{d}x^2} \quad (28)$$

where c_{V_s} is linear with x, $\frac{\mathrm{d}c_{V_s}}{\mathrm{d}x}$ is a constant greater than 0. We have $\frac{\partial C_{p_e}^{EC}}{\partial x} > 0$ and $\frac{\partial^2 C_{p_e}^{EC}}{\partial x^2} = 0$. The $C_{p_e}^{EC}$ is a monotonically increasing function subject to x. Thus, there exists a minimum cost to maximize the revenue of the operator. ■

Algorithm 1 Game Theoretical Offloading Strategy (GTOS)

Input: Quoted price for offloading π;
Offloading data size x;
Price charged for unit energy spent on communication and computation p_e;
Output: The total cost spent on the offloading task C_{tc};
The agreed price for offloading π^*;
The agreed offloading data size x^*;
The optimal revenue of the network operator Φ^*

Initialize $\pi_0, x_0, \alpha, \beta$;
while $C_{p_e} > \Phi(\pi^*, x^*, C_{tc}^*)$ *and* $C(\pi^*, E^*, T^*, x^*) < C(\pi^*, E, T, x)$ **do**
  if $C_{V_s}^{EC} < C_{V_s}^{L}$ **then**
    $f_{V_S} \leftarrow 1$, $\lambda \leftarrow 1$;
    $C_{tc} \leftarrow (\pi^*, x^*, E_{V_s}^{EC}, t_{V_s}^{EC})$; `// Eq. (13)`
    if $C_{p_e}(E_{V_s}^{EC}, p_e, x) < \Phi(\pi^*, E^*, T^*, x)$, $\Phi(\pi^*, x^*, C_{tc}^*) > \Phi(\pi, x^*, C_{tc}^*)$, $\Phi(\pi^*, x^*, C_{tc}^*) > \Phi(\pi^*, x, C_{tc}^*)$ **then**
      $x^* \leftarrow \text{argmin } C$;
      $\Phi^* \leftarrow \pi^*, x^*$;
      Break;
    else
      Not to converge (no Nash equilibrium point)
    end
  else
    $f_{V_S} \leftarrow 0$, not to offload, $\lambda \leftarrow 0$
  end
end

IV. The Proposed Neural Network for Efficiently Solving the Game Theoretical Model

Solving game models is an iterative game-playing process, which is time-consuming and may cause a high cost of computing resources to some extent. Yet, VEC networks change dynamically for the varying number of vehicles connected with a base station (further connecting to a MEC server if offloading occurs) according to road conditions. These situations bring about the changing inputs of game models and the solving processes need to be recalled to resolve the models each time a change occurs. To deal with these problems, we propose a new solution which adopts NN to efficiently solve the changing game-playing process with better adaptability due to the variation of game theoretic models caused by the changing situations of AVs.

The mean field variational (MFV) method can be used to infer the probability distribution with latent variables based on the observed datasets [32], by which the intractable requirement can be asymptotically satisfied in the learning process. Through the generated data $\mathcal{D}$ and the defined latent variables, the variable generated in the game playing process $S_i = \{(C_{tc})_i, \pi_i, x_i, \Phi_i\}$ can be extended to $\mathbb{S}$, where i means the number of iterations. The distribution of the extended S_i is expected to be estimated. Let s represent an element of the latent variable, $s \in \mathbb{S}$, we postulate the marginal distribution $p_i \in \Delta(\mathbb{S}_i)$ with respect to variable i in the set of all marginal distributions $\Delta(\mathbb{S}_i)$ over S_i, θ is the parameter to the distribution p. Then

$$\ell(S|\mathcal{D}, \theta) = \log(p(S|\mathcal{D}, \theta)) \quad (29)$$

Let $q(s)$ represent a variational approximation to the posterior distribution $p(S|\mathcal{D})$, the criterion can be expressed [19]

$$\mathcal{F} = \int \cdots \int q(s) \ell(s|\mathcal{D}, \theta) ds + \tau H(q) \quad (30)$$

where Shannon entropy $H(s) = -\int \cdots \int q(s) \log q(s) ds$ implies the uncertainty inherent in the outcomes of variables.

The Kullback-Leibler(KL) divergence can be depicted as

$$D_{KL}(p\|q) = -\mathcal{F} = \int \cdots \int q(s) \log \left(\frac{p(s|\mathcal{D}, \theta)}{q(s)^{\tau}} \right) ds \quad (31)$$

The distribution of observed data $\mathcal{D}$ is not confirmed in the above analysis. Therefore, we assume the existence of means and variances of the distribution according to the central-limit theorem [33] and then depict the distribution p with the Gaussian distribution $p_i(S_i|\mathcal{D}, \theta) \sim \mathcal{N}(S_i|\mu, \rho^2)$, which is parameterized with $\theta = (\mu, \rho^2)$. Then

$$p(S|\mathcal{D}) = \prod_{i=i}^{I} \ell(S_i|\mu, \rho) \quad (32)$$

where the $\nabla S(C_{ct}, \pi, x, \Phi)$ is taken to optimize the mean and variance of the distribution.

In order to construct a NN to learn the distribution, for the sets of parameters $\{(\mu_1, \rho_1), \ldots, (\mu_i, \rho_i), \ldots\}$, the complete likelihood can be formulated as

$$p(S_i, \mu, \rho | s) \propto \exp\left\{\int \cdots \int \ell(S_i|\mu, \rho^2) ds\right\} \tag{33}$$

where s is each sample of the strategy.

To maximize the likelihood function

$$\hat{\mu}, \hat{\rho}^2 = \underset{\mu, \rho^2}{\arg\max}\ \log p(S_1, \ldots, S_I | \mu, \rho^2) \tag{34}$$

Let

$$\frac{\partial \log p}{\partial \mu} = 0, \ \frac{\partial \log p}{\partial \rho} = 0,$$

$$\hat{\mu} = \sum_{i=1}^{I} S_i / I, \ \hat{\rho}^2 = \sum_{i=1}^{I} (S_i - \hat{\mu})^2 / I \tag{35}$$

Then, the update of the q is

$$q(s) \longleftarrow \frac{p(\mathcal{D}|s) p(s)}{\int_s p(\mathcal{D}|s) p(s) ds} \tag{36}$$

Based on the analysis of the posterior distribution p in Eq. (33) and the above updated q in Eq. (36), the expected log-likelihood can be concluded as

$$\mathcal{L} = \int \cdots \int q(s) \log(p(s|\mathcal{D}, \theta)) ds \tag{37}$$

which is the expected reward from the perspective of game theory [19]. The result of the solving process of the game theory model will be collected, based on which the output datasets $S_i(C_{tx}, \pi, x, \Phi)$ are used to train the constructed neural network model. The parameter of the neural network q is updated according to the performance of each training iteration. The game theoretic offloading-mean field variational (GTO-MFV) process is described as Algorithm 2.

Algorithm 2 Game Theoretic Offloading-Mean Field Variational (GTO-MFV) Process

Input: Datasets $S_i(C_{tx}, \pi, x, \Phi)$
Output: The mean of Gaussian distribution $\hat{\mu}$;
The standard deviation of Gaussian distribution $\hat{\rho}$;
The expected log-likelihood $\mathcal{L}$

```
Initialize S_i(C_tx, π, x, Φ), μ_0, ρ_0, i ∈ I;
if the D_KL has not converged then
    p_i(S_i|D, θ) ~ N(S_i|μ, ρ^2);
    if μ, ρ have not converged then
        μ̂, ρ̂ ← arg max_{μ,ρ^2} log p(S_1, ..., S_I|μ, ρ^2);
          // Eq. (34)
        ℓ(S|D, θ) ← log p(S|D, θ);     // Eq. (29)
        p(s|D, θ) ← ℓ(S|D, θ);         // Eq. (32)
        q(s) ← p(s|D, θ);              // Eq. (36)
    else
        Return μ̂, ρ̂;
        Update q;
    end
else
    L ← q(s);
    Return L
end
```

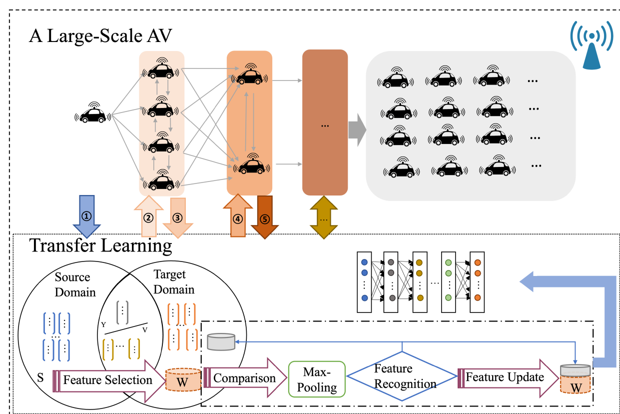


Fig. 3. The offloading decisions of a large-scale AV with the transfer learning approach.

V. Transfer Learning for Scalable Game Theoretical Offloading in Vehicular Edge Computing

Each AV on the road has exclusive features of its own and common features with others. However, collecting all of the AVs' data is impossible for making intelligent offloading decision-making strategies. Upon the request of the scalable offloading decision in VEC, transfer learning can make up for the above deficiencies for an effective solution. Because observed features are homogeneous in this scenario, and the distributions are independent and identically. Through hypothesising the training data independently and identically distributed (i.i.d.) with the test data, it can effectively achieve the knowledge transfer from the source domain to the target domain, so that the difficulties in insufficient training data can be overcome. The offloading decisions of a large-scale AV with the transfer learning approach are described in Fig. 3, where the upper part is the display of the transfer mechanism, and vehicles communicate with each other by a peer-to-peer mesh network.

A. Features Selection for Target Domain

1) Notations: The data $S_i(C_{tx}, \pi, x, \Phi)$ is denoted as the matrix $S = [S_1, S_2, \ldots, S_i] \in R^{h \times n}$, where $S_i = [C_{tx}^i, \pi_i, x_i, \Phi_i]^T$. The projection matrix $W \in R^{h \times m}$ projecting the input S into the common m-dimensional space for performing feature selection on different feature spaces. The dth row and jth column of the matrix W are represented with w_d and w_j, respectively. The Frobenius norm of the matrix S is defined as $\|W\|_F = \sqrt{\sum_{d=1}^{h} \|w_d\|_2^2}$, and the $L_{2,1}$-norm of W is expressed as $\|W\|_{2,1} = \sum_{d=1}^{h} \|w_d\|_2$.

2) Problem Formulation and Solution: The offloading decisions of a large-scale AV with the transfer learning process in Fig. 3 can be described in steps:

- *Step ①:* The output from the game theoretical model of an AV is treated as source data and sent to the transfer mechanism for processing.

- *Step ②:* The processed result from the mechanism is transferred to the nearby AVs for offloading references.
- *Step ③:* The received result and the features of the new nearby AVs are sent to the transfer mechanism for the evaluation of features and offloading references.
- *Step ④:* The offloading strategies of these new AVs with specific features are transferred to other AVs.
- *Step ⑤:* The strategies with new features of these AVs are sent to the transfer mechanism for features comparison with new AVs and updated for offloading decisions.

In the transfer mechanism, the data similarity between S and W is expected to be learned and predicted. Let Y be the similar semantics of a joint subspace between S and W. The optimal W can be formulated as a linear dimensional reduction problem and the feature of V_s with the space learning can be formulated as a convex optimization problem as

$$\begin{aligned} &\min_W \|W\|_{2,1} + \mu tr(W^T S L S^T W) \\ &s.t.\ W^T S V S^T W = I \end{aligned} \tag{38}$$

where I is the identity matrix. The graph Laplacian is expressed as $L = D - V$, $D_{ii} = \sum_j V_{ij}$. $V = (V_1, V_2, \ldots, V_c) \in R^{N\times c}$ is represented as the labeled data. μ is a regularization parameter. We take an example with kth class to demonstrate different V as

$$V_{i,j} = \begin{cases} 1/n_k, & \text{if } S_i \text{ and } S_j \text{ belong to the } k\text{th class} \\ 0, & \text{otherwise} \end{cases} \tag{39}$$

We suppose that there exists a matrix W whose each column is an eigenvector of $SYS^T = \lambda SVS^T$, λ is the eigenvalue, such that $S^T W = Y$. Therefore, we need to solve the problems: 1) work out Y according to $VY = \Lambda DY$, where Λ is a diagonal matrix; 2) find W which satisfies $S^T W = Y$. The optimization problem with non-convex and complex constraints can be converted to

$$\begin{aligned} &\min\ \ \|W\|_{2,1} \\ &s.t.\ \left\|S^T W - Y\right\|_F^2 \end{aligned} \tag{40}$$

Meanwhile, Eq. (38) has solutions, which means that the Laplacian function has solutions. Thus, we have

$$L_p(W) = \|W\|_{2,1} - \mu tr \left\|S^T W - Y\right\|_F^2 \tag{41}$$

which is convex. In order to solve this expression, we have

$$\frac{\partial L_p(W)}{\partial W} = GW + 2\mu(SS^T W - SY) = 0 \tag{42}$$

$$W = 2\mu(G + 2\mu SS^T)^{-1} SY \tag{43}$$

where G is a diagonal matrix described as

$$G_{ii} = \begin{cases} 0, & \text{if } w^i = 0 \\ \dfrac{1}{\|w^i\|_2}, & \text{otherwise} \end{cases} \tag{44}$$

which is dependent on the projection matrix W. The W can be reached by referring to [34]

$$\begin{aligned} W &= 2\mu G^{-1} SY - 2\mu G^{-1} S \left(I - \left(W^T G^{-1} S + \frac{1}{2\mu} I \right)^{-1} \right) Y \\ &= G^{-1} S \left(S^T G^{-1} S + \frac{1}{2\mu} I \right)^{-1} Y \end{aligned} \tag{45}$$

A transfer learning method that consists of feature selection, comparison and update processes is described in Algorithm 3.

Algorithm 3 Transfer Learning-Features Selection, Comparison and Update (TL-FSCU)

Input: The matrix with labeled and unlabeled S;
The low dimensional representation V;
The existing features F;
Output: The projection matrices W;
The Laplacian matrix L_p;
The matrix F_i;
1 Initialize W as identity matrix; $t \leftarrow 0$, μ;
2 **repeat**
3 $\quad W_{t+1} \leftarrow G_t^{-1} S (S^T G_t^{-1} S + \frac{1}{2\mu} I)^{-1} Y$;
4 $\quad G_{t+1} \leftarrow W_{t+1}$;
5 $\quad t \leftarrow t + 1$
6 **until** *Converges*;
/* compare and update feature clusters, in the following algorithm, G_{t+1} is treated as G to compare with $F_i \in F$ */
7 **while** G, F_i **do**
8 $\quad$ Compute similarity $\mathcal{S}$;
9 $\quad$ **if** $\mathcal{S}(G, F_i) > \epsilon$ **then**
10 $\quad\quad$ $i \leftarrow i + 1$ to initialize the new feature cluster $\mathcal{K}$;
11 $\quad\quad$ Update the final feature cluster $\mathcal{M}$;
12 $\quad\quad$ Repeat the line 7;
13 $\quad$ **else**
14 $\quad\quad$ Add G to existing features F_i and update the feature cluster of new AVs;
15 $\quad\quad$ Make a decision with the G;
16 $\quad$ **end**
17 **end**

B. Features Comparison and Update

There are two methods in terms of similarity comparison of features and we conclude them to our scenario:

1) Suppose that F is the features that G needs to compare with, and can be obtained by applying the basic Gaussian membership function $e^{-(\frac{G-F}{\delta})^2}$. The similarity of the features can be expressed as

$$\mathcal{S}(G, F) = \frac{e^{-\left(\frac{G-F}{\delta}\right)^2} + \eta}{1 + \eta} \le \epsilon \tag{46}$$

where $\delta = \frac{\epsilon}{\sqrt{ln\left(\frac{1}{abs(1-(1+\eta)*\epsilon)}\right)}}$, ϵ is the threshold. $\eta = 0.3679$ [35].

2) The similarity of the ϵ can be referred as

$$\mathcal{S} = D_{KL}(G\|F) = P(G)\log\left(\frac{P(G)}{P(F)}\right) \leq \epsilon \quad (47)$$

The selected features are compared against the stored features in the target domain: a) if the selected features do not exist, the features will be added to the target domain for feature mapping; b) if the similarity score is greater than a specific threshold ϵ, the features will not be updated. Then the offloading strategy of the AV will be inherited.

In the lower part of the Algorithm 3, the feature cluster $\mathcal{K}$ is inherited from existing feature F_i due to the greater similarity of G to F_i. $\mathcal{M}$ indicates the updated feature cluster of nearby AVs which wait for offloading decision-making strategies. $\mathcal{M}$ can be the same as $\mathcal{K}$.

VI. Experimental Results and Analysis

In this section, we conduct experiments on the proposed a two-stage game theoretical model for offloading decisions in VEC. Then the feasibility of efficiently solving the game theory model with the constructed NN is verified through KL divergence. Afterwards, the superiority of transfer learning on the scalable optimisation of offloading decisions is evaluated by comparing it with several related algorithms in terms of accuracy, features update ratio, runtime, etc.

A. Experiment Settings

The proposed algorithms, GTOS and GTO-MFV, are implemented via Matlab, and TL-FSCU is implemented via Python 3.7 on a GPU-based server. The server has a GPU of NVIDIA-SMI 470.57.02 with TITAN RTX and CUDA version 11.4. The CPU is Intel(R) Core(TM) i9-9960X CPU @ 3.10GHz. In the experiments, we consider a typical VEC scenario, where in a network coverage area, a base station provides the network services for AVs in D_s and D_T, which means that the V_s in D_s has the same network environment with the V_T in D_T.

We set the number of AVs in D_T as 2 initially to verify the feasibility of the algorithms, and then we increase it to 10 and 100, respectively, to compare the runtime performance of the proposed method with the other selected algorithms. The assigned bandwidth w will be calculated accordingly. The other parameters are presented in Table I.

B. Performance and Analysis

We firstly conduct the evaluation of GTOS (Algorithm 1) that achieves the optimal solution for the offloading strategy, and then we evaluate the GTO-MFV (Algorithm 2) to ensure that the solving process is more effective and efficient. After that, the TL-FSCU (Algorithm 3) is evaluated to show the improvement of scalability of the offloading optimization problems in VEC. The ablation study has been conducted by comparing the performances of our proposed method with baseline methods including iCaRL [36], LwF [37], and JFSSL [38], respectively, to verify the effectiveness of the proposed method.

TABLE I
Experiment Parameters

Parameter	Description	Value
B	The total channel bandwidth	10MHz
x	Data size to be determined whether to offload to VEC	10^6 bits
κ^L	Effective switched capacitance related to the chip structure of V_s	10^{-11}
κ^{EC}	Energy coefficient	10^{-27}
$\psi^L_{V_s}$	CPU frequency of the V_s	5×10^8Hz
$\psi^{EC}_{V_s}$	Assigned computational speed (CPU frequency) of the server to the V_s	6×10^9Hz
p_e	The price charged for unit energy spent on communication and computation	17.2 p/kWh
$p_{(V_n)}$	The power of vehicle p_{V_n}	20 dBm
$p_{(V_s,m)}$	The power of transmission between V_s and m	3W
$h_{(V_s,m)}$	The channel gain between V_s and m	-30 dBm
$h_{(V_s,V_n)}$	The channel gain between the vehicle V_s and the vehicle V_n	-30 dBm
ρ^2	Noise power	-180 dBm/Hz
ϵ	Threshold of the similarity	0.5

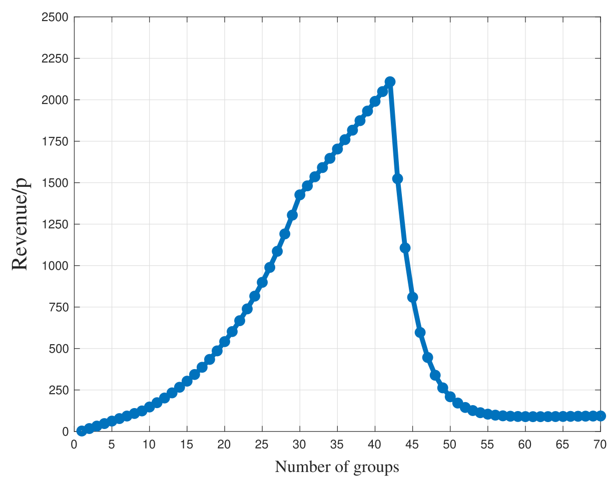


Fig. 4. Revenue performance through the game theoretical model.

1) The Revenue of the Network Operator: Fig. 4 shows the performance of the obtained revenue of network operators through the game theoretical playing between a vehicle and a network operator, where 70 feature groups containing $\{C, \pi, x, \Phi, \}$, captured from the playing process of GTOS, are taken to demonstrate the trend of the revenue change. It is worth noting that the revenue grows up to 2100p, along with the iteration to the 42nd group. After that, the revenue of the network operator is sharply decreased, which means that the more offloading data size or the increased service price would not improve the revenue of the network operator. This experiment shows that there exists a point satisfying the balancing strategy of the network operator and the edge AV.

2) The Performance of Offloading Latency and Energy Consumption: Fig. 5 presents the offloading latency energy

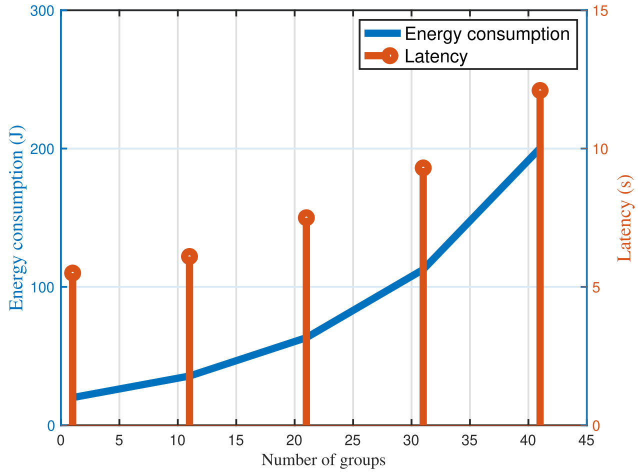


Fig. 5. The performance of the offloading latency and energy consumption.

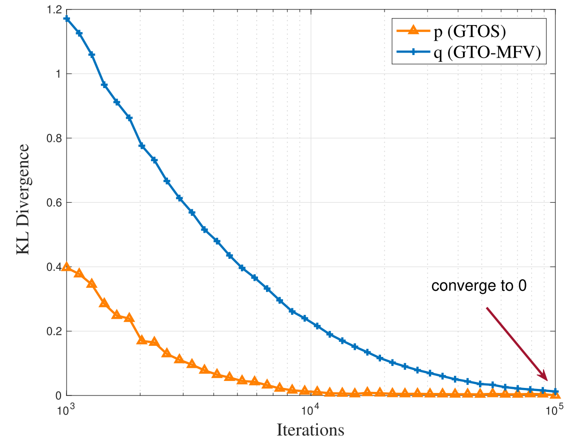


Fig. 6. The KL divergence for distributions p in game theoretical model and q for the proposed GTO-MFV.

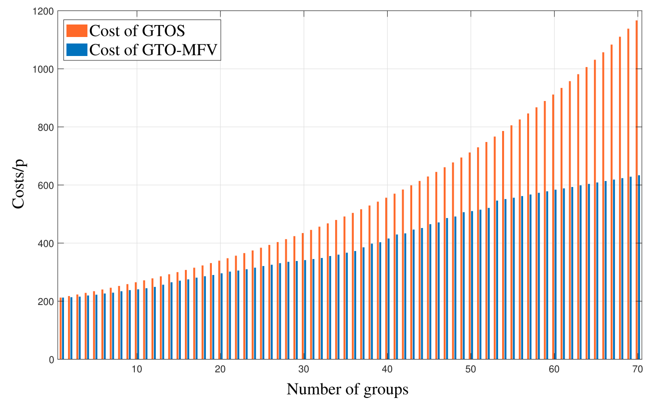


Fig. 7. The cost of GTOS vs. GTO-MFV.

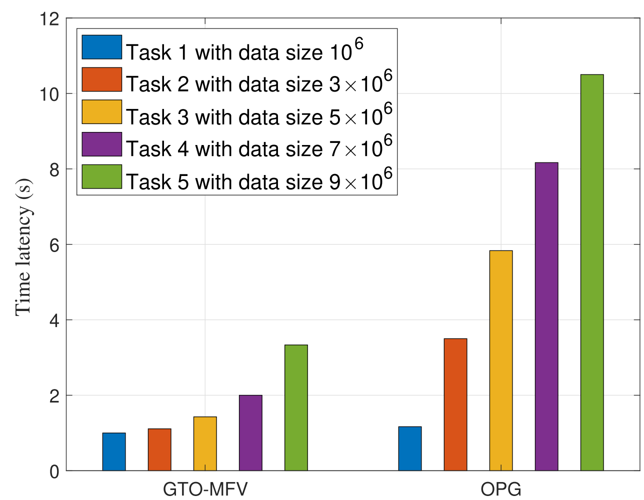


Fig. 8. The comparison of latency between GTO-MFV and OPG [2] with different tasks.

and consumption in addition to the revenue of the network operator, displayed in Fig. 4, where we collect 5 group results to show latency and energy consumption to demonstrate the performance of offloading. Along with the increase in offloading data size, the energy consumption shows an upward trend, which is displayed in the blue line to the left y-axis in the figure. Latencies, the lines marked with the red colour, also demonstrate an upward trend overall, but they go up with a slow scope as the increase in the offloading data size. The performance can be seen with reference to the right y-axis.

3) Verification of Constructed NN to the Game Theoretical Model: Fig. 6 reveals the KL divergence of the game theoretical model and the proposed GTO-MFV, where the data set $S_i(C_{tx}, \pi, x, \Phi)$ of the GTOS process is used to fit the distribution. q represents the variational approximation to the posterior distribution of p. We can see that the two lines converge at the number of 10^5 iterations, which indicates the equivalence of the constructed GTO-MFV to the GTOS. Therefore, the difference between the two formulated distributions can be ignored. In other words, the GTO-MFV can be seen as the substitution solution to the process of GTOS, so that the limitations of the traditional way of solving game theoretical offloading models such as high computational complexity and energy consumption can be overcome.

4) Cost Comparison of GTOS With GTO-MFV: Fig. 7 demonstrates the comparison of costs of solving game theory models with the GTOS and the GTO-MFV. We take the costs of 70 feature groups from the process of solving the game model through these two methods. It is obvious that the costs spent on the model solving with GTOS and GTO-MFV keep increasing. However, the cost with the GTOS increases sharply, but GTO-MFV shows its slow growth in cost. The reason for this difference is that continuous iterations of heuristic search in the traditional method consume more energy than the learning process since the GTO-MFV is constructed by approximation and inference. The cost generated by GTO-MFV in solving the game model is, therefore, reduced compared with the GTOS.

5) Comparison of GTO-MFV With OPG in Latency: We compare the proposed GTO-MFV with OPG in latency with different task sizes $\left[10^6, 3\times10^6, 5\times10^6, 7\times10^6, 9\times10^6\right]$, as shown in Fig. 8. When the offloading data size is 10^6, the GTO-MFV shares almost the same time latency with OPG when processing the data. Along with the increase in task size, the latencies caused by the proposed GTO-MFV and OPG increase accordingly. However, it can be seen from the figure that the latencies cost by the proposed GTO-MFV are much less compared to the latencies by using OPG, which shows the superiority of the machine learning used in game theoretic offloading.

6) The Accuracy of the Proposed TL-FSCU: To display the superiority of the proposed TL-FSCU, the comparisons of TL-FSCU with existing algorithms are conducted. We consider the following baseline methods:

- iCaRL was proposed for the incremental classifier and representation learning in the area of computer vision [36]. This method solves the problem of training data with a small number of classes, which is of a similar scenario to our case that has limited features to be represented for offloading decisions in a large number of AVs.
- LwF uses only new task data to train the network while preserving the original capabilities [37]. However, our proposed TL-FSCU method can strategically update existing features with new features for offloading decisions. It is worth expecting to observe the differences in the results by comparing "only new data training" and the "strategic update" in this scalable optimization VEC scenario.
- JFSSL maps multimodal data into a common subspace for feature selection and learning in [38]. For the cross-modal retrieval requirement, this method also takes the similarity between different modalities into account for data measurement. Therefore, it is necessary to compare this method with our proposed approach.

We compare the selected algorithms with the proposed TL-FSCU to our application scenario. Fig. 9 demonstrates the superiority of the constructed TL-FSCU in accuracy. The performance of iCaRL, LwF and JFSSL show their sensitivity for the slight fluctuations along with the increase in the number of features, while the constructed TL-FSCU maintains its accuracy performance with a positive correlation to the number of feature groups. It can be seen that the TL-FSCU is more stable compared with the other three algorithms in the number of 60 feature groups, with accuracy in the range of 68% to 80%. The remarkable difference in this experiment lies in iCaRL and TL-FSCU with the number of 60 feature groups, showing that the TL-FSCU is 37% more accurate than the iCaRL. This is caused by adding all of the features in D_T to the iCaRL for the training data. The proposed TL-FSCU only needs to compare the similarity of features, all of the training data are filtered through GTO-MFV. Therefore, the comparison results show the advantage of the TL-FSCU in accuracy.

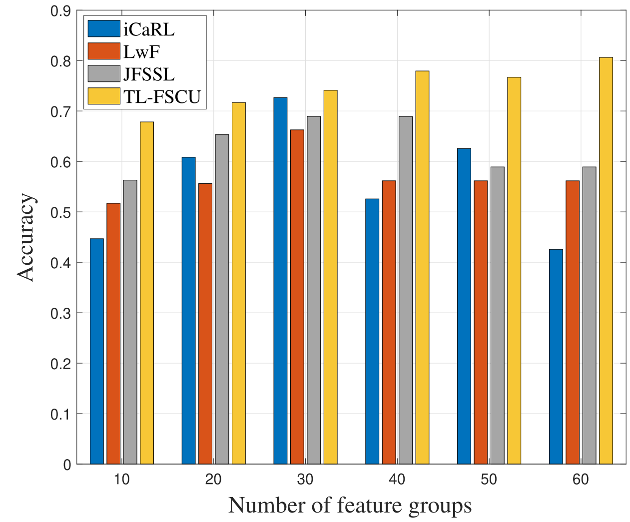


Fig. 9. Comparisons of TL-FSCU with different algorithms in accuracy.

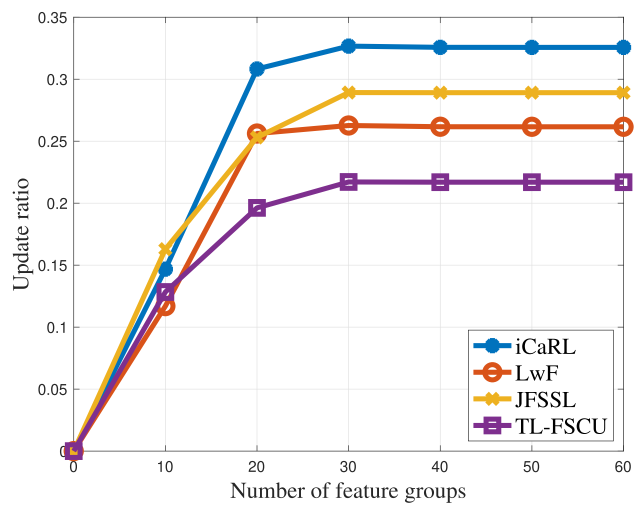


Fig. 10. Comparisons of TL-FSCU with different algorithms in features update.

7) Features Update Ratio of the Proposed TL-FSCU: Fig. 10 compares the feature update ratios in the four algorithms. With the number of 70 feature groups in this experiment, we verify the effectiveness of the TL-FSCU in scalable optimization through the transfer learning method. In the results, the features update ratio of the four algorithms shows their growth along with the increase in the number of feature groups. However, all update ratios of the algorithms keep flat after the point of 20 feature groups. The TL-FSCU displays the smallest ratio with 0.21 in the features update ratio, which means that the smallest number of features need to be updated to the transferred destination, and more resources will be conserved in the scalable optimization with this method. The other three algorithms show higher feature update ratios to different extents compared with the TL-FSCU. This is because the features need to be updated in these methods. Although the TL-FSCU considers the similarity of the selected features with the compared features before the complete update, once the similarity is confirmed, the features will not be updated and the offloading strategy of the target AVs will be determined based on the features of V_s, described by Algorithm 3. The update ratio of TL-FSCU, thus, is lower than the other methods.

8) Runtime Performance of the Proposed TL-FSCU: Table II shows the runtime performance of the three selected algorithms and the proposed TL-FSCU. We set the number of target transferred vehicles to be 2, 10 and 100, respectively. From the results, we can see that the TL-FSCU has the lowest

TABLE II
RUNTIME PERFORMANCE OF TL-FSCU WITH COMPARED ALGORITHMS

Methods	Runtime performance (s)		
	2	10	100
iCaRL	0.7286	5.2715	66.6346
LwF	0.6336	7.0447	81.2544
JFSSL	0.6624	7.8373	68.5206
TL-FSCU	0.3864	4.1972	32.1105

time cost in the process. This is because the extra time has been saved through the proposed TL-FSCU by the efficient feature-based transfer learning method.

For the same feature group of the offloading decision strategy, the TL-FSCU can recognize whether to keep the same offloading strategy or fast update features. The time cost of TL-FSCU is almost half of the time required by iCaRL and JFSSL. With the determined features from GTO-MFV, the proposed TL-FSCU can help reach the scalable optimization of game theoretical offloading in VEC rapidly.

VII. Distributed Deployment Proposal of the Proposed Method in Reality

The framework shown in Fig. 3 may not be scalable in actual scenarios due to the complexity of systems. In reality, the existing framework of federated learning can effectively help to deploy our algorithm in a distributed manner. The process can be described in Fig. 11, and steps are introduced as follows:

1) The roadside units (RSU) are stable and have high communication, computing, and cache capabilities. Therefore, they are normally used to collect traffic messages via AV-to-infrastructure communication, analyze the information locally, and then forward the analyzed result to the corresponding road users such as other AVs [39]. Such that, the NN training on solving the constructed game theoretical model is based on the local data.
2) After finishing the NN training (solving the game theoretical model with an NN) locally, the updated model is sent to the central server where no real vehicle data is involved, as step ①.
3) The transfer learning of the framework will happen in the central server where models from AVs are averaged and aggregated, as step ②.
4) The updated model is sent back to the AV and other AVs by step ③ for offloading decision-making strategies.

An AV may not just communicate with one base station when it drives on the road. In this case, if the model update is finished in (A), then the updated model will be sent back to an AV and other AVs through step ③. After that, the updated model will be distributed to new base stations/edge servers (B), (C) or (D) due to the moving of AVs. The next round of model aggregation and update will also happen in new edge servers such as (B), (C) or (D). Afterwards, the new updated model will be sent to around AVs. Therefore, the FL-FSCU, integrated with the federated learning framework, can be deployed in distributed to achieve the objective of

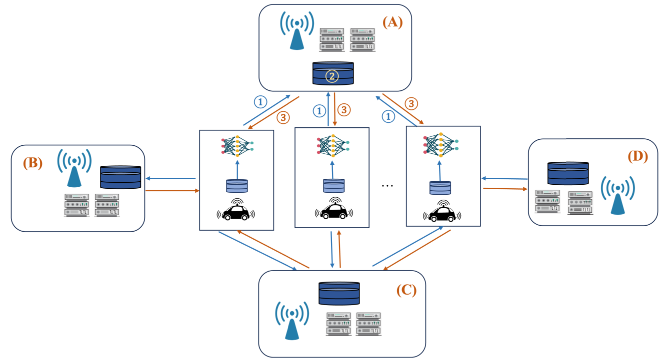


Fig. 11. The description of distributed deployment of the proposed algorithm (TL-FSCU) using a federated learning framework.

scalability in real scenarios. The distributed deployment of the proposed algorithm (TL-FSCU) using a federated learning framework is depicted in Fig. 11.

VIII. Conclusion

In this paper, we proposed a new neural network-based transfer learning approach towards scalable offloading in VEC. The offloading strategy was reached by the game theory model (GTOS). In order to reduce the computational complexity and energy consumption caused by continuous iterations in game playing and improve the scalability of offloading for the changing VEC environment, the game theoretical offloading-mean field variational process (GTO-MFV) was proposed for efficiently solving GTOS. The transfer learning framework with features selection, comparison and update (TL-FSCU) was designed to further enhance the scalability of offloading optimization in the face of new and unseen VEC environments. Experimental results demonstrated that TL-FSCU can achieve better performance in the update ratio and accuracy than existing frameworks and algorithms such as iCaRL, LwF and JFSSL.

This work can guide the market to satisfy the scalable offloading requirements through an efficient and effective approach in VEC. At the same time, the cost can be significantly reduced, and the decision features of offloading can be transferred rapidly in the scalability optimization. The efficiency of game theoretical offloading in VEC, therefore, can be improved and boosted.

In our future work, we will investigate the causes of affecting scalable optimization in offloading processes. The effective methods are expected to be further explored to deal with the challenges of scalability in offloading decisions in VEC.

Besides, we plan to employ the existing federated learning framework to implement our algorithm to achieve scalability in reality, by which the TL-FSCU can be deployed in a distributed manner effectively.

References

[1] F. Sun et al., "Cooperative task scheduling for computation offloading in vehicular cloud," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11049–11061, Nov. 2018.

[2] J. Zhang, Y. Wu, and G. Min, "System revenue maximization for offloading decisions in mobile edge computing," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–6.

[3] L. Mendiboure, M.-A. Chalouf, and F. Krief, "Edge computing based applications in vehicular environments: Comparative study and main issues," *J. Comput. Sci. Technol.*, vol. 34, no. 4, pp. 869–886, Jul. 2019.

[4] K. Li, "On the profits of competing cloud service providers: A game theoretic approach," *J. Comput. Syst. Sci.*, vol. 117, pp. 130–153, May 2021.

[5] Y. Tian, S. Min, and Q. Wu, "Application of neural network to game algorithm," *J. Comput. Commun.*, vol. 6, no. 2, pp. 1–12, 2018.

[6] H. Tembine, "Deep learning meets game theory: Bregman-based algorithms for interactive deep generative adversarial networks," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1132–1145, Mar. 2020.

[7] Md. Al Maruf, A. Singh, A. Azim, and N. Auluck, "Faster fog computing based over-the-air vehicular updates: A transfer learning approach," *IEEE Trans. Services Comput.*, vol. 15, no. 6, pp. 3245–3259, Nov. 2022.

[8] J. Guo, W. Luo, B. Song, F. R. Yu, and X. Du, "Intelligence-sharing vehicular networks with mobile edge computing and spatiotemporal knowledge transfer," *IEEE Netw.*, vol. 34, no. 4, pp. 256–262, Jul. 2020.

[9] T. Han, C. Liu, W. Yang, and D. Jiang, "Learning transferable features in deep convolutional neural networks for diagnosing unseen machine conditions," *ISA Trans.*, vol. 93, pp. 341–353, Oct. 2019.

[10] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[11] R. Assaad, M. O. Ahmed, I. H. El-adaway, A. Elsayegh, and V. S. S. Nadendla, "Comparing the impact of learning in bidding decision-making processes using algorithmic game theory," *J. Manage. Eng.*, vol. 37, no. 1, Jan. 2021, Art. no. 04020099.

[12] Y. Wang et al., "A game-based computation offloading method in vehicular multiaccess edge computing networks," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4987–4996, Jun. 2020.

[13] P. Lang, D. Tian, X. Duan, J. Zhou, Z. Sheng, and V. C. M. Leung, "Cooperative computation offloading in blockchain-based vehicular edge computing networks," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 783–798, Sep. 2022.

[14] P. Lang, D. Tian, X. Duan, J. Zhou, Z. Sheng, and V. C. M. Leung, "Blockchain-based cooperative computation offloading and secure handover in vehicular edge computing networks," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 7, pp. 3839–3853, Jul. 2023, doi: 10.1109/TIV.2023.3271367.

[15] J. Zhang, Y. Wu, G. Min, F. Hao, and L. Cui, "Balancing energy consumption and reputation gain of UAV scheduling in edge computing," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 4, pp. 1204–1217, Dec. 2020.

[16] N. Lin, H. Tang, L. Zhao, S. Wan, A. Hawbani, and M. Guizani, "A PDDQNLP algorithm for energy efficient computation offloading in UAV-assisted MEC," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 8876–8890, Dec. 2023, doi: 10.1109/TWC.2023.3266497.

[17] N. Cheng et al., "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[18] C. Chen, H. Li, H. Li, R. Fu, Y. Liu, and S. Wan, "Efficiency and fairness oriented dynamic task offloading in Internet of Vehicles," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 3, pp. 1481–1493, Sep. 2022.

[19] I. Rezek et al., "On similarities between inference in game theory and machine learning," *J. Artif. Intell. Res.*, vol. 33, pp. 259–283, Oct. 2008.

[20] H. Shiri, J. Park, and M. Bennis, "Massive autonomous UAV path planning: A neural network based mean-field game theoretic approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[21] M. Babar and M. S. Khan, "Scaledge: A framework for scalable edge computing in Internet of Things–based smart systems," *Int. J. Distrib. Sensor Netw.*, vol. 17, no. 7, 2021, Art. no. 15501477211035332.

[22] M. R. Khosravi, K. Rezaee, M. K. Moghimi, S. Wan, and V. G. Menon, "Crowd emotion prediction for human-vehicle interaction through modified transfer learning and fuzzy logic ranking," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15752–15761, Dec. 2023, doi: 10.1109/TITS.2023.3239114.

[23] P. Wang, H. Deng, J. Zhang, L. Wang, M. Zhang, and Y. Li, "Model predictive control for connected vehicle platoon under switching communication topology," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7817–7830, Jul. 2022.

[24] M. Shakeri, E. Miahi, A. Gupta, and Y.-S. Ong, "Scalable transfer evolutionary optimization: Coping with big task instances," *IEEE Trans. Cybern.*, vol. 53, no. 10, pp. 6160–6172, Oct. 2023, doi: 10.1109/TCYB.2022.3164399.

[25] P. Wang, Y. Wang, H. Deng, M. Zhang, and J. Zhang, "Multilane spatiotemporal trajectory optimization method (MSTTOM) for connected vehicles," *J. Adv. Transp.*, vol. 2020, pp. 1–15, Dec. 2020.

[26] S. A. Mohamed, S. Sorour, and H. S. Hassanein, "Group-delay aware task offloading with service replication for scalable mobile edge computing," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[27] L. Baresi, D. F. Mendonça, and M. Garriga, "Empowering low-latency applications through a serverless edge computing architecture," in *Proc. Eur. Conf. Service-Oriented Cloud Comput.* Cham, Switzerland: Springer, 2017, pp. 196–210.

[28] D. Satria, D. Park, and M. Jo, "Recovery for overloaded mobile edge computing," *Future Gener. Comput. Syst.*, vol. 70, pp. 138–147, May 2017.

[29] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102781.

[30] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[31] J. Zhang et al., "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[32] J.-D. Benamou, G. Carlier, and F. Santambrogio, "Variational mean field games," in *Active Particles*, vol. 1. Cham, Switzerland: Springer, 2017, pp. 141–171.

[33] S. G. Kwak and J. H. Kim, "Central limit theorem: The cornerstone of modern statistics," *Korean J. Anesthesiol.*, vol. 70, no. 2, p. 144, 2017.

[34] S. Uguroglu and J. Carbonell, "Feature selection for transfer learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2011, pp. 430–442.

[35] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty, and V. S. Kiran, "Similarity based feature transformation for network anomaly detection," *IEEE Access*, vol. 8, pp. 39184–39196, 2020.

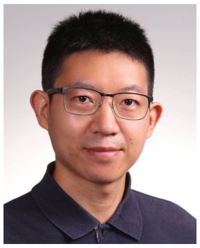
[36] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5533–5542.

[37] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[38] K. Wang, R. He, L. Wang, W. Wang, and T. Tan, "Joint feature selection and subspace learning for cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2010–2023, Oct. 2016.

[39] A. Guerna, S. Bitam, and C. T. Calafate, "Roadside unit deployment in Internet of Vehicles systems: A survey," *Sensors*, vol. 22, no. 9, p. 3190, Apr. 2022.

Juan Zhang (Member, IEEE) received the B.Eng. degree in mechanical design, manufacture and automation and the M.Sc. degree in control science and engineering from Beihang University, Beijing, China, and the Ph.D. degree in computer science from the University of Exeter, U.K., in 2022. She was a Post-Doctoral Associate with the Chair of High-Performance Computing, Helmut-Schmidt University, Hamburg, Germany. She is currently a Lecturer with the Department of Computer and Information Sciences, Northumbria University, U.K. Her research interests include mobile edge computing, decision-making strategies, the Internet of Things, and future mobile networks.



Yulei Wu (Senior Member, IEEE) received the B.Sc. degree (Hons.) in computer science and the Ph.D. degree in computing and mathematics from the University of Bradford, Bradford, U.K., in 2006 and 2010, respectively. He is currently an Associate Professor with the Faculty of Science and Engineering and the Bristol Digital Futures Institute, University of Bristol, Bristol, U.K. His research interests include digital twins and ethics-responsible decision-making and their applications on future networks, connected systems, edge computing, and digital infrastructure.

Geyong Min (Member, IEEE) received the B.Sc. degree in computer science from the Huazhong University of Science and Technology, China, in 1995, and the Ph.D. degree in computing science from the University of Glasgow, U.K., in 2003. He is currently a Professor of high-performance computing and networking with the Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, U.K. His current research interests include computer networks, wireless communications, parallel and distributed computing, ubiquitous computing, multimedia systems, modeling, and performance engineering.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University in 1985 and the Ph.D. degree in computer science from the University of Houston in 1990. He is currently a SUNY Distinguished Professor with the State University of New York. He is also a National Distinguished Professor with Hunan University, China. He has authored or coauthored more than 950 journal articles, book chapters, and refereed conference papers. He holds more than 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He is a member of the SUNY Distinguished Academy and an Academia Europaea (Academician of the Academy of Europe). He is a fellow of AAAS and AAIA. He received several best paper awards from international conferences, including PDPTA-1996, NAECON-1997, IPDPS-2000, ISPA-2016, NPC-2019, ISPA-2019, and CPSCom-2022. He is among the world's top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of the Scopus citation database. He was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis Who's Who in Science and Engineering, Who's Who in America, Who's Who in the World, and Who's Who in American Education for more than 20 consecutive years. He received the Distinguished Alumnus Award from the Department of Computer Science, University of Houston, in 2018; the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022; and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023.