Towards Efficient Federated Learning Using Agile Aggregation in Internet of Vehicles

Xin He, Xiaolin Hu, Guanghui Wang, Junyang Yu, Zhanghong Zhao and Xiaobin Lu

1School of Software, Henan University, Kaifeng, China
2Henan International Joint Laboratory of Intelligent Network Theory and Key Technology, Henan University, Kaifeng, China
3Aerospace Science and Industry Defense Technology Research and Test Center, Beijing, China
4Henan HMCY Technology Co, Ltd., Zhengzhou, Henan, China
5Yellow River Conservancy Technical Institute, Kaifeng, Henan, China
6PLA SSF Information Engineering University, Zhengzhou, China

Correspondence should be addressed to Guanghui Wang; gwang@vip.henu.edu.cn

Received 8 December 2022; Revised 31 May 2023; Accepted 21 August 2023; Published 1 September 2023

1.Introduction

Internet of vehicles (IoV) is a new paradigm in the future Internet of things, where vehicles sense external information through on-board equipment on the road [1–4]. Vehicles can obtain huge amounts of data (e.g., video data), and the circulation of data among vehicles is one of the most effective means to improve the efficiency of data utilization and fully explore the value of vehicular data [5–8]. However, due to concerns about personal privacy data leakage, vehicle owners are unwilling to upload data to the data center, which hinders the flow of data and affects the development of the IoV industry [9–12].

With the enhancement of public awareness of data privacy, federated learning methods are quickly applied to various fields [13–15] in the Internet of things. Based on the significant advantages of federated learning for data privacy protection [16–18], the purpose of collaborative training without the leakage of local data is achieved. High-dynamic vehicles in IoV only need to transmit the trained model parameters to road side units (RSUs) by federated learning. It does not need to share the entire original dataset, which reduces the risk of privacy leakage [14, 19, 20]. Vehicle-sensed data are trained locally, and only the trained model parameters are transmitted to the server, which improves privacy preservation during the learning process. Therefore, it is important to investigate federated learning for IoV [21, 22].

Existing federated learning methods have a long training time and poor collaborative training accuracy when applied...
to IoV scenarios due to their high-dynamic characteristics [23, 24]. Because of heterogeneity among vehicles, there are different computing and communication capabilities among different vehicles. RSUs can quickly receive local updates uploaded by some vehicles. They need to wait for local updates uploaded by other slower vehicles to aggregate, and the waiting time is long. Vehicles that have completed training cannot obtain the latest global model in time due to waiting. Therefore, vehicles that have completed training face both low computational efficiency and poor training accuracy. For example, IoV is more sensitive to data transmission delays during model training [25–27]. Based on the heterogeneity of vehicles in the IoV scenario, computational and communication capabilities are different. After the local update is uploaded to RSUs by vehicles with higher performance, RSUs also need to wait for vehicles with lower performance to complete the upload of locally updated training to aggregate the global model. Due to the existence of unnecessary waiting, the computational efficiency of the global model is low. However, the traditional federated learning algorithms (such as FedSGD and FedAVG) perform model training based on multiple participants. They perform model average aggregation on the server side, which takes a long time in model average aggregation [28, 29]. Therefore, traditional federated learning methods have limitations in the scenario of the high-dynamic vehicle training process because they cannot guarantee the efficiency of model training. It is necessary to study efficient federated learning for IoV.

This paper proposes an efficient federated learning algorithm based on an agile aggregation model, which reduces the waiting time for model aggregation. In particular, a multistep federated learning architecture is developed by including task release, vehicle model training, identity authentication, quality assessment, and weighted aggregation processes. The novel agile aggregation model is established, where RSUs aggregate the received model update into the global model according to model weights and immediately send the aggregated model state back to the corresponding high-dynamic vehicle. The privacy of vehicle data is preserved during the federal learning process. The RSU controls the processes of vehicle training through vehicle authentication, quality assessment, and agile aggregation. Because the vehicle only transmits the calculated model updates to the RSU, the privacy of the vehicle’s original data is protected. In conclusion, the contributions of this paper are summarized as follows:

1. We develop a multistep federated learning architecture, including task release, vehicle model training, identity authentication, quality assessment, and weighted aggregation processes, which introduces a novel agile aggregation model to the scenario of high-dynamic vehicles in IoV.

2. We propose an efficient federated learning (EFL) algorithm to reduce the model aggregation time, thus improving the training efficiency of the vehicle learning model for IoV.

3. We analyze the advantages of the EFL algorithm over the traditional FedAVG algorithm in terms of efficiency, training accuracy, and security. Based on the MNIST dataset, the effectiveness and efficiency of the EFL algorithm are evaluated to show its correctness and superiority.

The structure of the paper is as follows: In Section 2, related research on federated learning in related fields is reviewed. Section 3 presents the system model and problem setting. Section 4 proposes the multistep federated learning architecture and the EFL algorithm. Section 5 analyzes the advantages of the EFL algorithm over the traditional FedAVG algorithm. In Section 6, the effectiveness and efficiency of the EFL algorithm are evaluated using the MNIST dataset. Finally, Section 7 concludes the paper.

2. Related Work

Federated learning is a distributed machine-learning paradigm [30], where the client uses local data for training and acts as a computing node for the entire training process. Since the data do not leave the node, the client’s privacy is protected. Federated learning technology is widely used in various fields, such as smart medical care [13], smart finance [14], and smart transportation [15], to address the problems of data privacy leakage and data silos. When applying federated learning to IoV, more attention is paid to data privacy and model training efficiency. For example, W. Zhang and Li [31] proposed a federated migration-learning method for fault diagnosis, which allows different users to use different models to enhance data privacy. However, this method cannot be directly applied to the dynamic data in the IoV scenario. Qu et al. [32] proposed a data-trading mechanism based on the reverse game, which can improve equipment efficiency to some extent but is not suitable for specific scenes with high requirements for delay.

Existing researchers have studied the problem of low model training efficiency with spatiotemporal data. Lu et al. in [33] proposed the parameter compression mechanism. This mechanism can greatly reduce the amount of communication required for training. However, when some clients have network delays, there will still be the issue of waiting, leading to low efficiency. Yang et al. in [34, 35] proposed classification standards and international norms. According to the data characteristics of the client, federated learning is divided into horizontal federated learning, vertical federated learning, and migration federated learning. Several research institutions have jointly written the Federated International Standard for Learning. It focuses on the difference in data characteristics, thus ignoring real-time performance. Mcmaham et al. in [28] reduced the number of training cycles by local iteration, thereby improving the overall model’s training efficiency. However, the data processes are nondynamic real-time data, such as image pixels and corpus. Although some federated learning methods have improved in efficiency due to the influence of delay, the application of federated learning to IoV scenarios with spatiotemporal data still faces the problem of low
efficiency. Thus, the existing federated learning technology should be improved for the applications in the IoV scenario with high-dynamic data.

Several studies have paid attention to machine-learning problems in IoV. Y. Sun et al. in [36] analyzed major problems and challenges in autonomous vehicles. Bonawitz et al. in [37–39] analyzed the problems faced by vehicles from the perspective of data management. Wang in [40] analyzed the research process of vehicle communication. Although the above content fully analyzes the problems faced in the field of IoV, it is important to judge the efficiency of vehicle participation in training and protect the privacy of users’ data for the development of IoV. But there is no actual improvement of such problems. Yuan et al. in [41] proposed a deep reinforcement-learning method to determine the location of the vehicle. Its main purpose is to improve the efficiency of network resource allocation. However, the low training efficiency of some vehicles due to network delay has not been addressed. Wang et al. in [14, 42] proposed a content-based vehicle selection and resource allocation method. Training efficiency is improved by optimizing the node selection model. However, some clients still have the risk of network delay. Feng et al. in [43, 44] explored a minimum-maximum cost optimization problem. Zhou et al. in [45] explored how to achieve model fairness. Although it can improve the enthusiasm of the client, it does not change the efficiency of training. The training model has been tried many times to make IoV better, but the protection of local data privacy for vehicles has not greatly improved while training. The above research furthers the research on IoV efficiency and data privacy.

The efficient model training issue has been investigated while protecting data privacy in the scenario of high-dynamic data in IoV. Xiong et al. in [46, 47] proposed a vehicle crowdsensing system based on blockchain and a privacy protection method by IoV sensing. Cha et al. in [48] proposed a client selection scheme based on fuzzy logic. Lu et al. in [49, 50] utilized an asynchronous federated learning scheme to improve computational efficiency in vehicular networking scenarios. However, they are all based on the static data of vehicles to participate in federal learning and training, thereby improving training efficiency. In practice, vehicles are not static computers, and the data of vehicles are also dynamic. Meese et al. in [51] proposed a blockchain joint-learning architecture for online traffic flow prediction using real-time data and edge computing but did not focus on efficiency research. The timeliness of client data training is not considered in the above work. Local data are a real-time variable in IoV application scenarios. Therefore, algorithms that both meet the needs of highly dynamic data and protect data privacy still need further exploration. This paper makes an attempt to solve the problem of low training efficiency for federal learning in IoV.

3. Models and Design Goals

3.1. Federated Learning for IoV. The execution process in the federated learning for IoV is described in Figure 1. The RSU selects the task broadcast with the largest current weight based on task weight so that the vehicle can obtain relevant information. After the vehicle obtains and parses the task, it is calculated according to the local data. The updated model is sent to the RSU. After receiving the RSU, identity authentication and quality detection are carried out. Agile aggregation is carried out according to the model weight of the vehicle. After aggregation is completed, the model returns to the vehicle immediately.

In the federated learning model for IoV, $Q_k = (W_k, U_k), (\forall K = \{1, 2, \cdots, N\})$ is defined to represent the task items in the current task set, where $W_k$ represents the weight of the task and its size represents the weight. $U_k$ is defined as the specific content. $C_i (\forall i = \{1, 2, \cdots, M\})$ is represented as the corresponding vehicle. $P_k^i = (V_{k}^i / V_{K}^M), (P_k^i \in [0, 1])$ is denoted as the weight of model updates in the global model during agile aggregation, where $V_{K}^M$ is the sum of the security weights of all vehicles currently involved, namely, $V_{K}^M = \sum_{i=1}^{M} V_{k}^i$. The main symbols are shown in Table 1.

3.2. Multistep Federated Learning Architecture. The multistep federated learning architecture is described. The architecture includes task release, vehicle model training, identity authentication, quality assessment, and node-switching processes, where a novel agile aggregation model is introduced to the scenario of high-dynamic vehicles in IoV.

3.2.1. Task Release. RSUs collect and weight tasks according to their urgency. The tasks with the largest weight are trained. The RSU establishes the initial state $\omega_0$ of the model. It sends the training task $Q_k$ to the vehicle. After receiving, the vehicle parses the task content $U_k$. After comprehensive consideration by owners, if they decide to participate, the vehicle establishes network communication with the RSU. It establishes a temporary index after identity authentication is passed. The index is a dynamically changing collection $C_i$. The RSU is stored in the form of index tables. It sends the initial model parameters to the vehicle through the index table.

3.2.2. Local Training. The vehicle participating in the task is trained locally with the current model state through local data. If it is the first round of model training, the initial model parameter is $\omega_0$ to train with local data. The update of training is uploaded to the RSU such as the launch tower and base station. Finally, the vehicle uploads locally trained model updates to the RSU.

3.2.3. Model Aggregation. The microserver on the RSU aggregates the received vehicle local updates agilely. After the RSU receives the update of the model uploaded by the vehicle, the quality of the model update is evaluated by analyzing the update content of the model. After the uploaded model update is evaluated, the RSU weights the model update through $P_k^i$. The weighted model update is aggregated into the global model to generate a new current
model state. The RSU immediately returns the newly generated model state to the vehicle and waits for other vehicles.

3.2.4. Identity Authentication. The RSU determines whether the vehicle can participate in the learning process with identity authentication. For example, the vehicle can apply for registration to participate in task learning, and the initial security weight of the vehicle is \( V_i^0 = 1 \). With the contribution of participating in training, the value will be higher. When \( V_i^K > 0 \) is considered, the vehicle is authenticated. The gains from registration and the security weight improvement \( V_i^K \) are part of vehicle training incentives.

![Data interaction in federated learning for IoV.](image)

**Figure 1:** Data interaction in federated learning for IoV.

<table>
<thead>
<tr>
<th>Signs</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>The vehicle index</td>
</tr>
<tr>
<td>( J )</td>
<td>The number of flexible aggregates of RSUs</td>
</tr>
<tr>
<td>( M )</td>
<td>The total number of vehicles participating in training</td>
</tr>
<tr>
<td>( D )</td>
<td>The data on local vehicles</td>
</tr>
<tr>
<td>( \eta_i )</td>
<td>The local learning efficiency of vehicles</td>
</tr>
<tr>
<td>( V_i^0 )</td>
<td>The initial vehicle security weight. ( \forall K = {1, 2, \cdots, N} )</td>
</tr>
<tr>
<td>( V_i^K )</td>
<td>The current vehicle security weights. ( \forall K = {1, 2, \cdots, N} )</td>
</tr>
<tr>
<td>( P_i^K )</td>
<td>The current vehicle model weights. ( \forall K = {1, 2, \cdots, N} )</td>
</tr>
<tr>
<td>( \omega )</td>
<td>The state of the model after flexible aggregation</td>
</tr>
<tr>
<td>( \omega_0 )</td>
<td>The initial state of the model</td>
</tr>
<tr>
<td>( \omega_i )</td>
<td>The updating of the current vehicle model</td>
</tr>
<tr>
<td>( \omega_j )</td>
<td>The state of the model generated by the previous round of flexible aggregation</td>
</tr>
<tr>
<td>( \omega_{j+1} )</td>
<td>The last state of the model generated by flexible aggregation</td>
</tr>
<tr>
<td>( \omega^* )</td>
<td>The latest model update received by the road test team</td>
</tr>
<tr>
<td>( \omega^*_t )</td>
<td>The local model state before training. ( t ) is the number of local vehicle training sessions</td>
</tr>
<tr>
<td>( \omega^{t+1}_i )</td>
<td>The model update generated after local irradiation</td>
</tr>
</tbody>
</table>
3.2.5. Quality Assessment. Through the data surrounding the vehicle, the updated content of the model is evaluated to determine whether the updated model uploaded by the vehicle is reliable. If the vehicle is reliable, the RSU applies model update aggregation to the global model. If not reliable, the RSU gives up aggregation and sends back the certification results to the vehicle. The significance of identity authentication and quality assessment is to identify malicious vehicles and prevent them from affecting global model training. The vehicle uploads local updates to RSUs, and the RSUs aggregate the received local updates. In this process, authentication and quality assessment are needed to detect whether the received model updates are reliable and then determine whether local updates are aggregated into a global model.

3.2.6. Node Switching. The process of a vehicle switching is from one RSU node to another RSU node. As shown in Figure 2, the distance between RSU nodes is m meters, and the total length of the critical region is 3m/3 meters. When a vehicle enters the critical region, it stops sending model updates to the previous RSU node and starts sending model updates to the next RSU node, beginning the entire training process with the next RSU node. The previous RSU node deletes the vehicle’s information from the temporary index table, and the next RSU node updates the vehicle’s information in the temporary index table. The RSU nodes participate in the same training tasks through network transmission. As vehicles travel between different RSU nodes, a single RSU node in the entire traffic network can obtain all model updates. In special cases, if a vehicle sends a model update before entering the critical region but still has not received the global model sent by the RSU node when entering the critical region, the vehicle waits to receive the global model sent by the previous RSU node while in the critical region. If the vehicle still has not received the global model sent by the previous RSU node when leaving the critical region, it will no longer wait for that global model.

The mobility of vehicles between different roadside units refers to the process of vehicles switching between them. As roadside units only care about whether vehicles have uploaded model updates and do not pay attention to which specific vehicle has uploaded the updates, there will be no impact on model aggregation due to vehicle switching between different roadside units. Since vehicles switch between different roadside units, they bring the received models into the aggregation of the next roadside unit. Therefore, the overall model updates in the entire transportation network will be strengthened as a result.

3.3. Threat Model. Because the training data contain confidential client information, we focus on the confidentiality of the data of the participants in the threat model. No participant in training can access the data of other participants so that user privacy data can be preserved, even if there are internal adversaries (i.e., intruders) spying on privacy. Besides, malicious vehicles are prone to involve unqualified local updates into the global model, thus affecting entire federal learning and training. If there is a model update whose quality does not meet the standard, the client’s security weight will be reduced. The model update will not participate in flexible aggregation in this round. The local update of the vehicle is monitored by identity authentication and quality assessment to prevent malicious nodes from damaging the global training process, so the security of the vehicle when participating in model training is improved. We make the following assumptions:

1. In the system, we assume that vehicles will not be damaged during training. The channel between the vehicle and RSUs is protected by encryption.
2. Participants are willing to comply with the training rules, and they are curious about the data of other participants. There may be some users who want to get other people’s data to participate in the training in order to receive a reward. Some users may also intentionally use unrealistic data to participate in training, trying to prevent the completion of tasks.

3.4. Design Goals. This paper is dedicated to designing an efficient federated learning algorithm so as to protect the privacy of real-time data training in IoV. The design requirements are described as follows.

3.4.1. Efficiency. The EFL algorithm has low computational and communication overheads compared with the existing FedSGD and FedAVG algorithms. Asynchronous training methods are used for distributed model training. Agile aggregation is carried out on the RSU according to the model weight of the vehicle to help the vehicle obtain the latest information on model training in real time.

3.4.2. Security. The original data of vehicles cannot be obtained by other vehicles. Through RSU, the EFL algorithm establishes a joint training mode between vehicles. When the number of vehicles participating in training increases, difficulty increases exponentially to reverse the original data of other vehicles through global updating, and thus, overall security is ensured.

3.4.3. Accuracy. The local model update of the vehicle is updated in real time in the IoV scenario. The latest local model update in the EFL algorithm can ensure the accuracy of the learning model.

4. EFL Algorithm

4.1. Basic Ideas. EFL is an efficient federated learning algorithm that performs agile aggregation of model updates trained by the vehicle. The key idea is described as follows: The RSU sends task information to the vehicle. The vehicle performs local training based on the received task and local data. The vehicle sends the training results to the RSU. After receiving the vehicle’s model update, the RSU performs identity authentication and quality evaluation. When
verification is completed, the model update is weighted and aggregated according to the model weight. The training process is iteratively repeated to complete the entire model’s training.

When a vehicle switches from one RSU to another, the process involves the RSU authenticating the vehicle’s identity and establishing a temporary index table to dynamically store the vehicle’s information. This index table is stored in the form of a collection in the RSU. When the vehicle needs to receive model parameters, the RSU will send them to the vehicle through the index table. As the vehicle leaves one RSU and heads to another, the previous RSU will dynamically remove the vehicle’s information from the index table, while the next RSU will dynamically add the vehicle’s information to the index table. This way, all RSUs in the entire transportation network can receive all model updates, and vehicles can update models between different RSUs. It is worth noting that when a vehicle enters the critical zone, if it has not received the global model sent by the previous RSU yet, the vehicle will wait to receive the global model within the critical zone until it leaves. The EFL algorithm improves the efficiency of real-time data participating in federated learning and training. The training process is shown in Figure 3.

4.2. Local Training. Vehicles conduct calculations locally based on local data and their model state. In calculation, the EFL algorithm is considered applicable to the following objective function. The prediction loss of local data is carried out with the model parameter $\omega$:

$$\min_{\omega \in \mathbb{R}^d} f(\omega), f(\omega) = \ell(\omega, D).$$

Each vehicle $i$ calculates the current gradient decline $g^i = \nabla f_i(\omega)$, where $\omega$ is the global model state received by the vehicle. We assume that the learning efficiency of the vehicle is represented by $\eta_i$, $\omega^{t+1}_i \leftarrow \omega^t_i - \eta_i g^i_t$ is calculated according to the gradient descent. The latest local update of the vehicle is generated, which will be uploaded to the RSUs to participate in agile aggregation. After sorting out the above local calculation formulas, we obtain the following equation:

$$\omega^{t+1}_i \leftarrow \omega^t_i - \eta_i \nabla (\omega, D).$$

Each client uses its local data for locally gradient descent in the current model.

4.3. Agile Aggregation. After the RSU receives the local update uploaded by the vehicle, its first operation is authenticating the vehicle; that is, it determines whether there is $V^i_k > 0$. Then, a quality evaluation is conducted based on the uploaded local updates. If identity authentication and quality evaluation are qualified, the RSU updates the security weights for the vehicle and calculates the model weights. Among them, the calculation method of vehicle security weights is shown in (3). The RSU calculates the model weights based on the latest security weights of the vehicle. The calculation method is shown in (4):

$$V^i_{k+1} \leftarrow V^i_k + \frac{V^i_k}{M},$$

$$P^i_k = \frac{V^i_k}{\sum_{i=1}^{M} V^i_k}.$$
If vehicle verification fails, the RSU will reduce the weight of the vehicle and update the security weight. The assessed information is sent back to the vehicle. If verification fails, the calculation method for the security weight is shown in (5). Model weights are no longer updated. The central server stopped aggregating the model updates:

\[ V^i_k \leftarrow V^i_k - \frac{V^0_k}{M_i}. \]  

The weighted model updates are aggregated agilely on the RSUs. The RSUs update the weighted model in real time and update the state of the model generated after aggregation for the vehicle. The agile aggregation formula is shown as follows:

\[ \omega_{j+1} \leftarrow \omega_j + P_k(j \omega_j^*). \]  

4.4. Algorithm Description. Algorithm 1 gives a complete pseudocode and accurately describes the EFL algorithm based on the agile aggregation model. Algorithm 1 shows the processes performed separately on vehicles and RSUs. In Steps 1–17, RSU is considered a microsector to participate in the federal learning and training process. In Steps 2–4, the RSU first selects the training task and initializes the model. The vehicle security weight \( V^0_k \) is initialized. In Steps 5–14, the RSU participates in each iteration process and the RSU sends the initial model state to the vehicle by broadcasting. In Steps 7–14, the RSU performs identity authentication and quality assessment on the received information. If it does not pass authentication and assessment, the RSU will discard the information uploaded by the vehicle and update the security weight of the vehicle. If passed, the RSU will update the vehicle’s security weights and corresponding model weights. The model updates uploaded by the vehicle are aggregated into the global model according to model weights. In Steps 15–16, the RSU sends the aggregated global model to the current vehicle. In the meantime, the RSU prepares for receiving the aggregated global model of the next vehicle. In Steps 18–19, the process of transmitting client information between servers occurs. In Steps 20–22, the client participates in the local federal learning and training process. In Step 21, the vehicle participates in model training as a client using local data. In Step 22, the client sends the locally trained model update to the RSU.

5. Algorithm Analyses

5.1. Efficiency Analysis. The efficiency of model updating is used to quantify the efficiency of model updating by calculating the time interval between two model aggregations of RSUs. The efficiency is expressed by the average time of two adjacent global models obtained by all vehicles, which reflects the speed at which the vehicle completes local updates and uploads the global model. We assume that the efficiency of the updated model can be represented by the time interval between two global updates of the model. That is,
S = \{X \mid S_{11}, S_{21}, \ldots, S_{1i}, \ldots, S_{M1}\} + A$, where $S_{ij}$ denotes the time required for the client $i$ to obtain a round of model updating from the central server. “A” is a constant that indicates the time required for the client to train locally and aggregate in the central server model, and “S” denotes the time required for the client $i$ to obtain the global model updating round. The efficiency of the EFL algorithm in model training is higher than that of the FedAVG algorithm. The following theorem provides a theoretical basis for the efficiency of the EFL algorithm.

**Theorem 1.** In the federated learning training, there are $M$ clients, $i = 1, \ldots, M$. Let the time consumed by the client to train locally and aggregate in the central server model be constant $A$, $S_{ij}$ and $S_{ij}$ denote the time consumption updating model of EFL and FedAVG algorithms, respectively. If only $S_{ij}$ takes $\{S_{11}, S_{21}, \ldots, S_{1i}, \ldots, S_{M1}\} + A$, the following equation holds:

$$S_{1} \leq S_{2}. \quad (7)$$

**Proof.** First, the efficiency of the EFL algorithm is proved, namely, $S_{1} \leq S_{2}$ (if only $S_{1}$ takes $\{S_{11}, S_{21}, \ldots, S_{1i}, \ldots, S_{M1}\} + A$).

When the EFL algorithm updates the global model, each client contacts the central server immediately after calculating the data for updating the local model and further aggregating the model. That is,

$$S = \{X \mid S_{11}, S_{21}, \ldots, S_{1i}, \ldots, S_{M1}\} + A. \quad (8)$$

When the FedAVG algorithm updates the global model, all clients are weighted averages after local updates. The communication time between the client and the central server is consistent with that of the client that takes the longest time, namely,

$$S_{2} = \max_{x \in \{S_{1i}, \ldots, S_{M1}\}} x + A.$$

$$S_{1} = \frac{S_{1i} + A}{\max_{x \in \{S_{1i}, \ldots, S_{M1}\}} x + A} \cdot S_{2}. \quad (9)$$

The size relationship between $S_{1}$ and $S_{2}$ can be obtained by further reasoning based on the known conditions:

$$S_{1} \leq \max_{x \in \{S_{1i}, \ldots, S_{M1}\}} x, \quad \max_{x \in \{S_{1i}, \ldots, S_{M1}\}} x + A \leq 1, S_{1} \leq S_{2}. \quad (10)$$

According to the two formulas, it can be concluded that $S_{1} \leq S_{2}$ (if only $S_{1}$ takes $\{S_{11}, S_{21}, \ldots, S_{1i}, \ldots, S_{M1}\} + A$, the equality holds). Thus, Theorem 1 is proved.

Theorem 1 ensures the efficiency of the EFL algorithm, that is, $S_{1} \leq S_{2}$ (if only $S_{1}$ takes the equal sign $\{S_{11}, S_{21}, \ldots, S_{1i}, \ldots, S_{M1}\} + A$). Mobile vehicles can better protect the interests of clients with timely access to model update information. Therefore, it can be concluded that the EFL algorithm may be more suitable for the federated learning of vehicles on highways than the FedAVG algorithm. □

5.2. **Effectiveness Analysis.** Effectiveness refers to whether the training accuracy of the FedAVG algorithm in the global model can be achieved after using the EFL algorithm. The results show that the training accuracy of the EFL algorithm and the FedAVG algorithm in the global model is almost the same.
The training accuracy is the total amount of local updates (number and size) contained in the global model. The global model is composed of multiple local updates. Within a certain time, the total amount of local updates contained in the global aggregation is represented by \( P \). For example, the local update of vehicles is regarded as the "raw material" of the global model. For the global model, the EFL algorithm and the FedAVG algorithm use the same amount of local update data in aggregation. Finally, the components of the global model trained remain unchanged; that is, the training accuracy of the global model remains unchanged. It is assumed that from a global perspective, the updated model accuracy of the global model remains unchanged. It is assumed that from a global perspective, the updated model accuracy of the global model remains unchanged. That is, the training and the FedAVG algorithm use the same amount of local update data in aggregation. Finally, the components of the global model trained remain unchanged; that is, the training accuracy of the global model remains unchanged. It is assumed that from a global perspective, the updated model accuracy of the global model remains unchanged. For example, the local update of vehicles is regarded as the "raw material" of the global aggregation is represented by \( P_{11}, P_{21}, \cdots, P_{1i}, \cdots, P_{MM} \). Let \( P_1 \) and \( P_2 \) represent the overall accuracy of the EFL and FedAVG algorithms, respectively.

**Theorem 2.** At the moment \( S_2 \), under the EFL algorithm and the FedAVG algorithm, the total aggregation accuracy of the local training model can be received by the central server which remains unchanged, namely, \( P_1 = P_2 \).

**Proof.** When EFL and FedAVG participate in aggregation, the total model remains unchanged, that is, \( P_1 = P_2 \).

According to proof 1, \( S_1 \leq S_2 \). In the EFL algorithm, the most time-consuming local training model has been uploaded at \( S_2 \). The central server has received local training models from all clients. That is, \( P_1 = P_{11} + P_{21} + \cdots + P_{1i} + \cdots + P_{MM} \). In the FedAVG algorithm, when \( S_2 \), all local training models are uploaded together to the central server to participate in model aggregation. That is, \( P_2 = P_{11} + P_{21} + \cdots + P_{1i} + \cdots + P_{MM} \). The total amount of training received by the central server for the local training model remains unchanged; i.e., the overall accuracy of model aggregation remains unchanged. Thus, Theorem 2 is proved.

Theorem 2 shows that the overall accuracy of model aggregation remains unchanged. That is, the overall accuracy of two algorithms for global model aggregation is unchanged. \( \square \)

5.3. Security Analysis. Security means that vehicles in the EFL algorithm can protect the original data by uploading local updates without data out of the local area and protect the privacy of vehicle data from leakage. Based on the existing work, it is known that in the federated learning training process, the client can learn the original data of other clients according to the gradient information of those other clients [52]. This subsection mainly analyzes the impact of the increasing number of vehicles on the strength of privacy protection. The performance changes in different algorithms are analyzed when malicious nodes exist in the client group. We assume that the privacy protection of vehicle data is expressed as \( y = 1 - (1/x - 1) \cdot L \), where \( y \) is the strength of privacy protection, \( x \) is the number of vehicles, and \( L \) is the difficulty coefficient for the client to deduce the original data according to the global update. Thus, we have the following theorem.

**Theorem 3.** Under the federated learning in IoV scenarios, the strength of privacy protection of vehicle data can be enhanced.

**Proof.** In an IoV scenario, when the data between vehicles are trained by directly exchanging original data information, the privacy of vehicle data will be difficult to protect. In other words, privacy protection strength \( y = 0 \). When the data between vehicles are not exchanged, each vehicle is considered a "data island." Although it can protect the privacy of vehicle data, it will not achieve the purpose of common training, namely, that \( y = 1 \). The vehicle updates the local training model to RSUs by encryption. The difficulty for malicious nodes to learn the original data of other vehicles through global updating increases exponentially. The specific formula is as follows:

\[
y = 1 - \frac{1}{x - 1} \cdot L.
\]

When \( x \in [2, 5] \), \( L = \left(2^x - 1\right)^2 \). When \( x \in [5, +\infty) \), \( L = 1 \). Obviously, \( 0 < y = 1 - \left(1/x - 1\right) \cdot L < 1 \), federated learning technology is applied to IoV. Vehicles can participate in training together. The strength of vehicle data privacy protection is enhanced. Thus, Theorem 3 is proved. \( \square \)

**Lemma 4.** We set \( E_1 + E_2 = K \). Among them, \( E_1 \) and \( E_2 \) are known numbers; \( a \) is a positive number less than 1. \( K \) is an arbitrary constant greater than 0. Then, one has

\[
E_1 \cdot x + E_2 \cdot a = K.
\]

In accordance with the above formula, we can additionally conclude that

\[
x = 1 + \frac{E_2}{E_1} \cdot a.
\]

This formula is then applied to the proof of Theorem 5.

**Theorem 5.** When the vehicle queue contains malicious nodes, the EFL algorithm has better performance than the FedAVG algorithm, namely, \( H_1 \geq H_2 \), where \( H_1 \) and \( H_2 \) are the ability to cope with malicious nodes under the EFL and FedAVG algorithms, respectively.

**Proof.** We suppose that the contribution of the effective model updating of vehicle local training to global model aggregation is 1. The contribution of the disturbed model updating to global model aggregation is \(-1\) of \( B \) interference model updating, and \( B < M \). \( B \) is the total number of malicious nodes included in the vehicle queue. The following formulas can be obtained by combining Lemma 4:

\[
E_1 \cdot x + E_2 \cdot a = K.
\]
6. Performance Evaluation

6.1. Simulation Settings. In this section, simulation results demonstrate the effectiveness of the proposed EFL algorithm. For the learning evaluation part, we evaluate data classification on real-world datasets. The MNIST dataset [53] consists of 60,000 training examples and 10,000 test examples. The simulation experiment is run on the server with the following parameters: Lenovo Savior Ren 9000K, CPU 1910900K @ 3.70 GHz, memory RAM128G, graphics GPU RTX3090 24G, and software Visual Studio Code.

In the simulation, the EFL algorithm compares the FedAvg and FedSDG algorithms, respectively. The vehicles are set to 100, 200, and 500, respectively, to complete the training tasks of the CNN model and the 2NN model in IID data and NONIID data to test the program's performance under different conditions.

According to Algorithm 1, the vehicle performs single local training locally. It uploads the model updates obtained from the training to the RSUs for agile aggregation. In the simulations, the relationship between the learning efficiency of the three algorithms (EFL, FedAVG, and FedSDG) per unit time with the participation of different vehicles is reported. In order to calculate this point, the paper takes into account the experiment of using vehicles under the same learning ability conditions. It is known that in real-time data training, the accuracy rate of task results has obvious advantages over traditional federated learning algorithms. Based on the previous work, it can be seen that FedAVG works best when the number of local iterations is 5. Therefore, we set $m = 5$ in our simulations.

6.2. Results. The effectiveness of training is quantified by the time required for different iterations of the vehicle. The effectiveness of the algorithm training is estimated in Figure 4. Four groups of experiments were used for comparison. Figures 4(a)–4(d) are used to show the results of the CNN model trained on the NONIID dataset, 2NN model trained on the NONIID dataset, CNN model trained on the IID dataset, and 2NN model trained on the IID dataset, respectively. The horizontal coordinate is the number of iterations involved in training, and the vertical coordinate is the time accumulated by all vehicles participating in the training of the corresponding iteration rounds. Both are processed with $\log_2$ to facilitate a more intuitive observation of the performance comparison of different algorithms in the simulation. It can be seen that the EFL algorithm represented by the red lines consumes the least time when setting the same parameters. It means that the EFL algorithm is superior to the FedAvg algorithm and the FedSDG algorithm when evaluating the efficiency of federal learning training involving vehicles and RSUs. It can also be noted that with the significant increase in the number of vehicles, the gap between the red line and the other two color lines in the vertical direction becomes larger; that is, with the continuous increase in the number of vehicles participating in training, it can better reflect the EFL algorithm. The gap between the total time corresponding to the FedAvg algorithm and the total time corresponding to the FedSDG algorithm is expanding. In addition, it should be noted that the current federal learning method applied in the field of vehicle networking focuses on nondynamic data processing, and its core is still the application of the FedAvg algorithm. The efficiency of the EFL algorithm is still the best for dealing with dynamic data processing. Therefore, it is concluded that EFL can better guarantee user requirements for real-time data training in IoV training scenarios. The experimental results validate the correctness of Theorem 1.

The experiment considered the factor of vehicle mobility. The efficiency evaluation of algorithm training when vehicles switch between different roadside units is shown in the figure. Figure 5 evaluates the efficiency of algorithm training when considering vehicle mobility. Six groups of experiments are compared. Figures 5(a), 5(c), and 5(e) show the results of training a 2NN model on NONIID datasets with 50, 100, and 200 iterations. Figures 5(b), 5(d), and 5(f) show the results of training a 2NN model on IID datasets with 50, 100, and 200 iterations. The horizontal coordinate is the number of iterations involved in training, and the vertical coordinate is the accumulated time of all vehicles participating in the corresponding iteration rounds. The accumulated time of vehicles is processed using $\log_2$ for a more intuitive comparison of the performance of different algorithms in the simulation. It can be observed that when the same parameters are set, the EFL algorithm represented by the red line consumes less time than the FedAvg algorithm represented by the green line. Although the EFL algorithm represented by the red line consumes more time than the FedSDG algorithm represented by the blue line, there will be a significant improvement in accuracy, which will be described in detail later. This means that when evaluating the efficiency of federated learning training involving vehicles and RSUs, the EFL algorithm is superior to the FedAvg algorithm.

Model training is performed based on the NONIID dataset, and the accuracy is shown in Figure 6. The horizontal coordinate is the number of iterations involved in training, and the vertical coordinate is the accuracy of the model obtained by training for image data prediction. Through six sets of simulation experiments, it can be seen that when using the NONIID dataset for model training, the
red line indicates that the accuracy of the EFL algorithm is slightly higher than that of the FedAvg algorithm and significantly higher than that of the FedSGD algorithm. Even if the number of vehicles increases, the accuracy of the EFL algorithm is still high.

Model training is performed based on the IID dataset, and the accuracy is shown in Figure 7. The horizontal coordinate is the number of iterations involved in training, and the vertical coordinate is the accuracy of the model obtained by training for image data prediction. Through six sets of
Figure 5: Continued.
Figure 5: When the vehicle switches between different RSUs, the cumulative time of vehicles varies with the number of iterations. (a, c, e) The learning performance test of training 2NN models using the NONIID dataset when the number of iterations is 50, 100, and 200, respectively. (b, d, f) The learning performance test of training 2NN models using the IID dataset when the number of iterations is 50, 100, and 200, respectively.

Figure 6: Continued.
simulation experiments, it can be seen that when using the IID dataset for model training, the accuracy of the EFL algorithm represented by the red lines is slightly higher than that of the FedAvg algorithm and significantly higher than that of the FedSGD algorithm. Even if the number of vehicles increases, the accuracy of the EFL algorithm is still high. The overall trend is strikingly consistent with Figure 6. It can be concluded that the accuracy of the EFL algorithm is slightly higher than that of the FedAvg algorithm and significantly higher than that of the FedSGD algorithm under the same number of iterations when the model is trained on the NONIID and IID datasets. Because federal learning is applied to the field of vehicle networking, it pays more attention to efficiency than to accuracy, which is enough to prove that the EFL algorithm performs well in the field of vehicle networking. The experimental results validate the correctness of Theorem 2.

The study considered the mobility factor of vehicles and evaluated the accuracy of algorithm training when vehicles switch between different roadside units. Figure 8 evaluates the accuracy of algorithm training when considering the mobility of vehicles. Six sets of experiments were compared. Figures 8(a), 8(c), and 8(e) show the results of training the 2NN model using the NONIID dataset with 50, 100, and 200 iterations, respectively. Figures 8(b), 8(d), and 8(f) show the results of training the 2NN model using the IID dataset with
Figure 7: Continued.
Figure 7: Within a single RSU region using the IID dataset, accuracy changes with the number of iterations. (a, c, e) The learning performance test of training CNN models using the IID dataset when the number of iterations is 100, 200, and 500, respectively. (b, d, f) The learning performance test of training 2NN models using the IID dataset when the number of iterations is 100, 200, and 500, respectively.

Figure 8: Continued.
50, 100, and 200 iterations, respectively. The overall effect of training the CNN model and the 2NN model is consistent, so there is no further description here. The graph shows that the EFL algorithm has slightly higher accuracy than the FedAvg algorithm and significantly higher accuracy than the FedSGD algorithm when vehicles switch between different roadside units. Even with an increase in the number of vehicles, the accuracy of the EFL algorithm remains very high. Therefore, it can be concluded that, at the same number of iterations, the EFL algorithm has slightly higher accuracy than the FedAvg algorithm and significantly higher accuracy than the FedSGD algorithm, demonstrating the good overall performance of the EFL algorithm in the field of vehicular networks. This experimental result also further verifies the correctness of Theorem 2. The experimental results indicate that the application of the EFL algorithm has the potential to further improve the efficiency of federated learning in the field of vehicular networks.

The performance of data privacy in the simulations is shown in Figure 9. Using federated learning technology to protect the privacy and security of vehicle data can greatly reduce the probability of privacy disclosure and better protect the privacy of vehicle data. It can also be observed
that with the increasing number of vehicles participating in training, the privacy of vehicle data will be better protected. If the source data are located directly between the vehicles participating in training, the privacy protection force $y = 0$. If data are not transmitted between vehicles, the goal of joint training between vehicles cannot be achieved. The degree of privacy protection $y = 1$. Vehicles participate in training tasks through combined training, which can protect information about vehicle data. When the number of vehicles increases, the power of privacy protection will also increase. The experimental results validate the correctness of Theorem 3.

7. Conclusions

It is an important topic for research on federated learning in IoV. Traditional federated learning has efficiency limitations when it comes to protecting the privacy of high-dynamic vehicles. Aiming at the high-dynamic data scenario of IoV, a multistep federated learning architecture is proposed by including task release, vehicle model training, identity authentication, quality assessment, and weighted aggregation processes. A novel, agile aggregation model is introduced. RSUs perform the tasks of identity authentication, quality detection, and agile aggregation. Vehicles perform the task of model training. Data privacy is protected by only sharing model updates. Then, an efficient federated learning algorithm (EFL) is proposed using an agile aggregation model to improve the training efficiency of the vehicle-learning model for IoV. We theoretically analyze the advantages of EFL in terms of efficiency, training accuracy, and security over the FedAvg algorithm. Based on real datasets, simulations are conducted to verify the correctness of the theoretical results and demonstrate the effectiveness of the proposed algorithm.

Data Availability

The dataset used in this paper is publicly available on the Internet: MNIST (http://yann.lecun.com/exdb/mnist/).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was funded in part by the Major Science and Technology Special Project of Henan Province, China (grant no. 201300210400), the Key R&D and Promotion Special Project of Henan Province, China (grant nos. 222102210133 and 222102210055), the China Postdoctoral Science Foundation (grant no. 2020M672217), the Key Scientific Research Project of Colleges and Universities in Henan Province, China (grant no. 21AS20003), and the Elite Postgraduate Students Program of Henan University, China (grant nos. SYLYC2022149 and SYLYC2022147).

References


