ITC 4/52

Information Technology and Control
Vol. 52 / No. 4 / 2023
pp. 984-995
DOI 10.5755/j01.itc.52.4.34479

Weight Coefficient Based Adaptive Federated Learning for Vehicular Data Transmission

Received 2023/06/25 | Accepted after revision 2023/10/26

HOW TO CITE: Xie, H. (2023). Weight Coefficient Based Adaptive Federated Learning for Vehicular Data Transmission. *Information Technology and Control*, 52(4), 984-995. https://doi.org/10.5755/j01.itc.52.4.34479

Weight Coefficient Based Adaptive Federated Learning for Vehicular Data Transmission

Hui Xie
School of Computing and Information Science, Fuzhou Institute of Technology,
Fuzhou 350506, P. R. China; e-mail: 18231408@qq.com

Corresponding author: 18231408@qq.com

With the ever-increasing amount of vehicle data being generated, the collection and transmission of this data-to-data processing centers is consuming significant amounts of communication resources. The traditional method of compressing and transmitting the vehicle data is not effective in addressing the issue of efficient utilization of this data. In order to overcome this challenge, we propose an adaptive federated learning approach that avoids the need for transmitting data per vehicle. Our approach leverages the vehicle as a distributed training device node and enables the training of vehicle data using the vehicle's own computing power, thereby eliminating the need to transmit the data over the network. To further enhance the efficiency of the federated learning aggregation calculation, we introduce the information entropy function and cosine similarity calculation. By computing the similarity between the model and the benchmark model, we present a new round of model aggregation calculation weight. Finally, we validate the proposed algorithm using the actual datasets, demonstrating its high effectiveness.

KEYWORDS: Data Mining, Information Entropy, Federated Learning, Adaptive Weight Coefficient.

1. Introduction

With the rapid advancement of industrial manufacturing technology, the production of motor vehicles has witnessed an explosive growth [26, 41, 49]. Furthermore, with the significant improvement in calculation performance of smart sensor [5, 30, 31], these sensors have become smaller in size, thereby allowing for their widespread deployment in conventional motor vehicles to make them smarter. The increasing popularity of electric vehicles has further accelerated the trend towards intelligent motor vehicles [16, 17, 46], leading to an unprecedented surge in the amount of data generated by such vehicles [8, 14, 44].

Machine learning is being increasingly utilized to extract valuable insights from vehicular data, with the aim of improving the safety of intelligent motor vehicles [11, 22, 37, 40]. However, the collection, transmission, and storage of this large amount of motor vehicle data remains a major challenge. Although 5G networks have been deployed extensively, experiencing the advantages of fast transmission provided by these networks requires the development of large-scale, high-density base station networks [4, 20, 24, 32].

However, the deployment of 5G networks with strong network coverage will take a significant amount of time, which has hindered the practical applications of vehicular data [15, 19, 23, 45, 47]. To address the issue of high-throughput data transmission, people have turned to distributed computation, such as federated learning (FL), which allows for the collaborative training of deep learning prediction models using data sourced from multiple parties while avoiding massive data transmission [15, 21, 27, 51].

FL is an effective means of achieving distributed training of vehicular data and not only mitigates the network transmission burden caused by large-scale data transmission but also balances the data silos faced by different parties [3, 34]. Additionally, FL enables the sharing of data while ensuring privacy-preserving between participants' data, thereby helping users with small training datasets to achieve better results [7, 42]. However, several critical issues still need to be addressed. Firstly, the execution of data processing operations by different participants needs to meet temporal consistency in the acquired vehicular data to enable data owners to complete the effect of data aggregation simultaneously [2, 13, 43]. Secondly, the different data distribution from various participants necessitates the use of different processing strategies by the system. While current solutions focus on data redundancy and transferring data with redundant information removed to improve the efficiency of data transfer, more needs to be done to improve the transmission efficiency [1]. Even with the distributed approach to data transmission, challenges related to incomplete data-sharing applications still persist.

To enhance the efficiency of data sharing, improve fine-grained data application, and address the imbalance in multi-party data application, in this paper, we propose an adaptive federated learning approach using weight coefficients for high-throughput data transmission. This approach optimizes the transfer power of the network and improves the efficiency of data reuse between parties. Specifically, it facilitates efficient FL deep learning network training to enable participating parties to collaborate and achieve predictive model aggregation.

The contributions of our approach are summarized below.

1. We present a novel multi-party data transmission system that enables high-speed, collaborative data sharing without reliance on network transmission for deep network model training. To enhance the efficiency of multi-party data use, we employ a federated learning (FL) framework that circumvents network overload arising from transmission of large data quantities, while enabling the reuse of multi-party aggregates.
2. In order to expedite FL aggregation, we introduce an adaptive FL deep learning training framework that gauges the correlation of trained predictive models between different parties and allocates varying weights to the sharing of the aggregated model based on measures of information entropy and cosine similarity.
3. To evaluate our novel system, we perform simulation tests based on the CNN network. Results attest to the system's efficacy, which not only adapts to network conditions but also accelerates the speed of FL aggregation, achieving high-throughput data transmission.

The structure of this paper is as follows. We first introduce the related work in Section 2, followed by an overview of background knowledge in Section 3. Our proposed system, including each individual party, is presented in Section 4. Section 5 provides a detailed description of our proposed system and its ability to achieve adaptive federated learning training and high-throughput data transmission. Finally, in Section 6, we draw conclusions and suggest future research directions.

2. Related Work

Federated learning represents a distributed machine learning methodology capable of facilitating collaborative learning from diverse datasets without compromising the privacy of data owners. The FL architecture that comprises two levels of entity structure: numerous participating clients and central servers. In particular, a large number of local clients utilize their data to train corresponding local models, which are subsequently transmitted to the central server.

FL can significantly reduce data transmission by leveraging distributed collaborative training to achieve multi-party model integration. However, owing to the crucial role of numerous clients in training, in addition to their diverse communication capabilities and varying device computing power, ensuring consistency across local model training can present a challenging task in FL.

Considering a judicious utilization of the communication resources alongside new perceptive learning-oriented methods are vital, Taïk et al. [36] proposed an FL architecture that utilizes vehicular-to-vehicular resources to bypass the communication bottleneck where clusters of vehicles train models simultaneously and only the aggregate of each cluster is sent to the multi-access edge server. Elbir et al. [10] presented a federated learning framework that leveraged local computing power of vehicles in training models, reducing the need for data transmission. Nguyen et al. [27] proposed an adaptive federated learning approach that dynamically adjusts the aggregation calculation weight based on model similarity, enhancing the efficiency of model aggregation.

To mitigate the high bandwidth consumption, Xiao et al. [48] proposed a compression technique that effectively reduced the size of transmitted data while maintaining model accuracy. To ensure the robustness and security of the federated learning framework in vehicular environments, Du et al. [8] developed a secure and robust federated learning framework by incorporating secure aggregation, Byzantine fault tolerance, and verifiable model updates. Saputra et al. [33] utilized FL to accurately predict energy demand in electric vehicle networks with low communication overhead. In their model, charging stations act as clients and only exchange trained models with the charging station provider, ensuring the privacy of raw user data. Yu et al. [44] introduced a FL-based proactive content caching scheme for edge computing, in which mobile devices function as clients and the base station serves as the central server. Sozinov et al. [35] demonstrated the efficacy of FL in human activity recognition, highlighting its comparable accuracy to centralized learning.

Zhou et al. [52] proposed a FL-based real-time data processing architecture for multi-robot systems. Doku et al. [9] combined FL with blockchain to determine data relevance and store relevant data in a decentralized manner. Ren et al. [29] presented a FL-based framework for edge computing in large-scale network environments. Their approach focuses on jointly allocating communication and computing resources. Mowla et al. [25] introduced a FL-based jamming attack detection mechanism for flying ad hoc networks. They also utilized a client selection approach based on Dempster-Shafer theory to enhance the efficiency of FL.

Nguyen et al. [28] proposed a FL-based intrusion detection system that efficiently aggregated behavior profiles based on device-type-specific communication profiles. Notably, this system requires no labeled data for detection. Verma et al. [38, 39] proposed a web service-based implementation of FL for cross-domain enterprise data sharing. In a similar vein, Fantacci et al. [12] utilized FL to address the allocation of virtual machine replica copies in hybrid cloud mobile edge computing (MEC) networks. Their model leveraged FL to forecast user application demands and maximize the hit percentage. Lu et al. [18] proposed a privacy-preserving asynchronous FL mechanism for MEC. They introduce an asynchronous test process after each training round at a client, which determines whether the updates will be sent to the central server. Yan et al. [50] investigated a FL framework for power allocation in decentralized vehicular networks. They employed an online Actor-Critic algorithm for local training and achieve collaboration among clients by sharing gradients and weightages. Chen et al. [6] proposed a FL-based framework for minimizing "breaks in presence" in wireless networks. Their approach utilized FL to predict user location and orientation by enabling multiple clients to collaboratively train their deep echo state networks based on local data.

Despite these efforts, the computational burden of federated learning remains an undeniable challenge that warrants further optimization. Further, the efficiency of vehicular data also needs to be improved. To this end, we proposed an adaptive FL (AdaFL) approach, which offers an effective solution to address the communication burden arising from large data transfers.

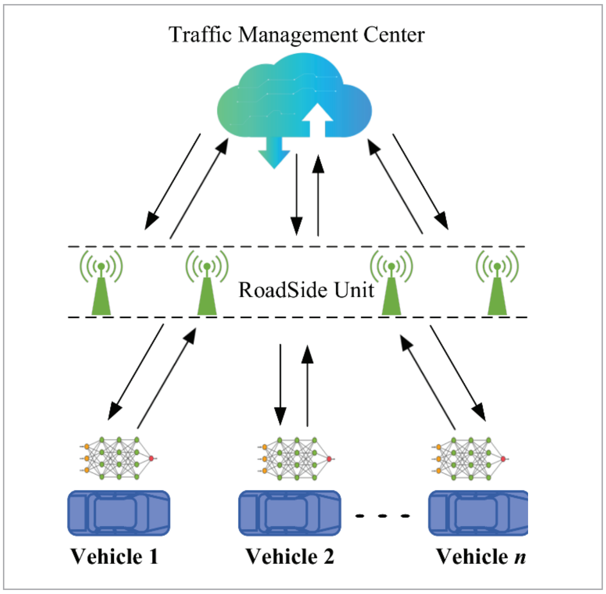
3. Adaptive Federated Learning Architecture

3.1. System Framework

To address the unique challenges posed by the varying communication capabilities and device computing power of local clients, we have developed an adaptive FL training framework, called AdaFL. This framework incorporates a weight coefficient for high-throughput data transmission, facilitating efficient training of neural network prediction models for autonomous vehicles under many motor vehicles and enhancing their robustness. The AdaFL system framework comprises four main components: the traffic management center (TMC), in-vehicle sensor vehicles (VSV), mobile vehicles central (MVC), and service provision content (SPC), as illustrated in Figure 1.

Figure 1

The system framework of our proposed AdaFL



Traffic Management Center (TMC): As the government department responsible for transportation, the TMC functions as the final aggregator of the numerous local models generated within AdaFL. Additionally, the TMC undertakes the task of neural network model initialization and distribution to the VSVs.

In-Vehicle Sensor Vehicle (VSV): As an essential component of the AdaFL system, the VSVs play a pivotal role in generating numerous local neural network prediction models. They use their data to iteratively optimize the model's predictive accuracy round after round, based on the received initialized model.

Roadside Server (RSS): As a roadside unit with enhanced computing power, the RSS not only facilitates the communication transit function of forwarding data to the TMC but also integrates the submitted local model within the specified communication radius.

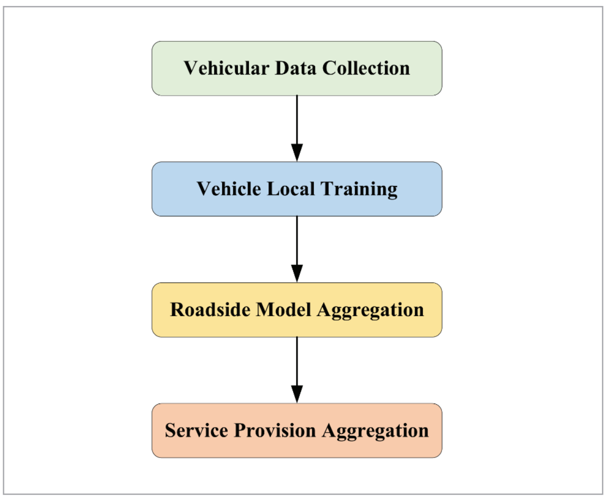
Service Provision Content (SPC): Once the final predictive model has been generated, the TMC not only transmits the aggregated predictive model to all participating vehicles in the AdaFL system but also improves the SPC of multiple applications while optimizing the traffic capacity of the entire transportation network.

3.2. System Overview

To achieve the high-throughput data transmission needed for vehicular data, our AdaFL system implementation relies on four key components: vehicular data collection, vehicle local training, radius model aggregation, and final model aggregation. The construction process of the entire AdaFL system is depicted in Figure 2. To enhance the aggregation effectiveness of the system on local models, we have introduced the Pearson correlation coefficient, allowing for the assignment of different weights to each local model based on its correlation with other local models during each training round.

Figure 2

The construction process of the entire AdaFL system



Vehicular Data Collection (VDC): VDC is the basic progress for realizing data transmission and data mining. The vehicular data is mainly collected by in-vehicle sensors and traditionally transmitted to a data terminal via wireless networks. Assume that there are m vehicles on the road where each vehicle $\mathcal{V}_i(1 \leq i \leq m)$. The vehicle $\mathcal{V}_i$ could generate a series of in-vehicle data that can reflect the driving status of the vehicle and the behavior of the driver and passengers, which can be represented by a dataset $\mathcal{X} = \{x_1, x_2, x_3, ..., x_k\}$ where $x_p(1 \leq p \leq k)$ represents the data of a certain type of information collected. Therefore, the data collected by the m vehicles be represented in the form of a matrix $\mathbb{A}$.

$$\mathbb{A} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mk} \end{bmatrix} \quad (1)$$

Vehicle Local Training (VLT): This progress in the AdaFL system is key to reducing the communication burden of data transmission on existing wireless networks. To utilize vehicular data without outsourcing data transfer, we introduce the FL framework to achieve the distribution training of the deep neural network model. Similarly, suppose that there are \$m\$ vehicles. Each vehicle $\mathcal{V}_i$ could train a predictive model per round using its computing device according to the training mechanism of FL. Assume that the vehicle $\mathcal{V}_i$ as the local client could generate the local model $\mathcal{M}_i$. Thereby, the local prediction models generated by \$m\$ vehicles passing through a round can be constructed as a prediction model set $\mathcal{Y} = \{\mathcal{M}_1^j, \mathcal{M}_2^j, ..., \mathcal{M}_m^j\}$, where $j(1 \leq j)$ represents the number of communication rounds in which FL training ultimately reaches the available model. Particularly when the vehicle $\mathcal{V}_i$ accomplishes each round of model $\mathcal{M}_i^j$ training, the vehicle $\mathcal{V}_i$ sends the $\mathcal{M}_i^j$ to the nearest roadside server.

Roadside Model Aggregation (RMA): As the second progress in AdaFL system, the main task of RMA is to aggregate local models submitted by clients within the communication radius R into an aggregated model. Benefiting from RSS's computing and storage capabilities, RMA mainly implements the aggregation task of some local models through RSS. Suppose that the $\mathcal{M}$ local clients are randomly averaged across \$U\$ number RSS, and each RSS contains K local clients within the communication radius R where $K \ll \mathcal{M}$. Therefore, the RSS utilizes the federated learning average (FedAvg) algorithm to accomplish the \$K\$ local clients aggregated. For instance, the RSS $\mathcal{P}(k \in [1, U]$) could generate the sub-aggregated model list $[M_{\mathcal{P}_1}, M_{\mathcal{P}_2}, M_{\mathcal{P}_3}, ..., M_{\mathcal{P}_U}]$.

Service Provision Aggregation: The service provision aggregation is the last progress to generate the final aggregated model. Mainly when the RSS receives the sub-aggregated model list $[M_{\mathcal{P}_1}, M_{\mathcal{P}_2}, M_{\mathcal{P}_3}, ..., M_{\mathcal{P}_U}]$, the SPC also uses the FedAvg algorithm to achieve the final aggregated model.

3.3. Design Goal

In this paper, we aim to achieve the following goals to balance the value of data and the amount of data transferred.

Our proposed scheme mainly could significantly reduce vehicular data transmission and fully use the intrinsic value of data.

Since our proposed strategy mainly relies on the distributed training mechanism of FL. Therefore, we need to compensate for the contribution of different participating vehicle clients to the aggregated prediction model so that it can converge the model as soon as possible.

4. Adaptive Federated Learning Algorithm

In this section, we mainly introduce the details of accomplishing the AdaFL algorithm. Based on the progress of the AdaFL system in the system overview, we also proceed through the following three processes: vehicle local model training, local model correlation measurement, and final model aggregation.

4.1. Vehicle Local Model Training

With the continuous enrichment of onboard sensors and the increasing intelligence of vehicles, vehicles are now like a comprehensive service platform. To fully tap the value of vehicular data, collecting vehicular data is the most basic operation. According to the above statement, there are m vehicles on the road, which are randomly assigned to a specific area within the scope of K RSU. The same assumption, each vehicle could generate its own dataset $\mathcal{X} = \{x_1, x_2, x_3, ..., x_k\}$, where each element in the $\mathcal{X}$ means a collection of certain types of in-vehicle data. The use of each data type is guided by the training purpose of the vehicle. To realize the availability of data, according to the different types of vehicle users who collect data, the data will be marked accordingly to realize the usability of the data. Similarly, assume that the label could be defined as the vector $\mathcal{L} = [l_1, l_2, l_3, ..., l_K]$, each element l_q,$(1 \le q \le K)$ in vector $\mathcal{L}$ corresponds to the generated dataset $\mathcal{X}$, such as.

$$\mathbb{D} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k}, & l_1 \\ x_{21} & x_{22} & \cdots & x_{2k}, & l_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mk}, & l_k \end{bmatrix}. \tag{2}$$

When the vehicle generates the dataset, the $\mathcal{X}$ will be stored in the data storage device inside the vehicle. Thanks to the increased computing power of in-vehicle devices, we can use the data without transferring the in-vehicle data generated by the vehicle. Therefore, based on the above introduction, we can minimize data transmission and significantly save the bandwidth of communication resources.

As the client node of local model training, the vehicle uses its own standard data to complete the training of the corresponding prediction model based on selecting a specific data type. Assume that the vehicle $\mathcal{V}_i$ uses the training dataset $[\mathcal{X} : l_i]$ based on the deep neural network learning algorithm f to generate the local model $\mathcal{M}_i$, such as the following:

$$|\mathcal{X}: l_i| \rightarrow f(\mathcal{M}_i). \tag{3}$$

Therefore, the m vehicles could generate m local predict model, which further is structured as a vector $M = [\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, ..., \mathcal{M}_m]$. Assuming that there is an ideal state, at a particular moment, m vehicles are randomly assigned to the range of the communication radius R of K the RSU ($\frac{m}{K} \in \mathrm{N}^*$). Therefore, the vector M also is divided into $\frac{m}{K}$ components, and further the vector M could be rewritten as $\widehat{\mathcal{M}}_1 = [\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, ..., \mathcal{M}_{\frac{m}{K}}]$. After all the vehicle nodes participating in the local client training have trained the local prediction model, the vehicle will select the adjacent RSU server base station and send the trained model parameters to RSU. To speed up the aggregation speed of local models, it is imperative to fully explore the weight between different model parameters and realize the weight analysis of local models submitted by different vehicles.

4.2. Local Model Correlation Measurement

We construct a model parameter weights allocation algorithm by introducing cosine similarity to complete the analysis of different weights between different model parameters. Firstly, we utilize the KL divergence based on information entropy to compute the between different model relationships. And then, the so-called cosine similarity refers to calculating the cosine angle between different values. The KL divergence could be shown as follows.

$$\begin{aligned}\mathcal{K}_{KL}(X|Y) &= \mathcal{H}(X) - \mathcal{H}(Y) \\ &= \textstyle\sum_{x\in X, y\in Y} p(x) \cdot \log\frac{p(x)}{p(y)}.\end{aligned} \tag{4}$$

In Equation (4), the Y are the random variable sets, X are the independent variable set, and $\mathcal{H}(X, Y)$ is the joint probability distribution function of X and Y. Therefore, the $\mathcal{H}(X, Y)$ could be calculated by the information entropy $\mathcal{H}(X)$ by the formula $\mathcal{H}(X) = -\sum_{x\in X} p(x)\log(p(x))$ and the $\mathcal{H}(Y)$ could be obtained by $\mathcal{H}(Y) = -\sum_{y\in Y} p(y)\log(p(Y))$.To illustrate how to calculate the KL divergence between different models and cosine similarity, we through a simple example to illustrate as follows.

Suppose there are 2-dimensional vectors $A = [a_1, a_1, a_2, a_3]$ and $B = [a_1, a_1, a_2, a_3]$, we can obtain that the probability of element a_1 is 0.5 in the vector A, the probability of element a_2 is 0.25 in the vector A, the probability of element a_3 is 0.25 in the vector A. Similarly, the probability of element a_1 is 0.25 in the vector B, the probability of element a_2 is 0.5 in vector B, and the probability of element a_3 is 0.25 in vector B. Therefore, the KL divergence of vectors A and B could be computed as $\mathcal{K}_{KL}(A|B) = 0.5\log\frac{0.5}{0.25} + 0.25\log\frac{0.25}{0.5} + 0.25\log\frac{0.25}{0.25} = 0.25$. Here, we stipulate that when the KL divergence of the parameters of the two prediction models is less than the threshold α, we aggregate the two models. When some model parameters with high distribution similarity are aggregated, in order to balance the contribution of the remaining model parameters to the final prediction model, we next calculate the discrete cosine angle of the models. Assume that there are two-dimensional vectors $T_1 = [t_1, t_2]$ and $W_2 =$

$[w_1, w_2]$, the cosine angle can be computed as follows:

$$\cos\theta = \frac{t_1w_1+t_2w_2}{\sqrt{(t_1)^2+(t_2)^2}+\sqrt{(w_1)^2+(w_2)^2}}. \tag{5}$$

In most cases, the dimension of the data is greater than 2, so we need to extend the dimension to N. Similarly, suppose that there are n dimensional vectors $A = [a_1, a_2, a_3, ..., a_N]$ and $B = [b_1, b_2, b_3, ..., b_N]$. Therefore, the cosine angle can also be computed as follows:

$$\cos\theta = \frac{\sum_{i=1}^{n}(a_i \times b_i)}{\sum_{i=1}^{n}(a_i)^2 \times \sum_{i=1}^{n}(b_i)^2} = \frac{A \cdot B}{|A| \times |B|}. \tag{6}$$

To calculate whether the local model submitted by each vehicle client deviates from the final aggregation direction, we first need to use the FedAvg algorithm to perform an aggregation calculation on the local model to obtain a baseline for calculating cosine similarity. Suppose that the aggregation model obtained by aggregation calculation is $\mathcal{M}o$, we need to calculate the cosine similarity between each local model and the benchmark aggregation model respectively to cosine similarity coefficient $\Theta = [\theta_1, \theta_2, ..., \theta_{\frac{m}{K}}]$ as weight values. Based on the obtained weight vector, we redesign the federal learning model aggregation algorithm with weight coefficients. The refactored formulas are shown in 7 and 8.

$$f(w) = \sum_{k=1}^{n} \theta_i \frac{n_k}{n} F_k(w) \tag{7}$$

$$F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w) \tag{8}$$

Algorithm 1: AdaFL Algorithm

Input: Vehicular Clients: $O = \{o_1, o_2, ..., o_N\}$. B is the local mini-batch size, E is the number of local epochs, α is the learning rate, $\nabla L(.;.)$ is the gradient optimization function.

Output: $\mathcal{M}_j$

Initialize ω^*.
for communication round $t = 1, 2, ...$ **do**
  $\{O_M\} \leftarrow$ select VCs from O to join in this round;
  TMC broadcasts global model ω^* to $\{O_M\}$ to;
  for each VC $o \in \{O_M\}$ **do**
    Initialize $\omega_{(o,t)} = \omega^*$;
    $\omega_{(i,(o,t+1))} \leftarrow LocalUpdate(o, \omega_{(o,t)})$;
    Calculate the KL divergence $K_{KL}(\omega_{(i,(o,t+1))} \,||\, \omega_{(j,(o,t+1))})$;
    if $K_{KL} \leq \alpha$ **then**
      Computing $\bar{\omega}_{i,j} \leftarrow FedAvg(\omega_{(i,(o,t+1))} \,||\, \omega_{(j,(o,t+1))})$;
    else
      $\cos\theta = \frac{\sum_{i=1}^{n}(a_i \times b_i)}{\sum_{i=1}^{n}(a_i)^2 \times \sum_{i=1}^{n}(b_i)^2} = \frac{A \cdot B}{|A| \times |B|}$
      $\Theta = [\theta_1, \theta_2, \theta_3, ..., \theta_K]$
      $\bar{\omega}_{(j,t+1)} \leftarrow \frac{\theta_i}{|\{O_M\}|} \sum_{o \in O_M} \omega_{(j,(o,t+1))}$;
Local Update $(o, \omega_{(o,t)})$:
$\mathcal{B} \leftarrow$ (split $\mathcal{D}_i$ into batches of size B);
if each local epoch i from 1 to E **then**
  if batch $b \in \mathcal{B}$ **then**
    $\omega \leftarrow \omega - \alpha \cdot \nabla L(\omega; b)$;

4.3. Predictive Model Aggregation

To be able to adapt to the rapid movement of vehicles, our aggregation calculations involve two stages. The first stage involves the initial local model aggregation task, which is primarily concentrated in the RSU. When the RSU receives local model submissions from various vehicles within its communication radius R, it performs aggregation calculations, generating partially aggregated sub-models, which are then sent to the SPC for further aggregation of all sub-models. This step generates the final predictive global model. Subsequently, the SPC transmits the generated global model to each participating vehicle in the training for a new round of model updates.

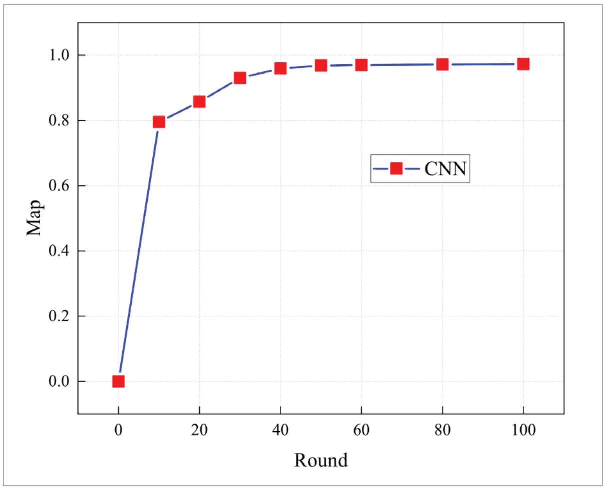
5. Simulation Experiment

To demonstrate the effectiveness of the AdaFL system, we utilized a CNN model to construct a prediction model based on the MNIST and FashionMNIST dataset, training data from 100 clients. Notably, the authors randomly assigned the dataset to each client based on the type of data they obtain to satisfy the characteristics of non-IID. The experiment environment employed an Intel(R) Core (TM), i7-12600 CPU@3.40GHZ, and 16.00GB of RAM, RTX3090 of GPU, to serve as our central server.

We first evaluated the performance of the CNN model based on 100 clients, as well as the performance via centralized training by mining the MNIST dataset, as shown in Figure 3. The results indicate that federated learning can achieve very stable results, even with a large number of training nodes. Specifically, when the number of training rounds reaches approximately 50, the CNN training method based on federated learning can achieve accuracy stability. Therefore, we can generate a model through the above process without collecting data generated by vehicles, and conduct mining training on vehicle data effectively.

Figure 3

Prediction accuracy by CNN model on MNIST test set

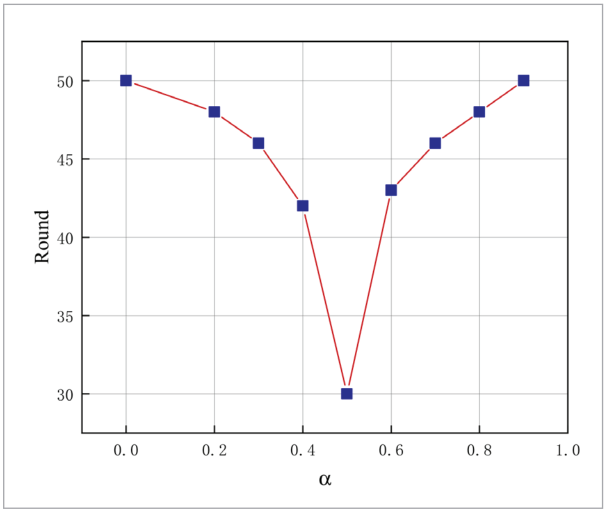


To further enhance the stability of local model prediction accuracy in federated learning, we used the KL divergence method based on information entropy to calculate the similarity between any two models. Specifically, we first calculated the KL divergence value between any two local model parameters and compared it with a preset threshold. If the KL divergence value between the model parameters was less than or equal to the threshold α, we used the traditional FedAvg algorithm to aggregate the corresponding model parameters. If the KL divergence value was greater than the threshold α, we did not aggregate its model parameters.

To identify a suitable threshold α, we varied values of α within the range [0, 1], and evaluated the number of rounds required for the entire algorithm to stabilize the prediction accuracy under different values. The results are shown in Figure 5, which shows that the

Figure 5

The choice of α



number of training rounds varies with the value of α in the form of a quadratic function with an upward opening. The curve has a minimum value, and when α = 0.5, the number of rounds required for the entire algorithm to stabilize is the least.

To further demonstrate the effectiveness of our proposed scheme in maximizing the saving of communication resources in the network, we compared the byte stream size (BSS) in different scenarios with or without vehicle data outsourcing, as shown in Figure 5. The results showcase that the algorithm proposed in this paper can significantly reduce communication resources consumption and enhance the utilization rate of vehicle data. In Figure 6, the green color represents the byte stream of communication required to collect vehicle data for centralized training in the traditional form. The consumption of this communication resource is mainly to complete the transmission of vehicle data. The orange color represents the consumption of communication resources required to use the method proposed in this paper. The consumption is primarily the resource consumption when participating in the model training and submitting the model by each client.

Lastly, to further validate the effectiveness of our proposed method, simulations were conducted to show that the cosine similarity calculation proposed in this study can reduce the number of interactive rounds of federated learning training, as shown in Figure 7.

Figure 6

Comparison of the communication burden with centralized data collection

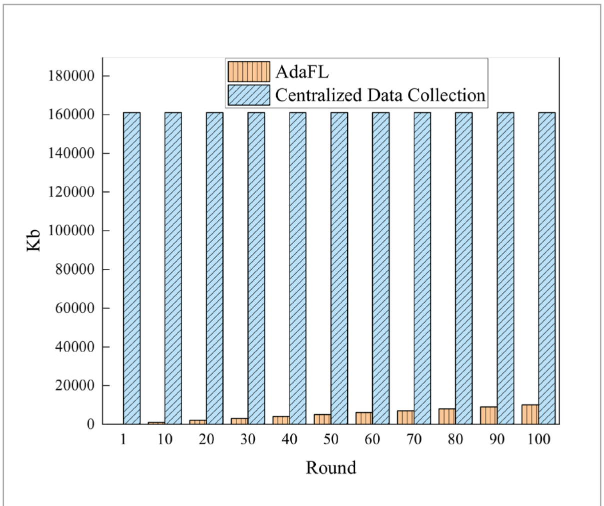
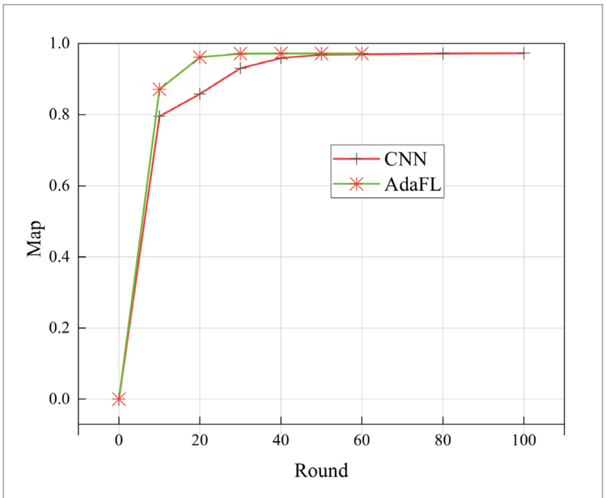


Figure 7

Comparison of cosine similarity calculation



The figure contains two curves, where the green line represents the result of the federal aggregation calculation using the cosine similarity proposed in this paper, while the red line represents the prediction result after the aggregation calculation using the traditional fed learning aggregation algorithm.

In Table 1, on FashionMNIST dataset, we also compared our proposed AdaFL with the centralized data collection (CDC) and CNN network with respect to the metrics byte stream size (BSS) and Map. Through comparisons, our proposed algorithm can achieve similar model prediction results with fewer rounds. Thus, our proposed cosine similarity calculation method has a positive impact and can improve the model aggregation calculation in federated learning training.

Table 1

Performance comparison between AdaFL and centralized data collection (CDC) and CNN network on FashionMNIST dataset

Round	BSS [Kb]		Map [%]	
	CDC	AdaFL	CNN	AdaFL
20	$1.6*10^5$	$0.06*10^5$	85.2	89.4
40	$1.6*10^5$	$0.11*10^5$	88.4	91.6
60	$1.6*10^5$	$0.14*10^5$	89.1	92.4
80	$1.6*10^5$	$0.17*10^5$	91.7	92.7
100	$1.6*10^5$	$0.21*10^5$	92.0	92.7

6. Conclusion

This paper presented a system designed for efficient utilization of vehicular data by utilizing adaptive federated learning via weight coefficient, without vehicular data transmission. To achieve this objective, we proposed two vehicle data usage mechanisms. The first mechanism involved using a vehicle node for distributed training of vehicular data, thus facilitating the mining process of the vehicular data. The second mechanism aimed to improve the efficiency of local model aggregation in federated learning by introducing information entropy and cosine similarity calculation into the system algorithm. In future research, we plan to further explore adaptive federated learning aggregation and develop adaptive aggregation computing for a wider range of scenarios. This will enable us to unlock the full potential of adaptive federated learning for various applications.

Acknowledgement

This work is supported by the Research Fund of Fuzhou Institute of Technology under Grant FTKY2022007.

References

1. Abdel-Basset, M., Moustafa N., Hawash H., Razzak, I., Sallam, K. M., Elkomy, O. M. Federated Intrusion Detection in Blockchain-Based Smart Transportation Systems. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(3), 2523-2537. https://doi.org/10.1109/TITS.2021.3119968
2. Abid, A., Yuksekgonul, M., Zou, J. Meaningfully Debugging Model Mistakes Using Conceptual Counterfactual Explanations. International Conference on Machine Learning. PMLR, 2022, 162, 66-88.
3. Blaine, N., Marco B., Fuching, J. C., Joseph, A. D., Rubinstein, B. I. P., Saini, U., Sutton, C., Tygar, J. D., Xia, K. Exploiting Machine Learning to Subvert Your Spam Filter. Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, 2008, 7, 1-9.
4. Chehri, A., Fortier, P., Saadane, R. A Framework for 5G Ultra-Reliable Low Latency for Industrial and Mission-Critical Machine-Type Communication. International Conference on Human-Centered Intelligent Systems, 2021, 99-109. https://doi.org/10.1007/978-981-16-3264-8_10
5. Chen, B., Wan, J., Shu, L., et al. Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. IEEE Access, 2017, 6, 6505-6519. https://doi.org/10.1109/ACCESS.2017.2783682
6. Chen, M., Semiari, O., Saad, W., Liu, X., Yin, C. Federated Echo State Learning for Minimizing Breaks in Presence in Wireless Virtual Reality Networks. IEEE Transactions on Wireless Communication, 2020, 19(1), 177-191. https://doi.org/10.1109/TWC.2019.2942929
7. Chepurko, N., Marcus, R., Zgraggen, E., Fernandez, R. C., Kraska, T., Karger, D. ARDA: Automatic Relational Data Augmentation for Machine Learning. arXiv preprint arXiv:2003.09758, 2020. https://doi.org/10.48550/arXiv.2003.09758
8. Dobson, R., Semple, S. Changes in Outdoor Air Pollution Due to COVID-19 Lockdowns Differ by Pollutant: Evidence from Scotland. Occupational and Environmental Medicine, 2020, 77(11), 798-800. https://doi.org/10.1136/oemed-2020-106659
9. Doku, R., Rawat, D. B., Liu, C. Towards Federated Learning Approach to Determine Data Relevance in Big Data. IEEE International Conference on Information Reuse & Integration, 2019, 184-192. https://doi.org/10.1109/IRI.2019.00039
10. Elbir, A. M., Soner, B., Çöleri, S., et al. Federated Learning in Vehicular Networks. IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, Greece, 2022, 72-77. https://doi.org/10.1109/MeditCom55741.2022.9928621
11. ElHalawany, B. M., El-Banna, A., Wu, K. Physical-Layer Security and Privacy for Vehicle-to-Everything. IEEE Communications Magazine, 2019, 57(10), 84-90. https://doi.org/10.1109/MCOM.001.1900141
12. Fantacci, R., Picano, B. Federated Learning Framework for Mobile Edge Computing Networks. CAAI Transactions on Intelligent Technology, 2020, 5(1), 15-21. https://doi.org/10.1049/trit.2019.0049
13. Garrido, G. M., Sedlmeir, J., Uludağ, Ö., Alaoui, I. S., Luckow, A., Matthes, F. Revealing the Landscape of Privacy-Enhancing Technologies in the Context of Data Markets for the IoT: A Systematic Literature Review. Journal of Network and Computer Applications, 2022, 207, 103465. https://doi.org/10.1016/j.jnca.2022.103465
14. Hoseinzadeh, N., Arvin, R., Khattak, A. J., Han, L. D. Integrating Safety and Mobility for Pathfinding Using Big Data Generated by Connected Vehicles. Journal of Intelligent Transportation Systems, 2020, 24(4), 404-420. https://doi.org/10.1080/15472450.2019.1699077
15. Huba, D., Nguyen, J., Malik, K., Zhu, R., Rabbat, M., Yousefpour, A., Wu, C.-J., Zhan, H., Ustinov, P., Srinivas, H., Wang, K., Shoumikhin, A., Min, J., Malek, M. Papaya: Practical, Private, and Scalable Federated Learning. Proceedings of Machine Learning and Systems, 2022, 4, 814-832.
16. Li, Q. Design of Automated Control System for Vehicle-mounted Communication Equipment Based on PLC and Shortwave Communication. International Conference on Electronics and Renewable Systems (ICEARS), IEEE, 2022, 796-799. https://doi.org/10.1109/ICEARS53579.2022.9752371
17. Liu, Y., Meng, X., Huang, X. Route Planning Strategy of Intelligent Car Based on Camera Sensor. IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), IEEE, 2022, 729-732. https://doi.org/10.1109/IPEC54454.2022.9777323
18. Lu, X., Liao, Y., Lio, P., Hui, P. Privacy-Preserving Asynchronous Federated Learning Mechanism for Edge Network Computing. IEEE Access, 2020, 8, 48970-48981. https://doi.org/10.1109/ACCESS.2020.2978082
19. Luping, W., Wei, W., Bo, L. CMFL: Mitigating Communication Overhead for Federated Learning. IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2019, 954-964. https://doi.org/10.1109/ICDCS.2019.00099

20. Kaur, K., Guillemin, F., Rodriguez, V. Q., Sailhan, F. Latency and Network Aware Placement for Cloud-Native 5G/6G Services. IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2022, 114-119. https://doi.org/10.1109/CCNC49033.2022.9700582

21. Khan, L. U., Saad, W., Han, Z., Hossain, E., & Hong, C. S. Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges. IEEE Communications Surveys & Tutorials, 2021, 23(3), 1759-1799. https://doi.org/10.1109/COMST.2021.3090430

22. Khayyam, H., Javadi, B., Jalili, M., & Jazar, R. N. Artificial Intelligence and Internet of Things for Autonomous Vehicles. Nonlinear approaches in engineering applications, 2020, 39-68. https://doi.org/10.1007/978-3-030-18963-1_2

23. Ma, C., Li, J., Ding, M., Yang, H. H., Shu, F., Quek, T. Q., & Poor, H. V. On Safeguarding Privacy and Security in the Framework of Federated Learning. IEEE Network, 2020, 34(4), 242-248. https://doi.org/10.1109/MNET.001.1900506

24. Moussaoui, M., Bertin, E., Crespi, N. Telecom Business Models for Beyond 5G and 6G Networks: Towards Disaggregation? International Conference on 6G Networking (6GNet), IEEE, 2022, 1-8. https://doi.org/10.1109/6GNet54646.2022.9830514

25. Mowla, N. I., Tran, N. H., Doh, I., Chae, K. Federated Learning-Based Cognitive Detection of Jamming Attack in Flying Ad-Hoc Network. IEEE Access, 2019, 8, 4338-4350. https://doi.org/10.1109/ACCESS.2019.2962873

26. Mu, X. Public Security Road Traffic Management Strategy Based on Big Data and Intelligent Dispatching System. International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), IEEE, 2022, 589-592. https://doi.org/10.1109/ICSCDS53736.2022.976075

27. Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., Poor, H. V. Federated Learning for Internet of Things: A Comprehensive Survey. IEEE Communications Surveys & Tutorials, 2021, 23(3), 1622-1658. https://doi.org/10.1109/COMST.2021.3075439

28. Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A. R. DIoT: A Federated Self-Learning Anomaly Detection System for IoT. IEEE International Conference on Distributed Computing Systems, 2019, 756-767. https://doi.org/10.1109/ICDCS.2019.00080

29. Han, Y., Li, D., Qi, H., Ren, J., Wang, X. Federated Learning-Based Computation Offloading Optimization in Edge Computing Supported Internet of Things. IEEE Access, 2019, 7, 69194-69201. https://doi.org/10.1109/ACCESS.2019.2919736

30. Rodríguez, A., Valverde, J., Portilla, J., Otero, A., Riesgo, T., De la Torre, E. FPGA-Based High-Performance Embedded Systems for Adaptive Edge Computing in Cyber-Physical Systems: The Artico3 Framework. Sensors, 2018, 18(6), 1877-1993. https://doi.org/10.3390/s18061877

31. Rohini, P., Tripathi, S., Preeti, C. M., Renuka, A., Gonzales, J. L. A., Gangodkar, D. A Study on the Adoption of Wireless Communication in Big Data Analytics Using Neural Networks and Deep Learning. 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), IEEE, 2022, 1071-1076. https://doi.org/10.1109/ICACITE53722.2022.9823439

32. Rossi, M. A. The Advent of 5G and the Non-Discrimination Principle. Telecommunications Policy, 2022, 46(4), 102279. https://doi.org/10.1016/j.telpol.2021.102279

33. Saputra, Y. M., Hoang, D. T., Nguyen, D. N., Dutkiewicz, E., Mueck, M. D., Srikanteswara, S. Energy Demand Prediction with Federated Learning for Electric Vehicle Networks. IEEE Global Communication Conference, 2019, 1-6. https://doi.org/10.1109/GLOBECOM38437.2019.9013587

34. Song, C., Shmatikov, V. Auditing Data Provenance in Text-Generation Models. ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019,196-206.https://doi.org/10.1145/3292500.3330885

35. Sozinov, K., Vlassov, V., Girdzijauskas, S. Human Activity Recognition Using Federated Learning. IEEE International Conference on Parallel Distributed Process, 2019, 1103-1111. https://doi.org/10.1109/BDCloud.2018.00164

36. Taïk, A., Mlika, Z., Cherkaoui, S. Clustered Vehicular Federated Learning: Process and Optimization. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(12), 25371-25383. https://doi.org/10.1109/TITS.2022.3149860

37. Tong, W., Hussain, A., Bo, W. X., Maharjan, S. Artificial Intelligence for Vehicle-to-Everything: A Survey. IEEE Access, 2019, 7, 10823-10843. https://doi.org/10.1109/ACCESS.2019.2891073

38. Verma, D., White, G., Mel, de. G. Federated AI for the Enterprise: A Web Services Based Implementation. IEEE International Conference on Web Services, 2019, 20-27. https://doi.org/10.1109/ICWS.2019.00016

39. Verma, D., White, G., Mel, de G. Federated Learning Based Proactive Content Caching in Edge Computing. IEEE Global Communication Conference, 2019, 1-6. https://doi.org/10.1109/GLOCOM.2018.8647616
40. Vishwakarma, L., Das, D. Smart Coin: A Novel Incentive Mechanism for Vehicles in Intelligent Transportation System Based on Consortium Blockchain. Vehicular Communications, 2022, 33, 100429. https://doi.org/10.1016/j.vehcom.2021.100429
41. Wang, K., Wang, X., Liu, X., Jolfaei, A. Task Offloading Strategy Based on Reinforcement Learning Computing in Edge Computing Architecture of Internet of Vehicles. IEEE Access, 2020, 8, 173779-173789. https://doi.org/10.1109/ACCESS.2020.3023939
42. Wang, Q., Yin, H., Chen, T., Yu, J., Zhou, A., Zhang, X. Fast-Adapting and Privacy-Preserving Federated Recommender System. The VLDB Journal, 2022, 31(5), 877-896. https://doi.org/10.1007/s00778-021-00700-6
43. Wang, Y., Su, Z., Ni, J., Zhang, N., Shen, X. Blockchain-Empowered Space-Air-Ground Integrated Networks: Opportunities, Challenges, and Solutions. IEEE Communications Surveys & Tutorials, 2021, 24(1), 160-209. https://doi.org/10.1109/COMST.2021.3131711
44. Wang, Z., Liao, X., Wang, C., Oswald, D., Wu, G. Boriboonsomsin, K., Barth, M. J., Han, K., Kim, B., Tiwari, P. Driver Behavior Modeling Using Game Engine and Real Vehicle: A Learning-Based Approach. IEEE Transactions on Intelligent Vehicles, 2020, 5 (4), 738-749. https://doi.org/10.1109/TIV.2020.2991948
45. Wu, Q., He, K., Chen, X. Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework. IEEE Open Journal of the Computer Society, 2020, 1, 35-44. https://doi.org/10.1109/OJCS.2020.2993259
46. Wu, X., Wang, P. Intelligent Transportation Information Interaction Technology. Intelligent Road Transport Systems. Springer, Singapore, 2022, 151-192. https://doi.org/10.1007/978-981-16-5776-4
47. Wu, X., Zhang, Y., Shi, M., Li, P., Li, R., Xiong, N. N. An Adaptive Federated Learning Scheme with Differential Privacy Preserving. Future Generation Computer Systems, 2022, 127, 362-372. https://doi.org/10.1016/j.future.2021.09.015
48. Xiao, H., Zhao, J., Pei, Q., Feng, J., Liu, L., Shi, W. Vehicle Selection and Resource Optimization for Federated Learning in Vehicular Edge Computing. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(8), 11073-11087. https://doi.org/10.1109/TITS.2021.3099597
49. Xu, D., Li, T., Li, Y., Su, X., Tarkoma, S., Jiang, T., Crwocroft, J., Hui, P., Edge Intelligence: Empowering Intelligence to the Edge of Network. Proceedings of the IEEE, 2021, 109(11), 1778-1837. https://doi.org/10.1109/JPROC.2021.3119950
50. Yan, M., Chen, B., Feng, G., Qin, S. Federated Cooperation and Augmentation for Power Allocation in Decentralized Wireless Networks. IEEE Access, 2020, 8, 48088-48100. https://doi.org/10.1109/ACCESS.2020.2979323
51. Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H. Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2019, 13(3), 1-207. https://doi.org/10.1007/978-3-031-01585-4
52. Zhou, W., Li, Y., Chen, S., Ding, B. Real-Time Data Processing Architecture for Multi-Robots Based on Differential Federated Learning. IEEE Proceedings of Smart World, 2018, 462-471. https://doi.org/10.1109/SmartWorld.2018.00106