

# RCFL: Redundancy-Aware Collaborative Federated Learning in Vehicular Networks

Yilong Hui<sup>✉</sup>, Member, IEEE, Jie Hu, Nan Cheng<sup>✉</sup>, Senior Member, IEEE, Gaosheng Zhao<sup>✉</sup>, Rui Chen<sup>✉</sup>, Member, IEEE, Tom H. Luan, Senior Member, IEEE, and Khalid Aldubaikhy<sup>✉</sup>, Member, IEEE

**Abstract**—In vehicular networks (VNets), vehicular federated learning (VFL) is a new learning paradigm that can protect data privacy of vehicle nodes (VNs) while training models. In VFL, the importance of data (IoD) is a key factor that affects model training accuracy. However, due to the heterogeneity of data in the VFL, it is a challenge to evaluate the quality of data owned by different VNs and design an efficient federated learning scheme to enable the VNs to complete learning tasks collaboratively. In this paper, we consider the IoD and propose a redundancy-aware collaborative federated learning (RCFL) scheme for the VFL. In the scheme, by jointly considering the data quality and the cooperation among VNs, we first design a redundancy-aware federated learning architecture to efficiently provide learning services in VNets. Then, we develop a data importance model that integrates the non-independent and identically distributed (non-IID) degree and the redundancy of data (RoD) to evaluate the data quality and formulate the cooperation of the VNs as a coalition game to improve their data importance, where the equilibrium of the coalition game is obtained by designing a coalition formation algorithm. After that, by considering the diversified characteristics of data and the available resources of different VNs in each coalition, a coalition-based federated learning algorithm is designed to enable the distributed coalitions to complete the learning task cooperatively with the target of improving the learning accuracy. The simulation results show that the proposed scheme outperforms the benchmark schemes in terms of the IoD obtained by the VNs and the training accuracy.

**Index Terms**—Federated learning, vehicular networks, data importance, coalition game.

## I. INTRODUCTION

WITH the rapid development of 6G and vehicular networks (VNets), a large number of machine learning

Manuscript received 24 February 2023; revised 19 July 2023 and 30 September 2023; accepted 7 November 2023. Date of publication 6 December 2023; date of current version 31 May 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1807700, in part by the China Post-Doctoral Science Foundation under Grant 2021TQ0260 and Grant 2021M700105, and in part by the Guangzhou Science and Technology Program under Grant 202201010870. The Associate Editor for this article was F. Xia. (Corresponding author: Nan Cheng.)

Yilong Hui is with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China, and also with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: ylhui@xidian.edu.cn).

Jie Hu, Nan Cheng, Gaosheng Zhao, and Rui Chen are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: 21181214042@stu.xidian.edu.cn; dr.nan.cheng@ieee.org; gszhao@stu.xidian.edu.cn; rchen@xidian.edu.cn).

Tom H. Luan is with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: tom.luan@ieee.org).

Khalid Aldubaikhy is with the Department of Electrical Engineering, Qassim University, Buraydah 52571, Saudi Arabia (e-mail: khalid@qec.edu.sa).

Digital Object Identifier 10.1109/TITS.2023.3336823

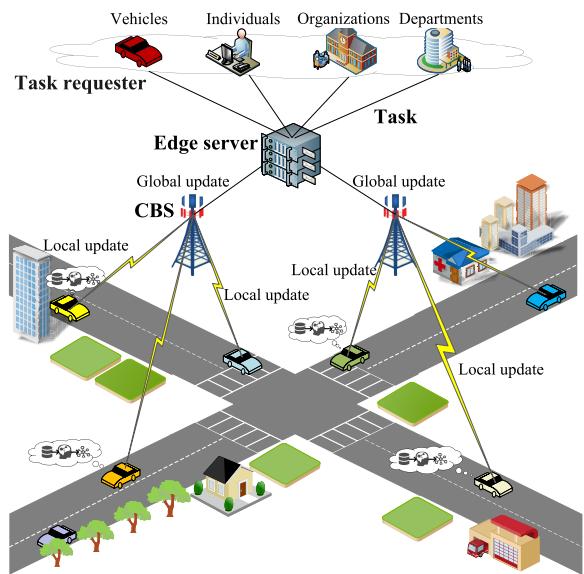


Fig. 1. The system model of the VFL.

tasks have been generated to facilitate intelligent transportation systems (ITS) [1]. As a new learning architecture, federated learning has attracted widespread attention to support various applications and services [2]. Different from traditional machine learning schemes which need to transmit the collected data to a central server for training, federated learning allows each data owner to locally train the collected data and upload the local parameters to the central server, reducing communication overhead and improving data security concurrently [3]. With these advantages, a new paradigm called vehicular federated learning (VFL) has been proposed to facilitate the applications and services in VNets [4], [5], [6].

In the VFL, as shown in Fig. 1, if a task requester intends to complete a learning task, it can send its request to the central server which will select a number of vehicle nodes (VNs) to train the learning task locally. After a round of training, the local updates generated by different VNs will be collected by the central server to update the global model. After this, the central server broadcasts the updated global model to the VNs, beginning the next round of local training. Repeat this process until a certain precision is obtained or the maximum number of the iteration is reached [7].

In order to further promote the application of VFL and improve the accuracy of model training, a large number of

VFL schemes from various perspectives have been proposed [8], [9], [10]. However, few of them consider the following three factors simultaneously to improve the model accuracy of VFL. First, in the model training process, the VNs with different data have different contributions to the global model. This is because the number and characteristics of data owned by different VNs may vary greatly if the VNs generate and collect data at different locations and in different ways. As a result, the data of VFL is non-independent and identically distributed (non-IID). For neural networks trained on highly skewed non-IID data, the accuracy will be significantly reduced [11]. Based on this fact, the current works usually use average earth mover's distance (EMD) to measure the importance of data (IoD). However, due to the constraints of road layout in the VNets, the VNs on the same road section may obtain the same data. In addition, a VN may share its data with other VNs. Obviously, the redundant data will reduce the IoD and the accuracy of the training model [12], [13], [14]. Therefore, a new data importance model needs to be designed to efficiently improve the model training accuracy. Second, in the VFL, the existing works typically think that the VNs independently train the model because they will train local models with their own data [15], [16], [17], which lacks the consideration of the cooperation among different VNs to increase their data importance. In general, the cooperation of VNs with different degrees of IoD will result in different model training accuracies [18]. Therefore, it is a key problem to design a mechanism for the VNs to facilitate the VFL by considering the IoD and the cooperation among VNs. Third, in the traditional federated learning architecture, the edge server needs to collect the local updates of all the VNs to complete a round of global aggregation [4]. However, VNs have various data volumes and available computing resources. Therefore, the time for each VN to complete a round of local training is different [19]. This leads to that the time of a round of local training depends on the VN that trains the slowest. Thus, based on the IoD and the cooperation among VNs, a new VFL algorithm needs to be designed to address this problem with the target of improving the model training efficiency.

To address the above problems, we considered the quality of data in our previous work [20] and proposed a collaborative data pricing scheme based on data quality. However, [20] only focuses on the impact of data quality on data transactions, while neglecting the training and communication capabilities of VNs. In addition, [20] lacks consideration for the training time of VNs, that is, it does not group VNs with similar training time to improve training accuracy. Therefore, on the basis of [20], this paper focuses on the impact of data importance on training accuracy, and proposes a redundancy-aware collaborative federated learning (RCFL) scheme in VNets. Specifically, we first establish a RCFL architecture in VNets by considering the IoD and the cooperation among VNs. Then, based on the non-IID of data and the redundancy of data (RoD), we design a new data importance model to evaluate the quality of data and model the interactions among the VNs as a coalition game to improve their data importance, where a coalition formation algorithm is designed to enable each VN to obtain its optimal coalition. After that, by considering the diversified

characteristics of data and the available resources of different VNs in each coalition, a coalition-based VFL algorithm is designed to facilitate the model training services completed by the distributed coalitions. The main contributions of our paper are four-fold.

- **Architecture Establishment:** We propose a RCFL architecture in VNets by considering the IoD and the cooperation among VNs. With the designed architecture, the VNs can efficiently cooperate with each other based on their data importance to provide learning services.
- **Game Formulation:** We develop a novel model to measure the IoD and formulate the interactions among the VNs as a coalition game. In addition, we design an importance-aware coalition formation algorithm to improve the data importance of each coalition.
- **Algorithm Design:** We propose a coalition-based VFL algorithm by jointly considering the diversified characteristics of data and the available resources of different VNs in each coalition. With the designed algorithm, the local updates of each learning task can be efficiently aggregated to increase the model training accuracy.
- **Performance Evaluation:** We evaluate the performance of the proposed RCFL using extensive simulations. The simulation results show that the proposed scheme can increase the average data importance of the VNs and improve the model training accuracy.

The remainder of this paper is organized as follows. Section II discusses the related works. Section III describes the system model. Section IV designs the proposed RCFL scheme in detail. The simulation results are presented in Section V. Finally, we conclude the paper in Section VI.

## II. RELATED WORK

### A. Federated Learning

Recently, the design of federated learning schemes has attracted a lot of attentions. Ye et al. [21] present a selective model aggregation approach for image classification tasks in VNets, where the accuracy and efficiency can be improved by evaluating the image quality and computation capability. Yang et al. [22] propose an asynchronous federated learning framework to improve the model learning accuracy. In this scheme, a device selection strategy is designed to keep the low-quality devices from affecting the learning accuracy. Zeng et al. [23] design a novel dynamic federated proximal algorithm to capture the diverse local data quality among VNs, where the unbalanced and non-IID data across VNs are considered. Ayaz et al. [24] propose a federated learning message dissemination solution to address the challenges caused by high vehicle density and mobility. The proposed solution can potentially lead to a more accurate model as compared to other consensus algorithms. Kang et al. [25] develop a reliable worker selection scheme to select trusted and reliable workers for federated learning tasks, where the reputation is introduced as a metric to select workers. Xiao et al. [26] formulate a min-max optimization problem to achieve the minimum cost in the worst case of VFL. To solve the problem, a greedy algorithm is designed to dynamically select VNs with higher

image quality. Elbir et al. [27] present the advantages of federated learning, centralized learning, and hybrid federated and centralized learning for training vehicular models, where the performance of different learning frameworks is evaluated on object detection problem in vehicular networks.

Unlike these studies on federated learning in VNets, in our paper, we focus on the data importance in the VFL and jointly consider the non-IID and RoD to design a novel data importance model for VFL. Based on the analysis of data importance, we further model the cooperation among VNs as a coalition game and design a coalition-based VFL algorithm to facilitate the local training, where the data importance of each VN can be improved and the model training accuracy of the VFL can be increased.

### B. Coalition Game

In VNets, the application of coalition game has been extensively studied to facilitate various applications and services. By jointly considering the throughput requirements and the transmission power constraints, Xu et al. [28] propose a two-stage framework to protect the legitimate links, where the secure transmission problem is formulated based on an overlapping coalition formation game. Hu et al. [29] present a novel autonomous client participation scheme to motivate the clients participating in the federated learning process. In their scheme, a coalition-based minority game is designed to reduce the volatility in each round. Based on a coalition game, Hui et al. [30] propose a distributed driving scheme to facilitate the collaborative autonomous driving. The simulation results show that the scheme can minimize the autonomous driving cost of each vehicle. Wu et al. [31] propose a cooperative content caching scheme based on a profit-aware coalition game. In the scheme, a dynamic coalition algorithm is designed to guide each member to join or leave the coalition. Su et al. [32] develop a novel vehicular content distribution scheme in 5G heterogeneous networks, where the coalition game is designed to determine the optimal content distribution strategy. Qi et al. [33] develop a coalition formation game in heterogeneous UAV networks. Through partial cooperation among coalition members, a bilateral mutual benefit transfer order is proposed to optimize the resource allocation. Ruan et al. [34] present a clustering-based localization scheme for UAV swarms. In their scheme, the clustering-based problem is formulated as a coalition formation game to evaluate the tradeoff between intracluster cooperation and intercluster packet loss.

Different from the coalition game models proposed in above schemes, our paper proposes a new data importance model and designs a coalition game to formulate the cooperation among the VNs to facilitate the VFL services. In addition, by using the proposed coalition formation algorithm, the VNs can cooperate with each other to maximize their data importance and improve the accuracy of federated learning tasks.

## III. SYSTEM MODEL

In the VNets, as shown in Fig. 1, the networks consist of task requesters (TRs), cellular base stations (CBSs), VNs and an edge server.

### A. TRs

In the VNets, the VNs, individuals, organizations, and departments can become the task requesters to request learning models. Let  $\mathbb{J} = \{1, \dots, j, \dots, J\}$  denote the set of the learning tasks in the VNets. If a task requester has a learning task needs to be completed, it can publish the task to the edge server. With the task request, the edge server will select a number of VNs to complete the learning task. After the task is completed, the trained model then can be delivered to the task requester to obtain rewards.

### B. CBSs

The CBS in the network is connected to the edge server through high-speed wired connection. In addition, within the coverage of each CBS, the VNs can communicate with the CBS through the vehicle to infrastructure (V2I) communication. In this way, the VNs can upload the trained local model parameters to the edge server through the CBS. At the same time, the edge server can distribute the global model to the VNs through the CBS to complete the local training.

### C. VNs

In the VNets, each VN refers to the vehicle that can cache the collected data and communicate with roadside infrastructures using the V2I communication. Let  $\mathbb{I} = \{1, \dots, i, \dots, I\}$  represent the set of VNs in the VNets. If a VN plans to participate in the training process, the VN sends its data information to the edge server. For VN  $i$  that intends to train learning task  $j$ , its dataset and the amount of data can be denoted as  $\mathbb{D}_i^j$  and  $|\mathbb{D}_i^j|$ . For data  $d_i^j$  ( $d_i^j \in \mathbb{D}_i^j$ ), it can be expressed as  $d_i^j = \{x_{d_i^j}, y_{d_i^j}\}$ , where  $x_{d_i^j}$  is the feature vector and  $y_{d_i^j}$  is the corresponding label. Let  $\phi_i$  denote the CPU cycle frequency of VN  $i$ . Then, the computation time spends on completing a local iteration can be expressed as [35]

$$\overline{t_i^{cmp}} = \frac{c_i |\mathbb{D}_i^j|}{\phi_i}, \quad (1)$$

where  $c_i$  is the number of CPU cycles for VN  $i$  to perform one data sample. Based on the above data information, VN  $i$  that intends to participate in training task  $j$  sends the information about its task training capability to the edge server. The training capability of VN  $i$  for task  $j$  can be expressed as

$$\Gamma_i^j = \left( \mathbb{L}_i^j || n_{l_i^j} || H(d_i^j) || t_i^j \right), \quad \forall l_i^j \in \mathbb{L}_i^j, \quad \forall d_i^j \in \mathbb{D}_i^j, \quad (2)$$

where  $\mathbb{L}_i^j = \{1, \dots, l_i^j, \dots, L_i^j\}$  is the set of labels owned by VN  $i$ .  $n_{l_i^j}$  is the number of data of label  $l_i^j$ , we have  $|\mathbb{D}_i^j| = \sum_{l_i^j \in \mathbb{L}_i^j} n_{l_i^j}$ .  $H(d_i^j)$  is the hash value of data  $d_i^j$ .  $t_i^j$  is the time for VN  $i$  to train the local dataset  $E^j$  times.

### D. Edge Server

The edge server is the controller of the VFL which caches the data information of each VN. If the edge server receives a learning task, it needs to select a number of VNs to train the

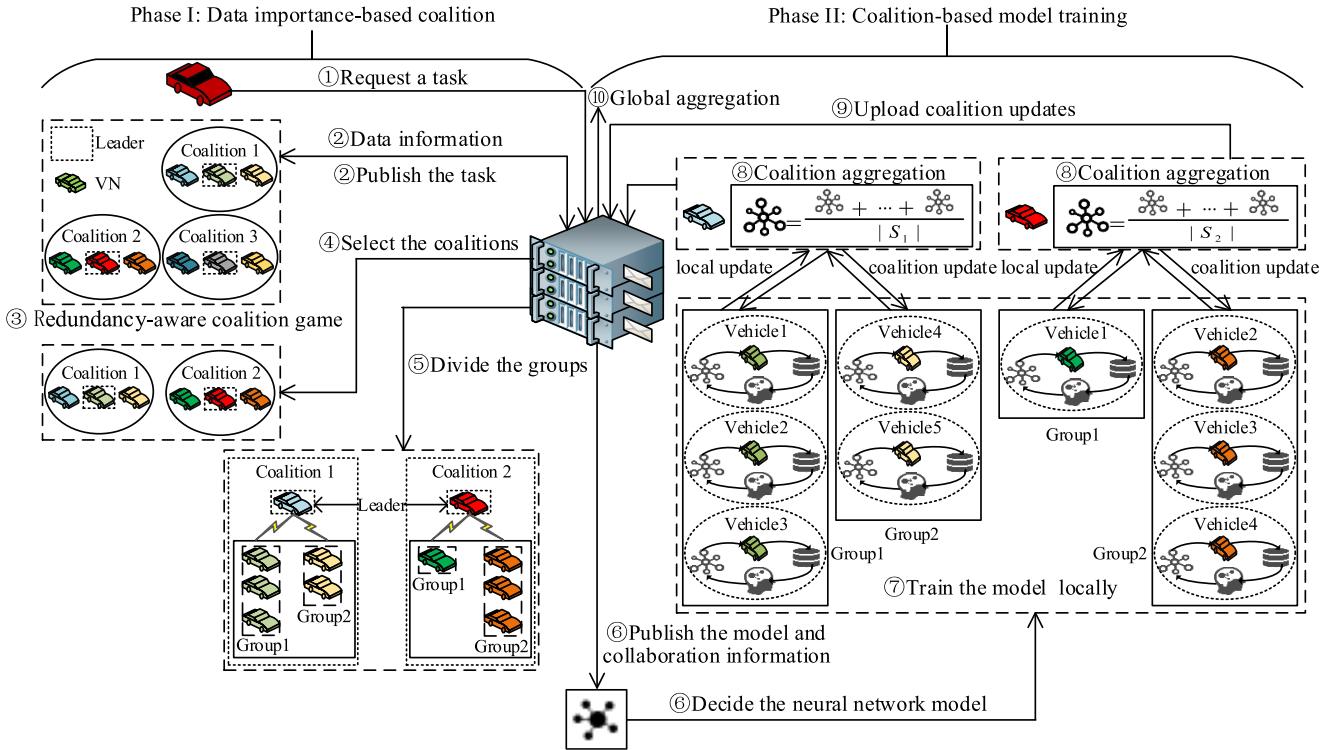


Fig. 2. The architecture of the RCFL.

model. Similar to existing studies [36], [37], [38], we adopt the free flow traffic model, where the speed and direction of each VN remain unchanged during the stay period in the coverage of the edge server. Specifically, the edge server covers a road section of length  $W$  and the VNs follow a Poisson process with a rate of  $\lambda$  to reach the covered road section. The distance from the position of VN  $i$  to the entrance of the communication coverage is  $\sigma_i$  ( $\sigma_i \in [0, W]$ ) in the  $k$ -th round of global training. The speed  $v_i$  of each VN  $i$  is generated by a truncated Gaussian distribution, where the minimum speed and the maximum speed are  $v_{\min}$  and  $v_{\max}$ , respectively. Therefore, its residence time in the coverage of the edge server can be expressed as

$$t_i^{stay} = \frac{W - \sigma_i}{v_i}. \quad (3)$$

Based on  $t_i^{stay}$ , the VN will make a decision on whether to join the current round of model training. Specifically, if  $t_i^{stay} \geq t_i^{cmp} + t_i^{down} + t_i^{up}$ , the VN sends training capability information to the edge server, where  $t_i^{cmp}$  is the calculation time spent on training the local dataset  $E^j$  times. It can be calculated by  $t_i^{cmp} = E^j t_i^{cmp}$ .  $t_i^{down}$  is the communication time of VN  $i$  to obtain the model parameters, and  $t_i^{up}$  is the communication time for VN  $i$  to upload model parameters. In contrast, if the VN leaves the coverage range in the next round of local training (i.e.,  $t_i^{stay} < t_i^{cmp} + t_i^{down} + t_i^{up}$ ), it will not be able to participate in the model training process. In addition, for VNs newly entering the coverage range of the edge server, they will make a decision on whether to join the model training in the next round based on their driving speed.

In the model training process, the edge server manages the local model parameters and generates the next round global model parameters. For learning task  $j$ , the initial model can be represented as

$$j = \langle \omega^0 || E^j || \eta^j \rangle, \quad (4)$$

where  $\omega^0$  is the initial parameters of the global model,  $E^j$  is the number of iterations to train the local dataset, and  $\eta^j$  is the learning rate.

#### IV. REDUNDANCY-AWARE COLLABORATIVE FEDERATED LEARNING

In this section, we first introduce the RCFL architecture. Then, we model the IoD and formulate the cooperation among VNs as a coalition game. After that, we design the coalition-based VFL algorithm to improve the model training accuracy.

##### A. RCFL Architecture

As shown in Fig. 2, the architecture consists of the following two phases, namely, data importance-based coalition phase and coalition-based model training phase.

###### *Phase I: Data Importance-Based Coalition:*

- 1) If a task requester plans to complete task  $j$ , it first sends task  $j$  to the edge server by connecting the CBS.
- 2) The edge server sends the task information to the VNs within the communication coverage. Then, the VNs choose whether to join the task training process based on their own data resources. If VN  $i$  plans to train task  $j$ , it sends its training capability information  $\Gamma_i^j$  to the edge server.

**3)** After receiving the training ability information of the VNs, the coalition game will be performed based on the data importance model. For each VN  $i$  ( $i \in \mathbb{I}$ ), it can form a coalition to train the model individually or with other VNs to increase its IoD. The details of the data importance model and the coalition game among the VNs will be discussed in Section IV-B. With the designed coalition game, we can obtain the final coalition partition  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_M\}$ , where  $M$  is the number of coalitions formed by the VNs.

**4)** Since different coalitions have different degrees of IoD, the edge server first sorts all the coalitions in  $\mathbb{S}$  in descending order according to the data importance of each coalition. Then, the edge server selects a part of coalitions or all the coalitions to form the optimal coalition set (OCS)  $\mathbb{S}^* = \{S_1^*, \dots, S_m^*, \dots, S_{\lfloor \xi^j M \rfloor}^*\}$ , where  $\xi^j$  ( $0 < \xi^j \leq 1$ ) is the proportion to select the coalitions. For coalition  $S_m^*$ , the VN that has the most computing resources will be selected as the leader of the coalition.

**5)** After obtaining the OCS, we divide all the VNs in set  $\mathbb{S}^*$  into several groups, where the VNs in the same group spend similar time to complete  $E^j$  local iterations. Note that the VNs in the same group may belong to different coalitions. Let  $\mathbb{G} = \{G_1, \dots, G_z, \dots, G_Z\}$  represent the set of all the groups, where  $Z$  denotes the number of groups formed by the VNs.

#### Phase II: Coalition-Based Model Training:

**6)** After determining the group and the coalition for each VN  $i$  ( $i \in \mathbb{I}$ ), the edge server initializes the federated learning model and broadcasts the model parameters, coalition information and group information to all the selected VNs.

**7)** In the  $k$ -th ( $1 \leq k \leq K$ ) round of global training, each VN  $i$  in coalition  $S_m^*$  ( $\forall S_m^* \in \mathbb{S}^*$ ) trains the local model with its dataset. If VN  $i$  completes  $E^j$  local iterations, the VN sends the local model parameters to the coalition leader.

**8)** Based on the training time of each group in coalition  $S_m^*$ , the coalition leader aggregates local model parameters. Specifically, if the coalition leader receives the parameters from only one group, it uses the intra-group aggregation to aggregate the local model parameters. After the parameters are aggregated, the coalition leader will send the updated model to the VNs in this group. In contrast, if the coalition leader receives the parameters from more than one group, the inter-group aggregation will be used and the new model will be delivered to the VNs in these groups.

**9)** After all the VNs in set  $\mathbb{S}^*$  have completed at least  $E^j$  local iterations (i.e., the group with the longest training time completes one round of  $E^j$  local iterations), the coalition leader of each coalition  $S_m^*$  receives the local training parameters of all the VNs in this coalition. At this time, the coalition leader then can aggregate these parameters into coalition parameters and send them to the edge server.

**10)** The edge server aggregates the coalition parameters from different coalitions to generate the global model. Furthermore, it distributes the global model to the VNs in set  $\mathbb{S}^*$  to start the next round of the global model training. The above coalition-based model training process will be discussed in detail in Section IV-C.

#### B. Importance-Aware Coalition Game

In the RCFL, the VNs with different degrees of IoD have different contributions to the global training. Therefore, we develop a model to evaluate the IoD before discussing the cooperation among the VNs. In general, the data owned by different VNs are non-IID [39]. Furthermore, the VNs on the same road section may obtain the same data or share data with each other, where the redundant data will reduce the IoD and the accuracy of the training model. Therefore, we define the data importance by jointly considering two factors, i.e., non-IID and RoD.

**1) Non-IID:** Based on the studies in [40], [41], and [42], EMD can be adopted to measure the distribution heterogeneity of the data among different VNs. We assume that task  $j$  is an  $L$ -class classification problem. Then, the non-IID of coalition  $S_m$  can be defined as

$$B_{S_m}^j = \sum_{l^j \in \mathbb{L}^j} ||p_{S_m}(l^j) - p(l^j)||, \quad (5)$$

where  $\mathbb{L}^j$  is the set of labels owned by all the VNs.  $p_{S_m}(l^j)$  is the distribution of the data with label  $l^j$  in coalition  $S_m$ . It can be expressed as

$$p_{S_m}(l^j) = \frac{\sum_{i \in S_m} \sum_{l_i^j \in \mathbb{L}_i^j} \gamma_{l_i^j l^j} n_{l_i^j}}{\sum_{i \in S_m} |\mathbb{D}_i^j|}, \quad (6)$$

where  $\gamma_{l_i^j l^j} = 1$  if label  $l_i^j$  of VN  $i$  and label  $l^j$  are the same. Namely, we have

$$\gamma_{l_i^j l^j} = \begin{cases} 1, & l_i^j = l^j, \\ 0, & l_i^j \neq l^j. \end{cases} \quad (7)$$

Similarly,  $p(l^j)$  in (5) is the distribution of the data with label  $l^j$  in set  $\mathbb{S}$ . We have

$$p(l^j) = \frac{\sum_{i \in \mathbb{I}} \sum_{l_i^j \in \mathbb{L}_i^j} \gamma_{l_i^j l^j} n_{l_i^j}}{\sum_{i \in \mathbb{I}} |\mathbb{D}_i^j|}. \quad (8)$$

**2) RoD:** In our paper, the RoD refers to the proportion of redundant data in the dataset of each coalition. For coalition  $S_m$ , the data redundancy can be defined as

$$R_{S_m}^j = 1 - \frac{|\mathbb{D}_{S_m}^*|}{\sum_{i \in S_m} |\mathbb{D}_i^j|}, \quad (9)$$

where  $\mathbb{D}_{S_m}^*$  is the non-duplicate dataset of coalition  $S_m$ ,  $|\mathbb{D}_{S_m}^*|$  is the amount of non-duplicate data. To calculate the RoD in coalition  $S_m$ , the edge server needs to compare the hash values of the data uploaded by the VNs in this coalition. If a data uploaded by VN  $i$  and a data uploaded by VN  $i'$  have the same hash value, the two data will be considered redundant. For data  $d_i^j$  owned by VN  $i$ , its hash value  $H(d_i^j)$  can be generated by a hash algorithm. Fig. 3 shows the number of redundant data by changing the RoD of each label. In this simulation, we consider the MINIST dataset and assume that both VN  $i$  and VN  $i'$  have 10 labels, where each label has 10 data.

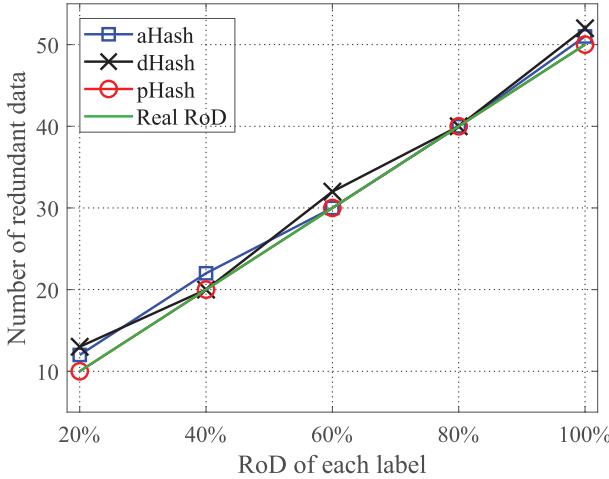


Fig. 3. The number of redundant data by changing the RoD of each label.

As shown in this figure, the green line represents the real RoD, and the blue line, the black line and the red line represent the RoD results obtained by average hash (aHash), difference hash (dHash) and perceptual hash (pHash), respectively. It can be seen from Fig. 3 that pHash can accurately obtain the true RoD between the two VNs. Thus, we use the pHash algorithm to generate the hash value of the data. In this way, by comparing the hash values of all the data uploaded by the VNs, the edge server can regard data with the same hash value as the redundant data without requiring the VNs to upload real data.

Based on the definitions of non-IID and RoD, the IoD of coalition  $S_m$  can be defined as

$$V_{S_m}^j = \alpha^j(-\log(\rho_B^j B_{S_m}^j + \varpi_B^j) + a^j) + \vartheta^j(-\log(\rho_R^j R_{S_m}^j + \varpi_R^j) + b^j), \quad (10)$$

where  $\varpi_B^j$  ( $\varpi_B^j > 0$ ) and  $\varpi_R^j$  ( $\varpi_R^j > 0$ ) are small positive numbers.  $\alpha^j$  ( $0 < \alpha^j < 1$ ) and  $\vartheta^j$  ( $0 < \vartheta^j < 1$ ) are the weights of the EMD and the RoD, respectively.  $\rho_B^j$  ( $0 < \rho_B^j < 1$ ) and  $\rho_R^j$  ( $0 < \rho_R^j < 1$ ) are the factors used to balance the influence of the EMD and the RoD on the IoD.  $a^j$  and  $b^j$  are the factors that ensure the positive impact of EMD and RoD on the IoD of the coalition.

According to (10), each VN can improve its IoD by forming a coalition with other VNs. Thus, the IoD of VN  $i$  can be defined as

$$V_i^j = \begin{cases} V_{S_m}^j, & i \in S_m, \\ V_{\{i\}}^j, & \text{else}, \end{cases} \quad (11)$$

where  $V_{\{i\}}^j$  denotes that VN  $i$  forms a coalition that only includes itself. According to the definition of  $V_i^j$ , the IoD of all the VNs can be defined as

$$V_{\mathbb{S}}^j = \sum_{i \in \mathbb{I}} V_i^j. \quad (12)$$

Based on the definitions of the IoD of each VN and each coalition, we then model the interactions among the VNs in the coverage of the CBS as a coalition game through which each VN can find its optimal coalition to enhance the IoD of each coalition.

**Definition 1 (Coalition partition):** The set of coalitions  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_M\}$  is called a coalition partition if  $S_m \cap S_{m'} = \emptyset, \forall m \neq m'$  and  $\bigcup_{m=1}^M S_m = \mathbb{I}$ .

**Definition 2 (Stable partition):** A coalition partition  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_M\}$  is a stable partition if no VN  $i$  ( $\forall i \in S_m$ ) has the incentive to change the current partition  $\mathbb{S}$  by leaving the current coalition  $S_m$  and joining another coalition  $S_{m'}$ , where  $S_m \cap S_{m'} = \emptyset, \forall m \neq m'$ .

**Definition 3 (Coalition condition):** For each VN  $i$  in coalition  $S_m$  and two coalition partitions  $\mathbb{S}' = \{S_1, \dots, S_m \setminus \{i\}, \dots, S_{m'} \cup \{i\}, \dots, S_M\}$  and  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_{m'}, \dots, S_M\}$ , if  $V_{\mathbb{S}'}^j > V_{\mathbb{S}}^j$ , then VN  $i$  will leave coalition  $S_m$  and join coalition  $S_{m'}$ .

Based on the above definitions, we then design an importance-aware coalition game to promote the cooperation among VNs and increase the IoD of each coalition. In the initial stage of the game, each VN forms a coalition independently. In other words, the VNs are divided into  $I$  coalitions, where each coalition has only one VN. We have  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_M\} = \{\{1\}, \dots, \{i\}, \dots, \{I\}\}$ . For each selected VN  $i$  ( $i \in S_m$ ) in set  $\mathbb{S}$ , it has a candidate coalition sequence  $SQ_i$ . To determine  $SQ_i$ , we consider the fact that VNs need to transmit model parameters to its coalition leader and define the constraint condition  $|\sigma_i - \sigma_{i'}| \leq \theta$  for dividing VN  $i$  and VN  $i'$  into the same coalition, where  $\theta$  is the effective transmission distance between the VNs [43]. Then, the sequence can be defined as a set of coalitions that contains all the coalitions in the current coalition partition  $\mathbb{S}$  except  $S_m$  and the coalitions that the leaders of these coalitions cannot satisfy condition  $|\sigma_i - \sigma_{i'}| \leq \theta$ , shown as

$$SQ_i = \{S_1, \dots, S_{m-1}, S_{m+1}, \dots, S_M\}. \quad (13)$$

Then, based on the IoD defined in (10) and the candidate sequence, the process of VN  $i$  joining a coalition can be divided into the following three cases.

**Case 1:** If no coalition satisfies the coalition condition in sequence  $SQ_i$ , VN  $i$  will stay in the original coalition  $S_m$ .

**Case 2:** If only one coalition  $S_{\bar{m}}$  satisfies the coalition condition in sequence  $SQ_i$ , VN  $i$  will leave the original coalition  $S_m$  and join the new coalition  $S_{\bar{m}}$ .

**Case 3:** If more than one coalition satisfies the coalition condition in sequence  $SQ_i$ , VN  $i$  will leave the original coalition  $S_m$  and join the coalition that maximizes the IoD of the new coalition partition. The coalition can be determined by

$$S_{\bar{m}} = \arg \max_{S_{m'} \in \mathbb{S}_{\text{temp}}} (\sum_{i \in \mathbb{I}} V_i^j), \quad (14)$$

where  $\mathbb{S}_{\text{temp}}$  is the set of coalitions that satisfy the coalition condition.

Based on the above cases, we then design an importance-aware coalition formation algorithm for obtaining the optimal coalition partition. The details of the coalition formation algorithm are shown in Algorithm 1.

**Proposition 1:** The coalition partition  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_M\}$  obtained by using Algorithm 1 is a stable partition that maximizes the IoD of the VNs.

**Algorithm 1** The Importance-Aware Coalition Formation Algorithm

```

1: Input:  $\mathbb{I}, \Gamma_i^j, \varpi_B^j, \varpi_R^j, \alpha^j, \vartheta^j, \rho_B^j, \rho_R^j, a^j, b^j$ 
2: Initialization:  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_M\} = \{\{1\}, \dots, \{i\}, \dots, \{I\}\}$ 
3: Loop:  $\varrho = 0, \mathbb{S}_{\text{temp}} = \emptyset$ 
4: Select a VN  $i$  ( $i \in S_m$ ) and obtain its candidate coalition sequence  $SQ_i$  according to (13)
5: Calculate  $V_i^j$  and  $V_{\mathbb{S}}^j$  of the current coalition partition  $\mathbb{S}$  according to (11) and (12), respectively
6: for  $S_{m'} : S_{m'} \in SQ_i$  do
7: Calculate  $V_{\mathbb{S}'}^j$  of the new coalition partition  $\mathbb{S}'$  according to (12), where  $\mathbb{S}' = \{\mathbb{S} \setminus S_m\} \cup \{S_m \setminus \{i\}\} \cup \{\mathbb{S} \setminus S_{m'}\} \cup \{S_{m'} \cup \{i\}\}$ 
8: if  $V_{\mathbb{S}'}^j > V_{\mathbb{S}}^j$  then
9:      $\varrho = \varrho + 1$ 
10:     $\mathbb{S}_{\text{temp}} = \mathbb{S}_{\text{temp}} \cup S_{m'}$ 
11: end if
12: end for
13: if  $\varrho = 1$  then
14:     The coalition selected by VN  $i$  is coalition  $S_{\bar{m}}$ , where  $\mathbb{S}_{\text{temp}} = \{S_{\bar{m}}\}$ 
15:     Update coalition partition  $\mathbb{S}' = \{\mathbb{S} \setminus S_m\} \cup \{S_m \setminus \{i\}\} \cup \{\mathbb{S} \setminus S_{\bar{m}}\} \cup \{S_{\bar{m}} \cup \{i\}\}$ 
16:      $\mathbb{S} = \mathbb{S}'$ 
17: end if
18: if  $\varrho \geq 1$  then
19:     The VN  $i$  selects coalition  $S_{\bar{m}}$  using (14)
20:     Update coalition partition  $\mathbb{S}' = \{\mathbb{S} \setminus S_m\} \cup \{S_m \setminus \{i\}\} \cup \{\mathbb{S} \setminus S_{\bar{m}}\} \cup \{S_{\bar{m}} \cup \{i\}\}$ 
21:      $\mathbb{S} = \mathbb{S}'$ 
22: end if
23: End loop: Until the coalition partition does not change
24: Output: The stable coalition partition  $\mathbb{S}$ 

```

The proof of this proposition is given in Appendix A.

After obtaining the final stable coalition partition  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_M\}$ , the edge server can select some or all the coalitions to form the OCS according to the training requirement. If the edge server plans to select some coalitions to form the OCS, it first arranges the coalitions in  $\mathbb{S}$  in descending order according to the IoD of each coalition. Then, the edge server selects a number of coalitions based on  $\lfloor \xi^j M \rfloor$  to form the OCS  $\mathbb{S}^* = \{S_1^*, \dots, S_m^*, \dots, S_{\lfloor \xi^j M \rfloor}^*\}$ .

### C. Coalition-Based VFL Algorithm

Based on the coalition partition formed by the VNs in Section IV-B, we first group all the VNs within the coverage of the CBS to prepare for the design of the coalition-based VFL algorithm. In the  $k$ -th round of global training, each coalition leader needs to aggregate the local model parameters of all the VNs. However, the computing resources and the data volume of the VNs in the coalition are different. Therefore, the VNs with short training time need to wait for the VNs with long training time before receiving the model parameters returned by the coalition leader. By considering this fact, we propose

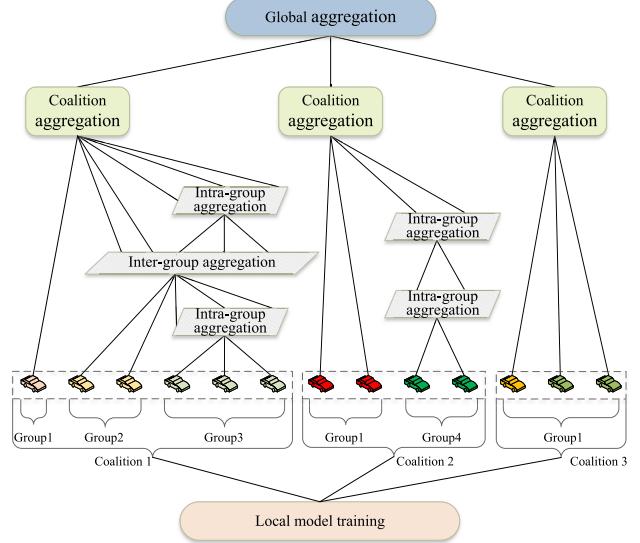


Fig. 4. The architecture of the coalition-based VFL algorithm.

a strategy based on training time to group all the VNs within the coverage of the CBS. Specifically, we use the number of local iterations  $E^j$  in a round of global training as the reference value to group the VNs. That is, each VN trains its own dataset at least  $E^j$  times in the  $k$ -th round of global training. Based on (1), the training time for VN  $i$  to complete  $E^j$  local iterations can be given by

$$t_i^j = t_i^{cmp} + t_i^{down} + t_i^{up}. \quad (15)$$

For the VNs in all the coalitions, the VN  $i^*$  with the longest training time only needs to complete  $E^j$  local iterations in each round of global training. We have

$$i^* = \arg \max_{i \in \mathbb{S}^*} t_i^j. \quad (16)$$

In contrast, the number of local iterations required by other VNs in each round is an integer multiple of  $E^j$ . For VN  $i$  ( $\forall i \neq i^*$ ), this multiple can be calculated by

$$\mu_i = \left\lfloor \frac{t_{i^*}^j}{t_i^j} \right\rfloor, \quad (17)$$

where  $\lfloor \cdot \rfloor$  is the floor function. Based on (17), if  $\mu_i = \mu_{i'}$ , then VN  $i$  and VN  $i'$  belong to the same group. Since the VNs are grouped by using the floor function, the training time of the VNs in the same group is similar but different. Therefore, for group  $G_z$ , we record the time of the VN with the longest training time as the training time of this group. Namely, we have

$$t_{G_z}^j = \max(t_i^j), \forall i \in G_z. \quad (18)$$

From (18), the local model parameters of the VNs in group  $G_z$  will be aggregated by the coalition leader every  $t_{G_z}^j$ .

After each VN determines its group, we then introduce the proposed coalition-based VFL algorithm. As shown in Fig. 4, the algorithm consists of the following three steps.

1) *Local Model Training*: In the  $k$ -th round of global training, if VN  $i$  completes  $E^j$  local iterations, the VN will send its local model parameters to the coalition leader. Therefore, each VN  $i$  will send its local parameters  $\mu_i$  times during the  $k$ -th round of global training. In each round of global training, we use  $e_i$  ( $e_i \in [1, \mu_i]$ ) to denote that VN  $i$  is executing the  $e_i$ -th round of  $E^j$  local iterations. Namely, the training sequence of VN  $i$  at this time is  $e_i$ . By doing this, the local model parameters of VN  $i$  can be updated by

$$\omega_{e_i}^k = \omega_{e_i-1}^k - \eta^j \nabla f_i(\omega_{e_i-1}^k), \quad (19)$$

where  $f_i(\omega_{e_i-1}^k)$  and  $\nabla f_i(\omega_{e_i-1}^k)$  are the loss function and the gradient of the loss function. We have

$$f_i(\omega_{e_i-1}^k) = \frac{\sum_{d_i^j \in \mathbb{D}_i^j} f_i(x_{d_i^j}, y_{d_i^j}, \omega_{e_i-1}^k)}{|\mathbb{D}_i^j|}, \quad (20)$$

where  $f_i(x_{d_i^j}, y_{d_i^j}, \omega_{e_i-1}^k)$  is the loss function of data  $d_i^j = \{x_{d_i^j}, y_{d_i^j}\}$ .

2) *Coalition Aggregation*: Let  $\mathbb{G}^m = \{G_1^m, \dots, G_z^m, \dots, G_{\bar{z}}^m, \dots, G_{\bar{Z}}^m\}$  be the set of the groups to which the VNs in coalition  $S_m^*$  belong, where  $G_z^m = G_z \cap S_m^*$  ( $G_z \in \mathbb{G}$ ). Since we record the time of the VN with the longest training time in group  $G_z^m$  as the training time of this group, we thus have  $\mu_i = \mu_{G_z^m}$  and  $e_i = e_{G_z^m}$  for VN  $i$  ( $i \in G_z^m$ ). For coalition  $S_m^*$ , its leader may receive parameters from one group or multiple groups. It will aggregate the parameters of the VNs in this group or the parameters of the VNs in these groups, and then send the aggregated model parameters to these VNs. Specifically, we divide the aggregation methods of each coalition leader into the following three types.

a) *Intra-group aggregation*: For training sequence  $e_{G_z^m}$  ( $e_{G_z^m} \in [1, \mu_{G_z^m}]$ ) of group  $G_z^m$  in coalition  $S_m^*$ , if any training sequence  $e_{G_{z'}^m}$  ( $e_{G_{z'}^m} \in [1, \mu_{G_{z'}^m}]$ ) of another group  $G_{z'}^m$  can not be found in group set  $\mathbb{G}^m$  to meet the condition  $e_{G_z^m} * t_{G_z^m}^j = e_{G_{z'}^m} * t_{G_{z'}^m}^j$ , the coalition leader will only aggregate the parameters of the VNs in group  $G_z^m$ . After aggregating the parameters, the coalition leader sends the parameters to the VNs in group  $G_z^m$  to start the next round of  $E^j$  local iterations. In particular, if the group has only one VN  $i$  and  $e_i < \mu_i$ , the VN does not upload local parameters to the coalition leader, and directly conducts the next round of  $E^j$  local iterations. The intra-group aggregation of the parameters trained by the VNs in group  $G_z^m$  can be expressed as

$$\omega_{G_z^m}^k = \sum_{i \in G_z^m} \frac{|\mathbb{D}_i^j| \omega_{e_i}^k}{|\mathbb{D}_{G_z^m}^j|} = \sum_{i \in G_z^m} \frac{|\mathbb{D}_i^j| \omega_{e_i}^k}{\sum_{i \in G_z^m} |\mathbb{D}_i^j|}. \quad (21)$$

b) *Inter-group aggregation*: For group set  $\bar{\mathbb{G}}^m = \{G_1^m, \dots, G_z^m, \dots, G_{\bar{z}}^m, \dots, G_{\bar{Z}}^m\}$ , which is subset of  $\mathbb{G}^m$ , if each group  $G_{\bar{z}}^m$  in the set can find a training sequence  $e_{G_{\bar{z}}^m}$  that satisfies  $e_{G_1^m} t_{G_1^m}^j = \dots = e_{G_z^m} t_{G_z^m}^j = \dots = e_{G_{\bar{z}}^m} t_{G_{\bar{z}}^m}^j$ , the coalition leader will aggregate the local parameters of the VNs in all the groups in set  $\bar{\mathbb{G}}^m$  at time  $e_{G_1^m} t_{G_1^m}^j$ . After completing the parameter aggregation between these groups, the coalition leader sends the parameters to each VN  $i$  ( $i \in \bar{\mathbb{G}}^m$ ), and VN  $i$  starts the next

---

**Algorithm 2** The Coalition-Based VFL Algorithm

---

```

1: Input:  $\mathbb{G}, \mathbb{S}^*, \omega^0, E^j, \eta^j$ 
2: for  $k : k \in K$  do
3:   for  $i : i \in \mathbb{S}^*$  in parallel do
4:      $e_i = 0$ 
5:      $\omega_{e_i}^k = \omega^{k-1}$ 
6:     for  $e_i : e_i \in [1, \mu_i]$  do
7:       The VN  $i$  computes the local model parameters  $\omega_{e_i}^k$  according to (19)
8:       if  $i = i^*$  and  $e_i = \mu_i$  then
9:         The edge server updates the global parameters  $\omega^k$  according to (24)
10:        break
11:      end if
12:      if  $i = \arg \max_{i \in \mathbb{S}_m^*} t_i^j$  and  $e_i = \mu_i$  then
13:        The leader of coalition  $S_m^*$  updates the coalition parameters  $\omega_{S_m^*}^k$  according to (23)
14:      end if
15:      if  $i \in G_z^m$  and  $i = \arg \max_{i \in G_z^m} t_i^j$  and  $|G_z^m| > 1$  then
16:        The leader of coalition  $S_m^*$  updates the intra-group parameters  $\omega_{G_z^m}^k$  according to (21)
17:        for  $i' : i' \in G_z^m$  do
18:           $\omega_{e_{i'}}^k = \omega_{G_z^m}^k$ 
19:        end for
20:      end if
21:      if  $i \in \bar{\mathbb{G}}^m$  and  $i = \arg \max_{i \in \bar{\mathbb{G}}^m} t_i^j$  then
22:        The leader of coalition  $S_m^*$  updates the inter-group parameters  $\omega_{\bar{\mathbb{G}}^m}^k$  according to (22)
23:        for  $i' : i' \in \bar{\mathbb{G}}^m$  do
24:           $\omega_{e_{i'}}^k = \omega_{\bar{\mathbb{G}}^m}^k$ 
25:        end for
26:      end if
27:    end for
28:  end for
29: end for

```

---

round of  $E^j$  local iterations based on the updated parameters. The inter-group aggregation of the parameters trained by the VNs in different groups can be expressed as

$$\omega_{\bar{\mathbb{G}}^m}^k = \sum_{i \in \bar{\mathbb{G}}^m} \frac{|\mathbb{D}_i^j| \omega_{e_i}^k}{|\mathbb{D}_{\bar{\mathbb{G}}^m}^j|} = \sum_{i \in \bar{\mathbb{G}}^m} \frac{|\mathbb{D}_i^j| \omega_{e_i}^k}{\sum_{i \in \bar{\mathbb{G}}^m} |\mathbb{D}_i^j|}. \quad (22)$$

*Proposition 2*: If the multiple of the training sequence  $e_{G_z^m}$  and the training time  $t_{G_z^m}^j$  of group  $G_z^m$  ( $G_z^m \in \mathbb{G}^m$ ) is equal to that of group  $G_{z'}^m$  ( $G_{z'}^m \in \mathbb{G}^m$ ), the VNs in this coalition need to perform the inter-group aggregation.

The proof of this proposition is given in Appendix B.

*Proposition 3*: If the least common multiple of the training time  $t_{G_z^m}^j$  of group  $G_z^m$  ( $G_z^m \in \mathbb{G}^m$ ) and the training time  $t_{G_{z'}^m}^j$  of group  $G_{z'}^m$  ( $G_{z'}^m \in \mathbb{G}^m$ ) is larger than the training time of the slowest VN in this coalition (i.e.,  $\max t_i^j$  ( $\forall i \in S_m^*$ )), then the

VNs in group  $G_z^m$  and group  $G_{z'}^m$  will not perform inter-group aggregation in each round of the global training.

The proof of this proposition is given in Appendix C.

*Proposition 4:* If the training time  $t_{G_z^m}^j$  of group  $G_z^m$  ( $G_z^m \in \mathbb{G}^m$ ) is a factor of the training time  $t_{G_{z'}^m}^j$  of group  $G_{z'}^m$  ( $G_{z'}^m \in \mathbb{G}^m$ ), and the VN  $i^*$  with the longest training time in coalition  $S_m^*$  is not in these two groups, then group  $G_z^m$  ( $G_z^m \in \mathbb{G}^m$ ) and group  $G_{z'}^m$  ( $G_{z'}^m \in \mathbb{G}^m$ ) will perform inter-group aggregation at least once.

The proof of this proposition is given in Appendix D.

c) *Coalition aggregation:* For coalition  $S_m^*$ , if each VN  $i$  ( $i \in S_m^*$ ) in the coalition has completed  $\mu_i$  rounds of  $E^j$  local iterations, the coalition leader will aggregate the local parameters of all the VNs in the coalition to form the coalition parameters. After this, the coalition leader sends the coalition parameters to the edge server to start the global parameters aggregation. The coalition aggregation of the parameters trained by the VNs in the coalition can be expressed as

$$\omega_{S_m^*}^k = \sum_{i \in S_m^*} \frac{|\mathbb{D}_i^j| \omega_{\mu_i}^k}{|\mathbb{D}_{S_m^*}^j|} = \sum_{i \in S_m^*} \frac{|\mathbb{D}_i^j| \omega_{\mu_i}^k}{\sum_{i \in S_m^*} |\mathbb{D}_i^j|}. \quad (23)$$

3) *Global Aggregation:* When each VN  $i$  in set  $\mathbb{S}^*$  completes  $\mu_i$  rounds of  $E^j$  local iterations in the  $k$ -th round of global training, each coalition leader sends the coalition model parameters to the edge server after completing the coalition aggregation. Then, the edge server aggregates the coalition model parameters from all the coalition leaders and generates the global model, ending the  $k$ -th round of the global training. The global model parameters of the  $k$ -th round can be updated by

$$\omega^k = \sum_{S_m^* \in \mathbb{S}^*} \frac{|\mathbb{D}_{S_m^*}^j| \omega_{S_m^*}^k}{|\mathbb{D}_{\mathbb{S}^*}^j|} = \sum_{S_m^* \in \mathbb{S}^*} \frac{|\mathbb{D}_{S_m^*}^j| \omega_{S_m^*}^k}{\sum_{S_m^* \in \mathbb{S}^*} |\mathbb{D}_{S_m^*}^j|}. \quad (24)$$

Based on the above steps, we then design a coalition-based VFL algorithm to describe the model training process. The details of the coalition-based VFL algorithm are shown in Algorithm 2.

#### D. System Cost Analysis

According to [44], the system overhead includes communication overhead for transmitting model parameters and computational overhead for training the model. We use the FedAvg algorithm as a benchmark and measure the total system overhead using communication time and computation time [45]. In the  $k$ -th round of global training of the FedAvg algorithm, the edge server sends global model parameters to all VNs and waits for all VNs to upload their local model parameters before performing the model aggregation operation [46]. Therefore, the transmission time of FedAvg depends on the slowest VN. In addition, since each VN independently completes local training, the calculation time of FedAvg also depends on the slowest VN [47]. Thus, the system overhead for each global round of FedAvg can be represented as

$$\mathcal{T}_{\text{FedAvg}} = \max_{i \in \mathbb{I}} (t_i^{\text{cmp}} + \overline{t_i^{\text{down}}} + \overline{t_i^{\text{up}}})$$

$$\begin{aligned} &\leq \max_{i \in \mathbb{I}} (t_i^{\text{cmp}}) + \max_{i \in \mathbb{I}} (\overline{t_i^{\text{down}}} + \overline{t_i^{\text{up}}}) \\ &= t_{\text{FedAvg}}^{\text{cmp}} + t_{\text{FedAvg}}^{\text{com}}, \end{aligned} \quad (25)$$

where  $t_i^{\text{cmp}}$  is the calculation time spent on training the local dataset  $E^j$  times,  $\overline{t_i^{\text{down}}}$  is the communication time of the edge server for sending global model parameters to VN  $i$ ,  $\overline{t_i^{\text{up}}}$  is the communication time of VN  $i$  for uploading local model parameters to the edge server,  $t_{\text{FedAvg}}^{\text{cmp}}$  is the calculation time of FedAvg, and  $t_{\text{FedAvg}}^{\text{com}}$  is the communication time of FedAvg.

In contrast, the system overhead of the RCFL needs to include the time for the edge server to distribute global model parameters to coalition leader  $i^{S_m^*}$  (i.e.,  $t_{i^{S_m^*}}^{\text{down}}$ ), the training time of each VN  $i$  for completing the  $E^j$  local iterations  $\mu_i$  rounds, and the time for coalition leader  $i^{S_m^*}$  to upload coalition parameters to the edge server (i.e.,  $t_{i^{S_m^*}}^{\text{up}}$ ). Therefore, in the  $k$ -th round of global training, the system overhead of the RCFL can be expressed as

$$\begin{aligned} \mathcal{T}_{\text{RCFL}} &= \max_{i \in \mathbb{I}, i^{S_m^*} \in \mathbb{I}} (t_{i^{S_m^*}}^{\text{down}} + \mu_i * t_i^j + t_{i^{S_m^*}}^{\text{up}}) \\ &\leq \max_{i \in \mathbb{I}} (\mu_i * t_i^j) + \max_{i^{S_m^*} \in \mathbb{I}} (t_{i^{S_m^*}}^{\text{down}} + t_{i^{S_m^*}}^{\text{up}}) \\ &= t_{\text{FedAvg}}^{\text{cmp}} + \max_{i^{S_m^*} \in \mathbb{I}} (t_{i^{S_m^*}}^{\text{down}} + t_{i^{S_m^*}}^{\text{up}}). \end{aligned} \quad (26)$$

By comparing the system overhead of FedAvg in (25) and the system overhead of RCFL in (26), we have

$$\begin{aligned} \mathcal{T}_{\text{RCFL}} - \mathcal{T}_{\text{FedAvg}} &= \max_{i^{S_m^*} \in \mathbb{I}} (t_{i^{S_m^*}}^{\text{down}} + t_{i^{S_m^*}}^{\text{up}}) \\ &\quad - \max_{i \in \mathbb{I}} (\overline{t_i^{\text{down}}} + \overline{t_i^{\text{up}}}). \end{aligned} \quad (27)$$

From (27), it can be seen that when the coalition leader with the longest communication delay with the edge server in the RCFL and the vehicle with the longest communication delay with the edge server in the FedAvg are the same vehicle (i.e.,  $\arg \max_{i^{S_m^*} \in \mathbb{I}} (t_{i^{S_m^*}}^{\text{down}} + t_{i^{S_m^*}}^{\text{up}}) = \arg \max_{i \in \mathbb{I}} (\overline{t_i^{\text{down}}} + \overline{t_i^{\text{up}}})$ ), we have  $\max_{i^{S_m^*} \in \mathbb{I}} (t_{i^{S_m^*}}^{\text{down}} + t_{i^{S_m^*}}^{\text{up}}) = \max_{i \in \mathbb{I}} (\overline{t_i^{\text{down}}} + \overline{t_i^{\text{up}}})$ . Accordingly, the system cost of RCFL is the same as FedAvg (i.e.,  $\mathcal{T}_{\text{RCFL}} = \mathcal{T}_{\text{FedAvg}}$ ). In comparison, when the coalition leader with the longest communication delay with the edge server in the RCFL and the vehicle with the longest communication delay with the edge server in the FedAvg are not the same vehicle (i.e.,  $\arg \max_{i^{S_m^*} \in \mathbb{I}} (t_{i^{S_m^*}}^{\text{down}} + t_{i^{S_m^*}}^{\text{up}}) \neq \arg \max_{i \in \mathbb{I}} (\overline{t_i^{\text{down}}} + \overline{t_i^{\text{up}}})$ ), we have  $\max_{i^{S_m^*} \in \mathbb{I}} (t_{i^{S_m^*}}^{\text{down}} + t_{i^{S_m^*}}^{\text{up}}) < \max_{i \in \mathbb{I}} (\overline{t_i^{\text{down}}} + \overline{t_i^{\text{up}}})$ . As a consequence, the system cost of RCFL is smaller than FedAvg (i.e.,  $\mathcal{T}_{\text{RCFL}} < \mathcal{T}_{\text{FedAvg}}$ ).

#### E. Convergence Analysis

In the neural network, the loss function is non-convex, so we need to prove that the first derivative of the loss function is approximately 0. We use the same method as [48] to analyze the convergence of the RCFL.

*Proposition 5:* Assuming  $f_i(\omega^k)$  is  $\beta$ -smooth, then if  $K \rightarrow \infty$ , we have  $\|\nabla f_i(\omega^k)\| \rightarrow 0$ .

TABLE I  
SIMULATION PARAMETERS

Parameters	MNIST	CIFAR10
The number of iterations: $E^j$	10	3
The learning rate: $\eta^j$	0.01	0.001
The number of VNs: $I$	30	10
The number of data of label $l_i^j$ : $n_{l_i^j}$	25	50
The weight of EMD: $\alpha^j$	0.5	0.5
The weight of RoD: $\vartheta^j$	0.5	0.5
The factor used to balance the influence of EMD on the data importance: $\rho_B^j$	1	1
The factor used to balance the influence of RoD on the data importance: $\rho_R^j$	0.125	0.035
The small positive number: $\varpi_B^j$	0.05	0.05
The small positive number: $\varpi_R^j$	0.05	0.05
The minimum speed of the VNs: $v_{\min}$	10km/h	10km/h
The maximum speed of the VNs: $v_{\max}$	60km/h	60km/h

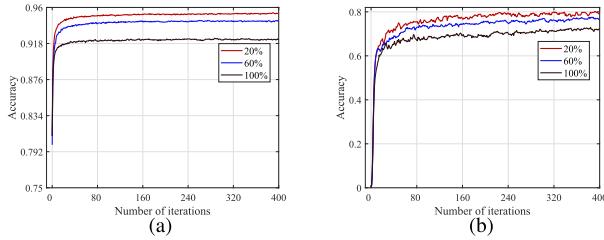


Fig. 5. The model accuracy by changing the RoD of each label. (a) MNIST; (b) CIFAR10.

The proof of this proposition is given in Appendix E.

After proving the convergence, we then discuss the convergence rate. According to [44], in the training process of FedAvg method, the convergence rate of each VN is  $O(\frac{E^j}{K})$ . In contrast, in the RCFL scheme, the convergence rate of each VN  $i$  participating in the training process to solve optimization problem (20) is  $O(\frac{(\mu_i E^j)^2}{K})$ , where  $\mu_i \geq 1$ . Therefore, the convergence rate of the RCFL is faster than the convergence rate of FedAvg.

## V. SIMULATION RESULTS

In this section, the performance of the proposed scheme is evaluated. In the simulation, we consider a scenario with a CBS that is connected to an edge server. The diameter distance of the communication coverage of the CBS is 5km. The minimum speed of the VNs is 10km/h and the maximum speed of the VNs is 60km/h [49]. There are a task requester and a number of VNs in the coverage of the CBS, where the task requester intends to train an image classification task  $j$ . For the datasets, similar to the existing references [8], [19], and [50], our paper uses MNIST and CIFAR10 with known labels for simulation. The MNIST is a 10 class grayscale image dataset with  $28 \times 28$  pixels, and CIFAR10 is a 10 class color image dataset with  $32 \times 32$  pixels. The parameters used in our paper are summarized in Table I.

With the above scenario and datasets, we first verify the impact of RoD on the model training accuracy. Specifically, we consider 3 VNs participating in the image classification task, where each VN has 10 labels in Fig. 5(a), and each VN has 5 labels in Fig. 5(b). Fig. 5 illustrates the model accuracy

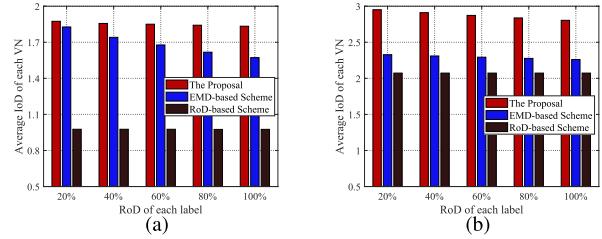


Fig. 6. The average data importance of each VN by changing the RoD of each label. (a) MNIST; (b) CIFAR10.

by changing the RoD of each label using different datasets. From both Fig. 5(a) and Fig. 5(b), it can be seen that the model accuracy decreases with the increase of the RoD of each label. This is because an increase in the RoD of each label will decrease the quality of data owned by the VNs. Consequently, the model accuracy will be reduced.

Then, we evaluate the average IoD of each VN and the accuracy by changing the RoD of each label and the maximum number of labels owned by each VN. The schemes used in the simulations are detailed as follows.

- The proposal: In our proposal, the importance-aware coalition formation algorithm is used by the VNs to form the stable partition. In addition, the coalition-based VFL algorithm is adopted to train the learning model.
- EMD-based Scheme: In this scheme, the VNs form the stable partition by considering the non-IID degree of data (i.e., EMD). Based on the coalition partition, the edge server uses the coalition-based VFL algorithm to train the learning model.
- RoD-based Scheme: In this scheme, the VNs form the stable partition by considering the redundancy of data (i.e., RoD). Specifically, in the RoD-based scheme, the redundancy of the local dataset owned by the VNs is 0. Due to the RoD-based scheme only considers the degree of data redundancy during the coalition process, each VN will not form a coalition with other VNs. In other words, similar to the FedAvg algorithm, in the RoD-based scheme, each VN completes the model training task independently. The difference is that the FedAvg algorithm randomly selects training VNs, while the RoD-based scheme selects training VNs based on the IoD.

Fig. 6 illustrates the average IoD of each VN by changing the RoD of each label, where the dataset of Fig. 6(a) is MNIST, and the dataset of Fig. 6(b) is CIFAR10. From Fig. 6, we can see that the average IoD of each VN decreases with the increase of the RoD of each label in the proposal and the EMD-based scheme. This is because the increase of the RoD of each label increases the RoD of each coalition. In addition, it can be seen that the increase in the RoD of each label hardly affects the RoD-based scheme. This is because each VN works independently and does not form a coalition with other VNs in this scheme. Namely, the RoD-based scheme only considers the RoD when playing the coalition game. As a result, the coalition partition where each VN forms a coalition individually is the stable partition. Therefore, the average IoD

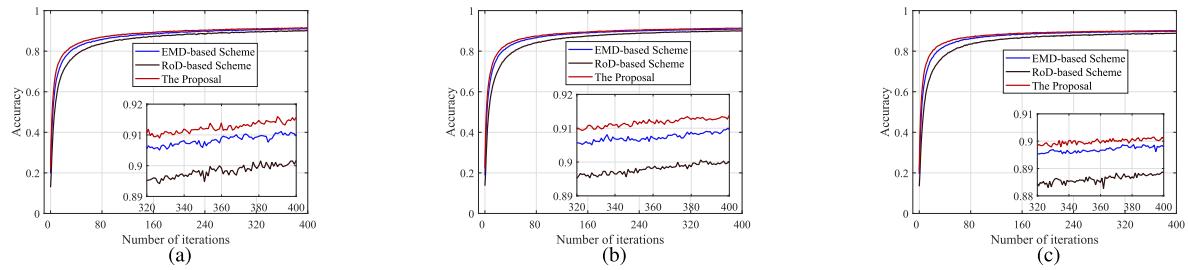


Fig. 7. The model accuracy of MNIST by changing the RoD of each label, where the number of labels owned by each VN is set to 2. (a) The RoD of each label is set to 20%; (b) The RoD of each label is set to 60%; (c) The RoD of each label is set to 100%.

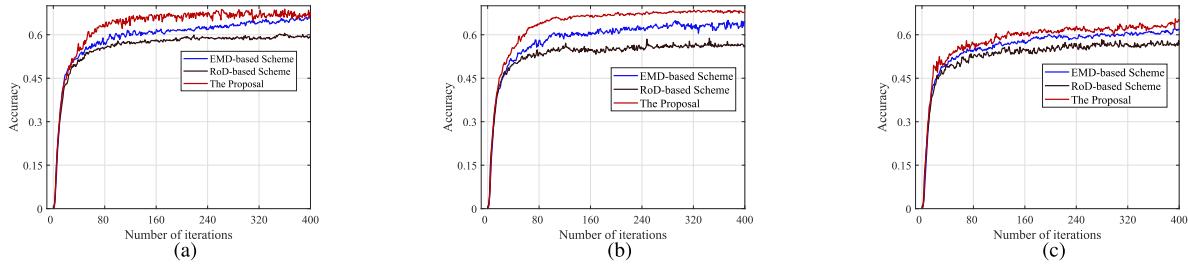


Fig. 8. The model accuracy of CIFAR10 by changing the RoD of each label, where the number of labels owned by each VN is set to 3. (a) The RoD of each label is set to 20%; (b) The RoD of each label is set to 60%; (c) The RoD of each label is set to 100%.

of each VN in the RoD-based scheme does not change with the change of the RoD of each label.

Fig. 7 shows the model accuracy of MNIST by changing the RoD of each label, where the number of labels owned by each VN is set to 2. Specifically, the RoD of each label in Fig. 7(a), Fig. 7(b), and Fig. 7(c) is set to 20%, 60%, and 100%, respectively. From Fig. 7, we can see that our scheme can obtain the highest model training accuracy. This is because in the proposal, the RoD and the EMD are jointly considered to determine the IoD and the coalitions with high data quality will be selected to complete the model training task. In addition, it can be seen that the change trend of the curves in Fig. 7(b) and Fig. 7(c) is basically the same as that in Fig. 7(a). However, the convergence accuracy of each scheme in Fig. 7(a) is higher than that in Fig. 7(b) and Fig. 7(c). This is because the increase of the RoD will decrease the data quality so that the model accuracy will be decreased.

Fig. 8 shows the model accuracy of CIFAR10 by changing the RoD of each label, where the number of labels owned by each VN is set to 3. The RoD of each label in Fig. 8(a), Fig. 8(b), and Fig. 8(c) is 20%, 60%, and 100%, respectively. From these figures, it can be seen that our scheme can achieve the highest accuracy. In addition, by comparing Fig. 8(a), Fig. 8(b), and Fig. 8(c), we can see that the convergence accuracy of all three schemes decreases with the increase of the RoD. Besides, by comparing Fig. 7 and Fig. 8, it can be seen that our scheme has a greater advantage in the CIFAR10 dataset than in the MNIST dataset. This is because compared with the grayscale images with  $28 \times 28$  pixels in the MNIST dataset, the color images with  $32 \times 32$  pixels in CIFAR10 have higher requirements for data quality. Therefore, the proposal can significantly improve the accuracy of model training by selecting high-quality data.

Fig. 9 illustrates the average IoD of each VN by changing the number of labels owned by each VN, where the dataset of

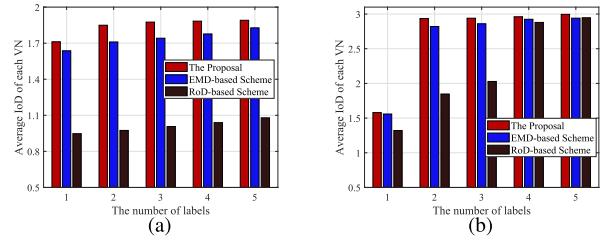


Fig. 9. The average data importance of each VN by changing the number of labels owned by each VN. (a) MNIST; (b) CIFAR10.

Fig. 9(a) is MNIST, and the dataset of Fig. 9(b) is CIFAR10. From Fig. 9, we can see that the average IoD of each VN increases with the increase of the number of labels in all the schemes. This is because the increase in the number of labels increases the local data diversity of each VN, thereby reducing the degree of EMD of data in the coalitions. Furthermore, it can be seen from this figure that the proposal can lead to the highest IoD for each VN. The reason for this is that the proposal jointly considers the EMD and the RoD to evaluate the IoD and form the stable coalition partition. In comparison, the EMD-based scheme and the RoD-based scheme only consider one factor to form the coalition partition. It is worth noting that as the number of labels increases, the gap between the traditional schemes and the proposal in Fig. 9(b) becomes small. The reason for this is that the CIFAR10 dataset is more sensitive to EMD than the MNIST dataset. Therefore, when the number of labels of each VN is sufficient, the EMD of data owned by each VN is relatively low. As a result, the average IoD of each VN is improved.

Fig. 10 shows the model accuracy of MNIST by changing the number of labels owned by each VN, where the RoD of each label is set to 50%. Specifically, the number of labels owned by each VN in Fig. 10(a), Fig. 10(b), and Fig. 10(c) is set to 1, 3, and 5, respectively. As can be seen

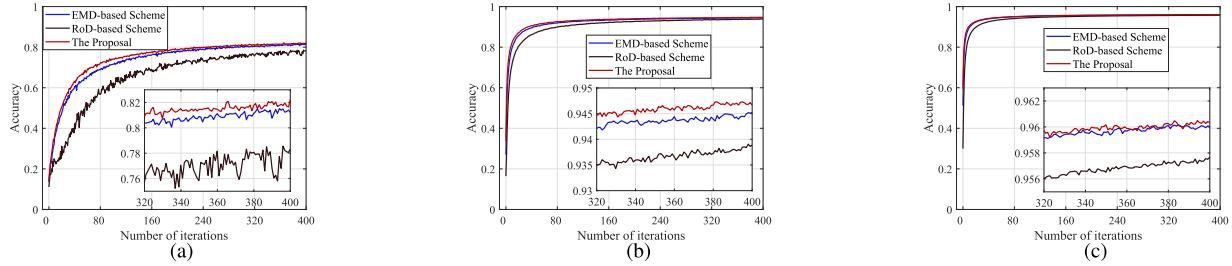


Fig. 10. The model accuracy of MNIST by changing the number of labels owned by each VN, where the RoD of each label is set to 50%. (a) The number of labels of each VN is set to 1; (b) The number of labels of each VN is set to 3; (c) The number of labels of each VN is set to 5.

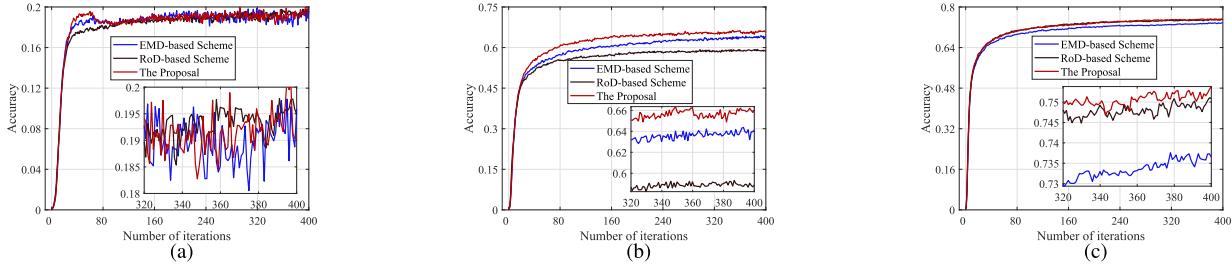


Fig. 11. The model accuracy of CIFAR10 by changing the number of labels owned by each VN, where the RoD of each label is set to 50%. (a) The number of labels of each VN is set to 1; (b) The number of labels of each VN is set to 3; (c) The number of labels of each VN is set to 5.

from the figure, our scheme can obtain the highest model training accuracy. The reason for this is that the RoD and the EMD are simultaneously considered in the proposed scheme. Consequently, the IoD of each coalition in our scheme is the highest compared to the conventional schemes. Furthermore, we can see that the change trend of the curves in Fig. 10(b) and Fig. 10(c) is basically the same as that in Fig. 10(a), where the convergence accuracy of each scheme in Fig. 10(a) is lower than that in Fig. 10(b) and Fig. 10(c). The reason for this is that the diversity of data and the EMD of the data owned by the VNs increase with the increase of the maximum number of labels. Thus, the model accuracy increases with the increase of the maximum number of labels.

Fig. 11 shows the model accuracy of CIFAR10 by changing the number of labels owned by each VN, where the RoD of each label is set to 50%. The number of labels of each VN in Fig. 11(a), Fig. 11(b), and Fig. 11(c) is 1, 3, and 5, respectively. From these figures, it can be seen that when the number of labels owned by each VN is 1, the convergence accuracy of the three schemes is almost the same. This is because the quality of data owned by each VN is extremely poor for training CIFAR10 tasks, so none of the three schemes can effectively improve data quality. In comparison, we can see from Fig. 11(b) that the accuracy of the proposal is higher than that of EMD-based scheme and RoD-based scheme. For Fig. 11(c), we can see that the proposal and the RoD-based scheme can obtain similar accuracy. This is because each VN in Fig. 11(c) has 5 labels, and the number of data of each label is 50. Thus, the non-IID degree is 0. This means that the EMD of the data owned by each VN is already optimal. Therefore, the main difference between the three schemes is the RoD of the data. In the proposal and RoD-based scheme, the data redundancy among VNs can be effectively reduced. Therefore, these two schemes exhibit similar accuracy and are both higher than the EMD-based scheme.

## VI. CONCLUSION

In this paper, we have proposed a RCFL scheme to facilitate the learning services in VNets. Specifically, we have jointly considered the IoD and the cooperation among VNs to design a redundancy-aware federated learning architecture. Then, based on the EMD and the RoD, we have developed the data importance model to evaluate the quality of data and formulated the cooperation of the VNs as a coalition game to improve their IoD. In addition, we have designed a coalition formation algorithm to obtain the stable coalition partition. After that, by considering the diversified characteristics of data and the available resources of different VNs in each coalition, we have designed a coalition-based federated learning algorithm to enable the distributed coalitions to complete the learning task cooperatively with the target of improving the learning accuracy. The simulation results have shown that the proposed scheme outperforms the benchmark schemes in terms of the IoD obtained by the VNs and the model training accuracy.

For future work, the communication environment of each VN will be considered to allocate the bandwidth resources of the CBS with the target of improving the communication efficiency of our proposed scheme.

## APPENDIX A PROOF OF PROPOSITION 1

Given any coalition partition  $\mathbb{S}' = \{S_1, \dots, S_{m'}, \dots, S_{M'}\}$ , if the IoD of a coalition partition is higher than the IoD of the current coalition partition, each VN will search for its optimal coalition and update the current coalition partition through Algorithm 1. Therefore, Algorithm 1 will generate a series of coalition partitions, in which the IoD of each coalition partition is no smaller than that of the last iteration. In addition, the total number of possible coalition partitions in Algorithm 1 is limited. Therefore, after a limited number of iterations,

the algorithm will eventually terminate in coalition partition  $\mathbb{S} = \{S_1, \dots, S_m, \dots, S_M\}$ , where the IoD of this coalition partition cannot be further increased by using Algorithm 1. At this time, no VNs can increase the IoD by changing the coalition partition which means that the partition is stable. Namely, the IoD of coalition partition  $\mathbb{S}$  is the highest. The proposition is proved.

#### APPENDIX B PROOF OF PROPOSITION 2

Based on Proposition 2, if

$$e_{G_z^m} t_{G_z^m}^j = e_{G_{z'}^m} t_{G_{z'}^m}^j, \quad (28)$$

the coalition leader will receive the  $e_{G_z^m}$ -th round of  $E^j$  iterations of  $G_z^m$  and the  $e_{G_{z'}^m}$ -th round of  $E^{j'}$  iterations of  $G_{z'}^m$ . Therefore, the coalition leader will aggregate the local training parameters of the two groups at the same time. Namely, the coalition leader will perform the inter-group aggregation.

#### APPENDIX C PROOF OF PROPOSITION 3

Let  $a$  and  $b$  be positive integers that are not zero. If

$$a \cdot t_{G_z^m}^j = b \cdot t_{G_{z'}^m}^j, \quad (29)$$

$a \cdot t_{G_z^m}^j$  is the common multiple of  $t_{G_z^m}^j$  and  $t_{G_{z'}^m}^j$ . Let  $a^*$  denote the smallest positive integer that satisfies (29). Then,  $a^* \cdot t_{G_z^m}^j$  is the least common multiple of  $t_{G_z^m}^j$  and  $t_{G_{z'}^m}^j$ . If

$$a^* \cdot t_{G_z^m}^j > \max_{i \in S_m^*} t_i^j, \quad (30)$$

then no  $a^*$  in the range of  $[1, \mu_{G_z^m})$  can satisfy (29). Namely,

$$\nexists e_{G_z^m} t_{G_z^m}^j = e_{G_{z'}^m} t_{G_{z'}^m}^j. \quad (31)$$

(31) indicates that the coalition leader cannot receive the local parameters of any round of  $E^j$  local iterations of the two groups  $G_z^m$  and  $G_{z'}^m$  at the same time before performing the coalition aggregation. Therefore, group  $G_z^m$  and group  $G_{z'}^m$  do not perform inter-group aggregation.

#### APPENDIX D PROOF OF PROPOSITION 4

If  $t_{G_z^m}^j$  is a factor of  $t_{G_{z'}^m}^j$ , then there must be an integer  $a$  ( $a > 1$ ) that satisfies

$$\frac{t_{G_z^m}^j}{t_{G_{z'}^m}^j} = a. \quad (32)$$

Based on the condition of the inter-group aggregation, if group  $G_z^m$  and group  $G_{z'}^m$  can perform the inter-group aggregation, the following condition should be satisfied. We have

$$e_{G_z^m} t_{G_z^m}^j = e_{G_{z'}^m} t_{G_{z'}^m}^j. \quad (33)$$

Namely, we have

$$\frac{t_{G_z^m}^j}{t_{G_{z'}^m}^j} = \frac{e_{G_z^m}}{e_{G_{z'}^m}}. \quad (34)$$

Based on (32) and (34), the condition for proving proposition 4 can be rewritten as

$$\frac{e_{G_z^m}}{e_{G_{z'}^m}} = a. \quad (35)$$

Because the training sequence of different groups is different. For different groups  $G_z^m$  and  $G_{z'}^m$ , we have  $e_{G_z^m} \neq e_{G_{z'}^m}$ . To prove (35), we set  $e_{G_{z'}^m} = 1$ . Since  $e_{G_z^m} \in [1, \mu_{G_z^m})$  and  $e_{G_{z'}^m} \in [1, \mu_{G_{z'}^m})$ , there must be a  $e_{G_z^m}$  that satisfies (35) if  $e_{G_z^m} > 1$ . Therefore, if  $t_{G_z^m}^j$  is the factor of  $t_{G_{z'}^m}^j$ , there must be  $e_{G_z^m} \in [1, \mu_{G_z^m})$  and  $e_{G_{z'}^m} \in [1, \mu_{G_{z'}^m})$  that satisfy the condition of the inter-group aggregation. Then the parameters of the two groups  $G_z^m$  and  $G_{z'}^m$  should be aggregated at least once. The proposition is proved.

#### APPENDIX E PROOF OF PROPOSITION 5

Because  $f_i(\omega^k)$  is  $\beta$ -smooth, we have

$$f_i(\omega^{k+1}) \leq f_i(\omega^k) - \frac{1}{2\beta} \|\nabla f_i(\omega^k)\|^2. \quad (36)$$

After  $K$  rounds of global training, the following condition is also satisfied.

$$f_i(\omega^K) \leq f_i(\omega^0) - \frac{1}{2\beta} \sum_{k=1}^K \|\nabla f_i(\omega^k)\|^2. \quad (37)$$

Namely, we have

$$\frac{1}{2\beta} \sum_{k=1}^K \|\nabla f_i(\omega^k)\|^2 \leq f_i(\omega^0) - f_i(\omega^K). \quad (38)$$

In addition, because  $\omega^*$  is the optimal value of the model parameters, we have

$$f_i(\omega^0) - f_i(\omega^*) \geq f_i(\omega^0) - f_i(\omega^K). \quad (39)$$

According to (38) and (39), (38) can be rewritten as

$$\sum_{k=1}^K \|\nabla f_i(\omega^k)\|^2 \leq 2\beta(f_i(\omega^0) - f_i(\omega^*)). \quad (40)$$

Obviously, we have

$$\sum_{k=1}^K \|\nabla f_i(\omega^k)\|^2 \geq K \left( \min_{k=1,\dots,K} \|f_i(\omega^k)\|^2 \right). \quad (41)$$

By comparing (40) and (41), (41) can be rewritten as

$$K \left( \min_{k=1,\dots,K} \|\nabla f_i(\omega^k)\|^2 \right) \leq 2\beta(f_i(\omega^0) - f_i(\omega^*)). \quad (42)$$

Namely, we have

$$\min_{k=1,\dots,K} \|\nabla f_i(\omega^k)\|^2 \leq \frac{2\beta(f_i(\omega^0) - f_i(\omega^*))}{K}. \quad (43)$$

From (43), we can know that if  $K \rightarrow \infty$ , we have  $\|\nabla f_i(\omega^k)\| \rightarrow 0$ . The proposition is proved.

## REFERENCES

- [1] Y. Hui et al., "Secure and personalized edge computing services in 6G heterogeneous vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5920–5931, Apr. 2022.
- [2] W. Liu, Y. Watanabe, and Y. Shoji, "Vehicle-assisted data delivery in smart city: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13849–13860, Nov. 2020.
- [3] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7751–7763, Aug. 2020.
- [4] X. Kong, H. Gao, G. Shen, G. Duan, and S. K. Das, "FedVCP: A federated-learning-based cooperative positioning scheme for social Internet of Vehicles," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 197–206, Feb. 2022.
- [5] A. A. Abdellatif, C. F. Chiasseroni, F. Malandrino, A. Mohamed, and A. Erbad, "Active learning with noisy labelers for improving classification accuracy of connected vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3059–3070, Apr. 2021.
- [6] X. Huang, P. Li, R. Yu, Y. Wu, K. Xie, and S. Xie, "FedParking: A federated learning based parking space estimation with parked vehicle assisted edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9355–9368, Sep. 2021.
- [7] Y. Hui et al., "Digital twins enabled on-demand matching for multi-task federated learning in HetVNets," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2352–2364, Feb. 2023.
- [8] B. Li, Y. Jiang, Q. Pei, T. Li, L. Liu, and R. Lu, "FEEL: Federated end-to-end learning with non-IID data for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16728–16740, Sep. 2022.
- [9] X. Zhou, W. Liang, J. She, Z. Yan, and K. I. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6G supported Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5308–5317, Jun. 2021.
- [10] Q. Zhang, H. Wen, Y. Liu, S. Chang, and Z. Han, "Federated-reinforcement-learning-enabled joint communication, sensing, and computing resources allocation in connected automated vehicles networks," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 23224–23240, Nov. 2022.
- [11] N. Ding, Z. Fang, and J. Huang, "Optimal contract design for efficient federated learning with multi-dimensional private information," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 186–200, Jan. 2021.
- [12] Z. Yan, M. Wang, Y. Li, and A. V. Vasilakos, "Encrypted data management with deduplication in cloud computing," *IEEE Cloud Comput.*, vol. 3, no. 2, pp. 28–35, Mar. 2016.
- [13] W. Wang and M. Zhang, "Tensor deep learning model for heterogeneous data fusion in Internet of Things," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 1, pp. 32–41, Feb. 2020.
- [14] S. Li, J. Huang, J. Hu, and B. Cheng, "QoE-DEER: A QoE-aware decentralized resource allocation scheme for edge computing," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 1059–1073, Jun. 2022.
- [15] J. Mills, J. Hu, and G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 630–641, Mar. 2022.
- [16] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu, "EdgeFed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209191–209198, 2020.
- [17] S. Wang, Y. Hong, R. Wang, Q. Hao, Y.-C. Wu, and D. W. K. Ng, "Edge federated learning via unit-modulus over-the-air computation," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3141–3156, May 2022.
- [18] Y. Deng et al., "Improving federated learning with quality-aware user incentive and auto-weighted model aggregation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4515–4529, Dec. 2022.
- [19] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.
- [20] Y. Hui, J. Hu, X. Xiao, N. Cheng, and T. H. Luan, "Value-aware collaborative data pricing for federated learning in vehicular networks," in *Proc. TridentCom*, Melbourne, VIC, Australia, 2022, pp. 289–300.
- [21] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23920–23935, 2020.
- [22] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, "Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3144–3159, Oct. 2021.
- [23] T. Zeng, O. Semiari, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road autonomous controller design for connected and autonomous vehicles," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10407–10423, Dec. 2022.
- [24] F. Ayaz, Z. Sheng, D. Tian, and Y. L. Guan, "A blockchain based federated learning for message dissemination in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1927–1940, Feb. 2022.
- [25] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.
- [26] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11073–11087, Aug. 2022.
- [27] A. M. Elbir, B. Soner, S. Çöleri, D. Gündüz, and M. Bennis, "Federated learning in vehicular networks," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*, Athens, Greece, Sep. 2022, pp. 72–77.
- [28] M. Xu, Y. Chen, and W. Wang, "A two-stage game framework to secure transmission in two-tier UAV networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13728–13740, Nov. 2020.
- [29] M. Hu, D. Wu, Y. Zhou, X. Chen, and M. Chen, "Incentive-aware autonomous client participation in federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 10, pp. 2612–2627, Oct. 2022.
- [30] Y. Hui et al., "Collaboration as a service: Digital-twin-enabled collaborative and distributed autonomous driving," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18607–18619, Oct. 2022.
- [31] R. Wu, G. Tang, T. Chen, D. Guo, L. Luo, and W. Kang, "A profit-aware coalition game for cooperative content caching at the network edge," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1361–1373, Jan. 2022.
- [32] Z. Su, M. Dai, Q. Xu, R. Li, and H. Zhang, "UAV enabled content distribution for Internet of Connected Vehicles in 5G heterogeneous networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5091–5102, Aug. 2021.
- [33] N. Qi, Z. Huang, F. Zhou, Q. Shi, Q. Wu, and M. Xiao, "A task-driven sequential overlapping coalition formation game for resource allocation in heterogeneous UAV networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4439–4455, Apr. 2023.
- [34] L. Ruan et al., "Cooperative relative localization for UAV swarm in GNSS-denied environment: A coalition formation game approach," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11560–11577, Jul. 2022.
- [35] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless federated learning (WFL) for 6G networks—Part II: The compute-then-transmit NOMA paradigm," *IEEE Commun. Lett.*, vol. 26, no. 1, pp. 8–12, Jan. 2022.
- [36] Q. Wu, Z. Wan, Q. Fan, P. Fan, and J. Wang, "Velocity-adaptive access scheme for MEC-assisted platooning networks: Access fairness via data freshness," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4229–4244, Mar. 2022.
- [37] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5341–5351, Aug. 2021.
- [38] Y. Hui, Z. Su, T. H. Luan, and J. Cai, "Content in motion: An edge computing based relay scheme for content dissemination in urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3115–3128, Aug. 2019.
- [39] R. Younis and M. Fisichella, "FLY-SMOTE: Re-balancing the non-IID IoT edge devices data in federated learning system," *IEEE Access*, vol. 10, pp. 65092–65102, 2022.
- [40] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021.
- [41] J. Wang, G. Xu, W. Lei, L. Gong, X. Zheng, and S. Liu, "CPFL: An effective secure cognitive personalized federated learning mechanism for industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7186–7195, Oct. 2022.
- [42] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.
- [43] H. Fatemidokht, M. K. Rafsanjani, B. B. Gupta, and C.-H. Hsu, "Efficient and secure routing protocol based on artificial intelligence algorithms with UAV-assisted for vehicular ad hoc networks in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4757–4769, Jul. 2021.

- [44] A. M. Elbir, S. Coleri, A. K. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "A hybrid architecture for federated and centralized learning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 3, pp. 1529–1542, Sep. 2022.
- [45] A. S. Berahas, R. Bollapragada, N. S. Keskar, and E. Wei, "Balancing communication and computation in distributed optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3141–3155, Aug. 2019.
- [46] S. S. Shinde, A. Bozorgchenani, D. Tarchi, and Q. Ni, "On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2041–2057, Feb. 2022.
- [47] Z. Ji, L. Chen, N. Zhao, Y. Chen, G. Wei, and F. R. Yu, "Computation offloading for edge-assisted federated learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9330–9344, Sep. 2021.
- [48] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6.
- [49] Q. Wu, Y. Zhao, Q. Fan, P. Fan, J. Wang, and C. Zhang, "Mobility-aware cooperative caching in vehicular edge computing based on asynchronous federated and deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 66–81, Jan. 2023.
- [50] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.



**Yilong Hui** (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Shanghai University, Shanghai, China, in 2018. He is currently an Associate Professor with the State Key Laboratory of ISN, Xidian University, Xi'an, China. He is also a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. His research interests include mobile edge computing, digital twins, space-air-ground integrated networks, and vehicular networks. He was a recipient of the Best Paper Award from the International Conference WiCon 2016 and the IEEE Cyber-SciTech 2017.



**Jie Hu** received the B.E. degree in network engineering from Hebei University, China, in 2021. She is currently pursuing the master's degree with Xidian University, China. Her current research interests include vehicular networks and federated learning.



**Nan Cheng** (Senior Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, in 2016. He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, from 2017 to 2019. He is currently a Professor with the State Key Laboratory of ISN and the School of Telecommunications Engineering, Xidian University, Shaanxi, China. His current research interests include B5G/6G, space-air-ground integrated networks, big data in vehicular networks, and self-driving systems.



**Gaosheng Zhao** received the B.E. degree in telecommunications engineering from Xidian University, Xi'an, China, in 2021, where he is currently pursuing the master's degree with the School of Telecommunications Engineering. His current research interests include digital twin enabled vehicular networks and intelligent transportation systems.



**Rui Chen** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in communications and information systems from Xidian University, Xi'an, China, in 2005, 2007, and 2011, respectively. From 2014 to 2015, he was a Visiting Scholar with Columbia University, New York, NY, USA. He is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University. He is also the Director of INFOHAND Tech.-Xidian Guangzhou Institute Joint Intelligent Transportation Research Center. He has published over 80 papers in international journals and conferences. His research interests include broadband wireless communication systems, array signal processing, and intelligent transportation systems. He is a member of the IMT-2030 Promotion Group.



**Tom H. Luan** (Senior Member, IEEE) received the B.E. degree in electrical and computer engineering from Xi'an Jiaotong University, Xi'an, Shaanxi, China, the M.S. degree in electrical and computer engineering from The Hong Kong University of Science and Technology, Hong Kong, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada. He is currently with the School of Cyber Science and Engineering, Xi'an Jiaotong University. He has authored or coauthored more than 150 peer-reviewed conferences. His research interests include content distribution in vehicular networks, the performance evaluation of digital networks, and edge computing. He was a recipient of the 2017 IEEE VTS Best Land Transportation Best Paper Award and the IEEE ICCS 2018 Best Paper Award.



**Khalid Aldubaikhy** (Member, IEEE) received the B.E. degree from Qassim University, Saudi Arabia, in 2008, the M.A.Sc. degree in electrical and computer engineering from Dalhousie University, Halifax, NS, Canada, in 2012, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2019. He is currently an Assistant Professor with Qassim University, Buraydah, Al-Qassim, Saudi Arabia. His research interests include millimeter-wave wireless networks, medium access control, impulse radio ultra-wideband, and millimeter-wave 5G cellular networks.