electronics

Article

A Federated Learning Framework Based on Incremental Weighting and Diversity Selection for Internet of Vehicles

Yuan Lei [1], Shir Li Wang [1,*], Minghui Zhong [2,3,*], Meixia Wang [3] and Theam Foo Ng [4]

1 Faculty of Art, Computing and Creative Industry, Universiti Pendidikan Sultan Idris, Tanjong Malim 35900, Perak, Malaysia
2 Faculty of Accountancy and Digital Business, Payap University, Chiang Mai 50000, Thailand
3 School of Information and Engineering, Guangxi University of Foreign Languages, Nanning 530001, China
4 Centre for Global Sustainability Studies, Universiti Sains Malaysia, Gelugor 11800, Penang, Malaysia
* Correspondence: shirli_wang@fskik.upsi.edu.my (S.L.W.); mhzhong2022@126.com (M.Z.)

Abstract: With the rapid increase of data, centralized machine learning can no longer meet the application requirements of the Internet of Vehicles (IoV). On the one hand, both car owners and regulators pay more attention to data privacy and are unwilling to share data, which forms the isolated data island challenge. On the other hand, the incremental data generated in IoV are massive and diverse. All these issues have brought challenges of data increment and data diversity. The current common federated learning or incremental learning frameworks cannot effectively integrate incremental data with existing machine learning (ML) models. Therefore, this paper proposes a Federated Learning Framework Based on Incremental Weighting and Diversity Selection for IoV (Fed-IW&DS). In Fed-IW&DS, a vehicle diversity selection algorithm was proposed, which uses a variety of performance indicators to calculate diversity scores, effectively reducing homogeneous computing. Also, it proposes a vehicle federated incremental algorithm that uses an improved arctangent curve as the decay function, to realize the rapid fusion of incremental data with existing ML models. Moreover, we have carried out several sets of experiments to test the validity of the proposed Fed-IW&DS framework's performance. The experimental results show that, under the same global communication round and similar computing time, the Fed-IW&DS framework has significantly improved performance in all aspects compared to the frameworks FED-AVG, FED-SGD, FED-prox & the decay functions linear, square curve and arc tangent. Specifically, the Fed-IW&DS framework improves the Acc (accuracy), loss (loss), and Matthews correlation coefficient (MCC) by approximately 32%, 83%, and 66%, respectively. This result shows that Fed-IW&DS is a more reliable solution than the common frameworks of federated learning, and it can effectively deal with the dynamic incremental data in the IoV scenario. Our findings should make a significant contribution to the field of federated learning.

Keywords: federated learning; incremental learning; Internet of Vehicles; diversity selection; arctangent curve; FED-AVG; FED-SGD; FED-prox

check for updates

Citation: Lei, Y.; Wang, S.L.; Zhong, M.; Wang, M.; Ng, T.F. A Federated Learning Framework Based on Incremental Weighting and Diversity Selection for Internet of Vehicles. *Electronics* **2022**, *11*, 3668. https://doi.org/10.3390/electronics11223668

Academic Editors: Muhammad Salman Haleem, Liangxiu Han, Ernesto Iadanza and Baihua Li

Received: 13 October 2022
Accepted: 7 November 2022
Published: 9 November 2022

1. Introduction

As an important application scenario of 5G (5th Generation Mobile Communication Technology), the Internet of Vehicles (IoV) has attracted wide research attention [1]. According to Intel's estimate, each intelligent vehicle will generate about 4000 GB of data every day, which is equivalent to the data generated by nearly 3000 mobile phone users. Due to this enormous growth, Brian Krzanich, CEO of Intel, said: "Data is the new oil in the future of automated driving!". Traditional machine learning is realized by uploading all data to a central server for processing. However, with growing significant data in loV, there are many shortcomings in centralized machine learning with car owner privacy leakage, high communication cost, and high transmission delay. The challenges emerged from these shortcomings, such as data isolated island, incremental data, and data diversity.

Several reasons for the isolated data island challenge in IoV include the car owner's unwillingness, strict policies, and technical difficulties. (1) Car owner's unwillingness. The IoV data contains many sensitive data about personal privacy, such as travel trajectory, navigation information, in-car recordings, camera images, etc. Most car owners are unwilling to share this sensitive data. (2) Strict policies. Once these sensitive data are stored and processed in the central server, the risk of data leakage will be greatly increased [2]. In 2018, the "General Data Protection Regulation" was implemented to protect user data privacy in Europe. In 2019, China launched the "Guide to Internet Personal Information Security Protection." The European Data Protection Committee (EDPB) adopted the "Guidelines on Processing Personal Data in the Context of Connected Vehicles and Mobility-Related Applications" in 2021. The guidelines introduced over the years explain the privacy protection, data risks, and countermeasures in different scenarios of IoV. These laws or guidance indicate that data owners must be supervised and be obligated to protect data. (3) Technical difficulties. Last, but not least, there were about 119 million intelligent vehicles in 2018, and by 2023, this number will nearly triple to 353 million. Meanwhile, as mentioned above, every smart vehicle will generate about 4000 GB of data every day. Therefore, even though 5G is widely applied on this planet, it is still impossible to realize data sharing and real-time calculation in IoV.

Federated learning provides a decentralized, distributed, and secure solution for IoV data sharing. In federated learning, the local data are only stored in the vehicle nodes, and the local machine learning (ML) parameters are trained [3]. Many local ML parameters are aggregated to the central server to train a more accurate global ML model. The data sharing problem was transformed into the problem of local ML parameter sharing, which solves data privacy and reduces transmission costs. However, the real-time incremental data generated in IoV is massive. Different road conditions will cause their incremental data, such as urban and suburban, uphill and downhill, day and night, rainy, snowy, and sunny days [4]. To better adapt to driving assistance or safety warning functions in IoV, it is necessary to update the old ML model. The key research has focused on fusing the incremental data with the old ML model and ensuring that all vehicles can effectively participate in the global ML model training. That's the incremental data challenge.

The incremental data challenge can be solved by incremental learning. Incremental learning means that a learning system can constantly learn new knowledge from new samples and can save most of the previously known knowledge [5]. However, the traditional federated learning method, which does not consider the weight of incremental data, largely depends on the repetition of the training process, and even leads to the severe decline of global ML model accuracy and ML model deviation. In addition, since both the common goal of federated learning and incremental learning is to obtain more reliable prediction results through local ML model parameters of multiple vehicles, theoretically, the greater difference of each local ML model parameter, the better the result will be. If local ML models are highly homogeneous and non-complementary, it does not make sense to train global ML models, but rather to duplicate them, which can increase computational costs. Our expected framework combines the strengths of different local ML models in loV to compensate for each other's deficiencies and better address the data diversity challenge [6].

Therefore, to solve the above challenges, this paper mainly focuses on the isolated data island, incremental data, and data diversity challenges in IoV. We proposed a federated learning framework based on incremental weighting and diversity selection (FED-IW&DS) that aims to achieve efficient computation and higher performance in IoV.

Our Contribution

This paper mainly focuses on the isolated data island, incremental data, and data diversity challenges faced by IoV. We propose a Federated Learning Framework Based on Incremental Weighting and Diversity Selection (Fed-IW&DS).

1. We first introduced federated learning in the IoV scenario to deal with the isolated data island challenge. Second, we combine incremental learning and federated learning to

address the incremental data challenge. More specifically, an algorithm was proposed based on incremental data weights and incremental parameter depth values. In this algorithm, the penalty factor is used to improve the arctangent function, and a weighted aggregation strategy is formed, which optimizes the problems of large steady-state errors and weak anti-interference ability in IoV.
2. We then propose a dynamic selection algorithm combining cosine distance and diversity score to deal with the data diversity challenges. The algorithm integrates the local ML model parameters from different vehicles, avoids homogenization computing, maintains independent complementary data diversity, and optimizes the over-fitting problem of the global ML model prediction model in IoV.
3. We finally validate the Fed-IW&DS framework on multiple datasets, adopting accuracy, loss value, and Matthews correlation coefficient (MCC) to evaluate the global ML model. The experimental results show that the Fed-IW&DS framework achieves higher accuracy, lower loss, and better MCC metric within the same computational time.

The remaining sections are organized as follows. Section 1 is a literature review. Section 2 introduces the system model and problem description. Section 3 introduces the FED-IW&DS framework in detail and expounds its mathematical principles. Section 4 conducts experiments using two datasets and compares the performance with other federated learning frameworks and decay functions.

2. Literature Review

The integration of machine learning and IoV is increasing as data grows exponentially. Traditional machine learning requires that the data is centralized in a server or data center. Obviously, this leads to high communication costs and low computational efficiency in IoV and many techniques have been introduced into IoV application scenarios. Therefore, we will analyze the literature layer by layer from the perspectives of distributed machine learning, federated learning, and incremental learning.

2.1. Distributed Machine Learning in IoV

Distributed machine learning is an effective method to alleviate the problems of low communication efficiency and limited computing resources in IoV. Distributed machine learning either divides the training data and distributes them to multiple vehicle nodes, or it divides the learning model, and different vehicle nodes are responsible for different model training [7–9]. For example, Al-Sharman et al. used deep learning to realize the prediction of vehicle braking pressure, and the experimental results show that its accuracy is very impressive [10]. However, Zhang et al. proposed that it is unrealistic to only consider the vehicle brake pressure data, because, in the IoV's practical application, the transmission and encryption of a large amount of data will bring huge pressure on the storage and calculation of the vehicle system. Therefore, they proposed that the problem of communication resource allocation in IoV can be effectively solved by transforming the centralized framework into a distributed framework [11]. However, Zhao et al. pointed out that it is not enough to only consider how to allocate resources reasonably in distributed machine learning, because the lack of mechanisms to protect user data security will have many limitations in practical applications. Therefore, they proposed a distributed machine learning-oriented data integrity verification scheme (DML-DIV) based on distributed machine learning, which realizes data integrity through the idea of provable data possession (PDP) sampling audit algorithm. Unfortunately, they did not give a specific reference to the effect of using this encryption method on the accuracy of the model [12]. In addition, Magdum et al. conducted a detailed study of distributed machine learning algorithms. Experiments on LightGBM, CatBoost, AdaBoost, and XGBoost show that CatBoost performs the best among other algorithms, with an accuracy rate of only 81.31% [13]. However, they still require significant improvements to be applied in IoV.

To sum up, distributed machine learning is indeed a method for multi-party data sharing in IoV. However, on the one hand, the accuracy of distributed machine learning is still far from a practical application of IoV. On the other hand, as mentioned above, in distributed machine learning, the central server has absolute control, and the data content and training process of vehicle nodes are deployed uniformly by it, which brings the risk of privacy leakage.

2.2. Federated Learning in IoV

Federated learning is a special kind of distributed machine learning. (1) Compared with traditional distributed machine learning, the central server in federated learning cannot directly or indirectly read the data on the client. The client can not only stop computing and quit learning at any time, but also have absolute control over the data. (2) Most distributed machine learning assumes that the data on different clients is average and randomly scrambled, which makes it easier to train efficient global models. However, the reality of IoV is often not so optimistic. When vehicle data is generated independently, their amount is always different and irregular, so it is very rare for the data to be independent and identically distributed (i.i.d.). Federated learning is born with the ability to process non-i.i.d. data. (3) Distributed machine learning is more based on stable connections. On the contrary, federated learning is usually applied to poor network connections. Vehicles are often disconnected from the central server, so there is an obvious requirement for a method that disconnecting has little impact on global training. Therefore, federated learning is more robust than traditional distributed machine learning.

These advantages of federated learning make it a research hotspot. Many scholars have made many achievements in many fields, such as 5G [14], Internet of Things [15] , UAV (Unmanned Aerial Vehicle) [16], Navigation [17], etc. The clients in these studies are independent, and the non-i.i.d. data they collect are different, making the traditional distributed machine applied poorly.

The emergence of federated learning provides a strong technical foundation for data sharing in the IoV. In federated learning, each vehicle node has data autonomy. The data is collected autonomously by the vehicle nodes, and the training process is not subject to mandatory management by the central server. In recent years, many scholars have proposed a variety of federated learning frameworks for IoV [18–20].

Wang et al. proposed an algorithm for vehicle selection and radio resource allocation based on data content, which maximizes the loss function decay of the global model. However, they are only tested on a single dataset, and the model robustness is still questionable [21]. Liang et al. proposed a semi-synchronous federated learning (Semi-SynFed) protocol and a dynamic aggregation scheme to aggregate model parameters in an asynchronous manner, making the federated learning process more efficient. However, using asynchronous aggregation makes it have to use more resources, and there are certain problems in process control [22]. Yu et al. proposed a Federated Learning Incentive (FLI), which achieves the purpose of dynamic balance by maximizing the collective joint data and minimizing the inequality among data owners within the same computing time [23]. However, Lyu et al. proposed that the central server of this scheme still has hidden dangers in terms of security. So they proposed a decentralized fairness and privacy-preserving deep learning (FPPDL) framework, designed a local reputation mutual evaluation mechanism to ensure fairness, and a three-layer onion-style encryption scheme to ensure privacy. They also claimed that under FPPDL, each participant receives a different version of the FL model whose performance is commensurate with their contribution [24]. However, for this FPPDL, Huang et al. proposed that the scenarios in which this framework can be applied are very limited, and the common progress of all parties cannot be achieved. Therefore, they proposed a 5G-V2X-oriented Asynchronous Federated Learning Privacy-Preserving Computation Model (AFLPC), which they claim can effectively guarantee the practicability and privacy of asynchronous federated learning in 5G-V2X scenarios [25].

Vehicles continuously collect incremental data while driving. Meanwhile, different road conditions and complexity will generate diverse incremental data. Unfortunately, most of the current federated learning frameworks are based on static datasets, and there is a lack of mechanisms for how to combine incremental data with current models.

2.3. Incremental Data and Data Diversity in IoV

If we only consider the current data, it is difficult for the trained model to adapt to incremental data. In turn, considering only incremental data makes the model prone to "catastrophic forgetting". In response to these problems, many scholars have proposed different incremental learning frameworks [26–28]. For example, Cui et al. proposed an adaptive feedback handover (SAFH) algorithm to solve the dynamic handover problem in IoV, using an incremental feedback mechanism to dynamically capture the splitting properties of decision trees [29].

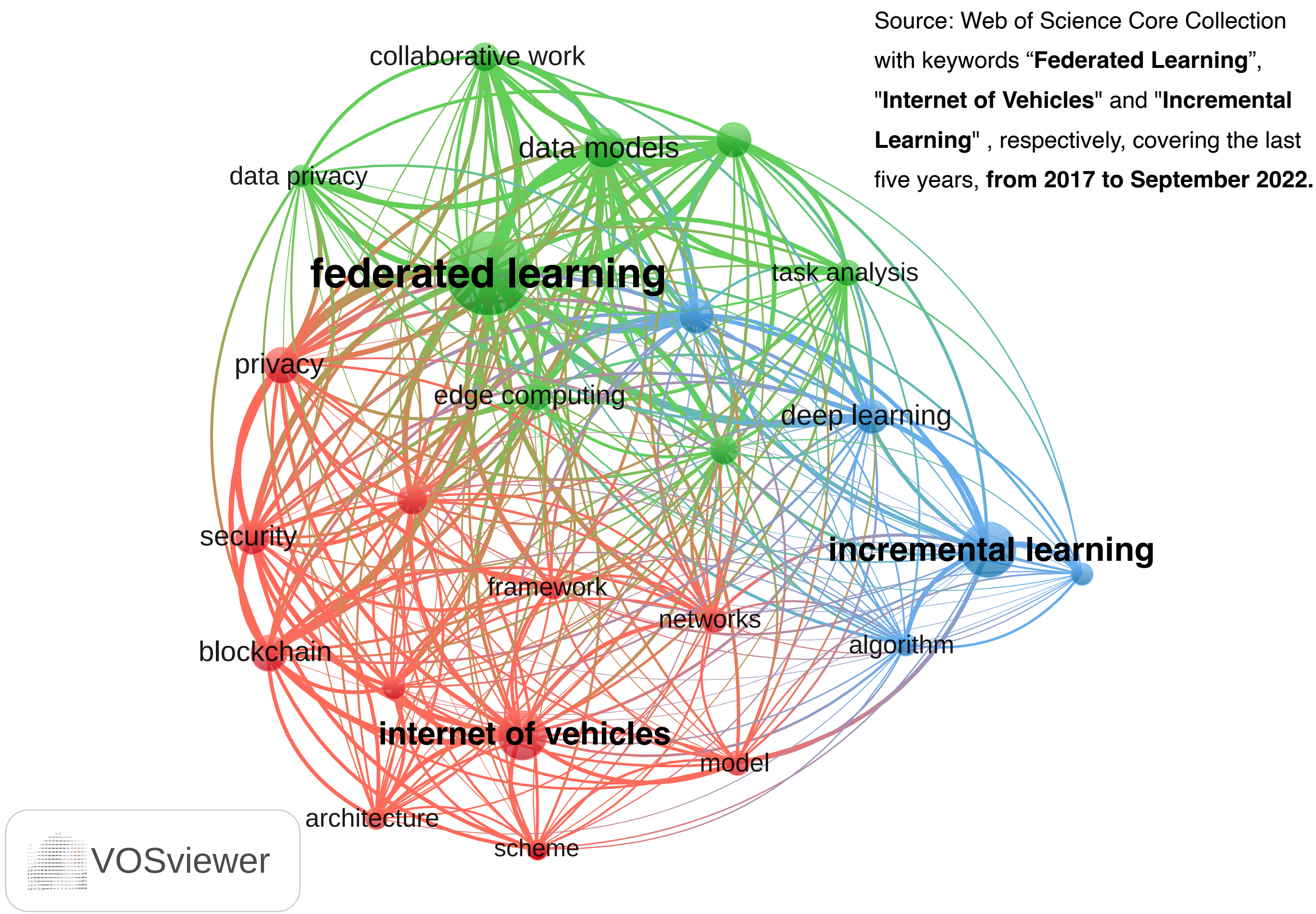
The algorithm proved to be less expensive than some existing ones. However, its data security cannot be guaranteed. In addition, Wang et al. proposed that the concept of a generalized learning system (BLS) be introduced into IoV, and the incremental learning algorithm is used to update and improve according to the newly generated data. This method has strong scalability [30]. Like Cui et al. [29], they didn't consider data security. Therefore, Zhu et al. proposed a novel attention-based federated incremental learning algorithm (Fed-SOINN). Their algorithm introduces an attention mechanism that increases the weight of ML model parameters uploaded by customers. Compared with baseline methods, Fed-SOINN improves detection accuracy by 3.1% and can reduce communication rounds by up to 73%. When faced with new traffic classes, the incremental learning mechanism in Fed-SOINN can also effectively identify unknown traffic classes [31]. However, they lack a discussion of computational time cost.

In summary, through the literature analysis above, there is much research on IoV at present. However, as shown in Table 1, how to solve the three major challenges of isolated data island, incremental data, and data diversity challenges caused by massive data in IoV at the same time are still missing. A more convincing framework is yet to be proposed.

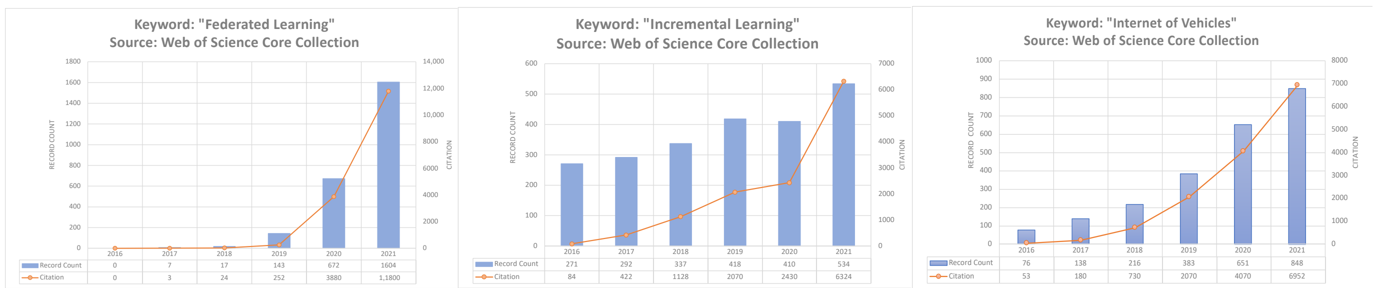
Table 1. Comparison of different frameworks.

Frameworks	Isolated Data Island Challenge	Incremental Data & Data Diversity Challenges	Computational Time Cost
Al-Sharman et al. [10]	×	×	×
Zhang et al. [11]	×	×	✓
Wang et al. [21]	×	×	✓
Liang et al. [22]	✓	×	✓
Hang et al. [25]	✓	×	×
Cui et al. [29]	×	✓	✓

In addition, the visual analysis of literature with the VOSviewer tool also confirms our view. As shown in Figure 1a, in the Web of Science Core Collection, we use "federated learning", "internet of vehicles" and "incremental learning" to search for the last five years, that is, from January 2017 to September 2022. Then we screened out the keywords that appeared more than 100 times and used the VOSviewer tool to generate the relationship diagram and number analysis. These 3 keywords are the hot research directions at present because their publication (record count in Figure 1b–d) and citation have been increasing in recent years. However, we can clearly see that these 3 keywords have become three relatively independent fields. Too little work has been devoted to their combinations, so this is a largely underexplored domain.



(a)



(b) (c) (d)

Figure 1. The citation relationship between "Federated Learning", "Incremental Learning" and "Internet of Vehicles", as well as the number of publications and citations in the last five years.

3. System Model and Problem Description

3.1. System Model

As shown in Figure 2, in the proposed Fed-IW&DS framework, we only upload local ML model parameters, and all data are kept in local vehicle nodes. There are three entities, which are the central server, the AP (Access Point), and the vehicle node. Communication between the three entities only transfers local ML model parameters, and valuable local data is only stored at the vehicle node. The functions of the three entities are as follows.

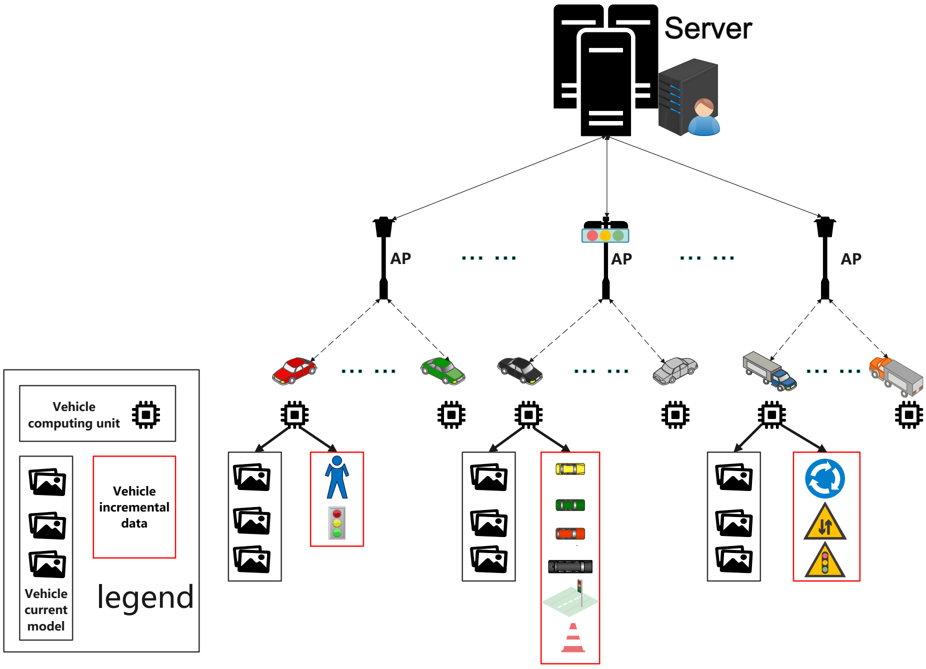


Figure 2. System Model.

1. Central server. It is the only core processing node. It does not receive local data collected by the vehicle, only local ML model parameters. In the Fed-IW&DS framework, it is responsible for the highest difficulty calculations, such as Incremental Weighted Calculation and Incremental Parameter Depth-Value Calculation, etc. We assume that the central server has unlimited computing power and storage space, in other words, it is capable of any computation.
2. AP (Access Point). There are multiple APs, all connected to the server. They can communicate with each other at any time. Likewise, it does not receive local data collected by the vehicle, only local ML model parameters. In the Fed-IW&DS framework, it is responsible for moderately difficult calculations such as Vehicle Diversity Selection and Diversity Score Calculation. We assume it has enough computing power and storage space to support these moderate calculations.
3. Vehicle node. There are many vehicles, and one vehicle will only be connected to one AP at a time. The vehicle node is responsible for collecting a large amount of incremental data (most of which are images from the camera) during the driving process, and only saves them in the local storage to form a local data set D_{local}. Then D_{local} is trained to form local ML model parameters W_{local}. While D_local will be saved locally, the vehicle node will only upload W_local. We define the collection of vehicles as V, and a vehicle as V_p, $p = \{1, 2, 3...n\}$. The sample data of vehicle V_p is represented as $D_p = \{X_1^{(p)}, X_2^{(p)}, \ldots, X_i^{(p)}\}$, the label set of these sample data is expressed as $C_p = \{Y_1^{(p)}, Y_2^{(p)}, \ldots, Y_i^{(p)}\}$. The amount of sample data of the vehicle V_p is S, and the sum of the number of training samples is $S_{\text{sum}} = \sum_{i=1}^{n}$ S. The sample size of vehicle V_p is S, and the total sample size of all vehicles is $S_{\text{sum}} = \sum_{i=1}^{n}$ S.During the first training process, the server will obtain a local ML model parameter in the existing local training set $D_{local} = \{(X_i, Y_i)\}_{n=1}^{|D_{\text{local}}|}$. At this stage, the existing local training sets (assuming they are class a) contain many training samples. When the global model is updated at round $t + 1$, the vehicle nodes will gradually collect a set of incremental data (assuming they are class b). Incremental learning is required to continuously update the global model. Incremental learning keeps learning new classes without

forgetting the learned old classes. That is, the global model can correctly identify images of class $a + b$.

The symbols and their meanings that we use subsequently are shown in Table 2.

Table 2. Symbols and Explanation.

Symbols	Explanation	Symbols	Explanation
D_{local}	vehicle local dataset	W_p	local ML model parameters for the p-th vehicle
W_{local}	local ML model parameters	W^i_{t+1}	local model parameters of the i-th vehicle in the (t+1)th round
p	total number of vehicles	$f_\theta(\cdot)$	extractor
p_i	i-th vehicle node	θ	extractor parameters
V_p	The p-th vehicle participating in the calculation	$C_\omega(\cdot)$	classifier
D_p	local dataset of the p-th vehicle	ω	classifier parameters
X^p_i	the i-th data in the p-th vehicle	L	number of neural network layers
C_p	local raw label dataset	$W^{(l)} \in R^K$	bias from layer l-1 to layer l
S	vehicle sample data volume	$Q^{(l)} \in R^K$	the input of the first layer neuron
N	number of sample categories	$F^{(l-1)} \in R^{M\times N\times D}$	l-1 layer parameter mapping result
X_i	sample data	$o^{(l)} \in R$	learning bias
Y_i	sample label	A	number of convolution kernels
D^{InDt}	incremental dataset of vehicle nodes at round t + 1	$\hat{C}_i$	image predicted category
W	all vehicle parameters	C_i	image real category

3.2. Problem Description

We assume that there are p vehicles participating in the global model, and each vehicle has its own local dataset at the beginning, denoted as a $\left\{\mathrm{X}_i^{(1)}\right\} \in \mathrm{D}_1, \left\{\mathrm{X}_i^{(2)}\right\} \in \mathrm{D}_2, \ldots, \left\{\mathrm{X}_i^{(\mathrm{p})}\right\} \in \mathrm{D}_\mathrm{p}, \mathrm{i} = \{1, 2, 3 \ldots\}$ To protect user privacy in federated learning, vehicle nodes do not directly send local data to the central server. Therefore, the central server can only update the global model by collecting the local ML parameters corresponding to each vehicle, which are $W_1, W_2, \ldots, W_p$. The set of all local ML model parameters is $W = W_1 \cup W_2 \cup \ldots \cup W_p$ Then in the $(t+1)$ round of global model, each vehicle has p_i training data and local ML model parameters W^i_{t+1}. The global model's weight update formula in the traditional FedAVG framework is as follows.

$$w_{t+1} \leftarrow \sum_{i=1}^{\mathrm{I}} \frac{p_i}{p} w^i_{t+1} \tag{1}$$

where w_{t+1} represents the global model parameter update, p_i represents a certain vehicle node, p represents the total number of vehicles, and w represents the local ML model parameters participating in the global model. In addition, there is obviously $\sum_{i=1}^{\mathrm{I}} n_k = n$.

However, the model weights are averaged in the traditional FedAVG framework. The FedAVG framework does not perform well with diverse incremental data (as our experiments will demonstrate). Diverse incremental data comes from the fact that vehicles will generate different incremental data in different road conditions. To efficiently fuse incremental data with existing models, we define a base model for incremental learning.

$$M = C_\omega(f_\theta(x)) \tag{2}$$

It consists of an extractor $f_\theta(\cdot)$ and a classifier $C_\omega(\cdot)$, where θ and ω represent the parameters of the extractor and classifier, respectively. The image to be classified is denoted as x, the extracted ML model parameters from x are denoted as $f_\theta(x)$, and the classification result of the ML model parameters $f_\theta(x)$ is denoted as $C_\omega(f_\theta(x))$.

In the federated learning + incremental learning scenario, the first training dataset of all vehicles involved in the computation is $\{\mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_\mathrm{p}\}$. Then, each vehicle will generate an incremental dataset after the first round of global training. After the $(t+1)$ round global training, the incremental data is represented as D^{InDt}. Therefore, when the global model updates the parameters in the t + 1 round, the dataset in the vehicle becomes $\{\mathrm{D}_1, D_1^{\mathrm{InD}\,1}, D_1^{\mathrm{InD}\,2}, \ldots, D_1^{\mathrm{InD}\,t}\}$. The vehicle first completes the training of all $\{\mathrm{D}_1, D_1^{\mathrm{InD}\,1}, D_1^{\mathrm{InD}\,2}, \ldots, D_1^{\mathrm{InD}\,t}\}$, and then sends the local ML model parameters to the AP.

After being processed by the AP (such as Vehicle Diversity Selection and Diversity Score Calculation), the parameters collection $W_{t+1} = W_1 \cup W_2 \cup \ldots \cup W_p$ are sent to the central server to calculate the new global parameter value W_s for the next t+2 round.

To better complete the above work, we decided to use an industry-proven & state-of-the-art model—Convolutional Neural Network (CNN). The convolutional structure of CNN can reduce the amount of memory occupied by the deep network, effectively alleviate the problem of model overfitting, and better extract image ML model parameters. CNN generally consists of convolutional layers, pooling layers, and fully connected layers.

In CNN, convolution is used instead of fully connected layers, and the net input $Q^{((l))}$ of the l layer is the activity value $a^{((l-1))}$ of the $L-1$ layer and the convolution kernel $\boldsymbol{W}^{(l)} \in R^K$ convolution, i.e.,

$$\boldsymbol{Q}^{(l)} = \boldsymbol{W}^{(l)} \otimes \boldsymbol{a}^{(l-1)} + o^{(l)} \tag{3}$$

where $\boldsymbol{W}^{(l)} \in R^K$ is the learning weight vector, and $o^{(l)} \in R$ is the learning bias. In the $(t+1)$ round of our federated learning + incremental learning framework, the $(l-1)$ layer of the CNN is a convolutional layer, and its input parameters are mapped as $F^{(l-1)} \in R^{M \times N \times D}$. The parameters of layer l calculated using convolution are mapped to $Q^{(l)} \in R^{M' \times N' \times I}$. Therefore, the vehicle calculation process is as follows.

$$Q^{(l,i)} = \sum_{d=1}^{N} W^{(l,i,d)} \otimes \left[D^{(l-1,d)} \cup D^{\mathrm{InD}(l-1,d)} \cup \ldots \cup D^{\mathrm{InDt}(l-1,d)}\right] + o^{(l,i)} \tag{4}$$

Among them, $W^{(l,i,d)}$ and $o^{(l,i)}$ are convolution kernels and learning bias. There are SNconvolution kernels and I bias in the l layer, and the gradient can be calculated by using the chain rule.

Furthermore, to avoid overfitting, the CNN model adds a pooling layer after the convolutional layer. In this paper, we use two common pooling functions, Max Pooling and Mean Pooling.

Maximum Pooling or Max Pooling:

$$y_{m,n}^{d} = \max_{i \in R_{m,n}^{d}} x_i \tag{5}$$

Mean Pooling:

$$y_{m,n}^{d} = \frac{1}{\left|R_{m,n}^{d}\right|} \sum_{i \in R_{m,n}^{d}} X_i \tag{6}$$

Each output parameter in CNN requires a convolution kernel and a bias, and we derive the prediction result in $\hat{C}_{n,i}$ from Formula (7).

$$\mathrm{\hat{C}_i} = \mathrm{Q_{n,i} A_n^T} + \mathrm{o_{n,i}} \tag{7}$$

Therefore, it can be deduced that the prediction accuracy of the federated learning + incremental learning framework is as follows.

$$\text{Accuracy } = \frac{1}{N} \sum_{i=1}^{N} \left(\mathrm{Q_{n,i} A_n^T} + \mathrm{o_{n,i}}\right) / C_i \tag{8}$$

Among them, $\hat{C}_i$ is the predicted category, and C_i is the real category of the image.

On these, we can calculate the local ML model loss of the federated learning + incremental learning framework, and the calculation formula is as follows.

$$f_n(w) = \log_2\left(1 + \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \mathrm{Q_{n,i} A_n^T} + \mathrm{o_{n,i}} W_n^T \left[D^{(l-1,d)} \cup D^{InD1(l-1,d)} \cup \ldots \cup D^{InDt(l-1,d)}\right]\right)\right) \tag{9}$$

To evaluate the performance of our framework more comprehensively, we decided to add MCC to evaluate the model in addition to the Acc and Loss indicators.

In the IoV scenario, MCC is defined as follows.

$$C_{k,k} = |\{s \in S \mid X_{sk} = Y_{sk} = 1\}| = \sum_{s=1}^{S} X_{sk} \cdot Y_{sk} \tag{10}$$

where S represents the total number of samples of all vehicle data, N represents the number of categories, and X, Y represent two S*N-dimensional matrices.

Therefore, the MCC calculation formula is as follows.

$$MCC = \frac{\sum_{k,l,m}^{N} \ldots C_{kk} \cdot C_{ml} - c_{lk} \cdot C_{km}}{\sqrt{\sum_{k=1}^{N}\left[\left(\sum_{k=1}^{N} C_{lk}\right)\left(\sum_{f,g=1\&f\neq g}^{N} C_{gf}\right)\right]}\sqrt{\sum_{k=1}^{N}\left[\left(\sum_{k=1}^{N} C_{lk}\right)\left(\sum_{f,g=1\&f\neq g}^{N} C_{gf}\right)\right]}} \tag{11}$$

3.3. Problem Summary

Through the above introduction of three indicators—Acc, Loss, and MCC, we can get the training target of our Fed-IW&DS framework, which is formulated as follows.

$$\max \ \text{Accuracy} \ = \max\left[\frac{1}{N}\sum_{i=1}^{N} \mathrm{Q_{n,i}A_n^T} + \mathrm{o_{n,i}}\right] \tag{12}$$

$$\min f(W) \triangleq \min\left[\frac{1}{N}\sum_{n=1}^{N} \ldots (W)\right] \tag{13}$$

$$MCC = \max \frac{\sum_{k,l,m}^{N} \ldots C_{kk} \cdot C_{ml} - c_{lk} \cdot C_{km}}{\sqrt{\sum_{k=1}^{N}\left[\left(\sum_{k=1}^{N} C_{lk}\right)\left(\sum_{f,g=1\&f\neq g}^{N} C_{gf}\right)\right]}\sqrt{\sum_{k=1}^{N}\left[\left(\sum_{k=1}^{N} C_{lk}\right)\left(\sum_{f,g=1\&f\neq g}^{N} C_{gf}\right)\right]}} \tag{14}$$

The isolated data island, incremental data, and data diversity challenges in the IoV are specifically quantified into the Formulas (12)–(14). With constantly diversifying data and updating incremental datasets, the framework we pursue should be able to achieve higher accuracy, lower loss, and better MCC values, to achieve a better solution between the local ML model and the global ML model, to significantly optimize the efficiency of federated learning + incremental learning in IoV.

4. The Proposed Fed-IW&DS Framework

4.1. Overall Introduction to the Framework

Our Fed-IW&DS framework is divided into 3 modules (shown in Figure 3), namely Vehicle Node-vehicle data collection (as the name suggests, the task of vehicle node is: collecting incremental data, training the local ML model & uploading it, waiting for the next round of global ML model sent by the server.), AP Node-vehicle diversity selection algorithm and Central Server-vehicle federated incremental learning algorithm.

- Vehicle diversity selection algorithm consists of diversity score calculation and diversity score sorting. Diversity score calculation first uses ACC, Loss, and MCC three vehicle performance indicators, and then calculates the difference of these three indicators through an improved cosine distance. Diversity score sorting is responsible for statistical calculation results and filtering vehicle nodes with high diversity scores.
- The vehicle federated incremental learning algorithm is executed by the central server. It is well known that vehicles in IoV drive freely (as long as they obey the traffic rules). Therefore, the incremental data are different from each other during driving. However, in the traditional federated learning framework, the central server collects the local ML model parameters of all vehicle nodes and then updates the global model with the same weights. This is obviously unreasonable. In response to the above problems,

our proposed vehicle federated incremental learning algorithm is divided into three sub-modules: incremental weighted calculation, incremental parameter depth-value calculation, and incremental parameter update.

- We introduce these three sub-modules: incremental weighted calculation, incremental parameter depth-value calculation, and incremental parameter update. (1) Incremental weighted calculation. The server constantly monitors the incremental learning status of vehicle nodes. The server explores the impact of incremental data on the current local ML model according to the ratio of the incremental sample number to the current total number of samples. Its purpose is to make the local ML model decay relatively stable and gentle. (2) Incremental parameter depth-value calculation. It refers to the incremental parameter correction of the global ML model based on the previous incremental weighted calculation results. More specifically, the impact of incremental data on the current local ML model will be reflected by a specific value—the incremental parameter depth value. The incremental parameter depth value enables reasonable, rapid, and stable incremental parameter correction for the global ML model. (3) Finally, only the global ML model updated with incremental parameters could be sent to every vehicle node for the next round of calculation. The above process will continue to repeat until the vehicle node leaves the AP coverage or parks or disconnects.

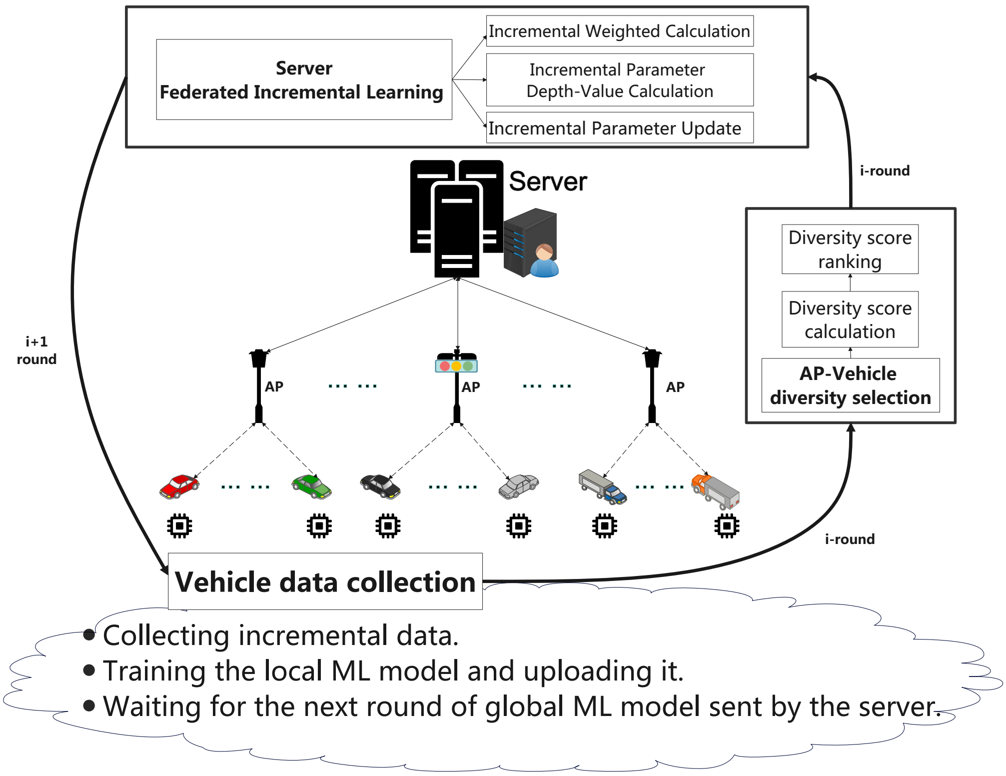


Figure 3. The proposed Fed-IW&DS Framework.

AP Node-vehicle diversity selection algorithm and Central Server-vehicle federated incremental learning algorithm are not independent of each other. The AP Node-vehicle diversity selection algorithm needs to obtain the aggregated global ML model parameters from the Central Server and collect the local ML model parameters to calculate the diversity score. The vehicle nodes selected by vehicle diversity affect the learning direction of the Central Server's global ML model, which in turn affects the results of the vehicle federated incremental learning algorithm.

4.2. AP Node-Vehicle Diversity Selection Algorithm

The vehicle diversity selection algorithm means that the AP node calculates the diversity score of the vehicle nodes participating in this communication round before uploading the local ML model parameters to the server, then sorts them according to the diversity score, and filters the vehicles that are eligible to participate in the server's global ML model. In traditional federated learning, most of the discriminant conditions for participating in the global ML model are based on the accuracy from accurate to inaccurate or setting the proportion of participating in the global ML model or setting a fixed threshold. These methods have the following shortcomings in IoV.

1. The vehicle's existing data and incremental data are unbalanced, resulting in different requirements for parameter change in the process of ML learning. Simply grading the accuracy of the local ML model from good to bad, or selecting participating vehicles proportionally, it is easy to ignore the useful ML parameters. This will lead to the deviation of the global ML model in the central server's training process, which cannot cope with the diversity of incremental data, thus affecting the accuracy of the global ML model.
2. In traditional federated learning, the selection of nodes is only based on a fixed threshold, which is artificially set and the value of this threshold is too subjective. However, the situation in IoV changes rapidly, and unreasonable fixed thresholds will make the central server to be time-consuming. This will cause the global ML model to fluctuate greatly and not easily converge, and it is also unable to cope with the diversity of incremental data.

In Fed-IW&DS, the central server considers the accuracy and diversity scores equally when selecting vehicle nodes to participate in the global ML model. We first calculate the diversity score of vehicle nodes and sort them, and then select the vehicle nodes participating in the global ML model training according to the set proportional coefficient Diversity-Ratio (D-R). Only the selected vehicle nodes can obtain the qualification to participate in the global ML model, otherwise, the ML parameter information is accumulated locally, and the final parameters will accumulate sufficient information to be uploaded to the central server. No matter whether the vehicle node is qualified for the global ML model of this round or not, the AP node must calculate the diversity score in each round, so the diversity score calculation will run through the whole learning process.

4.2.1. Diversity Score Calculation–Initial Process

In the Fed-IW&DS framework, the core method of diversity score calculation is cosine distance. Also known as cosine similarity, the cosine of the angle between two vectors is used as a measure of size between two individuals. Therefore, we apply it to the performance indicators of vehicle nodes in IoV, which can reflect the diversity of vehicle nodes in each round. However, the data in IoV are rich in diversity, and the traditional calculation of cosine distance in a single dimension will lead to inaccurate data similarity calculation results. To obtain a more comprehensive diversity score, we use more indicators, namely the Acc, Loss, and MCC of the vehicle nodes. From these three indicators, the sum of the cosine distances between a certain vehicle node and all other vehicle nodes is calculated, which is taken as the vehicle node's Diversity Score (DS_i).

We form these three indicators into a matrix $[u_{acc}, u_{\text{loss}}, u_{MCC}]$, that is, obtain the overall performance matrix $V_{i,j}$, as follows.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ \cdots & \cdots & \cdots \end{bmatrix} = V_{ij}[u_{acc}, u_{\text{loss}}, u_{MCC}] \tag{15}$$

The subscript n is the number of vehicle nodes. From n vehicle clients, we take any two vehicle clients $p_i[a_{i1}, a_{i2}, a_{i3}]$ and $p_j\left[a_{j1}, a_{j2}, a_{j3}\right](0 < i \leq n, 0 < j \leq n, i \neq j)$, and their cosine distances are as follows.

$$\cos\left(p_i, p_j\right) = \frac{\sum_{k=1}^{n} p_{i,k} p_{j,k}}{\sqrt{\sum_{k=1}^{n} p_{i,k}^2} \sqrt{\sum_{k=1}^{n} p_{j,k}^2}} \tag{16}$$

Simplify it to get the following formula, which represents the cosine distance between any two vehicle nodes.

$$\cos\left(p_i, p_j\right) = \frac{\sum_{k=1}^{n} p_{i,k} p_{j,k}}{\|p_i\| \cdot \|p_j\|} \tag{17}$$

4.2.2. Diversity Score Calculation–Improvement Based on Penalty Factor

- Motivation of Penalty Factor

The Fed-IW&DS framework faces the following problems. In IoV, with the increase of data diversity, the training dataset's error of the global ML model will gradually decrease, while the test dataset's error will generally decrease first and then increase. This is the overfitting problem of the global ML model.

Therefore, for the above-mentioned time difference between diversity and error rate, we need to select an appropriate complexity for the local ML model by introducing a penalty factor. Penalty factors can balance empirical risk and structural risk in ML models [32]. After many experiments, we found that when more vehicles are qualified to participate in the global ML model, it means that the complexity of the model is lower, the structural risk is lower, and the required penalty factor is smaller. On the contrary, the fewer vehicles participating in the global ML model, the greater the structural risk, the smaller the empirical risk, the easier it is to over-fit, and the greater the penalty factor required.

- Improved Diversity Score Based on Penalty Factor

We introduce a penalty factor to give proper complexity to the local ML model. All vehicles are expressed as V-Sum, the total number of vehicles eligible for global ML model training is expressed as V-Glb, and the penalty factor is Pen-Fac. The Pen-Fac formula is as follows.

$$\mathrm{Pen} - \mathrm{Fac} = 1 - \frac{\mathrm{V} - \mathrm{Glb}}{\mathrm{V} - \mathrm{Sum}} \tag{18}$$

After improvement on Pen-Fac, the similarity between vehicles X and Y is expressed as $\mathrm{sim}_{\cos}(X, Y)$.

$$\mathrm{sim}_{\cos}(X, Y) = \frac{\sum_{t \in R_{XY}} \mathrm{Pen} - \mathrm{Fac} \cdot \left(R_{X,t} - \bar{R}_X\right)\left(R_{Y,t} - \bar{R}_Y\right)}{\sqrt{\sum_{t \in R_X} \left(R_{X,t} - \bar{R}_X\right)^2} \sqrt{\sum_{t \in R_Y} \left(R_{Y,t} - \bar{R}_Y\right)^2}} \tag{19}$$

Hence, we get the vehicle client diversity score DS_i.

$$DS_i = \sum_{j=0}^{n} \mathrm{Sim}_{\cos}(X, Y); (i \neq j) \tag{20}$$

where n represents the number of all vehicle nodes. Obviously, the larger the diversity score DS_i, the smaller the similarity of performance indicators, and vice-versa. Finally, we select the vehicle nodes participating in the global ML model according to the factor D-R. Vehicle nodes that are not selected accumulate ML parameters locally and proceed to the next round's local ML model learning. Regardless of whether the vehicle node is eligible for the global ML model or not, the diversity score calculation is carried out throughout the whole learning process.

4.3. Central Server-Vehicle Federated Incremental Learning Algorithm

After the Fed-IW&DS framework has found a way to deal with data diversity, the next step is to incorporate it into a federated learning global ML model. However, what we must deal with next is the incremental data challenge. We are dealing with highly free vehicle nodes, and each vehicle node will have exponential incremental data. Most of the current federated learning frameworks are based on static datasets and lack the mechanism to combine incremental data with existing models. How to effectively fuse incremental data with existing models quickly and at the same time ensure that each vehicle effectively participates in global ML model training becomes the next research goal of the Fed-IW&DS framework.

4.3.1. Incremental Weighted Calculation

Vehicle incremental data has the characteristics of uneven learning samples and dynamic data increase. In the same training round of the global ML model, the sample size of incremental data is different. As an example, vehicle A doubles the existing data: if the amount of existing data is 200, the incremental data amount is 400. Another example is that vehicle B has only increased by 0.5 times the existing data: if the existing data amount is 200, the incremental data amount is 100.

The process of model training can be understood as the process of model "learning". As time goes by, the model approaches the optimal solution to the problem. However, in IoV, more incremental data will increase the distance between the global ML model and the optimal solution to the learning problem. Therefore, it is unreasonable to update the local ML models with equal weight. Our Fed-IW&DS framework introduces the strategy of incremental weighted aggregation, which solves the problem of unbalanced incremental data in federated incremental learning.

The FedAVG algorithm is given above, and its aggregation strategy is shown in formula x, which only considers the influence of local data's sample size on the global ML model, that is, a larger n will have a greater impact on the global ML model. However, it does not perform special treatment for incremental data. By calculating the incremental parameter depth-value of vehicle nodes, the incremental weight is introduced to influence the global ML model of the Fed-IW&DS framework.

The incremental weight indicates the proportion of the incremental samples in the total number of existing samples. In the (t + 1) round, the incremental weight ϕ_i of the vehicle node i can be obtained by the number of incremental samples and the total number of samples. The formula is as follows.

$$\phi_i = \frac{\left|\mathrm{D}_{\mathrm{i}}^{\mathrm{InDt}}\right|}{\left|\mathrm{D}_i\right| + \left|\mathrm{D}_{\mathrm{i}}^{\mathrm{InDt}}\right|} \tag{21}$$

$\left|\mathrm{D}_{\mathrm{i}}^{\mathrm{InDt}}\right|$ is the number of incremental data, and $|D_i|$ is the number of existing data.

4.3.2. Motivation of Adjustment Variable α for Parameter Depth-Value

The above calculated incremental weights fully reflect the proportion of the incremental data sample. However, the Fed-IW&DS framework is based on federated learning, which only uploads local ML model parameters. Therefore, we need to transform the proportion of the incremental data into an appropriate update of the incremental parameter's depth value. Based on these thoughts, we have formed the following motivations: the relationship between incremental data and parameter depth-value, massive incremental data, small incremental data, and incremental data trends.

- Relationship between incremental data & parameter depth-value. Obviously, the amount of incremental data is proportional to the depth value of the incremental parameters. That is to say, the more incremental data of a vehicle, the more we need to consider updating its parameters. Therefore, we first need to consider a "proportional decay function".

- Massive incremental data. In the early stage of global ML model training, the amount of existing data is small, and the proportion of incremental data is large. If blindly selecting vehicles with massive incremental data will lead to excessive consumption of resources and increase the calculation cost of the global ML model. Therefore, we next need to consider a proportional decay function with "reducing the large signal".
- Small incremental data. In addition, if we ignore vehicles with small incremental data, it will easily lead to the extreme competition phenomenon of "the strong are stronger, the weak are weaker". Therefore, we also need to consider a proportional decay function that can "amplifying the small signal + reducing the large signal".
- Incremental data trends. In the middle and late stages of global ML model training, the amount of existing data will increase, and the proportion of incremental data will decrease. Therefore, the more common situation is a small amount of incremental data. In other words, the key to our incremental parameter depth-value calculation is "amplifying the small signal".

We have considered many decay functions. However, neither theoretical calculations nor experimental attempts have achieved satisfactory results. (1) First, we try linear function. It cannot cope with the massive incremental data in the early stage of global ML model training, nor can it cope with the incremental data in the middle and late stages. (2) Second, we try the square curve function. When the variable takes very large positive or negative values, it will be supersaturated, which means that the function becomes steep and sensitive to small changes in input. (3) Then we try the arc tangent function, which performs well in massive incremental data. It can reasonably deal with vehicles with massive incremental data, calculate their corresponding parameter depth-value in a balanced way, and efficiently realize "reducing the large signal". However, when faced with a small amount of incremental data, it does not perform well in terms of "amplifying the small signal".

Therefore, in the next subsection we propose an adjustment variable α, which can significantly compensate for the inadequacy of the arctangent function in " amplifying the small signal ".

4.3.3. Incremental Parameter Depth-Value Calculation

It is necessary to convert the sample size ratio of incremental data into an update of the depth-value corresponding to the incremental parameters. The depth value of the incremental parameter indicates the influence of the local incremental data on the global ML model when the vehicle node finishes learning the local ML model and reflects the updated size of the local ML model parameters. In the process of parameter optimization, there are certain depth values, as follows.

$$\varepsilon_{t+1}^{\mathrm{i}} = \varepsilon_t^{\mathrm{i}} - \left(\phi_k * \varepsilon_t^{\mathrm{i}}\right) \tag{22}$$

Among them,$\varepsilon_t^{\mathrm{i}}$ is the incremental parameter depth-value, and ϕ_k is the incremental weight.

The greater the proportion of incremental data sample size, the greater the depth-value of the incremental parameters. We need to achieve the goal of " reducing the large signal ", so that the parameter depth-value of the vehicle node with huge incremental data is not so that huge, and the decay process is relatively gentle. As shown in Equation (23) below, we choose the arc tangent as the decay function for incremental weighting.

$$\mu_{t+1}^{\mathrm{i}} = \frac{2}{\pi} \arctan \varepsilon_{t+1}^{\mathrm{i}} \tag{23}$$

However, the traditional arctangent function has a large steady-state error and weak anti-interference ability [33]. Most importantly, it ignores the amplification of small signals. To solve this problem, we consider introducing an adjusting variable α. Reasonable selection of α value can make the algorithm not only ensure high accuracy, but also obtain faster convergence speed in practical application. We propose a calculation method for the adjustment variable α, to expect better calculation results for the global ML model.

We set O_{ij} to represent the ratio of incremental data D_i^{InDt} to existing data $|D_i|$, and ε_t^i to be the parameter depth-value before optimization. Our adjustment variable α is defined as follows.

$$\alpha = 1 + \varepsilon_t^{\mathrm{i}} \left(1 - \phi_{ij} / \sum_{i=1}^{c} O_{ij}\right)^2, 1 < \alpha < 2 \tag{24}$$

To avoid the extreme competition phenomenon of "the strong are stronger, the weak are weaker", we apply the adjustment variable α to the arctangent function interval where the incremental data is smaller than the existing data. As shown in Figure 4, the x-coordinate is the ratio of the incremental data to the existing data. The further to the right, the larger the incremental data ratio of the vehicle node; the y-coordinate refers to the calculation of the decay function—increment parameter depth-value. There are 3 vehicles from left to right in Figure 4, and the red color on the far left represents vehicles with incremental data smaller than the existing data. Originally its incremental parameter depth value should be the arc tangent curve of the black line. By the adjustment variable α to make its incremental parameter depth-value turn into a red curve, we achieve the effect of "amplifying the small signal".

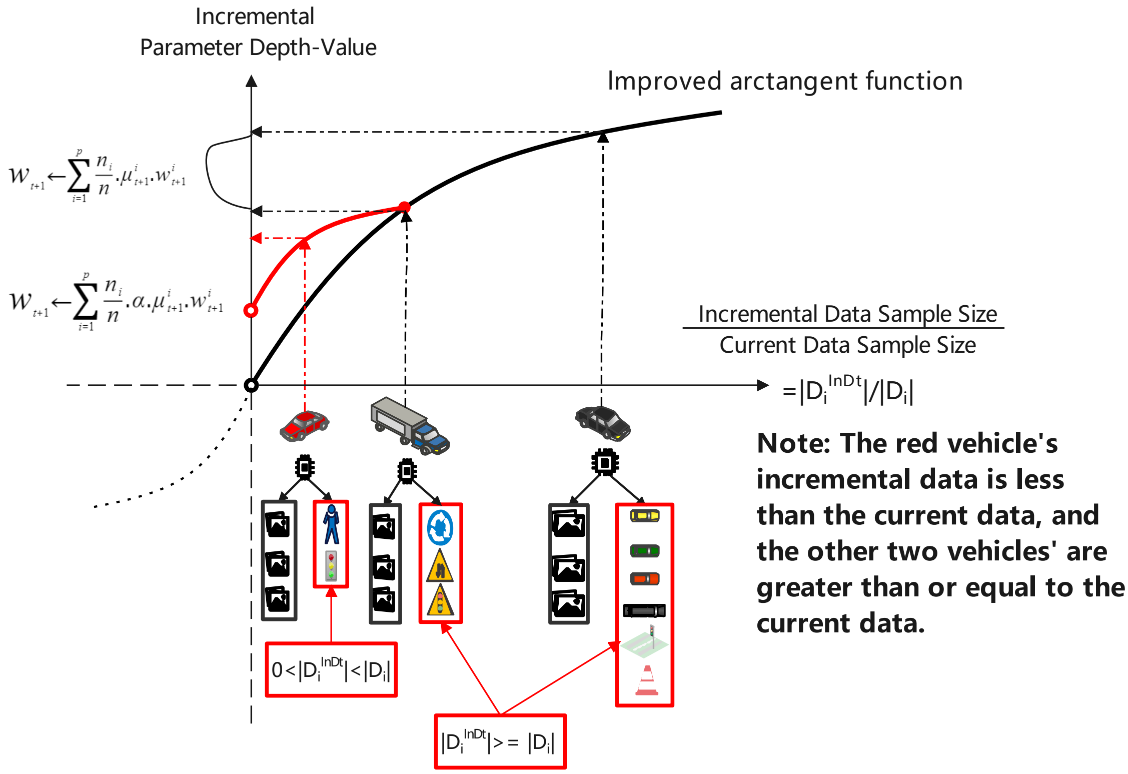


Figure 4. Incremental parameters based on improved arctangent functions Depth-value calculation.

In the Fed-IW&DS framework, only the vehicle nodes participating in the global ML model are updated in each round, and the contribution of the local ML models is determined according to the incremental parameter depth-value, which can effectively utilize historical information, distinguish the utilization value of local ML models, and significantly improve global ML model performance. Therefore, our proposed weighting strategy is as follows.

$$\begin{cases} \text{if} \quad 0 < \left|\mathrm{D_i^{InDt}}\right| < |\mathrm{D}_i| & \boldsymbol{w}_{t+1} \leftarrow \sum_{i=1}^{P} \cdot \frac{n_i}{n} \cdot \alpha \cdot \mu_{t+1}^{i} \cdot \mathrm{w}_{t+}^{i} \\ \text{if} \quad \left|\mathrm{D_i^{InDt}}\right| \gg |\mathrm{D}_i| & \boldsymbol{w}_{t+1} \leftarrow \sum_{i=1}^{P} \frac{n_i}{n} \cdot \mu_{t+1}^{i} \cdot \mathrm{w}_{t+1}^{i} \\ \text{if} \quad \mid \mathrm{D_i^{InDtI}} = 0 & \text{This round: This vehicle isn't participating global ML model} \end{cases} \tag{25}$$

In the federated incremental learning process, the local ML model parameters submitted by the vehicle nodes must be modified by incremental weighting before they can participate in the global ML model. The local ML model parameters are updated on the central server according to the optimization algorithm described above. The vehicle node will regain the latest global ML model parameters for the next round of local ML model training.

4.4. Pseudocode of the Fed-IW&DS Framework

This section presents the pseudocode of the Fed-IW&DS framework, including the Central Server-Vehicle Federated Incremental Learning Algorithm (see Algorithm 1) and the AP Node-vehicle diversity selection algorithm (see Algorithm 2). Algorithm 1 Central Server mainly includes initialization parameters and global ML model training. The 2nd line represents the initialization of the server global ML model parameters; the 3rd to 5th lines represents the initialization of the depth value of the incremental parameter of the vehicle node, and each sub-loop in the whole training; the 8th to 10th lines represent the calling of the sub-function Vehicle Update, In addition, the local ML model parameters and incremental parameter depth values of vehicle nodes are updated to obtain.

Algorithm 1: Central Server-Vehicle Federated Incremental Learning Algorithm

1 FUNCTION SERVER UNION EXECUTES;
2 Initialize w_0;
3 **for** *vehicle* $i \in 1, 2, ..., P$ **do**
4 $\quad \epsilon^i \leftarrow 0$;
5 **end**
6 **for** *each round* $t = 1, 2, ...$ **do**
7 $\quad m \leftarrow max(D - R \cdot P, 1)$;
8 $\quad$ **for** *each vehicle* $i \in 1, 2, ..., PinParallel$ **do**
9 $\quad\quad w_{t+1}^{i}, \mu_{t+1}^{i} \leftarrow VehicleUpdate(w_t, t, \epsilon_t^i)$;
10 $\quad$ **end**
11 $\quad M_t \leftarrow order(DS_1, DS_2, ..., DS_1)$;
12 $\quad S_t \leftarrow (topmsetofM_t vehicles)$;
13 $\quad$ **for** *each vehicle* $i \in S_t$ **do**
14 $\quad\quad$ **if** $|D_t^{Indt}| \gg |D_i|$ **then**
15 $\quad\quad\quad \boldsymbol{w}_{t+1} \leftarrow \sum_{i=1}^{P} \frac{n_i}{n} \cdot \mu_{t+1}^{i} \cdot \mathrm{w}_{t+1}^{i}$;
16 $\quad\quad$ **else if** $0 < |D_t^{Indt}|$ **then**
17 $\quad\quad\quad \boldsymbol{w}_{t+1} \leftarrow \sum_{i=1}^{P} \cdots \frac{n_i}{n} \cdot \alpha \cdot \mu_{t+1}^{i} \cdot \mathrm{w}_{t+1}^{i}$;
18 $\quad\quad$ **else if** $|D_t^{Indt}| = 0$ **then**
19 $\quad\quad\quad$ This round: This vehicle isn't participating global ML model ;
20 $\quad\quad$ **end**
21 $\quad$ **end**
22 **end**
23 END FUNCTION;

Algorithm 2: AP Node-Vehicle Diversity Selection Algorithm

Input: parameters w, t, S_t
Output: $ACC, Loss, MCC, w, \mu_{t+1}$
1 FUNCTION Vehicles Update:;
2 **if** $|D_i^{Indt}|$ **then**
3 $\quad \phi = \frac{|D_i^{Indt}|}{|D_i|}$;
4 $\quad \epsilon_{t+1} \leftarrow \epsilon_t - (\phi_i * \epsilon_t)$;
5 **else**
6 $\quad \epsilon_{t+1} \leftarrow t + 1$;
7 **end**
8 $\mu \leftarrow \frac{2}{\pi} arctan \epsilon_{t+1}^{i}$;
9 $\alpha = \epsilon_t^i (1 - \frac{1-\phi_{ij}}{\sum_{j=1}^{C}})^2$;
10 $D_i \leftarrow (D_i^{Indt}\ \&\ D_i)$;
11 **for** *each local epoch k from 1 to E* **do**
12 $\quad$ batches $\leftarrow (data D_i split into batches of size B)$;
13 $\quad$ **for** *batch b in batches* **do**
14 $\quad\quad$ $\max \text{Accuracy} = \max\left[\frac{1}{N}\sum_{i=1}^{N} \mathrm{Z_{n,i}A_n^T} + \mathrm{b_{n,i}}\right]$;
15 $\quad\quad$ $\min f(w) \triangleq \min\left\{\frac{1}{N}\sum_{n=1}^{N} \ldots (w)\right\}$;
16 $\quad\quad$ $MCC = \frac{\sum_{k,l,m}^{N} \ldots C_{kk} \cdot C_{ml} - c_{lk} \cdot C_{km}}{\sqrt{\sum_{k=1}^{N}\left[\left(\sum_{k=1}^{N} C_{lk}\right)\left(\sum_{f,g=1\&f\neq g}^{N} C_{gf}\right)\right]}\sqrt{\sum_{k=1}^{N}\left[\left(\sum_{k=1}^{N} C_{lk}\right)\left(\sum_{f,g=1\&f\neq g}^{N} C_{gf}\right)\right]}}$;
17 $\quad\quad$ $w_{t+1} \leftarrow w_t - \eta\partial[(D_p, C_p); w]$;
18 $\quad$ **end**
19 **end**
20 **return** $ACC, Loss, MCC, w$ to AP;
21 END FUNCTION;

Algorithm 2 AP Node-vehicle diversity selection algorithm. It takes ω, t and ϵ_t as input, where: w represents the server global ML model parameters, t represents the communication round, and ϵ_t represents the incremental parameter depth value in the t round. Let B represent the batch size of the vehicle node data, E represents the number of training iterations, and η represent the learning rate. Specifically, lines 2 to 7 represent the acquisition of the latest incremental parameter depth value. If there is incremental data, we use the above method to obtain the incremental parameter depth value, otherwise, the label is used as the latest value; The incremental weighting $u_{t+1}$is calculated by the depth value; the 9th line represents the acquisition of the latest data set; the 10th to 15th lines represent the use of the local gradient training method to train the local ML model parameters; the 19th line represents returning the vehicle local ML model parameter w, parameter increment weight $u_(t+1)$, and performance indicators Acc, Loss, MCC to AP.

5. Experimental Result

5.1. Dataset Description

We use the CIFAR-10 and vehicle image datasets for experiments, both of which are publicly available. CIFAR-10 is a popular dataset for experiments in the field of federated learning [34–36]. It contains 10 kinds of color pictures, such as planes, cars, ship, and truck. CIFAR-10 has a total of 50,000 training images and 10,000 test images. CIFAR-10 contains complex and diverse pictures in the real world. Not only are the pictures noisy, but also their proportions and characteristics of them are different, which brings great difficulty to the identification. These characteristics make this dataset more realistic than other datasets, so CIFAR-10 has been widely used in the field of federated learning. For example, Xiao et al. used the CIFAR-10 dataset to conduct experiments to verify the feasibility and performance gap between the proposed Accuracy Based Averaging(ABAVG) algorithm and the traditional federated algorithm in the Internet of Things [37]. To verify the performance of their proposed federated algorithm for intelligent machine collaboration, Sun et al. conducted experiments using the CIFAR-10 dataset [38]. Through experiments on the CIFAR-10 dataset, Wang et al. verified that their proposed method outperformed the basic method in terms of model accuracy and convergence rate in the vehicle selection and resource allocation algorithm for IoV [21].

We consider that although the CIFAR-10 dataset comes from real objects in the real world, it is not a dataset specific to the IoV scene. To further discuss the performance of the Fed-IW&DS framework in the IoV scene, we added a public IoV-specific dataset to conduct experiments to test the performance of the Fed-IW&DS framework. All the pictures in this vehicle image dataset are real IoV images collected from the real world, and the image format is 256 * 256. A total of 10 kinds of vehicle pictures were collected, including common vehicles such as buses, taxis, trucks, family sedans, minibuses, jeeps, SUVs, heavy trucks, racing cars, fire engines, and so on. Originally, there were 2 k pictures in the dataset, so the number allocated to each vehicle node would be relatively small. Therefore, we rotated this 911.3 m size dataset at different angles, and finally got 26,150 images, of which 18,305 were training images and 7845 were testing images.

Therefore, we adopt two datasets, CIFAR-10 and vehicle image, for the experimental validation of the Fed-IW&DS framework.

5.2. Experimental Parameter Settings and Experimental Platform

In this experiment, a convolutional neural network is used to validate the two datasets CIFAR-10 and vehicle image. The architecture of CNN is the input layer, convolution layer, fully connected layer, 64-node, arctangent activation function fully connected layer, and an output layer. The specific experimental parameters are shown in Table 3.

Table 3. Each specific parameter of the experiment.

Parameter Category	Parameter Name	CIFAR-10	Vehicle Image
CNN parameters	Number of input channels	3	3
	Number of output channels	5	1
	Convolution kernel size	5 * 5	2 * 2
	Fully connected layer 1	64 * 4 * 4	61* 61 * 64
	Fully connected layer 2	4 * 4 *16	4 * 4 * 16
	Convolutional Layer 1	3, 64, 5	3, 32, 5, 1
	Convolutional Layer 2	64, 64, 5	32, 64, 5, 1
	Stride	2	2
	Padding	0	0
	Dilation	1	1
Federated Learning Parameters	Global training rounds	10	
	Local training rounds	5	
	Initial vehicles	10	
	Data Distribution	Non-iid	
	Learning rate	0.01	
	Decay rate	0.1	
	Processor	Cpu	
Incremental data parameters	The number of images in the training set	50,000	18,305
	The number of images in the testing set	10,000	7845
	Initial allocation quantity for each vehicle (round 1)	2000	750
	Allocation quantity for each vehicle (2nd to 10th rounds)	Initial * (1/R) * Rnd_Int Initial is the initial allocation quantity for each vehicle (round 1). R is the global training round (1∼10). Rnd_Int is a random real number between 1∼2.	

The specific configuration of the experimental platform is: Intel(R) Core(TM) i7-10300H CPU@ 2.50 GHz 2.50 GHz, DDR4 2933 MHz 16 GB memory, Windows 10 Professional Edition 64-bit operating system. The experimental software uses python 3.6, PyTorch 1.4.0+CPU, and Keras 2.4.3.

5.3. Experimental Results

5.3.1. Diversity-Ratio Values

To verify the optimization effect of the Fed-IW&DS framework on vehicle node selection, the above dataset is randomly scrambled and divided 30% for testing, and the rest are randomly divided into 10 parts, indicating that 10 vehicle nodes are used for local training. We first calculate the diversity score D-R of the vehicle nodes participating in the global ML model training in the Fed-IW&DS framework and determine a better D-R value to better coordinate performance and efficiency. The performance indicators used in the experiments are as follows. (1) Fixed communication rounds. The different D-R numerical settings are all trained under the same 10-round global ML model training. (2) Accuracy. The highest accuracy that the Fed-IW&DS framework can achieve, including both training and testing sets. (3) Time per round. We repeated all experiments 10 times and then compared the average values of these indicators. The experimental results are shown in Table 4.

Table 4. Impact of different Diversity-Ratio values.

Dataset	D-R Value	Local ML Model Training Time Per Round (Unit: Seconds)	Training Set Accuracy	Testing Set Accuracy	Testing Set Loss	Testing Set MCC
CIFAR-10	0.2	4.1257	0.9265	0.8623	0.0002573	0.7738
	0.4	4.6481	0.9375	0.8845	0.0002467	0.6644
	0.6	4.8962	0.9414	0.9143	0.0002397	0.8014
	0.8	7.8413	0.9283	0.8926	0.0002372	0.8479
	1.0	10.9625	0.9224	0.8741	0.0002431	0.7738
Vehicle Image	0.2	6.2590	0.9073	0.8687	0.0002545	0.8638
	0.4	6.7105	0.9377	0.8520	0.0000284	0.8592
	0.6	6.2426	0.9631	0.9528	0.0000262	0.9172
	0.8	9.6470	0.9680	0.9461	0.0000478	0.9117
	1.0	13.0640	0.9279	0.9113	0.0004791	0.8633

Table 4 shows the performance impact of different Diversity-Ratio D-R values (i.e., the number of vehicle nodes participating in the global ML model per round) on various aspects of the global ML model. According to Table 4, when the D-R value is 0.2, 0.4, and 0.6, the accuracy of the global ML model increases with the increase of the D-R value, and the training time of the Local ML model does not increase significantly. However, when the D-R value is 0.8 and 1.0, we found that the Local ML model training time increases significantly, while the accuracy decrease. In other words, there are many meaningless global ML model training. To balance the problem of accuracy and computational time cost, we choose 0.6 as the D-R value of the following experiments.

5.3.2. The Acc, Loss, MCC Results and Computational Time Cost

- Accuracy Results

Next, we compare the Acc, Loss, and MCC values of FED-AVG [39], FED-SGD [40], Fed-prox [41], and Fed-IW&DS. More importantly, to verify the effect of our improved arctangent decay function in the global ML model, we also compared the performance when using the linear function, square curve function, and pure tangent function.

As can be seen from Figure 5, with the increase of global ML model rounds in all frameworks, only the accuracy of Fed-prox and Fed-IW&DS are continuously improving, but the improvement of fed-prox is very tiny. In CIFAR-10, the accuracy of the Fed-IW&DS model is about 30% higher than that of the Fed-prox model. This gap increases to 35% when the vehicle Image dataset is used. Moreover, the performance of FED-AVG and FED-SGD frameworks during the experiment is very unstable. The accuracy continues to decline. We believe that this is mainly due to the large amounts of increasing incremental data with the global ML model training rounds. The two frameworks, FED-AVG and FED-SGD, are incapable of handling large amounts of incremental data, resulting in poor model performance.

We continue to examine the accuracy performance of different decay functions. Linear functions are completely unable to handle incremental data challenges in the CIFAR–10 datasets as the accuracy rate are always maintained at a very low level, and even in the vehicle image dataset, there is a phenomenon of gradient explosion. The squared curve function showed a drop in accuracy after the fourth round, and the accuracy has not been stabilized since then. The arctangent function has better performance than them, but it is about 10–12% lower than the Fed-IW&DS framework. The results demonstrate that the unimproved tangent function has limited capability for incremental data in the later stage. The decay function of " amplifying the small signal " in the Fed-IW&DS framework is more in line with the reality of the IoV.

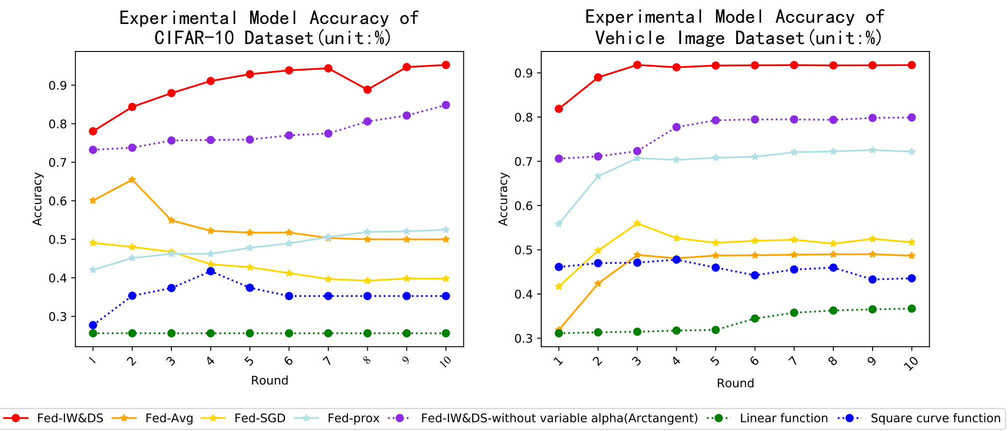


Figure 5. Comparison of Accuracy.

- Loss Results

Loss and accuracy in machine learning are relatively authoritative performance indicators. Therefore, we also compare the loss values of FED-AVG, FED-SGD, FED-prox, and the FED-IW&DS framework. As shown in Figure 6, we can see that the loss values of FED-AVG and FED-SGD are in a relatively unstable state with the increase of global ML model training rounds in both datasets, with no obvious improvement trend. Among the four frameworks, only FED-IW&DS and Fed-prox are successful, but it is obvious that there is still a big gap between Fed-prox and the Fed-IW& DS framework.

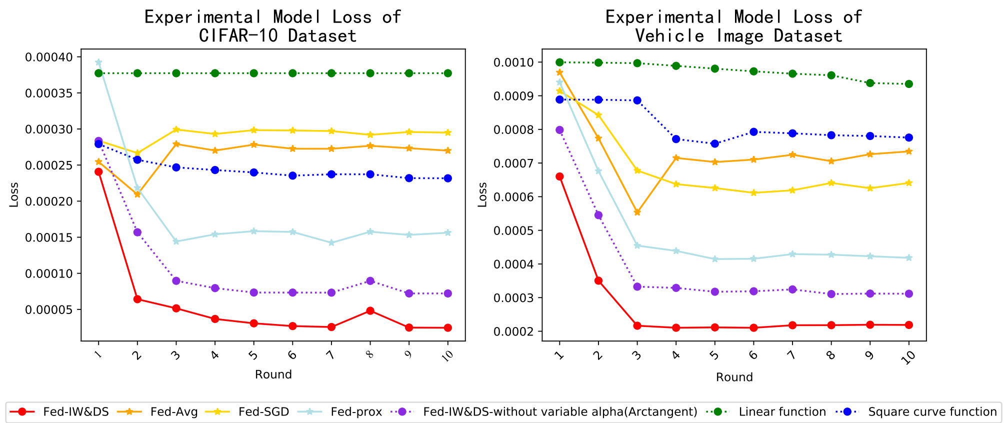


Figure 6. Comparison of Loss.

We continue to examine the loss performance of different decay functions. Linear functions perform poorly in CIFAR-10 and even exploding gradients again in the vehicle image dataset. Although the square curve function has a certain convergence trend, its convergence speed is extremely slow, and it cannot reach our goal for a long time. The arctangent function shows a relatively good and stable convergence trend, but its performance is not comparable to the Fed-IW&DS framework.

- MCC Results

MCC is generally accepted as a comprehensive evaluation indicator. MCC aggregates the confusion matrix into a single value and considers false positive and false negative error cases. Therefore, the MCC evaluation indicator is an excellent measurement even in unbalanced datasets.

We compare the MCCs of the above seven frameworks, as shown in Figure 7. The FED-AVG and FED-SGD are maintained between 0.5 and 0.6 in both datasets, proving that their performance is only marginally better than random prediction and cannot be further improved by local ML learning. Based on the frameworks' comparisons, only FED-prox and arctangent functions perform well, but their performance is still much weaker than the Fed-IW&DS framework. The MCC values of the linear function and the square curve function are even lower than 0.5, which shows that their performance is similar to the random prediction, and the learning efficiency is very low. While the traditional arctangent function is the closest to our results, its performance is still worse than the Fed-IW&DS framework. To further clarify, there was a drop in the 8th round of the Fed-IW&DS framework, which was caused by a large amount of unfamiliar incremental data. And in the 9th round, the Fed-IW&DS framework can automatically recover to a high MCC through diversity selection and incremental parameters. This shows that the Fed-IW&DS framework has strong robustness.

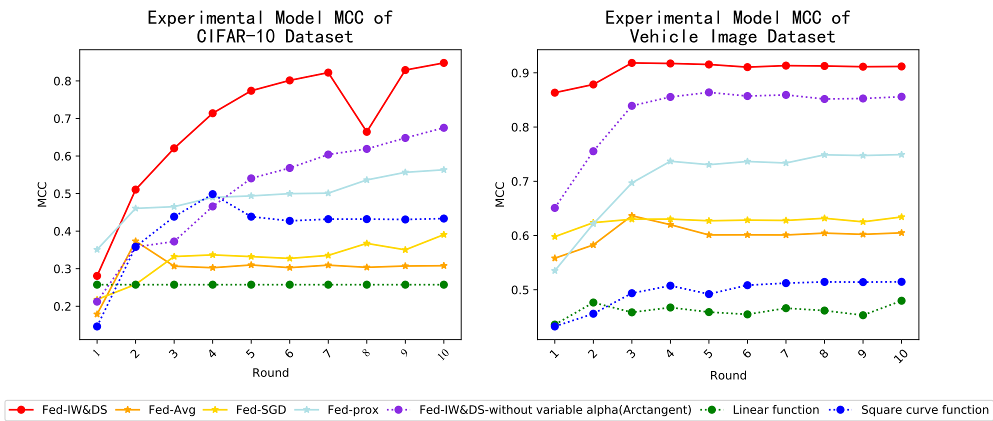


Figure 7. Comparison of MCC.

The Fed-IW&DS framework can maintain a high level of image recognition whether using the CIFAR-10 or vehicle image datasets, and its performance is far superior to several common federated learning frameworks such as FED-AVG, FED-SGD, and FED-prox. In terms of the decay function, the linear function has a gradient explosion phenomenon, so the constructed model cannot be used. The performance of the square curve function is unstable, so the model cannot be effectively learned. Fed-IW&DS framework performs better than the traditional inverse tangent function. Increasing the depth value of the incremental parameters effectively improves the performance of the Fed-IW&DS framework, successfully amplifying the small signal, which greatly widens the gap between the framework and other frameworks.

- Computational Time Cost

In addition to the above three indicators, the computing time cost is also an important indicator to measure performance. An excellent framework is needed to balance time cost and performance reasonably, to achieve better accuracy in a shorter time. We compare the average time for the Fed-IW&DS framework with the three federated learning frameworks, and the results are shown in Table 5.

Table 5. Time Cost.

Training Time for Local ML Models	CIARF-10 (Unit: Second)	Vehicle Image (Unit: Second)
Fed-AVG	3.4311	4.8437
Fed-SGD	4.9523	6.8524
Fed-prox	4.9354	6.1489
Fed-IW&DS	4.8962	6.2426

According to the results in Table 5, FED-AVG takes the shortest time among the four frameworks involved in the comparison, but the previous experiments show that its performance is very poor. The computation time required by FED-sgd and FED-prox is not much different from that of the Fed-IW&DS framework, but we have proved that our framework's performance is obviously better than the other two frameworks.

Moreover, vehicle manufacturers represented by Tesla, Audi, Mercedes Benz, etc. have increased their intelligent driving computing power to 500–1000 tops in the new generation of models (Tops is the abbreviation of tera operations per second. 1 tops means that the processor can perform one trillion operations (10^{12}) per second.). In September 2022, NVIDIA released a blockbuster autonomous driving chip, DRIVE Thor, whose computing power reached an astonishing 2000 TOPS, 8 times that of the previous Orin chip and 14 times that of Tesla's FSD chip. This experimental environment configuration is far below 200 tops computing power. Therefore, we believe that as the computing power of intelligent driving continues to grow, Fed -IW&DS framework has both theoretical and practical significance.

6. Conclusions

Faced with the challenges of incremental data and data diversity in IoV, the common baseline frameworks of federated learning are not performing well. In response to this, this paper proposes a Federated Learning Framework Based on Incremental Weighting and Diversity Selection, Fed-IW&DS. In this framework, we propose a scheme based on diversity selection and incremental parameter depth-value, which achieves "amplifying the small signal + reducing the large signal" for incremental data. To verify the performance of the Fed-IW&DS framework, we use the well-recognized CIFAR-10 and the vehicle image dataset for experiments. We compare the performance of this framework with common baseline frameworks and several decay functions. According to multi-indicators Acc, Loss, and MCC, the Fed-IW&DS framework outperforms other frameworks participating in the comparison within almost the same computational time cost. Therefore, our framework can solve the challenges of isolated data islands, data diversity, and incremental data in IoV, and the findings should make a significant contribution to the field of federated learning.

Author Contributions: Conceptualization, Y.L. and S.L.W.; methodology, Y.L.; software, Y.L. and M.W.; validation, T.F.N., S.L.W. and M.Z.; formal analysis, M.Z.; investigation, S.L.W.; resources, S.L.W.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, S.L.W.; visualization, T.F.N.; supervision, S.L.W.; project administration, M.Z.; funding acquisition, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Guangxi Zhuang Autonomous Region Education Department's New Engineering Project in 2022: Construction of the ASEAN New Engineering Education Community in Foreign Language Universities, No. XGK2022032.

Data Availability Statement: The following information was supplied regarding data availability: The raw data is available in the Supplemental Files. CIFAR-10 python is available at the University of Toronto (accessed on 8 April 2009): http://www.cs.toronto.edu/~kriz/cifar.html. The "vehicle classification dataset" (913 MB) is available at AIStudio and accessible using GitHub: (accessed on 19 June 2021) https://aistudio.baidu.com/aistudio/datasetdetail/95878?lang=en.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chaudhry, S.A. Designing an Efficient and Secure Message Exchange Protocol for Internet of Vehicles. *Secur. Commun. Netw.* **2021**, *2021*, 5554318. [CrossRef]
2. Zhou, Y.; Wang, J.; Wang, Z. Bearing Faulty Prediction Method Based on Federated Transfer Learning and Knowledge Distillation. *Machines* **2022**, *10*, 376. [CrossRef]
3. Majeed, U.; Hassan, S.S.; Hong, C.S. Cross-Silo Model-Based Secure Federated Transfer Learning for Flow-Based Traffic Classification. In Proceedings of the 35th International Conference on Information Networking (Icoin 2021), Jeju Island, Korea, 13–16 January 2021; IEEE: New York, NY, USA, 2021; pp. 588–593. [CrossRef]
4. Anowar, F.; Sadaoui, S. Incremental Neural-Network Learning for Big Fraud Data. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (Smc), Toronto, ON, Canada, 11–14 October 2020; IEEE: New York, NY, USA, 2020; pp. 3551–3557. [CrossRef]
5. Ge, S.; Luo, L.; Li, H. Orthogonal Incremental Non-Negative Matrix Factorization Algorithm and Its Application in Image Classification. *Comput. Appl. Math.* **2020**, *39*, 54. [CrossRef]
6. Jiang, Z.; Lin, R.; Yang, F. An Incremental Clustering Algorithm with Pattern Drift Detection for IoT-Enabled Smart Grid System. *Sensors* **2021**, *21*, 6466. [CrossRef]
7. Andrei, M.A.; Boiangiu, C.A.; Tarbă, N.; Voncilă, M.L. Robust Lane Detection and Tracking Algorithm for Steering Assist Systems. *Machines* **2022**, *10*, 10. [CrossRef]
8. Bzai, J.; Alam, F.; Dhafer, A.; Bojović, M.; Altowaijri, S.M.; Niazi, I.K.; Mehmood, R. Machine Learning-Enabled Internet of Things (IoT): Data, Applications, and Industry Perspective. *Electronics* **2022**, *11*, 2676. [CrossRef]
9. Gu, F.; Yang, X.; Li, X.; Deng, H. Computational Resources Allocation and Vehicular Application Offloading in VEC Networks. *Electronics* **2022**, *11*, 2130. [CrossRef]
10. Al-Sharman, M.; Murdoch, D.; Cao, D.; Lv, C.; Zweiri, Y.; Rayside, D.; Melek, W. A Sensorless State Estimation for a Safety-Oriented Cyber-Physical System in Urban Driving: Deep Learning Approach. *IEEE-Caa J. Autom. Sin.* **2021**, *8*, 169–178. [CrossRef]
11. Zhang, D.; Shi, W.; St-Hilaire, M.; Yang, R. Multiaccess Edge Integrated Networking for Internet of Vehicles: A Blockchain-Based Deep Compressed Cooperative Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1–15. [CrossRef]
12. Zhao, X.P.; Jiang, R. Distributed Machine Learning Oriented Data Integrity Verification Scheme in Cloud Computing Environment. *IEEE Access* **2020**, *8*, 26372–26384. [CrossRef]
13. Magdum, J.; Ghorse, R.; Chaku, C.; Barhate, R.; Deshmukh, S. A Computational Evaluation of Distributed Machine Learning Algorithms. In Proceedings of the 2019 IEEE 5th International Conference for Convergence in Technology (I2ct), 2019, Bombay, India, 29–31 March 2019; IEEE: New York, NY, USA, 2019. [CrossRef]
14. Wasilewska, M.; Bogucka, H.; Kliks, A. Federated Learning for 5G Radio Spectrum Sensing. *Sensors* **2021**, *22*, 198. [CrossRef]
15. Tahir, M.; Ali, M.I. On the Performance of Federated Learning Algorithms for IoT. *IoT* **2022**, *3*, 273–284.
16. Dahmane, S.; Yagoubi, M.B.; Brik, B.; Kerrache, C.A.; Calafate, C.T.; Lorenz, P. Multi-Constrained and Edge-Enabled Selection of UAV Participants in Federated Learning Process. *Electronics* **2022**, *11*, 2119. [CrossRef]
17. Hammedi, W.; Brik, B.; Senouci, S.M. Federated Deep Learning-Based Framework to Avoid Collisions Between Inland Ships. In Proceedings of the 2022 International Wireless Communications and Mobile Computing (IWCMC), Dubrovnik, Croatia, 30 May–3 June 2022; pp. 967–972. [CrossRef]
18. Gong, H.; Jiang, L.; Liu, X.; Wang, Y.; Wang, L.; Zhang, K. Recover User's Private Training Image Data by Gradient in Federated Learning. *Sensors* **2022**, *22*, 7157. [CrossRef]
19. Jeong, Y.; Kim, T. A Cluster-Driven Adaptive Training Approach for Federated Learning. *Sensors* **2022**, *22*, 7061. [CrossRef]
20. Sun, R.; Li, Y.; Shah, T.; Sham, R.W.H.; Szydlo, T.; Qian, B.; Thakker, D.; Ranjan, R. FedMSA: A Model Selection and Adaptation System for Federated Learning. *Sensors* **2022**, *22*, 7244. [CrossRef]
21. Wang, S.; Liu, F.; Xia, H. Content-Based Vehicle Selection and Resource Allocation for Federated Learning in IoV. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Nanjing, China, 29 March–1 April 2021; IEEE: New York, NY, USA, 2021. [CrossRef]
22. Liang, F.; Yang, Q.; Liu, R.; Wang, J.; Sato, K.; Guo, J. Semi-Synchronous Federated Learning Protocol With Dynamic Aggregation in Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4677–4691. [CrossRef]
23. Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D.; Yang, Q. A Fairness-Aware Incentive Scheme for Federated Learning. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA , 7–9 February 2020; pp. 393–399. [CrossRef]
24. Lyu, L.; Yu, J.; Nandakumar, K.; Li, Y.; Ma, X.; Jin, J.; Yu, H.; Ng, K.S. Towards Fair and Privacy-Preserving Federated Deep Models. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 2524–2541. [CrossRef]
25. Huang, J.; Xu, C.; Ji, Z.; Xiao, S.; Liu, T.; Ma, N.; Zhou, Q. AFLPC: An Asynchronous Federated Learning Privacy-Preserving Computing Model Applied to 5G-V2X. *Secur. Commun. Netw.* **2022**, *2022*, 9334943. [CrossRef]
26. Ayub, A.; Wagner, A.R. Cognitively-Inspired Model for Incremental Learning Using a Few Examples. In Proceedings of the 2020 IEEE/Cvf Conference on Computer Vision and Pattern Recognition Workshops (Cvprw 2020), Seattle, WA, USA, 14-19 June 2020; IEEE Computer Soc: Los Alamitos, CA, USA, 2020; pp. 897–906. [CrossRef]

27. Jaiswal, G. Performance Analysis of Incremental Learning Strategy in Image Classification. In Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence 2021), Noida, India, 28–29 January 2021; IEEE: New York, NY, USA, 2021; pp. 427–432. [CrossRef]
28. Yang, Q.; Gu, Y.; Wu, D. Survey of Incremental Learning. In Proceedings of the 2019 31st Chinese Control and Decision Conference (CCDC 2019), Nanchang, China, 3–5 June 2019; IEEE: New York, NY, USA, 2019; pp. 399–404. [CrossRef]
29. Cui, W.; Xia, W.; Lan, Z.; Qian, C.; Yan, F.; Shen, L. A Self-adaptive Feedback Handoff Algorithm Based Decision Tree for Internet of Vehicles. In Proceedings of the Ad Hoc Networks, Adhocnets, Cairns, Australia, 20–23 September 2018; Zheng, J., Xiang, W., Lorenz, P., Mao, S., Yan, F., Eds.; Springer International Publishing Ag: Cham, Switzerland, 2019; Volume 258, pp. 177–190. [CrossRef]
30. Wang, X.; Zhu, Y.; Han, S.; Yang, L.; Gu, H.; Wang, F.Y. Fast and Progressive Misbehavior Detection in Internet of Vehicles Based on Broad Learning and Incremental Learning Systems. *IEEE Internet Things J.* **2022**, *9*, 4788–4798. [CrossRef]
31. Zhu, M.y.; Chen, Z.; Chen, K.f.; Lv, N.; Zhong, Y. Attention-Based Federated Incremental Learning for Traffic Classification in the Internet of Things. *Comput. Commun.* **2022**, *185*, 168–175. [CrossRef]
32. Huang, Q.; Wang, D. Study on the Rumor Propagation Model with Penalty Factor. In Proceedings of the Proceedings of the 32nd 2020 Chinese Control and Decision Conference (Ccdc 2020), Hefei, China, 22–24 August 2020; IEEE: New York, NY, USA, 2020; pp. 5176–5181. [CrossRef]
33. Liu, J.; Shi, J.; Hao, F.; Dai, M.; Zhang, Z. Arctangent Entropy: A New Fast Threshold Segmentation Entropy for Light Colored Character Image on Semiconductor Chip Surface. *Pattern Anal. Appl.* **2022**, *25*, 1075–1090. [CrossRef]
34. Lei, Y.; Wang, S.L.; Su, C.; Ng, T.F. OES-Fed: A Federated Learning Framework in Vehicular Network Based on Noise Data Filtering. *PeerJ Comput. Sci.* **2022**, *8*, e1101. [CrossRef]
35. Gharibi, M.; Bhagavan, S.; Rao, P. FederatedTree: A Secure Serverless Algorithm for Federated Learning to Reduce Data Leakage. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; Chen, Y., Ludwig, H., Tu, Y., Fayyad, U., Zhu, X., Hu, X., Byna, S., Liu, X., Zhang, J., Pan, S., et al., Eds.; IEEE: New York, NY, USA, 2021; pp. 4078–4083. [CrossRef]
36. Ullah, S.; Kim, D.H. Federated Learning Using Sparse-Adaptive Model Selection for Embedded Edge Computing. *IEEE Access* **2021**, *9*, 167868–167879. [CrossRef]
37. Xiao, J.; Du, C.; Duan, Z.; Guo, W. A Novel Server-side Aggregation Strategy for Federated Learning in Non-IID Situations. In Proceedings of the 2021 20th International Symposium on Parallel and Distributed Computing (ISPDC), Cluj-Napoca, Romania, 28–30 July 2021; Potolea, R., Iancu, B., Slavescu, R.R., Eds.; IEEE: New York, NY, USA, 2021; pp. 17–24. [CrossRef]
38. Sun, Y.; Chong, N.S.T.; Ochiai, H. Information Stealing in Federated Learning Systems Based on Generative Adversarial Networks. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; IEEE: New York, NY, USA, 2021; pp. 2749–2754. [CrossRef]
39. Zhong, Z.; Zhou, Y.; Wu, D.; Chen, X.; Chen, M.; Li, C.; Sheng, Q.Z. FedAvg: Parallelizing Federated Learning with Theoretical Guarantees. In Proceedings of the IEEE Conference on Computer Communications (IEEE Infocom 2021), Vancouver, BC, Canada, 10–13 May 2021; IEEE: New York, NY, USA, 2021. [CrossRef]
40. Rjoub, G.; Wahab, O.A.; Bentahar, J.; Bataineh, A. Trust-Driven Reinforcement Selection Strategy for Federated Learning on IoT Devices. *Computing* **2022**, 1–23. [CrossRef]
41. Wang, J.; Liu, Q.; Liang, H.; Gauri, J.; Poor, H.V. A Novel Framework for the Analysis and Design of Heterogeneous Federated Learning. *IEEE Trans. Signal Process.* **2021**, *69*, 5234–5249. [CrossRef]