electronics

MDPI

Article

P2P Federated Learning Based on Node Segmentation with Privacy Protection for IoV

Jia Zhao [1,2], Yating Guo [1,2,*], Bokai Yang [1,2] and Yanchun Wang [1,2]

[1] Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China; zhaojia@bjtu.edu.cn (J.Z.)
[2] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
* Correspondence: 21120471@bjtu.edu.cn

Abstract: The current usage of federated learning in applications relies on the existence of servers. To address the inability to conduct federated learning for IoV (Internet of Vehicles) applications in serverless areas, a P2P (peer-to-peer) architecture for federated learning is proposed in this paper. Following node segmentation based on limited subgraph diameters, an edge aggregation mode is employed to propagate models inwardly, and a mode for propagating the model inward to the C-node (center node) while aggregating is proposed. Simultaneously, a personalized differential privacy scheme was designed under this architecture. Through experimentation and verification, the approach proposed in this paper demonstrates the combination of both security and usability.

Keywords: federated learning; IoV; differential privacy; P2P

1. Introduction

IoV (Internet of Vehicles) applications often require the utilization of big data for model training, which inevitably raises concerns about user privacy [1]. Therefore, employing a privacy-preserving algorithm becomes imperative. Federated learning [2,3] emerged as a privacy-preserving paradigm commonly utilized in machine learning. However, the prevalent C-S (client-server) architecture presents certain limitations, notably, the requirement for server involvement. This limitation restricts the application of federated learning in regions where servers are unavailable or not readily deployable, hindering its adoption for IoT applications in such areas. An alternative federated learning architecture, P2P (peer-to-peer), offers a solution to this challenge. However, P2P architectures are more complex, and most of the research on federated learning for P2P architectures is based on a complete node distribution network. The implementation of federated learning for P2P architectures in the context of vehicular networking must also consider the issue of the distance between vehicles, which need to be sufficiently close to each other to communicate and exchange model parameters. Therefore, it can be beneficial to group the vehicle nodes in proximity based on distance before federated learning, ensuring that vehicle nodes at each position in the node distribution network graph can form a P2P architecture with nearby vehicle nodes. Additionally, the system should be designed to be straightforward and efficient to implement, facilitating convenient application in real-world scenarios.

Federated learning, despite its privacy-preserving benefits, has certain security risks, such as inference attacks and poisoning attacks. Common privacy computation methods used to enhance the security of federated learning include encrypting the model or using differential privacy by adding noise. In the context of IoV, model parameters are interchanged between vehicle nodes in a P2P architecture, which introduces the possibility of inferring certain privacy information from the received model parameters if a malicious node is present. The use of differential privacy reduces this risk. However, the addition of noise affects the quality of the model. Malicious nodes often choose nearby nodes for inference attacks, and the risk of privacy information leakage decreases as the distance

check for updates

Citation: Zhao, J.; Guo, Y.; Yang, B.; Wang, Y. P2P Federated Learning Based on Node Segmentation with Privacy Protection for IoV. *Electronics* **2024**, *13*, 2276. https://doi.org/10.3390/electronics13122276

Academic Editor: Felipe Jiménez

Received: 29 March 2024
Revised: 20 May 2024
Accepted: 27 May 2024
Published: 10 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

between nodes increases. Therefore, the degree of differential privacy can be adjusted according to the geographic distance between nodes. When the distance between nodes is short, the risk of inference attacks is higher, necessitating the addition of more noise to the model parameters for protection. Conversely, when the distance between nodes is greater, the risk is reduced, allowing for less noise addition and a greater focus on maintaining model quality.

Therefore, to extend the provision of federated learning services to a broader customer base, employing federated learning with a P2P architecture can address the serverless scenario. However, the realization of P2P architecture federated learning on a complete node distribution graph is not suitable for IoV. We can consider grouping nodes in proximity based on distance, which can be used to partition the complete graph into subgraphs as the basis for grouping before federated learning. The data security of IoV applications is very important. To prevent vehicles participating in federated learning from inferring private data from the received model parameters, each vehicle should add noise to the propagation model to realize the differential privacy mechanism. Furthermore, the degree of differential privacy can be adjusted based on distance: when the distance between nodes is greater, the risk of data leakage is smaller, allowing more focus on the quality of the model. Conversely, when nodes are closer together, the risk of data leakage increases, necessitating the addition of more noise to the model parameters to enhance privacy protection.

This paper introduces a federated learning scheme specifically designed for the IoV within a peer-to-peer architecture. In utilizing the minimum spanning tree algorithm and centrality algorithm, the node distribution graph is segmented to control the diameter of the subgraph. This segmentation serves as the foundation for node grouping before federated learning, and the propagation process of federated learning was designed to align with the topology within the subgraph. To enhance the privacy protection capabilities of federated learning, a personalized differential privacy scheme is introduced based on distance adjustment. This scheme enables nodes to dynamically adapt the degree of differential privacy according to their environmental context.

The primary contributions include the following:

- An F-Prim algorithm, derived from Prim's algorithm and centrality algorithm, was devised to group nodes based on proximity while constraining the diameters of the subgraphs, thereby forming a P2P architecture.
- The propagation path of models within the P2P architecture is designed according to the node hierarchy, wherein nodes propagate from the periphery to the core of the subgraph for aggregation, facilitating the completion of the model aggregation process at the C-node (central node).
- A personalized differential privacy scheme was formulated, enabling each node to adjust the amount of noise added to the model parameters based on its distance from other nodes. This scheme aims to strike a balance between security and model quality.

The remainder of this article is structured as follows. In Section 2, existing studies that deployed P2P-based federated learning in vehicle networks are reviewed. Section 3 introduces the application scenarios of the scheme and the P2P architecture formed after node segmentation. In Section 4, the federated learning process and its privacy preserving scheme under P2P architecture are elaborated on. In Section 5, some experiments based on the scheme are designed to verify the proposed architecture. Section 6 show relevant proofs of some of the programs. Finally, Section 7 concludes this article.

A summary of the main notations is provided in Table 1.

Table 1. Main symbols.

Symbol	Meaning
$\hat{N}$	The maximum number of nodes in a single subgraph
$\lvert\cdot\rvert$	The cardinality of a set
v_i	The i-th node
D_i	The database held by the owner v_i
e_{ij}	The edge between v_i and v_j
d_{ij}	The distance between v_i and v_j
$d_{\max}$	The maximum of distance in G
$\hat{d}_i$	The maximum of the distance between v_i and v_i's neighbor node
l_{ij}	The minimum length between v_i and v_j
$center$	The ID of the C-node
$C_h(i)$	The value of the harmonic centrality of v_i
L_1	The set of one-layer node IDs
L_2	The set of two-layer node IDs
t	The index of the t-th aggregation
T	The number of aggregation times
ω	The vector of model parameters
ω^0	Initial parameters
ω_i^t	The local training parameters of the i-th node at the t-th aggregation
$\tilde{\omega}_i^t$	Local training parameters ω_i^t with noise n_i
$\omega_{v_i}^t$	Aggregated parameters on v_i
$n(\epsilon, \sigma, \triangle f)$	Gaussian noise function
n_i	The noise added by v_i
$\triangle f_i$	The sensitivity of v_i
σ	Sigma
C	Clipping threshold

2. Related Work

Federated learning research has been widely applied in the field of IoV. As a privacy paradigm for machine learning, federated learning helps to address the privacy issues of sensitive information such as of path and location in IoV applications without affecting the model training on IoV data. Samarakoon et al. [4] investigated the joint power and resource allocation problem for ultra-reliable and low-latency communication in vehicular networks and used Lyapunov optimization to derive a joint power and resource allocation strategy with ultra-reliable and low-latency communication in a distributed approach as well. Kong et al. [5] proposed a federated learning-based cooperative vehicle localization system, which takes full advantage of the Internet of Things and the potential of collaborative edge computing to ensure user privacy while providing highly accurate localization corrections.

Architectures for federated learning contain both C-S and P2P, and most of the research has focused on the C-S architecture. In IoV, V2V(Vehicle-to-Vehicle) communication is also possible, which allows P2P-based federated learning to be considered in IoV scenarios. Yuan et al. [6] introduced a novel framework named FedPC aimed at tackling driver privacy concerns stemming from in-cabin cameras in NDAR. FedPC employs a peer-to-peer federated learning approach coupled with continual learning to ensure privacy, enhance learning efficiency, and reduce communication, computational, and storage overheads. Barbieri et al. [7] investigated decentralized federated learning methods to enhance road user/object classification based on LiDAR data in smart connected vehicles. They proposed a consensus-driven FLapproach facilitating the collaborative training of deep ML models among vehicles by sharing model parameters via V2X links, eliminating the need for a central parameter server.

Privacy and security in IoV applications have been a major concern. In the domain of privacy-preserving federated learning within P2P networks under IoV, several noteworthy methodologies have emerged to address challenges in security, privacy preservation, and robustness. Lu et al. [8] proposed an asynchronous federated learning scheme for resource sharing in vehicular IoT. They used a local differential privacy technique to protect

the privacy of local updates. An asynchronous approach was employed with FL to enable distributed peer-to-peer model updates between vehicles, which is more suitable for a decentralized vehicular network. Chen et al. [9] proposed a novel decentralized federated learning method called BDFL (Byzantine Fault-Tolerance Decentralized Federated Learning), which combines Byzantine fault-tolerance mechanisms and privacy-preserving techniques to address security and privacy challenges. This method utilizes a P2P FL architecture and the HydRand protocol to establish a robust and fault-tolerant FL environment and adopts a PVSS (publicly verifiable secret-sharing) scheme to protect the privacy of autonomous vehicle models.

It is evident that federated learning with a P2P architecture is well suited for the connected car environment. However, the complexity of the P2P architecture poses challenges compared to the C-S architecture. Moreover, most existing studies on federated learning in IoV with a P2P architecture are based on a complete node distribution network. In real-world environments, the changing distances between nodes may lead to connection distances that are too far to facilitate the federated learning process effectively. Therefore, in this paper, a novel approach is proposed. A method is designed to partition the node distribution network graph and group nearby nodes into the same subgraph as the basis for federated learning. Subsequently, the federated learning process is deployed, and its associated privacy protection scheme is used in this framework to address the challenges posed by real-world IoV environments.

3. System Model

In this section, the operation of a P2P-based IoV scenario is described. And the F-Prim (Finite-length Prim) algorithm designed in this study is proposed for partitioning the node distribution graph to realize the P2P federated learning architecture.

3.1. P2P-Based IoV Scenarios

In the study, an IoV scenario was established to offer federated learning services within a server-less urban environment, as illustrated in Figure 1. In areas devoid of servers, solely vehicle terminals are observable, forming the entirety of the vehicle network. Each terminal is endowed with computational and storage capacities and is adept at communicating with other terminals.

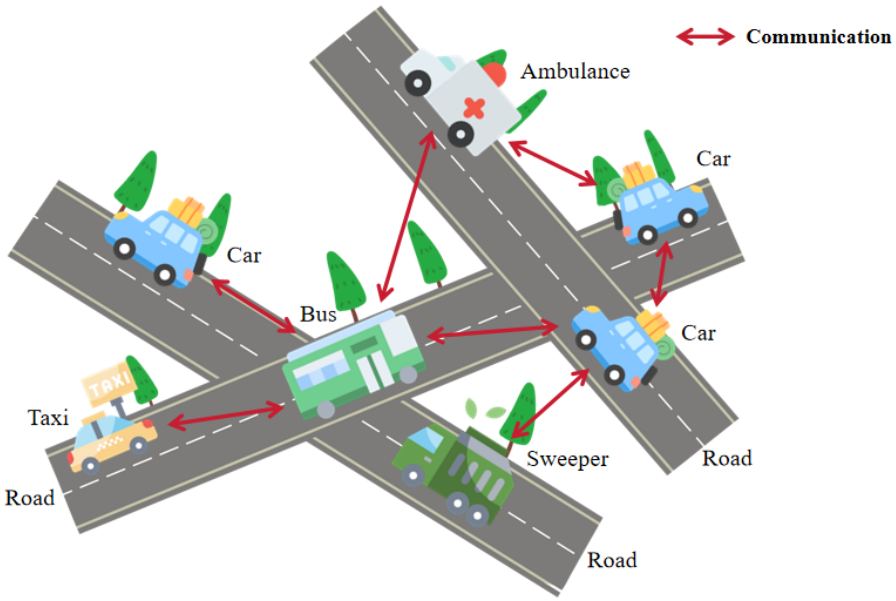


Figure 1. IoV scenario.

To represent the distribution of vehicle terminals, a terminal node network graph was constructed: $G = \{v, e, d \mid v \in V, e \in E, |V| = N\}$, as illustrated in Figure 2. In this

graph G, $V = \{v_1, v_2, \ldots, v_N\}$ denotes the set of vehicle terminal nodes. Each terminal v_i is associated with a local dataset D_i. If terminals v_i and v_j can communicate with each other, there exists an edge $e_{ij} \in E$ between the corresponding nodes, and the weight d_{ij} of edge e_{ij} represents the geographic distance between the two nodes. However, connections may become unstable if the distance between two terminals is too large. Therefore, consideration is given to two terminals capable of communication if the distance between them is less than $d_{\max}$. In summary, the weights of edges e_{ij} on the node distribution network graph are represented as $|d_{ij}| \in (0, d_{\max}]$.

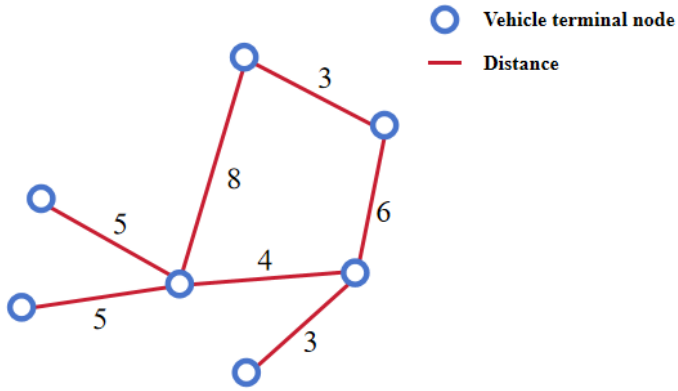


Figure 2. Node mapping network graph.

3.2. Node Segmentation to Build P2P-Based Federated Learning Architecture

In federated learning with a C-S architecture, clients typically prefer selecting nearby servers to serve themselves [10]. However, in serverless P2P architectures, which exhibit more complex topologies, special algorithms are required to determine the grouping method of nodes that are distributed in close proximity to each other. Additionally, in the context of IoV where vehicle locations are dynamic, grouping methods with too many conditions may become obsolete as node distributions change rapidly. Therefore, there is a need for a grouping algorithm with fewer conditions and reduced computational complexity to facilitate P2P node grouping in IoV scenarios.

In the architecture of IoV, the node distribution graph consists of nodes, edges, and weights. The algorithm designed for proximity grouping aims to minimize edge weights mapped to geographic location distributions. Thus, an algorithm from graph theory that minimizes the sum of edge weights while finding nodes with the shortest distances is desirable. The Prim algorithm, a classical algorithm for finding a minimum spanning tree in weighted connected graphs, serves this purpose. This algorithm retrieves the generated tree that minimizes the sum of edge weights, thereby facilitating the efficient proximity grouping of nodes in IoV scenarios.

Remark 1. *In IoV applications, the primary goal is to group individual nodes rather than connections. Prim's algorithm and Kruskal's algorithm are both classic approaches for generating minimum spanning trees. However, Prim's algorithm, which selects nodes based on the shortest distances from the current spanning tree, aligns more closely with the objective of node grouping in IoV scenarios. Therefore, it is more suitable as a basis for improvement in this context.*

The core of the Prim algorithm lies in finding the shortest pathways among all nodes. When directly applied to grouping, it tends to include all nodes in a single group. To ensure effective grouping, a threshold value $\hat{N}$ for the number of nodes in each group is necessary. If the group size surpasses $\hat{N}$, the group becomes saturated, signaling the end of grouping, and a new grouping process starts from another node.

The Prim algorithm, although effective at grouping nodes, does not inherently limit the diameters of subgraphs. In the context of federated learning in P2P architectures, excessively long subgraph diameters can lead to extended propagation chains between nodes,

resulting in prolonged propagation times unsuitable for vehicular networking. To address this, the Prim algorithm requires modifications to restrict the diameters of subgraphs.

To this end, this paper introduces the F-Prim algorithm, an adaptation of the Prim algorithm. In the C-S architecture, servers are designated as centers, and a C-node is established within each subgroup formed by grouping. This facilitates controlling the distances between remaining nodes and the C-node to limit the diameters of subgraphs. Ultimately, the number of nodes within each subgraph does not exceed $\hat{N}$.

Graph centrality serves as a vital metric in complex network analysis, quantifying the importance or influence of nodes in a graph. Various centrality algorithms exist, but for selecting a C-node to restrict the subgraph diameter, centrality algorithms related to shortest distances, such as harmonic centrality and betweenness centrality, are relevant. Their definitions are as follows.

Definition 1 (Harmonic Centrality)**.** *Harmonic centrality [11] is a metric used in network analysis to assess the importance of a node within a network. A higher harmonic centrality indicates that the node has shorter average distances than other nodes, implying greater influence within the network.*

$$C_h(i) = \sum_{j=1}^{N} \frac{1}{l(i,j)}, \tag{1}$$

Definition 2 (Betweenness centrality)**.** *Betweenness centrality [12,13] quantifies a node's importance by measuring how often it lies on the shortest paths between pairs of other nodes. It reflects a node's potential control over information or resource flow in the network.*

$$C_b(i) = \sum_{s \neq i \neq t} \frac{\beta_{st}(i)}{\beta_{st}} \tag{2}$$

where β_{st} represents the total number of shortest paths from node s to node t, and $\beta_{st}(i)$ denotes the number of those paths that pass through node i.

Given the constraints of IoV applications, harmonic centrality emerges as the more suitable choice for centrality computation due to its independence from knowledge of the entire graph. Unlike betweenness centrality, which requires information about all shortest paths between nodes, harmonic centrality only necessitates knowledge of distances from a node to others within its vicinity. This makes it compatible with P2P architectures where nodes have limited access to information beyond their local neighborhood.

Therefore, the F-Prim algorithm adopts harmonic centrality to identify the C-node of the subgraph. The algorithm aims to limit the subgraph diameter, thereby constraining the maximum propagation distance between nodes during model dissemination. By focusing on propagation length rather than physical distance between nodes, it better aligns with the dynamic nature of IoV applications. Using propagation length simplifies node segmentation, as nodes only need to assess connections with neighboring nodes without requiring stable distance information, which is impractical for vehicles whose positions change frequently.

To select the C-node accurately, the algorithm computes the harmonic centrality ($C_h(i)$) for each node within the subgraph as new nodes are added. According to the definition of harmonic centrality, the node with the highest centrality becomes the new C-node. With a predetermined upper limit of subgraph diameter set to 4, the algorithm ensures that the minimum propagation length from each newly added node to the C-node remains within two hops or less.

The complete procedure of the algorithm is delineated in Algorithm 1.

Algorithm 1: F-Prim Algorithm(G, V, d)

Input: A graph G, the set of nodes V in G, a matrix d representing the distances between nodes

```
sgNum ← 0;
while |V| ≠ 0 do
    for i ← 1 to |V| do
        disSG[i] ← ∞;
    end
    select v_r as root node;
    disSG[r] ← 0;
    center ← r;
    while True do
        minDis ← ∞;
        for v_i ∈ V do
            if dis[i] < minDis & l(center, i) ≤ 2 then
                minDis ← dis[i];
                rec ← i;
            end
        end
        for v_u ∈ G.Adj[v_rec] do
            if w(rec, u) < dis[u] then
                dis[u] ← w(rec, u);
            end
        end
        SG[sgNum].add(v_rec);
        V = V − {v_rec};
        center ← SG[sgNum].harmCenter;
        if |SG[sgNum]| ≥ K̂ then
            break
        end
    end
    sgNum ← sgNum + 1;
end
```

4. P2P-Based Federated Learning Process with Its Privacy Protection

In this study, the federal learning process and its privacy computation scheme were designed based on the P2P architecture proposed above.

4.1. P2P-Based Federated Learning Process

After partitioning the node graph, each subgraph is limited to a diameter of 4, and the distance from any node to the C-node is at most 2. Nodes are then organized into layers based on their distance from the C-node. Nodes at a distance of two hops from the C-node are labeled as two-layer nodes (set L_2), while those at a distance of one hop are labeled as one-layer nodes (set L_1). The C-node constitutes a separate layer. This layering facilitates efficient communication within the subgraph during federated learning.

4.1.1. Path Selection

In graph theory, directly connected edges do not always represent the shortest distance paths. In the hierarchical division discussed here, a two-layer node must pass through a one-layer node to propagate the model to the C-node for aggregation. However, a one-layer node connected to the center may also be connected to other one-layer nodes. Choosing the shortest distance path needs to consider the actual IoV application environment.

As illustrated in Figure 3, where the C is the central node, A and B are one-layer nodes, and a, b, and c represent geographical distances from B to the C-node, A to B, and A to the C-node, respectively. According to the triangle inequality theorem, $a + b > c$, in the actual geographic distance, the path from each node to the C-node via the minimum number of hops, i.e., the minimum length, must also be the path of the shortest distance. This means that the path from A to the C-node via the C-node directly represents the shortest distance path (Path ②) rather than the path via B first (Path ①).

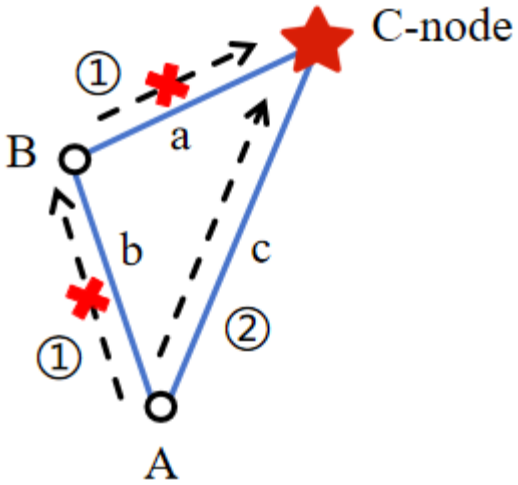


Figure 3. Path selection.

4.1.2. Aggregated Simultaneous Transmission

The federated learning subgraph topology of the P2P model forms a complex, multi-layered structure, requiring the careful consideration of propagation and aggregation operations within this topology.

In the general P2P model of federated learning, there are two potential approaches for propagating aggregation. The first involves each node exchanging locally trained models with neighboring nodes, resulting in potentially different global models on each node. The second approach entails finding a path in the topology where each node along the path can aggregate all models to obtain a global model on a specific node. In this study, with a C-node present in the subgraph itself, propagation and aggregation were designed based on the second method.

Following the hierarchical structure of the federated learning subgraph in the P2P model, propagation occurs from the inside out, starting from two-layer nodes, then moving inward through one-layer nodes, and finally reaching the C-node for aggregation.

In this propagation mode, all nodes initially train their local models. Once training is complete, two-layer nodes transmit their local models to all connected one-layer nodes. If network failure prevents global model aggregation at the C-node, models aggregated at one-layer nodes can temporarily replace global models to serve surrounding nodes. Therefore, one-layer nodes propagate to two-layer nodes while passing their local models to connected one-layer nodes, enhancing model availability.

Increased aggregations expedite model convergence. Consequently, after receiving all models from connected two-layer and one-layer nodes, a one-layer node aggregates them to lighten the C-node's workload and reduce the propagation time within the subgraph. After all one-layer nodes perform aggregation, they transmit models to the C-node. The C-node conducts a final aggregation by combining models from all one-layer nodes with locally trained models, resulting in the global model within the P2P federated learning subgraph. This transmission approach is termed AST (Aggregated Simultaneous Transmission).

4.1.3. Model Weight Adjustment

In the FedAvg algorithm, the weights are determined based on the data volume in the model, and the client c_i passes its local data volume $|D_i|$ to the server to compute its model weight. As shown in Equation (3), where p_i represents the weight of v_i's model ω_i in the global model ω,

$$p_i = \frac{|D_i|}{\sum_{k=1}^{K} |D_k|} \tag{3}$$

However, in this propagation mode, adjustments for the information on the amount of data transmitted are necessary to maintain fairness in the final model, as discussed in Section 6.1.

After calculations, the nodes pass the models while transmitting the amount of dataset $|D_i|_\alpha$ to calculate the model weights:

$$|D_i|_\alpha = \frac{|D_i|}{\alpha_i} \tag{4}$$

where α_i is the following value:

$$\begin{cases} \alpha_i = \left| \sum\limits_{e_{ij} \in E} e_{ij} \right| & i \in L_2 \\ \alpha_i = \left| \sum\limits_{e_{ij} \in E} e_{ij} \right|, j \notin L_2 & i \in L_1 \\ \alpha_i = 1 & i = \text{center} \end{cases} \tag{5}$$

4.2. Personalized Differential Privacy

In the proposed scheme, differential privacy is employed to ensure privacy protection during the propagation of federated learning model parameters.

Definition 3 (Differential privacy [14])**.** *For a function $f : D \to \mathbb{R}^d$, the mechanism M satisfies (ε, δ)-differential privacy if for all adjacent datasets D and D' that differ in at most one element and for all measurable sets S in the output space,*

$$\Pr[M(f(D)) \in S] \leq e^{\epsilon} \Pr\left[M\left(f(D')\right) \in S\right] + \delta, \tag{6}$$

Gaussian noise is a type of noise characterized by a probability density function equivalent to a normal distribution, also known as a Gaussian distribution. In essence, it manifests as values distributed according to this specific distribution pattern.

According to the definition of differential privacy, when equations

$$\sigma \geq \frac{c\Delta f}{\varepsilon} \tag{7}$$

$$c^2 > 2 \ln \frac{1.25}{\delta} \tag{8}$$

are satisfied, the algorithm $M(f(D)) = f(D) + N\left(0, \delta^2\right)$ satisfies (ε, δ)-differential privacy, where $N\left(0, \delta^2\right)$ is a random vector sampled from the Gaussian distribution.

If choosing to add noise after local training to streamline privacy calculations, the training process of the vehicle terminal node v_i on the local dataset D_i during the t-th round of federal learning can be represented as

$$f_i \triangleq f(D_i) = \omega_i^t = \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} \arg\min F_i(\omega_i^{t-1}, D_{i,j}) \tag{9}$$

The sensitivity of the vehicle terminal node v_i is calculated based on this training process function. To streamline privacy calculations, Gaussian noise was chosen to be added after local training. In this case, the sensitivity of v_i [15] can be expressed as

$$\begin{aligned} \triangle f_i &= \max_{D_i, D_i'} \left\| f(D_i) - f(D_i') \right\| \\ &= \max_{D_i, D_i'} \left\| \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} \arg\min F_i(\omega_i^{t-1}, D_{i,j}) - \frac{1}{|D_i'|} \sum_{j=1}^{|D_i'|} \arg\min F_i(\omega_i^{t-1}, D_{i,j}') \right\| \\ &= \frac{2C}{|D_i|} \end{aligned} \tag{10}$$

So, when σ_i satisfies

$$\sigma_i \geq \frac{c\Delta f}{\varepsilon} \geq \frac{2C\sqrt{2\ln\left(\frac{1.25}{\delta}\right)}}{\varepsilon |D_i|} \tag{11}$$

the algorithm complies with the (ε, δ)-differential privacy mechanism.

Differential privacy was selected as the model protection algorithm primarily to address the risk of curious nodes inferring the dataset from received models. Compared to encryption or other algorithms requiring extensive privacy calculations, it offers a more suitable solution for environments requiring rapid responses in IoV.

During federated learning, noise added to the model is determined based on inter-node distance before the propagation of local models. This approach aims to add more noise when vehicle nodes are close, thereby reducing the risk of data leakage from nearby nodes. Conversely, less noise is added when nodes are farther apart, balancing privacy protection with model quality considerations.

However, each node is generally connected to multiple nodes, and the distances between these nodes are not equal. If the added noise is computed multiple times, it will necessitate additional privacy computations. Hence, the minimum distance $\hat{d}_i$ of the node's neighbors is selected, calculated using Equation (12), as the parameter to compute the value of ε. This ensures that the node only needs to compute the noise once, and irrespective of the node to which it is passed, it is guaranteed to satisfy (ε, δ)-differential privacy.

$$\hat{d}_i = \min\{d_{ij} | v_j \in SG[h]\}, i \neq center \tag{12}$$

The function $f(x) = \ln(x+1)$ is utilized to compute the ε value, where the function monotonically increases in the range $(0, \varepsilon_{\max}]$. Therefore, $0 < x \leq e^{\varepsilon_{max}} - 1$ in this function. In this design, the range of values for the distance between two nodes is $(0, d_{\max}]$, and the privacy budget on node v_i is given by

$$\varepsilon_i = \ln\left(\frac{\hat{d}_i}{\frac{d_{max}}{e^{\varepsilon_{max}}-1}} + 1\right) = \ln\left(\frac{(e^{\varepsilon_{max}}-1)\hat{d}_i}{d_{max}} + 1\right) \tag{13}$$

In this way, the noise added by node v_i after local training is

$$n_i = n(\varepsilon_i, \delta, \triangle f_i) = n(\ln\left(\frac{(e^{\varepsilon_{max}}-1)\hat{d}_i}{d_{max}} + 1\right), \delta, \frac{2C}{|D_i|}) \sim N(0, \sigma_i^2) \tag{14}$$

In accordance with Equations (8), (10), and (7), σ_i needs to satisfy

$$\sigma_i \geq \frac{2C\sqrt{2\ln\left(\frac{1.25}{\delta}\right)}}{\ln\left(\frac{(e^{\varepsilon_{max}}-1)\hat{d}_i}{d_{max}} + 1\right)|D_i|} \tag{15}$$

Based on Equations (14) and (15), it can be observed that the amount of added noise decreases as the distance between nodes increases.

The entire federated learning process with differential privacy is presented in Algorithm 2.

Algorithm 2: Federated learning process

for $t \leftarrow 1$ **to** T **do**
  for $i \leftarrow 1$ **to** N **do**
    if $t = 1$ **then**
      get ω^0
    end
    $\omega_i^t \leftarrow \mathrm{Train}\left(\omega^{t-1}, D_i\right)$
    if $i \neq center$ **then**
      $\omega_i^t \leftarrow \frac{\omega_i^t}{\max\left(1, \frac{\omega_i^t}{C}\right)}$
      $\tilde{\omega}_i^t \leftarrow \omega_i^t + n_i$
    end
  end
  for $i \in L_2$ **do**
    for $j \in L_1$ **do**
      if $e_{ij} \in E$ **then**
        $v_i \rightarrow v_j : \tilde{\omega}_i^t$
      end
    end
  end
  for $i \in L_1$ **do**
    for $j \in L_1$ **do**
      if $e_{ij} \in E$ **then**
        $v_i \rightarrow v_j : \tilde{\omega}_i^t$
      end
    end
    $\omega_{v_i}^t \leftarrow \mathrm{Agg}\left(\tilde{\omega}_i^t, \sum \tilde{\omega}_{L_1}^t, \sum \tilde{\omega}_{L_2}^t\right)$
    $v_i \rightarrow v_{center} : \omega_{v_i}^t$
  end
  $v_{center} : \omega^t \leftarrow \mathrm{Agg}\left(\omega_{center}^t, \sum \omega_{v_{L_1}}^t\right)$
  for $i \leftarrow 1$ **to** N **do**
    $v_{center} \rightarrow v_i : \omega^t$
  end
end

5. Experimentation and Analysis

In this study, the scheme presented in this paper was experimented with and analyzed.

Remark 2. *In the experiments conducted, it was observed that some subgraphs contain too few nodes. To better show the impact of node segmentation, subgraphs with insufficient nodes were split. The internal nodes of these split subgraphs were redistributed to the subgraphs closest to them, while any isolated nodes still remaining after the splitting process were discarded.*

Remark 3. *Because node partitioning generates multiple subgraphs, the results in the experiment were obtained by averaging the results of multiple subgraphs.*

5.1. Node Segmentation

The distribution of vehicle nodes at a given moment was simulated in two ways. The first method was implemented using the networkx package, while the second method utilized OpenStreetMap.

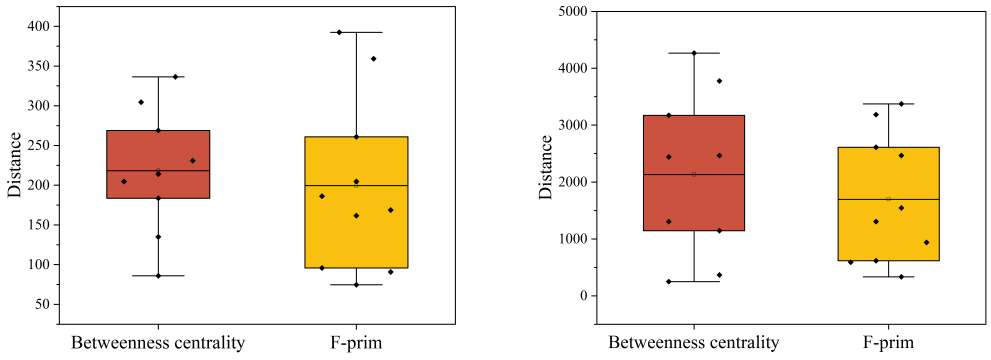
In the first approach, generating nodes with random horizontal and random vertical coordinates in a 100 × 100 square area, called an RC (randomized coordinate). In the second approach, nodes are randomly selected on the actual road map provided by OpenStreetMap, called MVN (moving vehicle nodes).

The sum of the shortest distances *cnodeDis* from the C-node to the remaining nodes $V - \{v_{center}\}$ within the subgraph is computed according to Equation (16) as a measure.

$$cnodeDis = \sum_{j=1, i=center}^{N} \min(\{d_{im} + d_{mj}, d_{ij}\}) \tag{16}$$

Under the two node simulation schemes, the total number of nodes $N = 160$ and the upper limit of the number of nodes within each subgraph $\hat{K} = 20$ were set to compute *cnodeDis*, for comparing the betweenness centrality and the F-Prim algorithm designed in this study.

As shown in Figure 4, the dots in it represent the data for each subplot, the line in the middle of the box represents the data mean, and the top and bottom edges represent the maximum and minimum values of the data. Because of the uneven distribution of nodes, the data distribution is more discrete between subgraphs, but most nodes are distributed around the mean. The difference between the values of the two centrality algorithms is not really significant, as nodes that are at more intersections of shortest paths within a better connected subgraph are also more likely to be nodes closer to the rest of the nodes within that subgraph, i.e., when a node's harmonic centrality is high, its harmonic centrality is likely to be relatively high as well. However, it is still possible to see from the mean median that F-Prim yields smaller *cnodeDis* values, i.e., the C-node computed under this algorithm is closer to the rest of the nodes. Moreover, compared to the RC simulation, the effect of F-Prim is a little more obvious under the MVN simulation, which may indicate that the algorithm is more effective on the actual vehicle distribution along the road, while it is weaker on the simulation with only a random distribution, implying that the algorithm is more suitable for practical IoV applications.



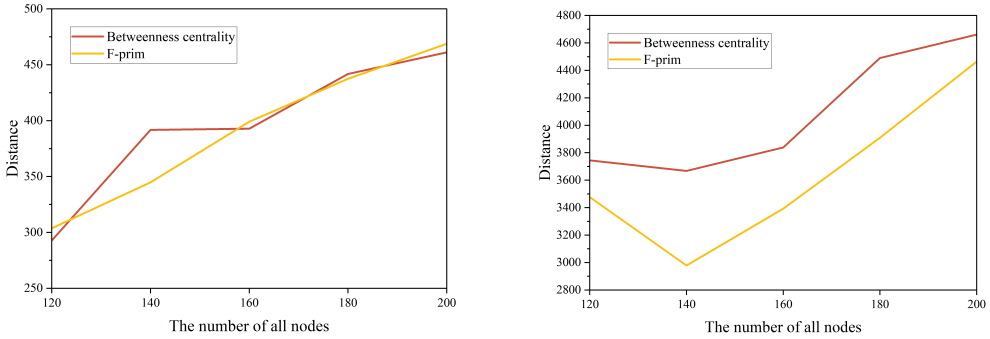
(**a**) RC (**b**) MVN

Figure 4. Comparison of centrality algorithms.

We fixed the upper limit of the number of nodes within each subgraph $\hat{K} = 20$, adjusted the value of the total number of nodes N to 120, 140, 160, 180, and 200, and compared the resulting *cnodeDis* values.

As indicated in Figure 5, the *cnodeDis* values of both algorithms exhibit an overall upward trend as the total number of nodes increases. However, this trend is not strictly linear and fluctuates, reflecting the randomness and unpredictability of node distribution.

Overall, the betweenness centrality utilized by the F-Prim algorithm yields better results in subgraph segmentation by positioning the C-node closer to other nodes. While the betweenness centrality algorithm may occasionally outperform other methods, its computation typically requires access to information about all nodes in the graph, rendering it unsuitable for P2P mode.



(**a**) RC (**b**) MVN

Figure 5. *cnodeDis*.

5.2. Aggregated Simultaneous Transmission

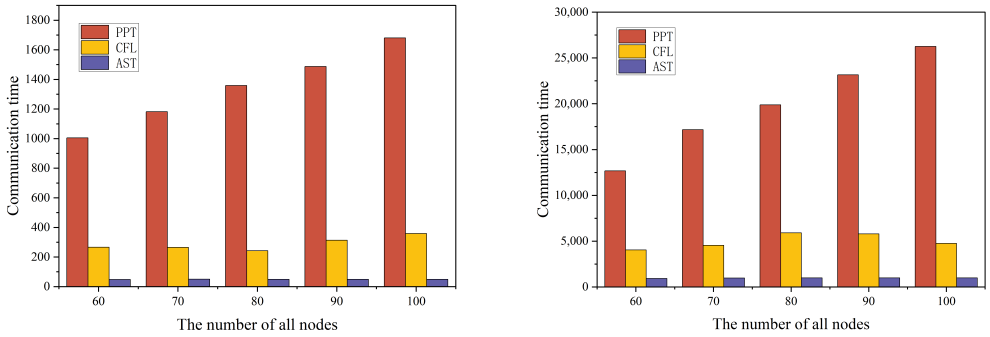
To demonstrate the accelerated intra-subgraph propagation process facilitated by AST in the IoV environment, a comparison is drawn with propagation patterns in the PPT algorithm [16] and the CFL algorithm [17]. Both algorithms are grounded in P2P-mode federated learning and employ a single-line propagation mode of depth-first traversal. In the PPT algorithm, depth-first traversal occurs within all nodes participating in federated learning. Each propagation involves a single aggregation, followed by a backtrack to identify unaggregated nodes after surrounding nodes have been aggregated. The CFL algorithm builds upon the PPT algorithm by implementing subclustering. It conducts depth-first traversal within a single cluster, resulting in more localized aggregation operations.

5.2.1. Communication Time

In the PPT and CFL algorithms, only the target node participates in federated learning, and in order to facilitate a comparison among the three propagation modes, all nodes were designated as target nodes to partake in the federated learning process in P2P mode.

Given that the model propagated in the three modes remains unchanged, the model parameters are set to propagate at an identical rate across all modes. Consequently, the total communication time for model propagation between subgraph nodes is primarily determined using the distance of the paths traversed by the model during propagation.

As illustrated in Figure 6, with the same propagation speed, the communication time of the cluster-medium propagation model is notably shorter compared to the propagation models of the PPT and CFL algorithms. The cluster-based propagation approach prioritizes efficiency by distributing nodes in layers based on clustering within the CFL algorithm. This enables subgraph propagation and interlayer model propagation to occur in parallel, minimizing serial transmission between nodes and reducing the overall communication time. In contrast, the propagation modes employed in the PPT and CFL algorithms are more suited for scenarios with fewer nodes and simpler topologies. They are less effective in environments with a large number of nodes and complex distribution and connectivity patterns. Overall, the cluster-medium propagation model demonstrates superior capability in obtaining the global model and promptly responding to application requirements by efficiently propagating under the distribution of vehicle networking nodes.



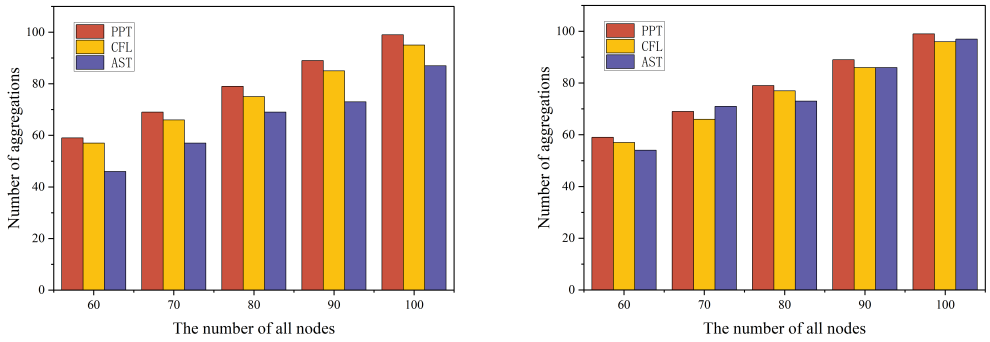
(**a**) RC (**b**) MVN

Figure 6. Communication time.

5.2.2. Number of Aggregations

We adjusted the number of vehicle terminal nodes N and compared the number of aggregations.

Figure 7 also presents the number of aggregations under all three methods. As the total number of nodes increases, the number of aggregations rises for all propagation modes. However, the cluster-in-propagation mode consistently requires fewer aggregations compared to the other two methods. Despite augmenting the number of aggregations on layer-1 nodes to expedite global model aggregation within the subgraph, the cluster-in-propagation mode maintains the number of aggregations within a reasonable range. This number does not exceed the one-step-at-a-time aggregation mode of the PPT and CFL algorithms. Additionally, aggregation operations on layer-1 nodes can be performed in parallel. In contrast, aggregation operations in the PPT and CFL propagation modes must be executed serially alongside model propagation between nodes. Consequently, even with the same number of aggregations, the cluster-in-propagation mode is expected to require less time than the PPT and CFL propagation modes.



(**a**) RC (**b**) MVN

Figure 7. Number of aggregations.

5.3. Personalized Differential Privacy

This section will show the impact of personalized differential privacy on the algorithm.

Three datasets were utilized to simulate federated learning in this study, including the Mnist dataset, Cifar10 dataset, and a specialized vehicle image dataset, Car. This specific vehicle image dataset is categorized into six classes: garbage trucks, buses, trucks, cars, pickups, and dump trucks. The training set encompasses approximately 5500 images, while the test set comprises approximately 750 images. For example, images of the cars are shown in Figure 8.

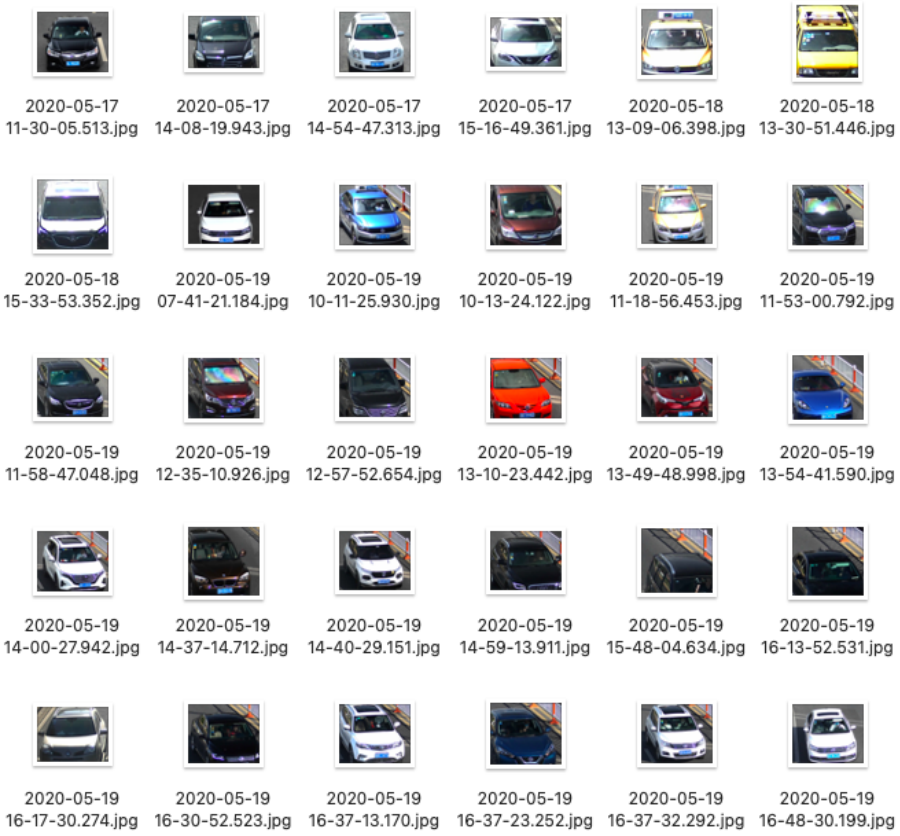
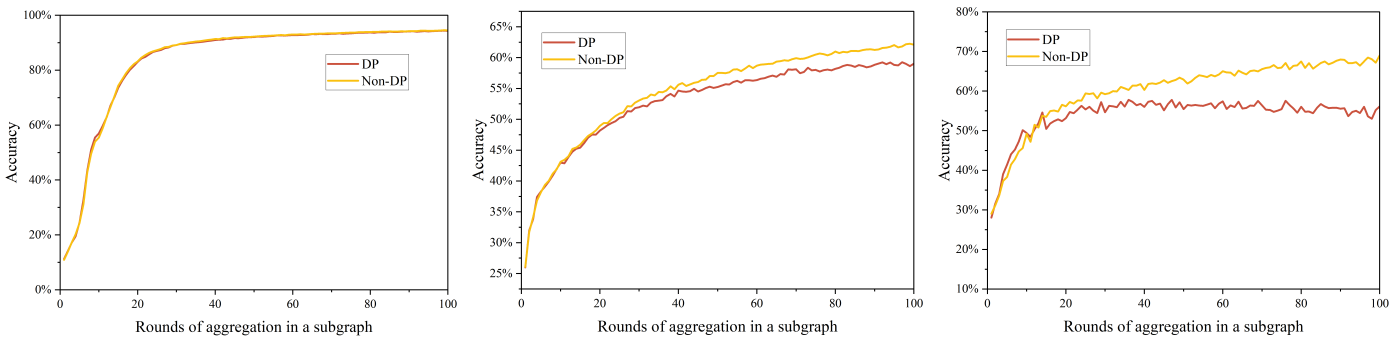


Figure 8. Special vehicle image dataset.

Three datasets were experimented with using specific CNN models and used for the same federated learning experiment. The results of the accuracy rate are shown in Figure 9, and loss is shown in Figure 10.

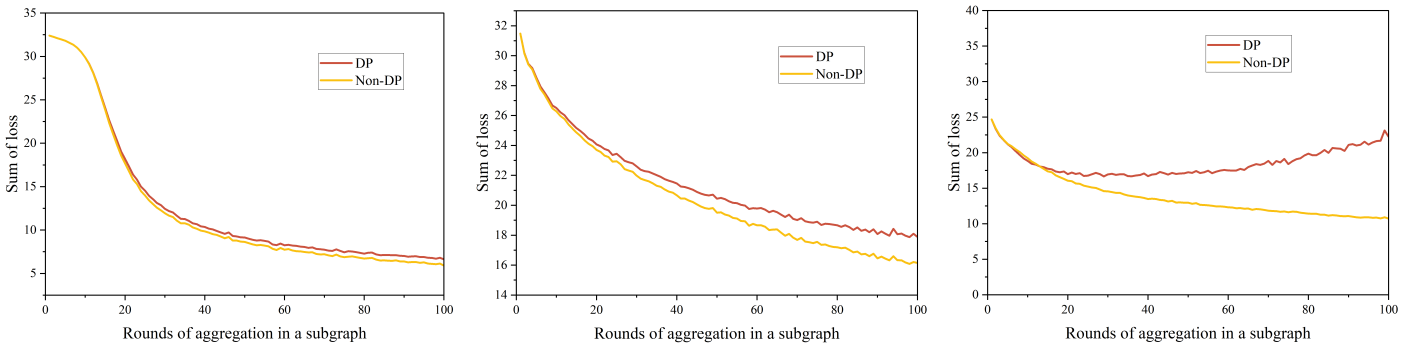


(**a**) MNIST (**b**) CIFAR-10 (**c**) Car

Figure 9. Accuracy.

The impact of personalized differential privacy on the generation of the global model is evident due to the addition of noise, resulting in decreased accuracy when tested across three types of data. However, the extent of this impact may vary slightly under the same node distribution and privacy strategy in federated learning due to differences in datasets. The MNIST dataset experiences the least impact on model accuracy owing to its simple image color and abundant data. Conversely, the CIFAR-10 dataset, with its more complex RGB images and larger dataset for local training, is more sensitive to noise. Moreover, the Car dataset is the most susceptible to noise due to its intricate image hierarchies and smaller data volume. When no noise is added, the test accuracy of the Car dataset steadily increases with the number of aggregations. However, upon introducing noise, the test

accuracy remains relatively consistent with minor fluctuations, and the upward trend in accuracy is not consistently maintained with an increase in the number of aggregations.



(**a**) MNIST (**b**) CIFAR-10 (**c**) Car

Figure 10. Sum of loss.

The impact of the personalized differential privacy scheme on the total loss value of federated learning follows the descending order of the Car, CIFAR-10, and MNIST datasets. The smaller number of instances in the self-constructed Car dataset results in significant fluctuations and challenges in convergence after introducing Gaussian noise to the model. Conversely, experiments conducted on larger datasets like MNIST and CIFAR-10 exhibit a smaller influence on the total loss value, allowing the trend of a decreasing total loss value with the number of aggregations to remain unaffected. This ensures the convergence of the model under the personalized differential privacy strategy, thereby safeguarding user security while maintaining the usability of federated learning in the foggy environment of connected cars.

6. Algorithm-Related Proofs

This section provides some proofs relevant to the scheme of this study.

6.1. Proof of P2P Architecture

To demonstrate that the segmentation performed by the F-Prim algorithm ensures a subgraph diameter of no more than 4, consider the hypothetical scenario illustrated in Figure 11. Suppose that there exists a subgraph with a diameter of 4, implying that at least two nodes within the subgraph are separated by a distance that requires four hops to traverse between them. Let us denote this path as A-B-C-D-E for the purpose of the proof.



Figure 11. Assumptions in node segmentation.

Indeed, if adding a node F such that the diameter of the subgraph exceeds 4, it implies that the C-node for this subgraph must be positioned at either D or B to maintain the condition that the distance from F to the C-node is at most 2. However, this contradicts the F-Prim algorithm's methodology.

According to the F-Prim algorithm, after the previous round of node additions, a C-node is selected through a harmonic centrality computation. Given that the computation involves nodes A, B, C, D, and E, C is determined to be closer to the center, resulting in a higher centrality value for C compared to D. Thus, the C-node should logically be located at node C rather than node D, which contradicts the assumptions.

Therefore, it can be concluded that the F-Prim algorithm effectively limits the diameter of the subgraph to 4.

6.2. Aggregation Weight Proof

Above, in FedAvg, it is noted that the weights of node v_i's local training model, denoted as ω_i, in the global model ω can be represented by Equation (3). Consequently, the global model yields

$$\omega_{FedAvg} = \sum_{i=1}^{K} p_i \omega_i = \sum_{i=1}^{K} \left(\frac{|D_i|}{\sum_{k=1}^{K} |D_k|} \cdot \omega_i \right) = \frac{\sum_{i=1}^{K} |D_i| \omega_i}{\sum_{k=1}^{K} |D_k|} \tag{17}$$

However, under the architecture, if not adjusted accordingly, there may be an imbalance in model weights. Within this architecture, the adjustment of model weights is delved into, simulating both a simple node network, as in Figure 12a, and a more complex node network, as in Figure 12b, and the global model ω was calculated without differential privacy.

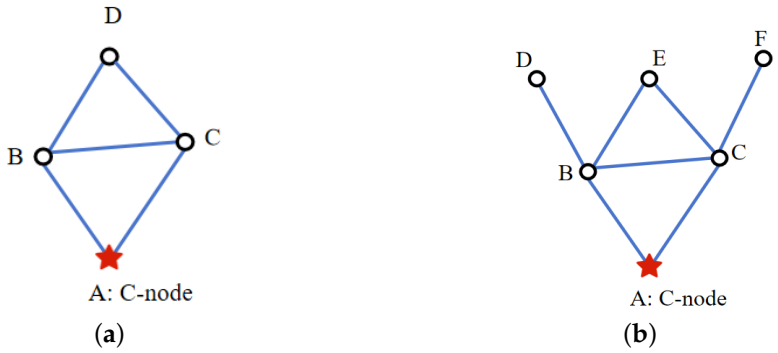


Figure 12. Node network graph, (**a**) simple node network and (**b**) complex node network.

6.2.1. Calculation of Weight

It is assumed that the amount of data collected on each node is different. So right now, in Figure 12a, $|D_A| \neq |D_B| \neq |D_C| \neq |D_D|$, and in Figure 12b, $|D_A| \neq |D_B| \neq |D_C| \neq |D_D| \neq |D_E| \neq |D_F|$.

In Figure 12a, after the one-layer nodes, that is, node v_B and node v_C, receive the model from the two-layer nodes and one-layer nodes, aggregation is completed on v_B and v_C. The result on node B and node C is

$$\omega_{v_B} = \omega_{v_C} = \frac{\omega_B |D_B| + \omega_C |D_C| + \omega_D |D_D|}{|D_B| + |D_C| + |D_D|}$$

At present, the amount of the dataset used for the aggregation in ω_{v_B} and ω_{v_C} is

$$|D_{v_B}| = |D_{v_C}| = |D_B| + |D_C| + |D_D|$$

Thus, after aggregating ω_{v_B}, ω_{v_C}, and v_A's model ω_A, the global model ω on v_A is

$$\begin{aligned} \omega =& \frac{\omega_{v_B} \cdot |D_{v_B}| + \omega_{v_C} \cdot |D_{v_C}| + \omega_A \cdot |D_A|}{|D_{v_B}| + |D_{v_C}| + |D_A|} \\ =& \frac{\omega_A |D_A| + 2\omega_B |D_B| + 2\omega_C |D_C| + 2\omega_D |D_D|}{|D_A| + 2|D_B| + 2|D_C| + 2|D_D|} \end{aligned}$$

Also calculated in the same manner, in Figure 12b, the global model ω on v_A is

$$\omega = \frac{\omega_A |D_A| + 2\omega_B |D_B| + 2\omega_C |D_C| + \omega_D |D_D| + 2\omega_E |D_E| + \omega_F |D_F|}{|D_A| + 2|D_B| + 2|D_C| + |D_D| + 2|D_E| + |D_F|}$$

6.2.2. Analysis and Solutions

Through the above calculations, under the designed architecture, it can be summarized that without tuning, the global model, ω, obtained with FedAvg as the aggregation algorithm is

$$\omega = \frac{\sum_{i=1}^{N} \alpha_i \omega_i |D_i|}{\sum_{n=1}^{N} \alpha_n |D_n|} = \sum_{i=1}^{N} \left(\frac{\alpha_i |D_i|}{\sum_{n=1}^{N} \alpha_n |D_n|} \cdot \omega_i \right) = \sum_{i=1}^{N} \acute{p_i} \omega_i \tag{18}$$

where the value of α_i is determined using Equation (5).

So, it becomes evident that there is a discernible pattern in the contribution of a node to the global model ω:

$$\acute{p_i} = \frac{\sum_{i=1}^{N} \alpha_i |D_i|}{\sum_{n=1}^{N} \alpha_n |D_n|} \tag{19}$$

This will cause an imbalance in node v_i's contribution to the global model ω, and the weight p_i is intended to be determined solely through the amount of the dataset D_i. Therefore, in order to ensure consistent contributions, a node v_i transmits its model ω_i with the amount of dataset as in Equation (4).

After the adjustment, the weight share $\tilde{p}_i$ of v_i's model ω_i is

$$\tilde{p}_i = \frac{\alpha_i |D_i|_\alpha}{\sum_{n=1}^{N} \alpha_n |D_i|_\alpha} = \frac{\alpha_i \frac{|D_i|}{\alpha_i}}{\sum_{n=1}^{N} \alpha_n \frac{|D_i|}{\alpha_n}} = \frac{|D_i|}{\sum_{n=1}^{N} |D_n|} = p_i \tag{20}$$

which is the same in FedAvg.

6.3. Convergence Proof for Global Model

When Gaussian noise is added by the nodes except the C-node, the global model's results in this study were

$$\begin{aligned} \omega &= \sum_{i=1}^{N} \tilde{p}_i \tilde{\omega}_i (i \neq center) + \tilde{p}_{center} \omega_{center} \\ &= \sum_{i=1}^{N} p_i (\omega_i + n_i)(i \neq center) + p_{center} \omega_{center} \\ &= \sum_{i=1}^{N} p_i n_i (i \neq center) + \sum_{i=1}^{N} p_i \omega_i \end{aligned} \tag{21}$$

The sum of the Gaussian noise added to the global model is

$$\sum_{i=1}^{N} n_i \sim N(0, \sum_{i=1}^{N} \sigma_i^2), i \neq center \tag{22}$$

where the value of σ_i needs to satisfy Equation (7).

According to Equation (22), $\sum_{i=1}^{N} n_i$ satisfies the mean value of 0. So, Equation (23) can be obtained as follows:

$$\mathrm{E}[F(w)] = \mathrm{E}[F(w_{FedAvg})] \tag{23}$$

Since the FedAvg converges [18], the global model ω discussed in this paper also converges.

7. Conclusions

In summary, a P2P federated learning scheme was implemented for IoV applications, integrating personalized differential privacy and a special way of communication after partitioning nodes into multiple subgraphs. Through experiments and analyses, it was demonstrated that the design of the scheme prepares a good grouping of nodes for federated learning in P2P architectures, the inward aggregation propagation to C-nodes speeds up

the process of federated learning within the grouping, and the introduction of personalized differential privacy provides privacy preservation without affecting the effect of federated learning too much.

This study can also be improved by thinking about the following aspects:

- The current design will limit the diameter of each subgraph; there is a better solution through implementing geolocation-based grouping.
- How other privacy-preserving algorithms should be implemented in this specific architecture.

Author Contributions: Conceptualization, J.Z. and Y.G.; methodology, J.Z. and Y.G.; validation, Y.G. and B.Y.; formal analysis, Y.G.; investigation, Y.W.; resources, J.Z.; writing—original draft preparation, Y.G.; writing—review and editing, J.Z.; supervision, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by systematic major projects of China State Railway Group Co., Ltd. (No. P2023W001).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study.

Abbreviations

The following abbreviations are used in this manuscript:

IoV	Internet of Vehicles
C-S	Client–server
P2P	Peer-to-peer
C-node	Center node

References

1. Feng, Y.; Mao, G.; Chen, B.; Li, C.; Hui, Y.; Xu, Z.; Chen, J. MagMonitor: Vehicle speed estimation and vehicle classification through a magnetic sensor. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1311–1322. [CrossRef]
2. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
3. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
4. Samarakoon, S.; Bennis, M.; Saad, W.; Debbah, M. Distributed federated learning for ultra-reliable low-latency vehicular communications. *IEEE Trans. Commun.* **2019**, *68*, 1146–1159. [CrossRef]
5. Kong, X.; Gao, H.; Shen, G.; Duan, G.; Das, S.K. Fedvcp: A federated-learning-based cooperative positioning scheme for social internet of vehicles. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 197–206. [CrossRef]
6. Yuan, L.; Ma, Y.; Su, L.; Wang, Z. Peer-to-peer federated continual learning for naturalistic driving action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5249–5258.
7. Barbieri, L.; Savazzi, S.; Brambilla, M.; Nicoli, M. Decentralized federated learning for extended sensing in 6G connected vehicles. *Veh. Commun.* **2022**, *33*, 100396. [CrossRef]
8. Lu, Y.; Huang, X.; Dai, Y.; Maharjan, S.; Zhang, Y. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2134–2143. [CrossRef]
9. Chen, J.H.; Chen, M.R.; Zeng, G.Q.; Weng, J.S. BDFL: A byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8639–8652. [CrossRef]
10. Niu, M.; Cheng, B.; Feng, Y.; Chen, J. GMTA: A geo-aware multi-agent task allocation approach for scientific workflows in container-based cloud. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1568–1581. [CrossRef]
11. Dekker, A. Conceptual distance in social network analysis. *J. Soc. Struct.* **2005**, *6*, 31.
12. Freeman, L.C. Centrality in social networks: Conceptual clarification. *Soc. Netw.* **2002**, *1*, 238–263. [CrossRef]
13. Freeman, L.C. A set of measures of centrality based on betweenness. *Sociometry* **1977**, *40*, 35–41. [CrossRef]
14. Dwork, C. Differential privacy. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Venice, Italy, 10–14 July 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–12.
15. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.; Poor, H.V. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]

16. Chen, Q.; Wang, Z.; Zhang, W.; Lin, X. PPT: A privacy-preserving global model training protocol for federated learning in P2P networks. *Comput. Secur.* **2023**, *124*, 102966. [CrossRef]
17. Chen, Q.; Wang, Z.; Zhou, Y.; Chen, J.; Xiao, D.; Lin, X. CFL: Cluster federated learning in large-scale peer-to-peer networks. In Proceedings of the International Conference on Information Security, Taipei, Taiwan, 23–25 November 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 464–472.
18. Li, X.; Huang, K.; Yang, W.; Wang, S.; Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv* **2019**, arXiv:1907.02189.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.