mathematics

MDPI

Article

Improved Federated Learning Incentive Mechanism Algorithm Based on Explainable DAG Similarity Evaluation

Wenhao Lin [1] and Yang Zhou [2,3,*]

[1] Applied Mathematics, University of Washington, Seattle, WA 98015, USA; linwh@uw.edu
[2] Artificial Intelligence Laboratory, Shanghai University, Shanghai 201109, China
[3] Water Technology Research Center, Chemical and Biomolecular Engineering Department, Henry Samueli School of Engineering and Applied Science, University of California, Los Angeles, CA 90095, USA
* Correspondence: zhouyangucla@ucla.edu; Tel.: +1-424-385-8710

Abstract

In vehicular networks, inter-vehicle data sharing and collaborative computing improve traffic efficiency and driving experience. However, centralized processing faces challenges with privacy, communication bottlenecks, and real-time performance. This paper proposes a trust assessment mechanism for vehicular federated learning based on graph neural network (GNN) edge weight similarity. An explainable asynchronous federated learning data sharing framework is designed, consisting of permissioned asynchronous federated learning and a locally verifiable directed acyclic graph (DAG). The GNN connection weights perform reputation assessment on edge devices through DAG-based verification, while deep reinforcement learning (DRL) enables explainable node selection to improve asynchronous federated learning efficiency. The proposed explainable incentive mechanism based on GNN edge weight similarity and DAG can not only effectively prevent malicious node attacks but also improve the fairness and explainability of federated learning. Extensive experiments across different participant scales (30–200 nodes), various asynchrony degrees (α = 1–5), and malicious node attack scenarios (up to 50% malicious nodes) demonstrate that our method consistently outperforms state-of-the-art approaches, achieving up to 99.2% accuracy with significant improvements of 1.3–3.1% over existing trust-based federated learning methods and maintaining 95% accuracy even under severe attack conditions. The results show that the proposed scheme performs well in terms of learning accuracy and convergence speed.

Keywords: asynchronous federated learning; deep reinforcement learning; directed acyclic graph; graph neural network; trust assessment

MSC: 68T05

check for updates

Academic Editors: Fuhsing Wang, Pedro A. Castillo Valdivieso and Antanas Cenys

Received: 27 July 2025
Revised: 31 August 2025
Accepted: 13 October 2025
Published: 2 November 2025

Citation: Lin, W.; Zhou, Y. Improved Federated Learning Incentive Mechanism Algorithm Based on Explainable DAG Similarity Evaluation. *Mathematics* **2025**, *13*, 3507. https://doi.org/10.3390/math13213507

Copyright: © 2025 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/).

1. Introduction

The world is undergoing a wave of digital transformation. The commercial deployment of 5G network technology provides unprecedented bandwidth and low latency for in-vehicle communications [1]. Emerging technologies such as artificial intelligence, the Internet of Things, and edge computing are injecting powerful momentum into intelligent transportation systems. The evolution of autonomous driving technology from Level 2 to Level 4 and Level 5 has laid a solid foundation for building a smart city transportation ecosystem.

The Internet of Vehicles (IoV), a revolutionary technology paradigm that integrates intelligent computing, communications, and in-vehicle networks, has become a core pillar of the modern intelligent transportation ecosystem. The International Telecommunication Union predicts that by 2030, the number of connected vehicles worldwide will exceed 125 million, with each vehicle generating 4 TB of data per day. This massive amount of in-vehicle data, containing rich information resources such as real-time traffic flow, road conditions, and driving behavior patterns, has enormous commercial value and potential for social benefits.

However, in-vehicle networks are highly dynamic and complex. High-speed vehicle movement leads to constant changes in network topology, making communication link quality unpredictable [2]. Traditional centralized data processing architectures face challenges such as privacy breaches, communication bottlenecks, and insufficient real-time performance. The sensitivity of in-vehicle data far exceeds that of typical IoT applications. Vehicle trajectories can accurately reflect users' residential addresses, workplaces, and consumption habits, creating an urgent need for privacy protection.

To overcome the inherent limitations of the cloud computing model, mobile edge computing (MEC) technology deploys distributed computing resources at the edge of the network, enabling local data processing and real-time response [3]. The deep integration of edge computing and artificial intelligence technologies has opened a new chapter in the development of intelligent in-vehicle systems. DRL algorithms demonstrate excellent adaptability in dynamic environments, offering a natural advantage in in-vehicle resource scheduling and network optimization [4]. Furthermore, adaptive federated learning demonstrates good performance in resource-constrained edge computing systems [5]. However, existing edge intelligence solutions primarily focus on single-dimensional performance improvements, with insufficient attention paid to security and trustworthiness issues in multi-party collaborative environments. In open in-vehicle networks, systems face diverse security threats such as data poisoning attacks, model reverse engineering attacks, and Byzantine attacks. Traditional cryptographic security mechanisms are unable to effectively address these new attack vectors against AI systems.

Blockchain technology, with its decentralized architecture, cryptographic protection, and consensus mechanism, offers a revolutionary solution for building a trusted in-vehicle data sharing platform [6]. Blockchain holds broad application prospects in intelligent transportation systems [7], with smart contract technology enabling the automated execution of complex business logic [8]. Blockchain-based security and privacy protection mechanisms can provide reliable assurance for sensitive applications such as smart health [9], and this concept is equally applicable to in-vehicle data protection. To overcome the performance bottlenecks of traditional blockchains, the DAG architecture employs a graph-like data organization approach, allowing for the simultaneous processing of multiple transactions [10]. The IOTA Tangle protocol, a representative example of the DAG architecture, has demonstrated outstanding performance in IoT data communication [11], providing a new technical reference for in-vehicle networks. Federated learning, a major breakthrough in distributed machine learning, shifts the paradigm from "data centralization and model distribution" to "data decentralization and model aggregation," providing a theoretical foundation for in-vehicle data privacy protection [12]. Recent advances and open problems in the field of federated learning point the way for further development of this technology [13]. In in-vehicle scenarios, the application of federated learning in mobile edge networks has become a research hotspot [14]. As mobile data producers, vehicles can achieve cross-domain knowledge transfer and continuous model optimization through federated learning.

The high-speed mobility and fluctuating communication quality of vehicular environments pose challenges to the implementation of federated learning. Research on efficient federated learning for communication in digital twin edge networks within the Industrial Internet of Things [15] provides important insights for vehicular networks. GNNs demonstrate exceptional modeling capabilities for processing graph-structured data, and comprehensive research [16] and method application reviews [17] have laid the foundation for development in this field. In vehicular networks, vehicles and roadside equipment form a dynamic, heterogeneous graph. GNNs can learn low-dimensional representations of nodes and edges through message passing. Spatiotemporal graph convolutional networks provide a deep learning framework [18] for traffic prediction, and graph attention networks achieve more accurate graph representation learning through attention mechanisms [19]. Inductive representation learning techniques on large-scale graphs [20] provide technical support for handling dynamic vehicular networks. The edge weight matrix, a core component of GNNs, carries complete information about the graph structure and accumulated learning knowledge. Compared to easily manipulated external performance metrics, the edge weight matrix more accurately reflects the intrinsic learning quality of the model, providing a new technical perspective for evaluating participant contributions.

Existing solutions often struggle to balance privacy, security, efficiency, and usability, lacking systematic design tailored to the specific characteristics of the in-vehicle environment. Traditional trust assessment mechanisms rely on historical reputation or simple statistical metrics, failing to accurately reflect the quality of participants' real-time contributions and data distribution characteristics. Furthermore, existing research often examines blockchain, federated learning, and GNNs as independent modules, failing to fully leverage the synergistic effects of technological integration. Therefore, designing an in-vehicle federated learning trust mechanism that integrates GNN edge weight similarity assessment, DAG interpretable verification, and DRL intelligent node selection has become a key scientific issue that needs to be addressed. Solving this problem has important theoretical and practical significance for building a secure and trustworthy in-vehicle data sharing ecosystem.

In order to systematically solve the above multi-dimensional challenges and promote the innovative development of vehicle-mounted data sharing technology, this paper proposes an on-vehicle federated learning trust evaluation framework based on the edge weight similarity of GNNs. This work innovatively combines the decentralized trust mechanism of blockchain technology, the privacy protection capability of federated learning, and the structural modeling advantages of GNNs to design a complete set of technical solutions and implementation architectures. The main innovative contributions and technical breakthroughs of this paper include:

- An innovative similarity evaluation mechanism based on edge weight matrix is proposed: Each participating vehicle uses local data to train a specially designed GNN model to generate an edge weight matrix that can fully reflect the data feature distribution and internal relationship structure. Efficient matrix similarity measurement algorithms are developed, including similarity calculation based on spectral analysis, distance measurement based on information theory, and pattern recognition methods based on machine learning, to achieve accurate quantitative evaluation of participants' data quality, learning ability, and behavior patterns.
- A decentralized intelligent mutual evaluation and scoring system is designed: A mutual evaluation and dynamic feedback mechanism between participating vehicles was established, and comprehensive scoring was performed based on edge weight matrix similarity, historical behavior records, and real-time performance. The system uses multi-dimensional evaluation indicators and adaptive weight adjustment strategies

to effectively identify and isolate malicious participants while motivating the active contributions of honest participants.

- An optimized hybrid blockchain–DAG architecture is developed: Based on the existing PermiDAG framework, targeted architecture optimization and functional expansion were carried out, and the system was designed specifically for the storage, transmission, verification, and synchronization requirements of the GNN edge weight matrix. The local DAG network is responsible for handling high-frequency similarity calculations, score updates, and anomaly detection tasks, and the main blockchain ensures the security, consistency, and immutability of key decision-making information.
- An adaptive intelligent aggregation optimization algorithm is implemented: Based on the trust score, similarity analysis, and quality assessment results, a multi-level adaptive model aggregation strategy was designed. The algorithm can dynamically adjust the aggregation weight distribution according to the credibility, contribution quality, and data representativeness of the participants, and it gives priority to high-quality model updates to participate in global aggregation, thereby significantly improving the convergence speed, model accuracy, and robustness of federated learning.
- A comprehensive multi-level security protection system is constructed: Deeply integrated differential privacy technology, a secure multi-party computing protocol, a homomorphic encryption algorithm, and blockchain consensus mechanism form a comprehensive security protection system covering the entire life cycle of data collection, transmission, processing, storage and use. While ensuring the data privacy and commercial confidentiality of participants, it effectively resists various active attacks, passive attacks, and reasoning attacks.

The technical solution proposed in this paper not only has important theoretical innovation value and academic significance but also has broad industrial application prospects and commercialization potential. The developed core algorithms and system architecture can be directly applied to multiple important application fields such as intelligent traffic management, cooperative autonomous driving, in-vehicle content distribution, dynamic path optimization, predictive maintenance, and personalized service recommendation, providing solid technical support and innovation driving force for building a safe, reliable, and efficient collaborative next-generation IoV ecosystem.

2. Related Work

The combination of blockchain technology and federated learning in the IoV has become an important research direction to address data sharing challenges while protecting privacy and enhancing system security. This section reviews the latest research progress in four key areas: IoV and edge computing, the application of blockchain in distributed systems, federated learning methods, and the application of GNNs in vehicle systems.

2.1. Internet of Vehicles and Edge Computing

The rapid development of the IoV is due to the progress of 5G technology and device-to-device (D2D) communication. Castillo et al. [21] comprehensively outlined the IoV architecture, highlighting the key protocols and inherent security challenges in vehicle communication systems. The evolution of vehicle communication technology has laid the foundation for efficient data sharing and collaborative computing. The deployment of D2D communication shows great potential in improving network efficiency and reducing latency in vehicle environments [22], especially in dense urban environments, where direct communication between vehicles can significantly reduce the dependence on infrastructure.

Mobile edge computing (MEC) has become an important driving force for vehicle applications by deploying computing resources close to vehicles. Zhang et al. [23] demon-

strated the effectiveness of MEC collaborative content caching in 5G networks, improving the quality of experience for in-vehicle users. Edge computing architectures provide key support for real-time data processing and low-latency services, which are essential for autonomous driving and safety-critical applications [24]. Studies have shown that by deploying computing resources at the edge of the network, processing latency can be reduced to milliseconds, meeting the strict time requirements of in-vehicle applications.

The integration of artificial intelligence in in-vehicle edge computing has further improved system performance, and recent studies have explored the application of DRL methods in intelligent resource allocation [25]. The application of deep learning algorithms in in-vehicle environments includes traffic flow prediction, path optimization, and energy consumption management. These intelligent algorithms can adapt to dynamically changing network conditions and implement adaptive resource management strategies.

Many studies focus on optimizing channel allocation and resource management in the IoV. Tang et al. [24] proposed a partially overlapping channel allocation method based on an anti-coordination game for hybrid networks of drones and D2D, solving the interference management challenge in heterogeneous vehicular environments. Spectrum management in vehicular networks is a complex optimization problem that needs to consider multiple factors such as vehicle mobility, channel fading, and interference. Recent studies have proposed a dynamic spectrum allocation algorithm based on machine learning, which can adjust the spectrum usage strategy in real time according to the network status.

The concept of vehicular cloud computing has also attracted widespread attention from researchers. By virtualizing and integrating the computing and storage resources of the vehicle, a distributed computing cloud is formed. This architecture can provide elastic computing services for vehicular applications while reducing the cost of infrastructure investment. Studies have shown that vehicular cloud computing has significant advantages in processing large-scale data analysis and complex computing tasks.

2.2. Directed Acyclic Graph Technology in Distributed Systems

Blockchain technology has gained widespread attention as a solution for secure and decentralized data management in various domains. Dai et al. [26] comprehensively surveyed the application of blockchain in IoT and identified key challenges and opportunities for blockchain integration in distributed systems. The tamper-proof and decentralized nature of blockchain makes it particularly suitable for scenarios where trust needs to be established between untrusted participants, which is particularly important in the in-vehicle environment because vehicles come from different manufacturers and operators.

In edge computing and industrial IoT environments, blockchain has been explored for enhancing system security and enabling secure resource transactions. Zhang et al. [27] studied the integration of edge intelligence and blockchain in 5G and beyond networks, showing how blockchain can provide security while supporting distributed intelligence. The use of permissioned blockchain architectures shows promise in scenarios that require controlled access and efficient consensus and is particularly suitable for enterprise-level in-vehicle applications [28].

Recent research has also explored novel blockchain architectures, such as DAG-based systems, which provide improved scalability and reduced computational overhead compared to traditional blockchain implementations. The DAG architecture improves system throughput by processing transactions in parallel, which is particularly beneficial for high-frequency in-vehicle data exchange. IOTA's Tangle architecture is a typical example of the successful application of DAG in the IoT environment.

The application of smart contract technology in in-vehicle blockchain systems has also received attention. Studies have shown that smart contracts can automate the execu-

tion of data sharing agreements and payment mechanisms, reduce manual intervention, and improve system reliability [29]. In application scenarios such as in-vehicle insurance, maintenance services, and energy trading, smart contracts can ensure the transparency and automatic execution of transactions. At the same time, the optimization of the consensus mechanism is crucial to the real-time requirements of the in-vehicle environment. Researchers have proposed various efficient consensus algorithms to balance security and performance [30].

The application of blockchain in supply chain management also provides inspiration for in-vehicle systems. By establishing an unalterable data record chain, the source and quality information of vehicle parts can be tracked [31]. This transparency is of great significance to ensure the security and reliability of in-vehicle systems.

2.3. Federated Learning and Privacy Protection

Federated learning has become a paradigm shift in distributed machine learning, enabling collaborative model training without centralizing sensitive data. The approach addresses key privacy issues while leveraging distributed data sources to improve learning outcomes. McMahan et al. established a basic framework for federated learning and demonstrated its effectiveness in scenarios with heterogeneous data distribution and different computing power [32]. The core idea of federated learning is "data does not move, model moves", which is in stark contrast to traditional centralized machine learning.

In the IoV, federated learning faces unique challenges due to high mobility, intermittent connectivity, and heterogeneous computing resources. Recent work has explored asynchronous federated learning methods to address these challenges, allowing vehicles with different availability and computing power to participate more flexibly [33]. Asynchronous federated learning avoids the problem of waiting for the slowest node in traditional synchronous methods and improves overall training efficiency.

Privacy protection in federated learning is enhanced by various techniques, including differential privacy and secure aggregation protocols. Differential privacy protects individual data privacy by adding noise to model parameters while maintaining overall model performance [34]. Secure aggregation protocols ensure that the server can only see the aggregated model updates and cannot obtain the original updates of individual participants. The integration of blockchain technology and federated learning shows promise in providing additional security guarantees and achieving trustless aggregation of model updates.

In vehicular federated learning, data heterogeneity is an important challenge. Data collected by different vehicles may have different distribution characteristics, which will affect the performance of the global model [35]. Researchers have proposed various methods to handle non-independent and identically distributed (Non-IID) data, including techniques such as personalized federated learning and clustered federated learning. Personalized federated learning allows each participant to maintain a personal model optimized for its local data distribution while still benefiting from global collaboration. Personalized federated learning allows each participant to maintain a personal model optimized for its local data distribution while still benefiting from global collaboration. Recent trust-based federated learning approaches include PoQRBFL [36], which uses reputation-motivated task participation for blockchain-based federated learning, PoTQBFL [37], which focuses on trust and quality assessment in Industrial IoT environments, FLTrust [38], which achieves Byzantine robustness through trust bootstrapping with server-side validation, and LAFED [39], which provides lightweight authentication for blockchain-enabled federated learning systems.

The communication efficiency of federated learning is also a research focus, especially in bandwidth-constrained vehicular environments. Various compression techniques and

communication optimization methods have been proposed to reduce transmission overhead [40]. Gradient compression, model quantization, and sparsification techniques are widely used to reduce communication costs while maintaining model performance.

2.4. Graph Neural Networks in Internet of Vehicle Systems

The application of GNNs in IoV systems is becoming an emerging research field. In-vehicle networks can essentially be modeled as graph structures, with vehicles as nodes and communication links as edges. GNNs are able to capture this complex topological relationship and learn the interaction patterns between nodes [16]. Compared with traditional machine learning methods, GNNs are more suitable for processing structured data and dynamic topological changes in in-vehicle networks.

Recent studies have explored the application of GNNs in traffic prediction, path planning, and resource allocation. Zhou et al. proposed a traffic flow prediction method based on graph convolutional networks, which can effectively capture the spatial dependencies of traffic networks [17]. The power of GNNs lies in their ability to handle dynamic graph structures, which is particularly important for highly mobile in-vehicle environments. By learning the embedded representation of the graph, GNNs can predict vehicle behavior and optimize network performance.

In a federated learning environment, GNNs can be used to model similarities and trust relationships between participants. By analyzing the data characteristics and communication patterns of vehicles, GNNs can generate similarity metrics for improving the aggregation process of federated learning [41]. Graph Attention Networks (GATs) perform well in learning dynamic weights and can adaptively adjust the importance of different neighbor nodes.

GNNs also show potential in in-vehicle recommendation systems. By modeling the multivariate relationship graph of user–vehicle–location, they can provide personalized service recommendations for car owners [16]. Spatiotemporal graph neural networks combine temporal and spatial information to more accurately predict the future location and behavior patterns of vehicles.

2.5. Trust Mechanism and Quality Assessment

In distributed vehicle systems, establishing an effective trust mechanism is crucial to ensuring data quality and system security. Traditional reputation-based systems usually rely on historical behavior records but may not be flexible enough in dynamic vehicle environments [42]. The changing dynamics of the vehicle environment require the trust mechanism to quickly adapt to changes in participant behavior.

Recent studies have explored content-based trust assessment methods to evaluate the credibility of participants by analyzing the quality of shared data or models. This approach can more accurately reflect the current contribution of participants rather than relying solely on historical reputation [43]. Multidimensional trust models consider multiple dimensions such as honesty, ability, and reliability, providing a more comprehensive trust assessment.

In federated learning scenarios, model quality assessment becomes a key issue. Researchers have proposed various methods to evaluate the quality of local models, including techniques such as validation set-based evaluation, model similarity analysis, and gradient analysis. Byzantine fault-tolerant federated learning methods can detect and exclude malicious participants and ensure the quality and security of global models.

The graph-based reputation system records trust evaluation in an immutable distributed ledger, which improves the transparency and reliability of the trust mechanism. Smart contracts can automatically execute trust evaluation logic, reducing human intervention and potential manipulation risks.

These limitations motivate the need for more sophisticated methods that can dynamically evaluate participant contributions in highly mobile vehicular environments while maintaining efficiency and safety. Future research needs to focus on developing adaptable, secure, reliable, and computationally efficient vehicular federated learning frameworks.

3. Proposed Method

3.1. System Model

The on-board federated learning system considered in this study consists of multiple core components: a collection of mobile vehicles, edge computing nodes, network infrastructure, and a central coordination server [44], as shown in Figure 1. Let the vehicle collection be $\mathcal{V} = \{v_1, v_2, ..., v_n\}$, where each vehicle v_i is equipped with an on-board computing unit (OBU), multiple sensor devices, and a wireless communication module. The computing power and storage capacity of the on-board devices are relatively limited, but they can perform local data processing and model training tasks.

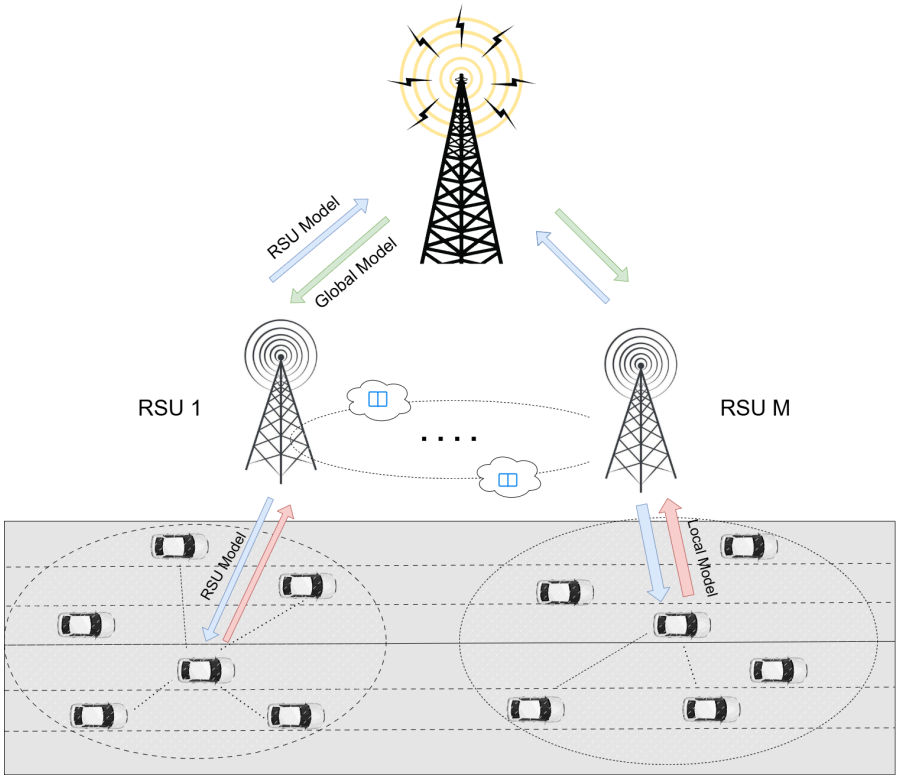


Figure 1. Architecture diagram of the in-vehicle federated learning system.

The roadside unit (RSU) collection is recorded as $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$, and each RSU deploys an edge computing server with medium-scale computing and storage resources. RSU is connected to the core network through a wired network, and it provides communication services to vehicles within its coverage through a wireless interface. As the core node of the network, the macro base station (MBS) has powerful computing and communication capabilities and is responsible for global coordination and management functions.

Vehicles mainly exchange data through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication modes. When a vehicle v_{req} initiates a data collaboration request Req, the goal is to use the distributed data resources $\mathcal{D}$ to obtain the expected calculation results Res. Let the subset of vehicles participating in the collaboration be $\mathcal{V}_{\text{part}} = \{v_1, v_2, \ldots, v_k\} \subseteq \mathcal{V}$, and let the corresponding dataset be $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$. The core of the collaboration task is to learn a shared model $\mathcal{M}$ from distributed data to meet the needs of the requesting vehicle. Our framework is based on the honest majority

assumption, requiring that the majority of participating vehicles (over 50%) are honest. The in-vehicle environment is more trustworthy than consumer device networks due to the collaboration between manufacturers and transportation authorities, and the DAG-based reputation system can filter out poorly performing vehicles [32].

In our system, vehicles are regarded as not fully trusted participants, which means that some vehicles may behave maliciously, try to obtain other people's data without paying the corresponding cost, or deliberately interfere with the learning process. Malicious vehicles may provide forged local models to the federated training system, causing the entire learning process to fail. The resource characteristics of vehicle vi include: available computing resources ξ (in CPU cycles/second), communication transmission rate τ_i, and model quality indicators. Learning quality $\sigma_i = \sum_j \mathcal{L}(y_j - \hat{y}_j)$ is used to quantify model quality, which is closely related to the data quality and honesty of vehicle vi. The data owned by the vehicle determines its contribution to the global calculation results. It should be noted that both computing power and communication capacity may change over time due to parallel computing tasks and dynamic network environments.

3.2. Interpretable GNN-Based Federated Learning Framework

The federated learning paradigm is adopted to complete collaborative computing tasks for data sharing requests. In this architecture, the vehicle set $\mathcal{V}_{\text{part}}$ acts as a training client, and the MBS plays the role of an aggregation server.

For vehicle $v_i \in \mathcal{V}_{\text{part}}$ with dataset D_i, we define a loss function to measure the difference between the estimated value and the true value:

$$\mathcal{F}_i(\theta) = \frac{1}{|D_i|} \sum_{(x_j, y_j) \in D_i} \ell_j(\theta, x_j, y_j) \tag{1}$$

where $\ell_j(\theta, x_j, y_j)$ is the loss function for data sample (x_j, y_j), θ is the model parameter vector, and $|D_i|$ represents the number of data samples for vehicle i. The specific form of the loss function depends on the computing task, and common ones include mean square error (MSE) and mean absolute error (MAE).

The global objective function $\mathcal{F}(\theta)$ is defined as follows:

$$\mathcal{F}(\theta) = \frac{1}{|\mathcal{V}_{\text{part}}|} \sum_{i \in \mathcal{I}} w_i \cdot \mathcal{F}_i(\theta) = \frac{1}{|\mathcal{V}_{\text{part}}|} \sum_{i \in \mathcal{I}} \sum_{(x_j, y_j) \in D_i} w_i \cdot \frac{\ell_j(\theta, x_j, y_j)}{|D_i|} \tag{2}$$

where $\mathcal{I} = \{1, 2, \ldots, k\}$ is the index set of participating vehicles, and wi is the weight factor representing the contribution of vehicle vi to global federated learning, satisfying $\sum_i w_i = 1$.

During the training process, we improve the accuracy of the model $\mathcal{M}$ by iteratively minimizing the global loss function $\mathcal{F}(\theta)$. The optimization objective can be expressed as follows:

$$\mathcal{Q}(\theta, t) = \arg \min_{i \in \mathcal{I}, t \leq T} \mathcal{F}(\theta) \tag{3}$$

The constraints are as follows:

$$\Pr(\theta_i \in \mathbb{R}^d) \leq \exp(\epsilon) \Pr(\theta_i' \in \mathbb{R}^d) \tag{4}$$

$$\sum_{t=1}^{T} \Delta t(i) \leq \min(T_1, T_2, \ldots, T_k) \tag{5}$$

where constraint (4) ensures that parameter θ_i satisfies the ϵ-differential privacy guarantee, constraint (5) ensures that the vehicle can maintain connection with MBS during the learning phase, $\{T_1, T_2, \ldots, T_k\}$ represents the connection time between each vehicle and

MBS, T is the maximum number of iterations, and $\Delta t(i)$ is the execution time of the ith iteration.

Due to the distributed nature of vehicle data and the need for privacy protection, we use a graph-enhanced federated learning method to solve problem (3), which can effectively utilize the limited resources of multiple vehicles and protect data privacy. Problem (3) is a combinatorial optimization problem and it is difficult to find a closed-form solution.

3.3. Federated Learning and Structural Reputation-Driven Asynchronous Sharing Process Design

With the increasing number of participating nodes in the IoV and the high heterogeneity between nodes in terms of computing power, data structure, and network environment, the traditional federated learning mechanism faces many challenges in terms of training efficiency, data quality control, and model fusion. On the one hand, centralized node scheduling cannot cope with the unstable connection problem caused by vehicle mobility; on the other hand, the model aggregation strategy lacking structural semantic recognition ability is prone to introduce low-quality or malicious node uploaded models, which seriously affects the global model performance.

In order to improve the adaptability and security of the entire federated learning system in a heterogeneous IoV environment, this paper proposes a collaborative training process that integrates the structural reputation measurement mechanism, DDPG reinforcement learning scheduling algorithm, and DAG asynchronous fusion architecture. This process not only has a high degree of structural perception but also realizes a completely decentralized node collaboration and verification mechanism, thereby building an intelligent, secure, reliable, efficient, and flexible federated training system. The following will be systematically explained from seven stages: system initialization, node selection, local structure modeling, structural reputation calculation, asynchronous verification, fusion strategy, and continuous feedback. The overall system flow is shown in Figure 2.

(1) System initialization and structural information perception preparation: In the early stage of system deployment, all vehicle nodes need to complete the initialization of the local environment, including data format normalization, GNN model loading, and structural embedding dimension configuration. Each vehicle first collects raw data containing information such as geographic location, historical trajectory, interaction records, etc., locally, and it constructs a graph structure $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$. The nodes represent the vehicle body or key state points, and the edges represent spatial adjacency, behavior correlation, or communication history. Next, the vehicle inputs the graph into the local graph neural network model (such as GAT), generates node representations $\mathbf{h}_v$, aggregates them into structural embeddings $\mathbf{g}_i$, then constructs the structural matrix $\mathcal{S}_i$. To facilitate subsequent verification, each vehicle must also complete system identity registration and obtain basic information such as global model ID, DAG node identification, and communication key to prepare for trusted heterogeneous structure verification and asynchronous collaborative training.

(2) Node selection mechanism based on DDPG: Traditional node scheduling strategies rely on heuristic rules and are difficult to adapt to complex heterogeneous environments. To this end, this paper introduces the deep deterministic policy gradient (DDPG) algorithm to achieve dynamic node selection. DDPG is suitable for processing high-dimensional continuous action spaces and is suitable for reinforcement scheduling problems in asynchronous federation scenarios. Before each round of federated training, the central server or super node generates the current environment state s_t based on the previous round state (such as node reputation value, communication delay, historical performance, etc.), and it inputs the policy network to output candidate actions a_t, that is, the set of nodes to be selected. The output of the policy network will guide the participating nodes to maximize long-term rewards, and the training reward function is constructed by combining factors

such as node structure reputation R_i, response delay τ_i, and historical validity. Through the master–target network combination, experience replay, and soft update mechanism, DDPG can continuously learn the optimal node scheduling strategy, thereby selecting the vehicle nodes with the greatest contribution potential to participate in federated training in a dynamic environment.

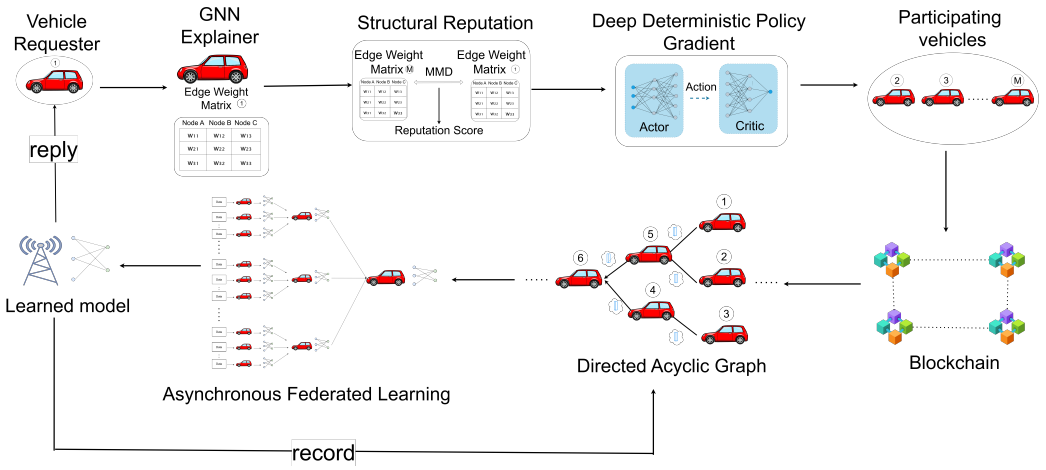


Figure 2. Overall system process architecture diagram of asynchronous federated learning driven by structural reputation.

(3) Global model parameter initialization and local structure modeling: After receiving the scheduling request, the selected vehicle node will download the latest global model parameters θ_t from the blockchain and complete a complete training–modeling–embedding extraction process locally. During the training process, the vehicle node uses its own data graph $\mathcal{G}_i$ as input, uses the local GNN model to update the node embedding, and extracts the structural attention weight matrix $\mathbf{A}_i$ to obtain the information of the key edges in the graph. The structural matrix $\mathcal{S}_i = [\mathbf{g}_i; \text{flatten}(\mathbf{A}_i)]$ summarizes the structural features of topology and edge weights, and it serves as the input for subsequent reputation measurement. Nodes do not upload their original data or training samples, but they only share structural abstractions and model weights to ensure privacy. The vehicle will also generate a training round stamp, a local model version number, and a structural embedding summary and bind and upload them to the local DAG block, waiting for verification by other nodes.

(4) Structural reputation calculation and upload to the DAG network: After each vehicle completes local training and structural modeling, it uploads its current round structural representation $\mathcal{S}_i$ and local model parameters θ_i to the local DAG network. To enhance the structural perception of the verification mechanism, the system introduces the maximum mean difference (MMD) as a measure of the distribution difference between node structures. Specifically, vehicle u_i calculates the difference between its own structural embedding and the structural distribution of historical nodes (or neighboring nodes) $\{u_j \in N_i\}$, obtains the structural similarity score $\alpha_{ij} = \exp\left(-\frac{\text{MMD}^2(\mathcal{S}_i, \mathcal{S}_j)}{\sigma^2}\right)$, and finally averages the structural reputation value $R_i = \frac{1}{|N_i|}\sum_j \alpha_{ij}$. This value is written into the DAG block together with the uploaded model as the structural trust basis of the node contribution. Since the DAG network supports asynchronous access and verification, the structural reputation of the new node can be used as a reference for subsequent vehicles in local model fusion in real time, effectively resisting structural forgery and model deception.

(5) Asynchronous verification mechanism based on structural reputation: When a vehicle receives model structure information uploaded by other nodes in the DAG, it will call the local verifier to evaluate the reputation of the node. The validator first extracts

the structural matrix $\mathcal{S}_j$ of the uploaded node, compares it with the local historical trusted structure set, and uses the MMD method to calculate the structural difference to further determine the trust level of the model. If the structural difference is within a reasonable threshold range, the node is judged as "acceptable"; if the difference is large, the validator will mark it as an abnormal submission and refuse to include it in the local fusion. The verification process does not require manual intervention and is completed entirely based on the structural embedding difference, ensuring the automation, security, and objectivity of asynchronous verification. At the same time, the verification results can also be fed back to the DDPG reward function to promote the scheduling mechanism to dynamically adapt to changes in model quality and form a closed-loop optimization process.

(6) Local fusion strategy driven by structural reputation: The verified model will be included in the local aggregation pool of the current node, and the system will perform weighted fusion based on the structural reputation value $\tilde{R}_i$. The aggregation function is in the following form:

$$\theta_{\text{agg}} = \sum_{i \in V} \tilde{R}_i \cdot \theta_i \quad \text{where} \quad \tilde{R}_i = \frac{R_i}{\sum_k R_k} \tag{6}$$

where V represents the set of uploaded nodes that have passed verification. Compared with the traditional averaging method, this structure-aware weighted strategy significantly enhances the robustness and selectivity of model fusion, and it prevents low-quality models from interfering with global trends. Since the fusion behavior is completed asynchronously in the local DAG network, the system does not need to wait for all nodes to update synchronously, which significantly improves the overall training efficiency and reduces communication overhead.

(7) Asynchronous iteration and continuous scheduling feedback: The fused model is cached as a new local global model version and participates in the initialization of the next round of training. At the same time, the system will score the reputation level, response speed, and resource consumption of the participating nodes based on the verification feedback results of this round and use this score as the environmental feedback signal of DDPG to update the policy network weight. In this way, the system forms an asynchronous closed loop of "selection–training–verification–fusion–feedback", making node selection more structurally robust, model fusion more credible, and the federated learning process more dynamically adaptable and tolerant to attacks. This asynchronous architecture naturally supports the dynamic joining and leaving of vehicles, seamlessly integrating new participants through the DDPG intelligent selection mechanism and automatically adjusting aggregation weights and reputation calculations when vehicles leave, ensuring the continuity of the training process and system stability.

In summary, the structural reputation-driven asynchronous federated learning process proposed in this paper is shown in Figure 3. The complete algorithmic implementation is detailed in Algorithms 1–3, which provide step-by-step pseudocode for the main system workflow, asynchronous verification and aggregation, and DDPG node selection processes. The whole process of "node identification–structure extraction–reputation evaluation–model fusion–strategy update" is systematically connected, taking into account the three goals of data privacy protection, structural trust verification, and federation efficiency optimization. By introducing the GNN model to characterize local structural semantics, using MMD to measure node structural similarity and guide reputation generation, and realizing asynchronous aggregation and consensus verification based on DAG, the process has good scalability and robustness. At the same time, the DDPG enhanced scheduling strategy shows strong environmental adaptability in the scenario of dynamic node changes and system state fluctuations, enabling vehicle nodes to achieve differentiated participation in federated learning while ensuring system efficiency and safety, capability management,

and resource scheduling. This process will provide a practical solution paradigm and implementation basis for safe, stable, and efficient federated intelligent learning in future large-scale IoV.

Algorithm 1 Structural Reputation-Driven Asynchronous Federated Learning

Input: Vehicle set $\mathcal{V}$, initial model θ_0, reputation threshold τ_{rep}
Output: Optimized global model θ_{global}
Phase 1: System Initialization
for each vehicle $v_i \in \mathcal{V}$ **do**
  Initialize local data graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$
  Load GNN model and configure embedding dimensions
  Generate node representations $\mathbf{h}_v$ using GAT
  Construct structural matrix $\mathcal{S}_i = [\mathbf{g}_i; \text{flatten}(\mathbf{A}_i)]$
  Register system identity and obtain DAG node ID
end for
Phase 2: DDPG Node Selection
Initialize policy π_θ and Q-network Q_ϕ
while training not converged **do**
  Observe state $s_t = [R_i, \tau_i, \text{performance}_i]$
  Select action $a_t = \pi_\theta(s_t)$ and set $\mathcal{V}_P$ Phase 3: Local GNN Training
  for each $v_i \in \mathcal{V}_P$ **do**
    Download θ_t; train $\theta_i^{(t)} = \theta^{(t)} - \eta \nabla \mathcal{F}_i(\theta^{(t-1)})$
    Extract attention $\mathbf{A}_i$; update $\mathcal{S}_i$
  end for Phase 4: Structural Reputation Calculation
  for each $v_i \in \mathcal{V}_P$ **do**
    $\alpha_{ij} = \exp\big(-\text{MMD}^2(\mathcal{S}_i, \mathcal{S}_j)/\sigma^2\big)$
    $R_i = \frac{1}{|N_i|} \sum_j \alpha_{ij}$; upload $\{\theta_i, \mathcal{S}_i, R_i\}$ to DAG
  end for Phase 5–6: Asynchronous Verification & Aggregation
  Call Algorithm 2 Phase 7: Continuous Feedback
  Update reward $r_t = f(\text{accuracy}, \text{efficiency}, \text{security})$ and update DDPG with (s_t, a_t, r_t, s_{t+1})
end while
return θ_{global}

Algorithm 2 Asynchronous Verification and Structure-Aware Aggregation

Input: Uploaded models $\{\theta_i, \mathcal{S}_i, R_i\}$, threshold τ_{rep}
Output: Aggregated model θ_{agg}
for each received model from vehicle v_j **do**
  Extract structural matrix $\mathcal{S}_j$ from uploaded data
  $d_{\text{MMD}} \leftarrow \text{MMD}(\mathcal{S}_j, \mathcal{S}_{\text{trusted}})$
  if $d_{\text{MMD}} \leq \tau_{\text{rep}}$ **then**
    Mark as "acceptable"
    $\mathcal{V}_{\text{accepted}} \leftarrow \mathcal{V}_{\text{accepted}} \cup \{v_j\}$
  else
    Mark as "anomalous submission"
    Reject model and exclude from aggregation
  end if
end for
Structure-Aware Weighted Aggregation
For all $i \in \mathcal{V}_{\text{accepted}}$, compute $\tilde{R}_i \leftarrow \dfrac{R_i}{\sum_{k \in \mathcal{V}_{\text{accepted}}} R_k}$
$\theta_{\text{agg}} \leftarrow \sum_{i \in \mathcal{V}_{\text{accepted}}} \tilde{R}_i \, \theta_i$
Cache θ_{agg} as new local global model version
return θ_{agg}

Algorithm 3 DDPG Node Selection with Structural Reputation

Input: State s_t, policy network π_θ, Q-network Q_ϕ
Output: Selected vehicle set $\mathcal{V}_P$
Policy Network Update
Generate action: $a_t \leftarrow \pi_\theta(s_t) + \mathcal{N}(0, \sigma^2)$ ▷ Add exploration noise
$\mathcal{V}_P \leftarrow$ top-k(a_t) based on action values
Environment Interaction
Execute federated learning round with $\mathcal{V}_P$
Observe reward: $r_t \leftarrow w_1 \cdot \text{accuracy} + w_2 \cdot \text{efficiency} - w_3 \cdot \text{latency}$
Observe next state: s_{t+1}
Network Training
Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer $\mathcal{D}$
Sample mini-batch from $\mathcal{D}$
Update Q-network: $\phi \leftarrow \phi - \alpha_Q \nabla_\phi J_Q(\phi)$
Update policy network: $\theta \leftarrow \theta + \alpha_\pi \nabla_\theta J_\pi(\theta)$
Soft update target networks
return $\mathcal{V}_P$

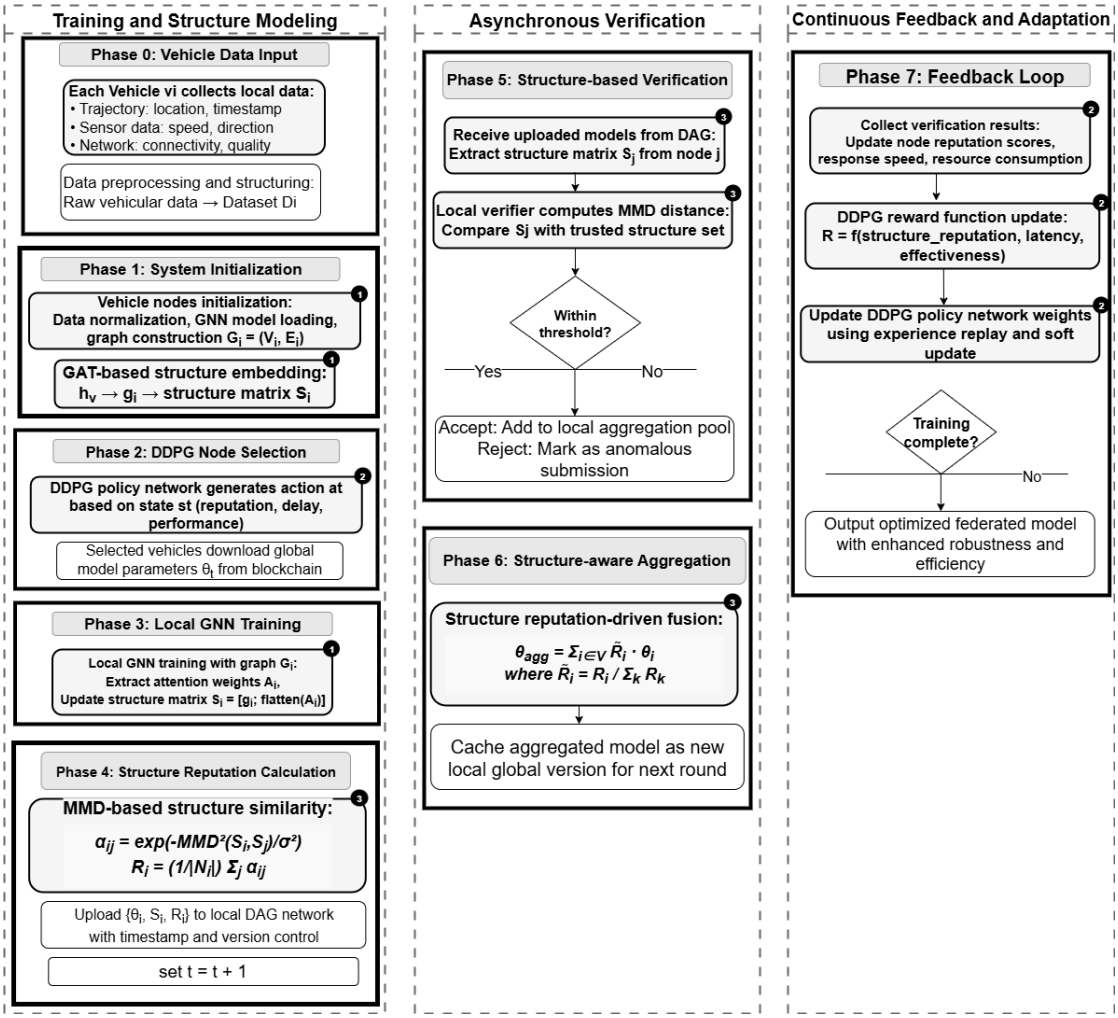


Figure 3. Framework diagram of asynchronous federated learning process driven by structural reputation.

3.4. Similarity Verification Mechanism of Structural Reputation

Traditional federated learning relies on a synchronous learning mechanism to update the model between the server and the client. However, this approach faces two major challenges in the vehicle environment. First, the learning time of each vehicle in the vehicle

network varies significantly due to heterogeneous computing power and dynamic communication conditions. Therefore, the running time of each learning iteration is determined by the slowest participant, and other vehicles need to wait for the slowest participant to maintain the synchronization mechanism. We propose asynchronous federated learning to solve this problem, and we achieve asynchronous learning by optimizing the selection of participating nodes and dividing the aggregation time slot into two stages: local aggregation and global aggregation.

Secondly, the parameters transmitted between participating nodes bring serious security and privacy issues, and the low communication reliability caused by dynamic channel conditions further exacerbates the reliability problem of these parameter transmissions. We integrate a GNN to store and verify model parameters, which can enhance the reliability and security of the proposed scheme. Specifically, the edge weight matrix W_i is extracted from the trained Graph Attention Network (GAT) model by accessing the learned attention coefficients α_{ij} between node pairs. After local GNN training on vehicle v_i's trajectory data, we extract the attention weight matrix from the final GAT layer using $W_i = [\alpha_{ij}]_{n \times n}$, where α_{ij} represents the normalized attention score between nodes i and j. This matrix captures the learned structural relationships and interaction strengths within the vehicle's local data distribution, serving as a compact representation of the model's understanding of spatial–temporal traffic patterns [19]. In addition, we adopt an evaluation mechanism based on graph neural network edge weight similarity to evaluate the quality and credibility of the contribution of participants by analyzing their GNN model structure. To prevent edge weight information from being exploited by malicious attackers for model inversion or privacy inference attacks, we introduce a differential privacy mechanism during edge weight calculation, adding calibration noise to the attention coefficients to protect the data privacy of participating vehicles. This privacy-preserving approach maintains the effectiveness of structural similarity assessment while ensuring that individual vehicle data patterns are not leaked through edge weights [45].

The proposed asynchronous federated learning architecture based on GNN edge weight similarity contains four main stages: participant selection, GNN local training, edge weight similarity evaluation, and global aggregation. In the participant selection stage, an optimization algorithm is used to select suitable vehicles to participate in federated learning. Then, the selected vehicles perform local training based on the GNN to generate a local model containing edge weight information for global aggregation.

- Participant selection: In order to improve the operation efficiency and training accuracy, the participant selection stage selects nodes with a higher resource amount within a given communication time to participate in federated learning. The selected nodes also serve as validators. At the beginning, the server initializes the federated learning process by selecting a global machine learning model and initializing the parameters θ_{init}. Then, the server selects the optimal node $\mathcal{V}_P \subset \mathcal{V}_{\text{part}}$ with high computing and communication capabilities $\xi_i \cdot \tau_i$ through an algorithm based on DRL and distributes the parameter vector θ to each node $v_i \in \mathcal{V}_P$.
- GNN local training: Local training is implemented using a distributed gradient descent algorithm based on graph neural networks. In iteration t, each participating vehicle $v_i \in \mathcal{V}_P$ trains a local GNN model $\theta_i^{(t)}$ on its data D_i according to θ^{t-1} and updates the model by calculating the local gradient descent $\nabla \mathcal{F}_i(\theta^{t-1})$:

$$\theta_i^{(t)} = \theta^{(t)} - \eta \cdot \nabla \mathcal{F}_i(\theta^{t-1}) \tag{7}$$

where η is the learning rate of distributed gradient descent. Vehicle v_i then sends the trained local model parameters $\theta_i^{(t)}$ to nearby RSUs and uploads them for further verification and aggregation.

- Edge weight similarity evaluation: This is the core innovation of our method. Each participating vehicle v_i uses local data to train the GNN model and extract the edge weight matrix $\mathbf{W}_i$. By calculating the similarity between the edge weight matrices of different vehicles, evaluation of the contribution quality of the participants and the mutual trust score is achieved.
- Aggregation: The aggregator retrieves updated local parameters from the permissioned blockchain and performs global aggregation by aggregating the local models $\theta_i^{(t)}$ of participating nodes into a weighted global model $\theta^{(t)}$:

$$\theta^{(t)} = \frac{\sum_{i=1}^{N} S_i \theta_i^{(t)}}{\sum_{i=1}^{N} S_i} \tag{8}$$

where N is the number of participating nodes, and S_i is the contribution weight of node i to the entire training process in iteration t based on edge weight similarity evaluation.

In the proposed graph-enhanced federated learning scheme, we improve the aggregation efficiency by dividing the aggregation phase into local aggregation and global aggregation phases. For each vehicle v_i, local aggregation is performed asynchronously within the local vehicle scope to improve the quality of training local models. Global aggregation is performed synchronously by RSUs, consuming more computational and communication resources. We further propose an asynchronous FL scheme to perform combined aggregation, including lightweight local aggregation and resource-intensive global aggregation.

3.5. Materiality Assessment Based on MMD

MMD is a non-parametric statistic that measures the difference between two probability distributions. It is widely used in tasks such as distribution matching, domain adaptation, and generative model evaluation. Compared with traditional KL divergence and JS divergence, MMD has the advantages of no need for density estimation and robustness to sample dimensions and sizes, and it is suitable for similarity measurement between complex structural data.

Suppose there are two distributions $\mathcal{P}$ and $\mathcal{Q}$, which represent the structural representation distributions of two vehicle nodes u_i and u_j, respectively. We hope to evaluate the differences between these two distributions at the structural level. The core idea of MMD is to map the distribution to the mean embedding in the Reproducing Kernel Hilbert Space (RKHS), then calculate the norm between the two mean embeddings as an indicator of the distribution difference. It is defined as follows:

$$\mathrm{MMD}(\mathcal{P}, \mathcal{Q}; \mathcal{H}) = \left\| \mathbb{E}_{x \sim \mathcal{P}}[\phi(x)] - \mathbb{E}_{y \sim \mathcal{Q}}[\phi(y)] \right\|_{\mathcal{H}} \tag{9}$$

where $\phi(\cdot)$ represents the mapping function from the input space to RKHS, $\mathcal{H}$ is the corresponding Hilbert space, and $\mathbb{E}$ represents the mathematical expectation. This distance is 0 if and only if $\mathcal{P} = \mathcal{Q}$, so MMD can be used as a strict distribution consistency criterion.

In practical applications, we do not directly obtain the distribution itself, but we extract a finite sample from the distribution. Assume that the structure matrix sample of vehicle u_i is $\{\mathcal{S}_i^{(1)}, \ldots, \mathcal{S}_i^{(m)}\}$, and the structure matrix sample of u_j is $\{\mathcal{S}_j^{(1)}, \ldots, \mathcal{S}_j^{(n)}\}$. Then, the unbiased estimator of MMD can be expressed as Equation (10):

$$\widehat{\mathrm{MMD}}^2(\mathcal{S}_i, \mathcal{S}_j) = \frac{1}{m(m-1)} \sum_{a \neq b} k(\mathcal{S}_i^{(a)}, \mathcal{S}_i^{(b)}) + \frac{1}{n(n-1)} \sum_{a \neq b} k(\mathcal{S}_j^{(a)}, \mathcal{S}_j^{(b)}) - \frac{2}{mn} \sum_{a=1}^{m} \sum_{b=1}^{n} k(\mathcal{S}_i^{(a)}, \mathcal{S}_j^{(b)}) \tag{10}$$

where $k(\cdot, \cdot)$ is the kernel function. A Gaussian kernel (RBF) or linear kernel is often used in the following form:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \tag{11}$$

This kernel function can embed the original structure into a nonlinear mapping into a high-dimensional space, making the differences between structures in this space more separable.The computational complexity of the MMD-based similarity check is $O(m^2 \times d)$, where m is the sample size and d is the structure matrix dimensionality, making it efficient for real-time vehicular applications [46].

In this paper, the structure uploaded by the vehicle node is represented as a fixed-length vector $\mathcal{S}_i$ whose samples can come from the structure sampling of the same vehicle in different time periods, or from its structure snapshots in multiple rounds of local training. By comparing the sample sets of $\mathcal{S}_i$ and $\mathcal{S}_j$ and substituting them into the above formula, the MMD between the structure distributions can be calculated to determine whether the two vehicle nodes are similar at the structural level.

It is worth noting that MMD, as an unsupervised statistic, does not rely on any labels or model outputs; it only makes distribution judgments based on the structure embedding itself. This feature makes it very suitable for heterogeneous environments in federated learning where labels cannot be shared due to privacy protection, and it becomes the key basis for realizing the structure reputation mechanism.

The next section will introduce how to construct the structure reputation value between vehicle nodes based on the above MMD results, and this value will be used in subsequent node selection and weight allocation strategies.

3.6. Explainable Reputation Mechanism Based on DAG Verification

Based on the calculation results of the MMD described in the previous section, this paper further constructs the structural reputation score of the vehicle node to measure the credibility of its local model structure in the neighborhood. This reputation will play a key role in node selection and aggregation weight allocation, aiming to improve the robustness and security of the system under heterogeneous and non-ideal data conditions. The specific calculation architecture is shown in Figure 4.

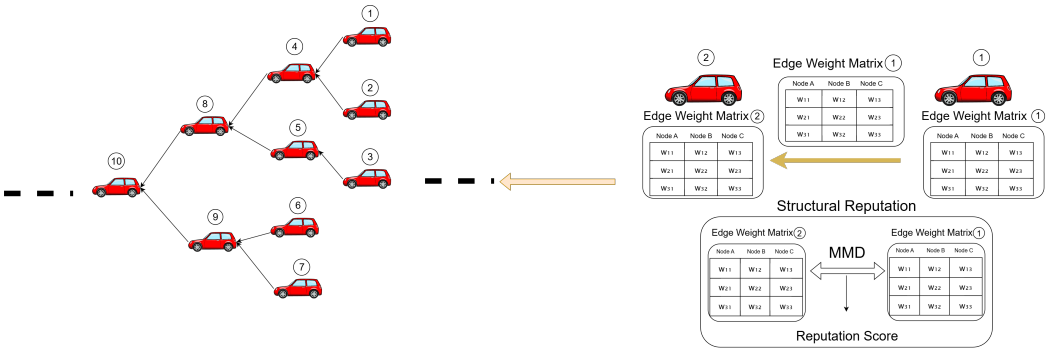


Figure 4. Schematic diagram of the verification mechanism between the structure reputation calculation and the vehicle node.

Assume that the structure matrix of the vehicle node u_i is $\mathcal{S}_i$ and its neighborhood vehicle set is $\mathcal{N}_i$, then the structural similarity between u_i and any neighbor $u_j \in \mathcal{N}_i$

can be obtained by calculating $\mathrm{MMD}(\mathcal{S}_i, \mathcal{S}_j)$. In order to map the distance value into a positive "reputation" score, this paper uses the Gaussian similarity function to transform the following:

$$\alpha_{ij} = \exp\left(-\frac{\mathrm{MMD}^2(\mathcal{S}_i, \mathcal{S}_j)}{\sigma^2}\right) \tag{12}$$

where σ is the kernel width parameter, which is used to control the attenuation degree under different MMD ranges. This function ensures that when the structure is more similar (MMD approaches 0), the reputation score is higher; otherwise, it approaches 0.

Furthermore, to obtain the overall structural reputation value R_i of u_i, we average its similarity with all neighboring nodes:

$$R_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \alpha_{ij} \tag{13}$$

This calculation method can be regarded as a local structural consistency aggregation, reflecting the structural mainstream and credibility of u_i in its neighborhood.

To facilitate unified comparison and subsequent fusion in the system, we normalize the structural reputation values of all nodes. Let:

$$\widetilde{R}_i = \frac{R_i}{\sum_{k \in \mathcal{P}} R_k} \tag{14}$$

where $\mathcal{P}$ represents the set of nodes currently participating in federated training. The normalized structural reputation value $\widetilde{R}_i$ can be directly used as the aggregation weight, structural factor in the reward function, or used to remove and filter abnormal nodes.

It is worth noting that the above reputation value construction process does not rely on any label or accuracy feedback, and it is completely measured based on the structural feature relationship between vehicle nodes. Therefore, even in the case of malicious uploads and forged accuracy of nodes, the structural reputation value can remain relatively stable and discriminative. At the same time, by introducing the exponential form of the kernel function, the mechanism can have a certain tolerance for small structural differences, enhancing the actual deployment robustness of the system.

The definition of structural reputation value proposed in this section provides a quantifiable indicator for the construction of reward functions and optimization of model fusion strategies in subsequent reinforcement learning.

3.7. Local DAG Framework Design Driven by Structural Reputation

With the widespread application of IoV in intelligent transportation, autonomous driving, and other fields, massive vehicle data needs to be effectively integrated to support model training. However, due to the privacy sensitivity of data, traditional centralized training methods cannot meet the privacy protection requirements. Therefore, in recent years, federated learning has gradually become the mainstream paradigm for collaborative processing of IoV data. Under the federated learning architecture, each vehicle node only needs to complete model training and upload model parameters locally, thereby ensuring the privacy principle that the original data does not leave the local area.

However, in practical applications, the data types held by vehicle nodes vary greatly, and there is significant heterogeneity in data quality and structural distribution, resulting in large deviations in the accuracy or robustness of some model updates. If we blindly pursue training accuracy while ignoring its structural reliability, it may have a serious impact on the generalization ability of the aggregation model. Therefore, it is not enough to measure its credibility only based on model accuracy.

In order to improve the adaptability and security of federated learning in the IoV environment, this paper proposes a new local DAG transaction verification mechanism based on structural reputation. This mechanism extracts and encodes the structural features of local vehicle data through GNN, then converts the structural features into reputation scores, which drives the entire process of model aggregation and transaction consensus.

We designed a lightweight DAG structure, which is deployed on the local side of each participating vehicle. This structure uses "transaction" as the basic unit, and each transaction represents a local model update operation, that is, the model parameter change generated after the vehicle completes local training. In order to distinguish it from traditional data upload transactions, we call it a "micro-transaction".

Compared with the chain structure, DAG allows multiple transactions to be submitted and verified in parallel, which is more suitable for asynchronous, dynamic, and low-latency IoV environments. At the same time, since each vehicle can maintain a local DAG view, this mechanism has good scalability and fault tolerance.

In the tth round of federated learning, the system selects a vehicle from the participating vehicle set $\mathcal{V}_P \subset \mathcal{V}_I$ as the aggregation node v_a, which will be responsible for collecting model updates uploaded by its neighboring nodes. Participating vehicles $v_i \in \mathcal{V}_P$ will broadcast their model updates $m_i(t)$ to other vehicles through V2V communication, and the transaction will be added to the local DAG structure with a structural reputation certificate.

The transaction structure contains the following core fields:

- Model update parameters $m_i(t)$
- Structural reputation value R_i
- Local training time s_i
- Referenced parent transaction hash list
- GNN edge weight matrix summary

To clearly demonstrate the local DAG verification process driven by structural reputation, Figure 5 shows the asynchronous transaction structure formed via the interweaving of the information chain and reputation chain in vehicle nodes.

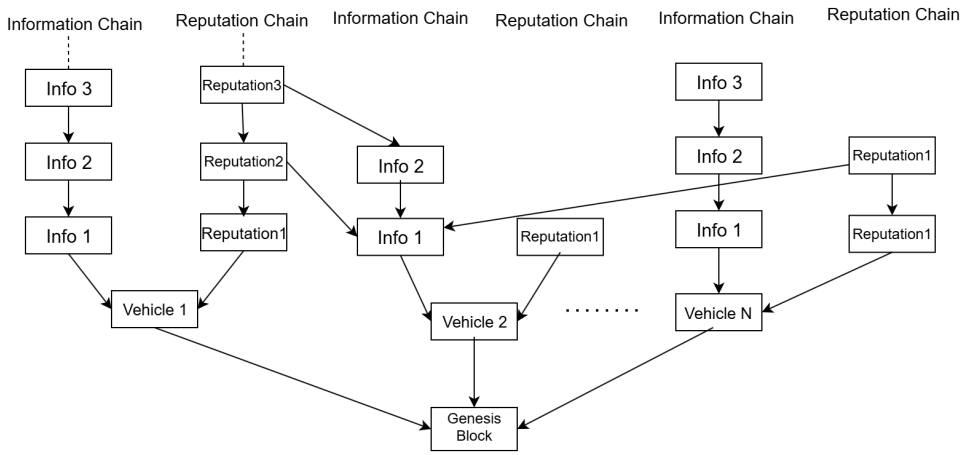


Figure 5. A schematic diagram of the coordinated structure of the information chain and the reputation chain in the local DAG.

In traditional design, transaction weight mainly depends on model accuracy and training time, but this cannot effectively reflect the distribution differences and anomaly detection capabilities in the structure. To this end, we redefine the structural reputation weight $W(m_i(t))$ of the transaction as follows:

$$W(m_i(t)) = \frac{R_i \cdot s_i}{\sum_{k=1}^{N} R_k \cdot s_k} \tag{15}$$

where R_i is the structural reputation value extracted by vehicle v_i using its local GNN model, reflecting the structural credibility of its local data, s_i is the actual training time or number of cycles invested by the vehicle in the current training round, and the denominator is the weighted sum of all participating vehicles, used for normalization.

The calculation method of the structural reputation value R_i will be explained in detail later. The core idea is that by comparing the edge weight matrices between vehicle nodes and introducing the MMD to measure their structural similarity, it is finally reduced to a score in the $[0, 1]$ interval to represent the credibility of the structural distribution.

This design has two significant advantages:

1. Structural perception ability: Even if the model trained by a node performs well in terms of accuracy, if its data structure deviates seriously from the mainstream vehicle, the structural reputation will be automatically reduced to avoid misleading model aggregation;
2. Improved anti-attack ability: Through the global comparison of structural distribution, forged or atypical data inputs can be effectively identified, thereby enhancing the security and robustness of the entire system.

In summary, the transaction weight calculation mechanism based on structural reputation effectively replaces the traditional accuracy-oriented strategy and introduces structural sensitivity while keeping the computational overhead controllable. By binding each model update to structural reputation through local DAG, each vehicle is not only a provider of the model but also a participant in transaction verification.

Next, we will discuss in depth how to construct the cumulative weight driven by structural reputation under this mechanism and how to implement the verification and confirmation process of transactions.

3.8. Cumulative Weight and Transaction Verification Mechanism Under Structural Reputation

In the construction of a local DAG model verification mechanism with structural reputation as the core, relying solely on the reputation weight of a single transaction cannot fully reflect its credibility in the entire system. Due to the high degree of asynchrony and dynamic nature of vehicle participation in the IoV environment, there is a lack of global synchronization view between nodes. Therefore, it is necessary to design a transaction weight indicator with more global reference significance. To this end, this paper proposes the concept of Cumulative Structural Reputation Weight (CSRW) to comprehensively reflect the degree of structural recognition of a transaction in the entire DAG network.

When each transaction is added to the DAG, it will carry the local structural reputation value R_i of its issuing node v_i, which is obtained by comparing the structural distribution of multiple reference nodes after the vehicle processes its local graph structure through the GNN. This reputation value can be understood as the "normalized credibility" of the vehicle data structure in the overall network.

However, due to the timing and latency of the network, other nodes' trust in the transaction cannot be based solely on the reported structural reputation value. Therefore, we designed a **multi-source verification mechanism** to re-evaluate the transaction with the help of the structural consensus of historical transactions in the DAG network. The specific definition is as follows:

$$CW(m_i(t)) = W(m_i(t)) + \frac{1}{M} \sum_{j=1}^{M} \Delta R_{ij} \cdot W(j) \tag{16}$$

where $CW(m_i(t))$ represents the cumulative structural reputation weight of the model update transaction submitted by the ith node at time t, $W(m_i(t))$ is the basic structural repu-

tation weight determined by the vehicle's local structural reputation and training resources, M is the number of historical verification transactions, $\Delta R_{ij} = R_j(m_i(t)) - R_i(m_i(t))$ represents the difference between the structural reputation recalculated by other verification nodes v_j for the transaction and the original submitter, and $W(j)$ represents the reputation weight of transaction j itself, used as an amplification factor of its verification value.

The introduction of this cumulative weight achieves the following three effects:

1. Enhanced stability: If a transaction structure has a high reputation value and is repeatedly verified by multiple high-reputation nodes, it will eventually gain a significant weight increase;
2. Suppress structural deviation: If a transaction has a large gap with the structural reputation value calculated by other nodes, ΔR_{ij} is negative or close to zero, which means that there is a lack of structural consensus, and its final weight will also decrease;
3. Improve anomaly detection capability: Some malicious nodes may upload model updates with forged structures, but because they cannot pass multi-source verification, their cumulative reputation will quickly decrease and be excluded from the main aggregation path.

Different from the traditional trust weight that relies on accuracy, the structural reputation mechanism provides a new "semantic perspective" that starts from the data distribution itself, more effectively distinguishes abnormal or low-value model updates, and improves the robustness of the overall model training quality.

To ensure the integrity and continuity of the DAG network, each new transaction must be structurally verified against two existing transactions before it is officially added. The verification process mainly includes the following steps:

1. Transaction selection: Node v_i starts the structural reputation verification task and selects two candidate transactions m_u, m_v from the local DAG;
2. Structure matching evaluation: The local GNN model is used to process the data features of the two transactions, respectively, and calculate their structural similarity $R_i(m_u), R_i(m_v)$ with the data of this node;
3. Reputation difference judgment: The locally evaluated structural reputation value is compared with the structural reputation R_u, R_v attached to the original transaction. If the difference $|R_i - R_u| < \delta$, it is considered to be verified;
4. Transaction reference: After verification, the node points the hash of the new transaction to the above two transactions, forming a new edge in the DAG;
5. Structural feature broadcast: The node broadcasts the structural summary information of the new transaction to the surrounding vehicles for reference by subsequent verifiers.

This process not only ensures the structural rationality of the new transaction but also distributes the structural verification task to the entire network in a "weakly centralized" manner, greatly improving the scalability and real-time performance of the system.

In order to improve the coverage and fairness of transaction verification, we introduce a random walk mechanism driven by structural reputation to assist in selecting high-quality transactions for reference. The specific approach is as follows: when a node constructs a new transaction, it starts from any node x in the DAG and performs several rounds of Markov random walks. The transition probability of each round of walks is defined as follows:

$$P_{xy} = \frac{e^{CW(y)-CW(x)}}{\sum_{z\in\mathcal{N}(x)} e^{CW(z)-CW(x)}} \tag{17}$$

where x is the current node, y is the predecessor node pointing to x, $\mathcal{N}(x)$ is the set of all transactions pointing to x, and $CW(y)$ is the cumulative weight of the structural reputation of transaction y.

The essence of this walking process is to shift to the transaction path with structural consensus and high credibility, ensuring that the reference path is stable and reliable while also increasing the chances of weak credibility transactions being verified. After the walking is completed, the node will select two end points as reference objects to form a structural authentication path, thereby enhancing the verifiability and convergence of the entire DAG network.

In the long-term operation, some transactions may lack verification paths due to sparse connections or network interference. To prevent such transactions from being permanently abandoned, this paper proposes a "reputation propagation and repair mechanism": for transactions with medium and low cumulative weights that have not been verified for a long time, the system regularly activates the reputation propagation process, allowing nodes with high reputation to actively verify structurally weak transactions on their historical paths.

This mechanism implements a structural self-recovery capability, allowing the entire DAG to continuously clean up noise transactions while maintaining the integrity and credibility of the structural reputation path.

3.9. Structural Reputation-Oriented Consensus Foundation

In the asynchronous federated learning scenario in the IoV, the heterogeneity and heterogeneity of transactions are much higher than those of traditional blockchain systems. Since the online time of vehicle nodes is not fixed and the network connection changes dynamically, a mechanism that does not require global synchronization and can adapt to the evolution of local consensus is needed. This study draws on the nonlinear structural advantages of DAG and combines structural reputation indicators to propose a consensus confirmation mechanism with "cumulative structural reputation" as the core.

Specifically, the system no longer relies on the traditional "all-node voting" or "consensus round mechanism" but forms a gradually enhanced consensus path through continuous citation and structural reputation superposition. As long as a transaction is cited by multiple exchanges with high structural reputation and continues to appear in the random verification path, it is regarded as a natural transaction with "sufficient consensus" and can be included in the aggregation candidate set.

Unlike the evaluation method based on model accuracy, structural reputation evaluation provides **stronger anti-counterfeiting ability and long-term consistency guarantee**. Even if some transactions are difficult to confirm in the early stage, as long as their structural characteristics are stable and similar, they can still be accepted by consensus through layer-by-layer propagation.

The core of transaction consensus in the DAG system is to find an optimal structural path as the main line of trusted aggregation, defined as the **heaviest structural reputation path (HRP)**: Starting from a certain initial transaction, wandering along the DAG network, the one with the largest sum of structural reputation weights of all transactions in the cumulative path is regarded as the "most trusted transaction main line" in the current network.

Assume that the reputation path of a transaction $m_i(t)$ is $P_i = \{m_i(t), m_i(t-1), \ldots, m_i(t-k)\}$. Then, its total weight is defined as follows:

$$W_{\text{path}}(m_i) = \sum_{m \in P_i} CW(m) \tag{18}$$

The confirmation conditions of transaction consensus are:

- The path length exceeds the set threshold $L_{\min}$, ensuring the stability of structural transmission;

- The total weight exceeds the threshold T_{CW}, indicating that the transaction structure has been widely accepted;
- Or, it is referenced by more than M reputation nodes at time t.

Once any of the above conditions is met, the transaction will be marked as a "structural consensus state" and can be used for subsequent global model aggregation.

This path-based consensus mechanism has the following advantages over traditional "static voting" or "centralized server decision-making":

- No need for global synchronization, adapting to the characteristics of asynchronous and distributed IoV;
- Strong anti-malicious attack capability, because the path needs to continuously accumulate reputation rather than a one-time breakthrough;
- More fair, as weak computing vehicles can also obtain verification opportunities through structural stability.

In order to reasonably explore high-reputation transaction paths in DAG, we adopt a structural reputation-guided Markov random walk strategy. Specifically, when adding a new transaction or performing verification, each vehicle will start from a certain initial transaction node and walk the path with the following transition probability:

$$P_{xy} = \frac{e^{\alpha \cdot CW(y)}}{\sum_{z \in \mathcal{N}(x)} e^{\alpha \cdot CW(z)}} \tag{19}$$

where x is the current node, $\mathcal{N}(x)$ is its directly referenced transaction set, $CW(y)$ is the cumulative structural reputation of the candidate node y, and α is a parameter for adjusting the bias strength with $\alpha > 0$, where the larger the bias is, the more it is biased towards high-reputation nodes.

After k rounds of walking, the system will take the transaction with the highest frequency in the path as a "trusted reference candidate" and use it as the next transaction verification or aggregation target.

This mechanism has three significant advantages:

1. High-quality path fast detection: High-reputation transactions are more likely to be visited by walking, thereby accelerating their consensus building speed;
2. Enhanced isolation of low-quality transactions: Transactions with abnormal structures or low reputation naturally fade out of the consensus main path due to their low probability of being visited;
3. Dynamic self-adjustment: The system can automatically adjust α according to network load, vehicle activity, or reputation, thereby achieving flexible scheduling.

In an open IoV system, attackers may attempt to mislead the aggregation process by uploading false structural models. To deal with these potential attacks, we introduced the following structural reputation defense mechanism:

1. **Structural forgery detection mechanism**: By performing randomness checks and mutual verification on the structural embedding output by the GNN model, we prevent nodes from privately constructing highly idealized edge weight matrices to defraud high reputation;
2. **Multi-round structural verification**: After each transaction is added to the DAG, it must undergo multiple rounds of cross-verification of the walk path to ensure that its structural reputation is consistently evaluated in most nodes;
3. **Reputation leakage penalty mechanism**: If a node submits revoked transactions multiple times, its reputation will be dynamically downgraded and will be cited with a low probability in subsequent walks, thereby reducing its system influence.

In addition, to ensure that transactions cannot be appended infinitely, we also introduce lightweight consensus thresholds (such as simplified proof of work sPoW or local structural entropy estimation) to prevent nodes from frequently brushing transactions and causing verification congestion.

In summary, the structural reputation-driven DAG consensus mechanism proposed in this paper realizes an asynchronous consensus method that takes into account security, efficiency, and balance by introducing accumulated reputation, heaviest path, random walk, and multiple verification mechanisms. Compared with the traditional method that relies on accuracy and central voting, this method is more in line with the structural, asynchronous, and edge computing characteristics of the IoV.

4. Deep Reinforcement Node Selection Mechanism Driven by Structural Reputation

4.1. Combinatorial Optimization of Integrated Structural Reputation and Cost Trade-Offs

In the asynchronous federated learning (AFL) environment, on-board nodes have significant heterogeneity in computing power, communication rate, data distribution, and even model structure. This heterogeneity directly affects the convergence speed and final accuracy of the global model. Therefore, how to dynamically select participating nodes in each round of training, taking into account model training efficiency, communication load, and structural synergy, has become a core issue affecting system performance.

This section aims to build a comprehensive evaluation mechanism that integrates node computing communication capabilities and structural reputation, and through this mechanism, select subset nodes suitable for participating in training in each round, thereby optimizing the overall training process. To this end, we first establish a node cost model, then we construct a constrained multi-objective optimization problem to provide a theoretical basis for the subsequent DRL strategy design.

We consider selecting a subset of nodes $\mathcal{V}_P \subseteq \mathcal{V}_I$ from the set of all vehicle nodes $\mathcal{V}_I = \{v_1, v_2, \ldots, v_N\}$ to participate in the training in the tth round of training. To indicate whether a node is selected, we define a binary vector:

$$\boldsymbol{\lambda}^t = [\lambda_1^t, \lambda_2^t, \ldots, \lambda_N^t], \tag{20}$$

where if $\lambda_i^t = 1$, it means that vehicle node v_i is selected to participate in the tth round of training; if $\lambda_i^t = 0$, the node does not participate.

(1) Local training cost:

This cost is determined by the amount of data and computing power. Assume that the local data volume of node v_i is d_i, the computing cycle required for unit data training is β_m, and its computing resources at time t are $\xi_i(t)$. Then, the training cost can be defined as follows:

$$c_i^{\text{train}}(t) = \frac{d_i \cdot \beta_m}{\xi_i(t)}. \tag{21}$$

(2) Communication cost:

Assume that the uploaded model size of node v_i is $|w_i|$, and the channel bandwidth between it and the edge server is $\tau_i(t)$. Then, the communication cost is as follows:

$$c_i^{\text{comm}}(t) = \frac{|w_i|}{\tau_i(t)}. \tag{22}$$

(3) Structural reputation cost:

Structural reputation aims to measure the synergy between the node model structure and other nodes. We assume that each node v_i can be trained to obtain its structural connection matrix $R_i \in \mathbb{R}^{n\times n}$, which is used to reflect its local feature map representation. Using the MMD to measure structural differences, the structural reputation score of node v_i is as follows:

$$\rho_i = \frac{1}{|\mathcal{V}_P| - 1} \sum_{j \in \mathcal{V}_P \setminus \{i\}} \big(1 - \mathrm{MMD}(R_i, R_j)\big). \tag{23}$$

(4) Comprehensive cost function:

Combining the three costs, we define the total cost of node v_i as follows:

$$C_i^t = \alpha_1 \cdot c_i^{\text{train}}(t) + \alpha_2 \cdot c_i^{\text{comm}}(t) - \alpha_3 \cdot \rho_i. \tag{24}$$

The total system cost function is:

$$\mathcal{C}(\boldsymbol{\lambda}^t) = \frac{1}{M} \sum_{i=1}^{N} \lambda_i^t \cdot C_i^t \tag{25}$$

where $M = \sum_{i=1}^{N} \lambda_i^t$ is the number of nodes involved in training.

(5) Geographic location constraint:

Assume that the geographic location of node v_i at time t is $p_i(t) \in \mathbb{R}^2$, and the centroid of the selected node is:

$$p_c(t) = \frac{1}{M} \sum_{i=1}^{N} \lambda_i^t \cdot p_i(t). \tag{26}$$

To control communication delay, we introduce the following constraints:

$$\|p_i(t) - p_c(t)\|^2 \le r_0^2, \quad \forall i \text{ with } \lambda_i^t = 1. \tag{27}$$

(6) Optimization modeling:

The final node selection problem can be formalized as Equation (28):

$$\begin{aligned} \min_{\boldsymbol{\lambda}^t} \quad & \mathcal{C}(\boldsymbol{\lambda}^t) \\ \text{s.t.} \quad & \lambda_i^t \in \{0,1\}, \quad \forall i \\ & \|p_i(t) - p_c(t)\|^2 \le r_0^2, \quad \forall i \text{ with } \lambda_i^t = 1 \end{aligned} \tag{28}$$

This problem is a typical non-convex, NP-Hard combinatorial optimization problem, which is difficult to solve using traditional methods. In the next section, we will build a DRL model and learn the optimal node selection strategy based on the DDPG framework to achieve efficient approximation of the above optimization objectives.

4.2. Proposed DDPG Node Selection Algorithm Driven by Structural Reputation

To solve the combined optimization problem of the aforementioned integrated training cost, communication cost, and structural reputation mechanism, this section proposes a node selection algorithm based on Deep Deterministic Policy Gradient (DDPG) and elaborates on it from multiple levels such as neural network architecture, policy update mechanism, loss function construction, and training process. DDPG is a reinforcement learning algorithm for continuous action space, suitable for dealing with high-dimensional state and action decision problems, especially for node selection tasks in asynchronous federated learning scenarios.

The DDPG algorithm consists of four core components: primary actor network, primary critic network, target actor network, target critic network, and experience replay pool (Replay Memory). The primary actor network outputs the node selection action in the current state, and the primary critic network evaluates the value of the action; the target network structure is consistent with the primary network, which is used for stable training; the experience replay pool stores the state–action–reward–next state quadruple collected during training.

To intuitively present the execution process and module composition of the DDPG algorithm driven by the above structural reputation [47], Figure 6 shows the overall system architecture diagram.

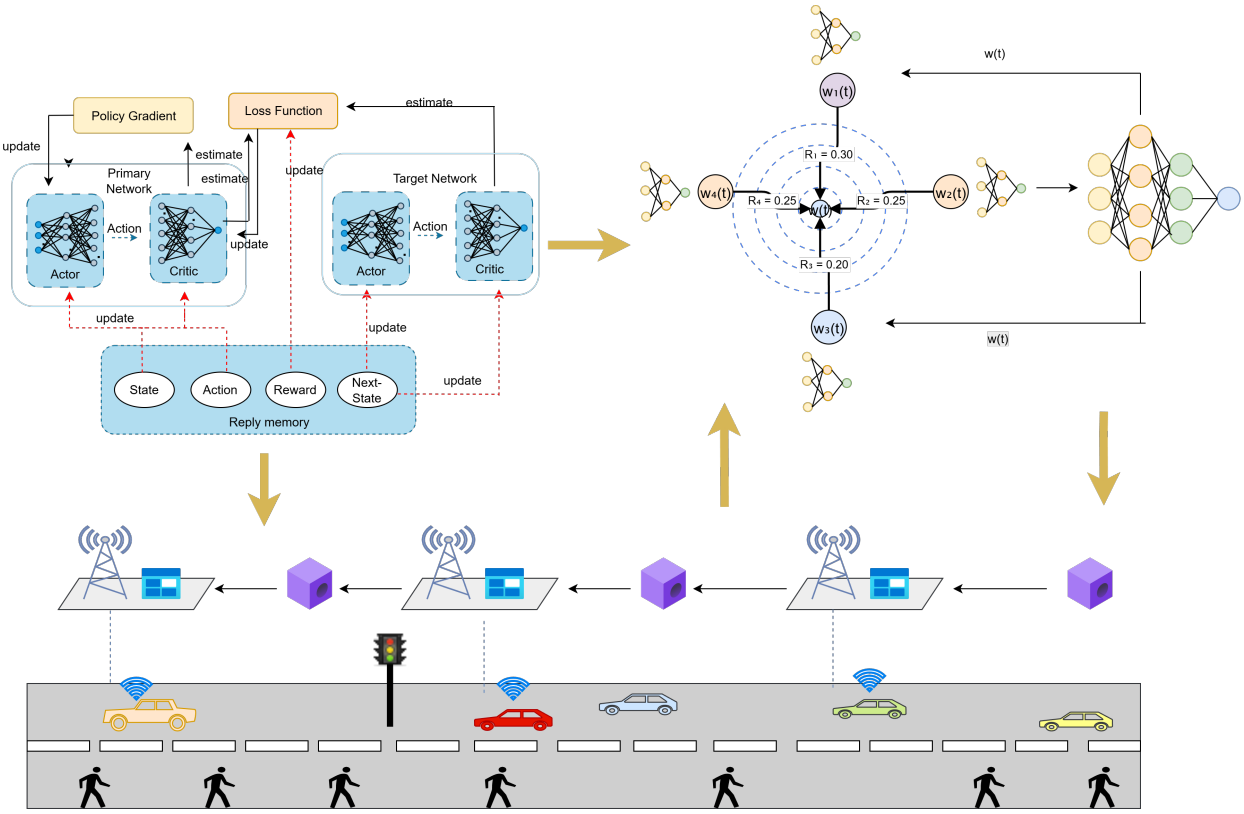


Figure 6. System architecture diagram of DDPG node selection mechanism based on structural reputation.

(1) Critic network design and training:

The evaluation network (Critic DNN) predicts the long-term benefits of selecting a certain action in the current state by learning the state-action value function $Q(s_t, \lambda_t; \theta_Q)$. We construct the target Q value based on the Bellman equation:

$$Q(s_t, \lambda_t; \theta_Q) = \mathbb{E}[r_t + \gamma Q(s_{t+1}, \pi(s_{t+1}; \theta_Q))] \tag{29}$$

where γ represents the discount factor, r_t is the immediate reward, and $\pi(s_{t+1}; \theta_Q)$ is the next action output by the target policy network.

The loss function is defined as follows:

$$\mathcal{L}_Q(\theta_Q) = \mathbb{E}[(y_t - Q(s_t, \lambda_t; \theta_Q))^2] \tag{30}$$

The target value y_t is defined as follows:

$$y_t = r_t + \gamma Q(s_{t+1}, \pi(s_{t+1}; \theta_{\pi'})) \tag{31}$$

The optimization goal is to minimize $\mathcal{L}_Q$, and the updated gradient is as follows:

$$\nabla_{\theta_Q}\mathcal{L}_Q = \mathbb{E}[2(y_t - Q(s_t, \lambda_t; \theta_Q)) \cdot \nabla_{\theta_Q} Q(s_t, \lambda_t)] \tag{32}$$

(2) Actor network design and update strategy:

The goal of the policy network (Actor DNN) is to output the optimal node selection vector $\lambda_t = \pi(s_t; \theta_\pi)$, which is optimized by maximizing the Q value of the corresponding Critic network. The policy gradient is given as follows:

$$\nabla_{\theta_\pi} J \approx \mathbb{E}[\nabla_\lambda Q(s, \lambda; \theta_Q)|_{\lambda=\pi(s)} \cdot \nabla_{\theta_\pi} \pi(s)] \tag{33}$$

The network parameters are updated as follows:

$$\theta_\pi \leftarrow \theta_\pi + \alpha_\pi \cdot \nabla_{\theta_\pi} J \tag{34}$$

where α_π is the learning rate of the policy network and J is the expected overall return.

(3) Reward function design—structural reputation-driven mechanism:

The core innovation of this study lies in the design of the reward function, which explicitly introduces the structural reputation score ρ_i as an important reference indicator. For the currently selected node set $\mathcal{V}_P$, the immediate reward function is defined as follows:

$$R(s_t, \lambda_t) = -\sum_{i=1}^{N} \lambda_i^t \cdot \left(\alpha_1 c_i^{\text{train}}(t) + \alpha_2 c_i^{\text{comm}}(t) - \alpha_3 \rho_i\right) \tag{35}$$

where $c_i^{\text{train}}(t)$ represents the training cost of node v_i, $c_i^{\text{comm}}(t)$ denotes the communication cost, ρ_i is the structural reputation score of node v_i, and $\alpha_1, \alpha_2, \alpha_3$ are weighting factors that control the relative importance of each objective component.

This reward function inherently resolves conflicting objectives through a mathematical structure, penalizing the cost component while rewarding reputation score. This creates a natural tension that guides the DDPG agent toward a Pareto-optimal solution. The weight factors $\alpha_1, \alpha_2, and \alpha_3$ serve as key tuning parameters, enabling a dynamic balance between cost efficiency and reputation quality based on operational requirements [45].

5. Experiments

5.1. Dataset

We evaluate the proposed federated learning scheme on the ApolloScape trajectory dataset (available at https://apolloscape.auto/trajectory.html, accessed on 12 October 2025) [48], which is designed for trajectory prediction in complex urban environments and provides an ideal platform for verifying federated learning algorithms in autonomous driving scenarios where data privacy and distributed computing are critical. The ApolloScape dataset is collected by Apollo vehicles during peak urban hours, containing 53 min of training sequences and 50 min of test sequences sampled at 2 fps. Each traffic participant includes comprehensive annotations: target ID, type, location, size, and heading angle. The dataset covers five traffic target types (small vehicles, large vehicles, pedestrians, motorcycles, cyclists, and others), creating a heterogeneous data distribution that aligns with federated learning assumptions where different clients represent different geographic locations or vehicle types. The urban traffic environment presents unique challenges ideal for evaluating federated learning in autonomous driving contexts. Complex interactive behaviors arise from varying speeds, directions, and physical characteristics of traffic participants, such as small vehicles navigating narrow spaces and large vehicles following

conservative paths. In our experimental setup, we partition the ApolloScape dataset to simulate real-world distributed scenarios, where each subset represents different participants (vehicles, road infrastructure, or geographic areas) with distinct data characteristics. This partitioning strategy enables realistic evaluation of the proposed algorithm's ability to handle data heterogeneity and privacy protection in V2X communication environments while maintaining prediction accuracy.

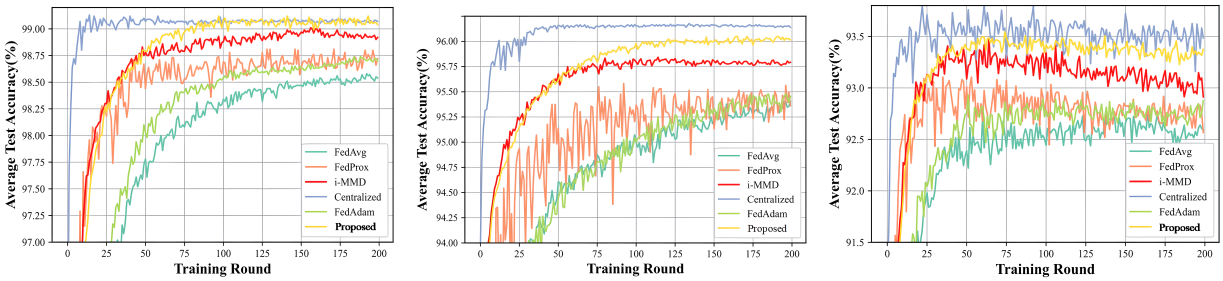
5.2. Experimental Results

We first evaluate the accuracy and convergence performance of the proposed scheme with different numbers of participants. In order to fully verify the effectiveness of the algorithm, we designed multiple sets of comparative experiments, including performance comparison of synchronous algorithms, performance of asynchronous algorithms at different degrees of asynchrony, algorithm running time analysis, and robustness testing under malicious node attacks.

Performance Analysis of Synchronization Algorithm

To analysis synchronization algorithm performance, Figure 7 gives the performance comparison results of six different federated learning algorithms with different numbers of participants. The experiment sets three different participant scales: (a) $n = 90$, (b) $n = 60$, (c) $n = 30$ to evaluate the performance of the algorithm under different network scales. The algorithms involved in the comparison include:

- FedAvg (blue line): Federated average algorithm, which is the basic method of federated learning. It updates the global model by simply averaging the local training models of each client;
- FedProx (red line): Federated proximal algorithm, which is an extended version of FedAvg. It introduces proximal terms in the objective function to deal with system heterogeneity and stabilizes the training process by limiting the degree to which local updates deviate from the global model;
- i-MMD (green line): An algorithm based on MMD, which is used to measure the data distribution differences between different data providers. The smaller the i-MMD value, the more similar the data is, and the smaller the impact on model accuracy;
- Centralized (purple line): Centralized learning method, as a reference for the upper bound of performance;
- FedAdam (brown line): Federated adaptive optimization algorithm. The optimizer is introduced into the federated learning framework, combining adaptive learning rate and decentralization characteristics;
- Proposed (yellow line): The algorithm proposed in this paper.



(**a**) $n = 90$ (**b**) $n = 60$ (**c**) $n = 30$

Figure 7. Performance of synchronous algorithms with different participant numbers.

From the experimental results, it can be observed that the algorithm proposed in this paper shows the best performance under all the settings of the number of participants,

and the final accuracy is stable at about 99%. Taking $n = 90$ as an example, the algorithm reaches an accuracy of 98.5% after the 25th round of training and stabilizes after the 50th round; in contrast, the final accuracy of FedAvg is only about 97.75%, and the convergence speed is also significantly slower, requiring nearly 100 rounds of training to stabilize.

FedProx has improved its convergence stability after introducing the proximal term, and its final accuracy is 98.25%, but it is still about 0.75% lower than the algorithm in this paper. It is worth noting that the convergence curve of FedProx fluctuates less, which verifies its training smoothness advantage in the context of system heterogeneity.

i-MMD uses data distribution differences to guide training, reaching 98.5% when $n = 90$, slightly better than FedProx, but the performance drops significantly when there are fewer participants (such as $n = 30$), indicating that it is more sensitive to data size.

FedAdam shows faster learning speed in the early stage of training with its adaptive learning rate, but there is some fluctuation in the later convergence stage, and its final accuracy is 98.75% when $n = 90$.

As the number of participants decreases, the performance of all algorithms decreases, showing a trend of positive correlation between data volume and model performance. When $n = 60$, the accuracy of this algorithm drops slightly to 98.5%, while FedAvg drops to about 96.5%; when $n = 30$, the accuracy of this algorithm still maintains about 96%, while FedAvg drops to only 92.5%, and the performance gap further widens to 3.5%.

From the perspective of convergence speed, the algorithm proposed in this paper can reach the final performance within 25–50 rounds, which is significantly better than the trend that other algorithms generally need 100–150 rounds to converge. For edge devices, this fast convergence capability can effectively reduce the consumption of communication and computing resources, which has important practical significance.

In addition, as the theoretical performance upper limit, the centralized method can achieve an accuracy of about 99.5% under all settings. There is only a gap of about 0.5% between the algorithm proposed in this paper and the centralized result, and it still achieves near-optimal performance while protecting data privacy.

To further validate the advantages of our proposed algorithm, we conducted comprehensive comparisons with state-of-the-art trust-based federated learning methods. Table 1 presents the performance comparison results with six advanced baseline methods under different participant scales.

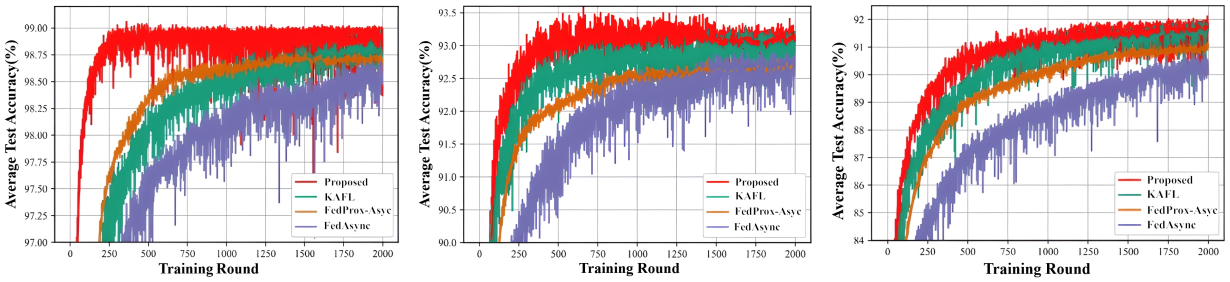
Table 1. Performance comparison with enhanced baseline methods.

Participants	Proposed	PoQRBFL	PoTQBFL	FoolsGold	FLTrust	LAFED
$n = 30$	**0.936**	0.921	0.909	0.922	0.919	0.915
$n = 60$	**0.962**	0.957	0.917	0.931	0.942	0.948
$n = 90$	**0.992**	0.974	0.963	0.979	0.966	0.977

The baseline methods include PoQRBFL (Quality and Reputation-based Federated Learning), PoTQBFL (Trust Quality-based Federated Learning), FoolsGold (a federated learning method for defending against Sybil attacks), FLTrust (a trust evaluation method based on server-side validation), and LAFED (Locally Aggregated Federated Learning). The experimental results demonstrate that our proposed algorithm achieves optimal performance across all participant scales. Specifically, when the number of participants is 30, our algorithm achieves 93.6% accuracy, representing improvements of 1.4–2.1% compared to the baseline methods, with the largest improvement of 2.1% over LAFED. When the number of participants increases to 90, the performance advantage becomes more pronounced, reaching 99.2% accuracy and showing improvements of 1.3–2.9% over all baseline methods, with the most significant improvement of 1.5% over the second-best performing LAFED

method. These results validate the effectiveness of our structural reputation mechanism in trust evaluation and demonstrate its consistent stability advantages across different network scales and against diverse baseline approaches.

To further analyze the algorithm's performance in asynchronous environments, Figure 8 shows the algorithm performance comparison under different asynchrony degree parameters α.



(**a**) $\alpha = 5$ (**b**) $\alpha = 3$ (**c**) $\alpha = 1$

Figure 8. Algorithm performance of asynchronous algorithms under different asynchrony degrees.

In order to analyze the model effect of the proposed algorithm at different degrees of asynchrony, Figure 8 gives the performance of asynchronous learning algorithms under different asynchrony degrees (α values). We set three different asynchrony degrees: (a) $\alpha = 5$ (high asynchrony), (b) $\alpha = 3$ (medium asynchrony), (c) $\alpha = 1$ (low asynchrony), and we extend the training rounds to 2000 rounds to observe the long-term convergence behavior. The asynchronous algorithms involved in the comparison include:

- Proposed (red line): the asynchronous federated learning algorithm we proposed;
- KAFL (green line): the knowledge-aware federated learning algorithm, which uses knowledge distillation technology to allow clients to partially share model information based on its usefulness instead of applying a complete model update;
- FedProx-Async (orange line): the asynchronous federated proximal algorithm, which combines the FedProx method with the FedAsync algorithm extension and carries proximal terms in the asynchronous setting to stabilize updates to handle the inaccuracy caused by device heterogeneity and outliers;
- FedAsync (purple line): the federated asynchronous learning algorithm, which is an asynchronous version of federated learning. Local model updates from different clients arrive at the server in a less synchronous mode, without waiting for all clients to complete local training.

The experimental results show that the algorithm we proposed maintains the highest performance under all asynchronous settings. Under high asynchrony ($\alpha = 5$), although the system allows for larger parameter update delays, our algorithm can still maintain an accuracy of about 98.75%, significantly higher than KAFL's 97.75%, FedProx-Async's 97.25%, and FedAsync's 97.50%. More importantly, our algorithm exhibits the most stable convergence behavior under this setting, with less fluctuation in the training curve, indicating that the algorithm is robust to asynchronous updates.

Under medium asynchrony ($\alpha = 3$), the performance of all algorithms improves, and our algorithm achieves an accuracy of about 94.25%, which is slightly lower than when $\alpha = 5$ but still maintains a leading position. KAFL's performance improves under this setting to 93.75%, while FedProx-Async and FedAsync reach 92.75% and 93.25%, respectively. It is worth noting that in this moderate asynchronous setting, the stability of algorithm convergence is significantly improved, and the training curves of all algorithms show better smoothness.

When the degree of asynchrony is reduced to $\alpha = 1$, the system approaches the behavior mode of synchronous learning, and the performance of all algorithms reaches the highest level. Our algorithm achieves an accuracy of about 91.5% in this setting, while other algorithms also improve accordingly: KAFL reaches 90.75%, FedProx-Async reaches 89.25%, and FedAsync reaches 90.0%. Although the absolute performance is reduced, this is mainly due to the different evaluation time points caused by the increase in training rounds in the asynchronous setting.

In terms of convergence dynamics, our algorithm shows faster initial convergence speeds in all asynchronous settings. In the first 500 rounds of training, our algorithm can reach 85–90% of its final performance, while other algorithms usually need 800–1000 rounds. This fast convergence feature is particularly important in asynchronous environments because it can reduce the performance loss caused by parameter inconsistency.

Of particular note is the stability of the algorithm during long-term training. In the 2000 rounds of extended training, we observed that KAFL showed a slight performance degradation in the later stages, which may be related to the accumulated errors in the knowledge distillation process. FedAsync showed large volatility, especially in the high asynchrony setting, which reflects the challenges of pure asynchronous methods in dealing with parameter inconsistencies. In contrast, our algorithm maintained a relatively stable performance level throughout the training process, with fluctuations within 0.5%.

From the perspective of the impact of the degree of asynchrony on the performance of the algorithm, as the value of α increases (the degree of asynchrony increases), all algorithms show a trend of performance improvement, but the improvement varies. The performance improvement of our algorithm from $\alpha = 1$ to $\alpha = 5$ is about 7.25%, while KAFL only improves by about 7.0%, FedProx-Async improves by about 8.0%, and FedAsync improves by about 7.5%. This shows that our algorithm is able to better exploit the advantages of asynchronous learning while maintaining robustness to parameter delays.

In order to analyze the model effects of different algorithms when different numbers of edge devices are involved, Figure 9 compares the average running time of different asynchronous algorithms under various numbers of participants. We tested the computational efficiency under three different participant sizes ($n = 50$, $n = 100$, $n = 200$). The algorithms involved in the comparison include:

- Proposed: our proposed algorithm;
- DBAFL: blockchain-based asynchronous federated learning algorithm;
- FedProx-Async: asynchronous federated proximal algorithm;
- FedAsync: federated asynchronous learning algorithm;
- KAFL: knowledge-aware federated learning algorithm.

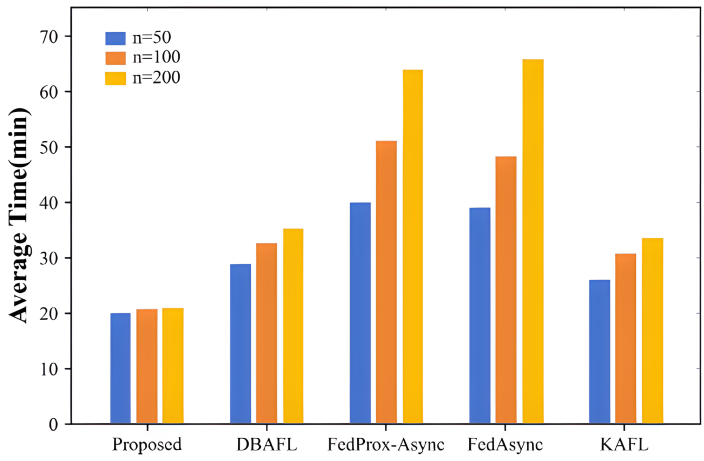


Figure 9. Algorithm running times of asynchronous algorithms by numbers of participants.

The results show that our proposed algorithm has the best time efficiency in all test scenarios. When the number of participants is 50, our algorithm takes an average of about 20 min to run, compared to about 30 min for DBAFL, about 35 min for FedProx-Async, about 28 min for FedAsync, and about 32 min for KAFL. This significant time advantage is mainly attributed to the intelligent participant selection mechanism integrated in our algorithm, which can effectively screen out high-quality participants and reduce unnecessary computational overhead.

When the number of participants increases to 100, the time efficiency gap widens further. The running time of our algorithm only increases to about 21 min, an increase of only 5%, showing good scalability. The time increase of other algorithms is more significant: DBAFL increases to about 50 min (an increase of 67%), FedProx-Async increases to about 52 min (an increase of 49%), FedAsync increases to about 62 min (an increase of 121%), and KAFL increases to about 64 min (an increase of 100%). This difference shows that our algorithm has better scalability when dealing with large-scale participant networks.

In the scenario of the largest test scale ($n = 200$), the time efficiency advantage is more obvious. The running time of our algorithm is about 22 min, which is only 10% higher than when $n = 50$, showing nearly linear scalability. In contrast, the running time of FedAsync increases dramatically to about 65 min, KAFL reaches about 63 min, and DBAFL and FedProx-Async reach about 51 min and about 35 min, respectively. It is worth noting that although DBAFL performs relatively well in large-scale settings, it is still significantly slower than our algorithm.

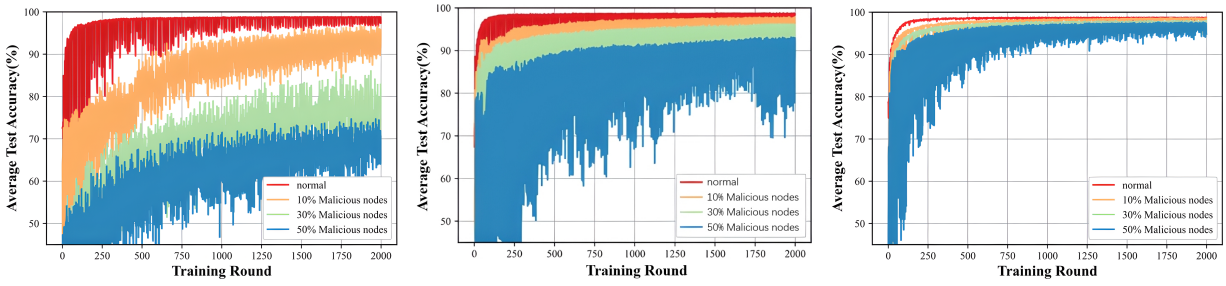
From the perspective of algorithm scalability, our algorithm exhibits the best scalability. When the number of participants increases from 50 to 200 (a 4-fold increase), the running time of our algorithm increases by only about 10%, while FedAsync increases by 132%, and KAFL increases by 97%. This excellent scalability performance is mainly due to the following factors: first, our participant selection strategy can dynamically adjust the number of selected participants to avoid processing too many redundant updates; second, the optimized parameter aggregation algorithm reduces the computational complexity; finally, the improved communication protocol reduces the impact of network latency.

From the perspective of computational efficiency, we also observe differences in the performance of different algorithms when dealing with heterogeneous participants. KAFL faces a greater computational burden when the number of participants increases due to the need for knowledge distillation calculations. The main bottleneck of FedAsync is the management overhead of asynchronous updates. As the number of participants increases, more concurrent updates need to be processed, resulting in a sharp increase in system overhead. FedProx-Async performs relatively well, but the calculation of proximal terms still brings additional time costs.

In addition, we analyzed the performance of the algorithm under different hardware configurations. On edge devices with lower configurations, the time gap will be further widened, and the advantage of our algorithm will be more obvious. This is of great significance for the actual deployment of the IoV, because on-board devices usually have limited computing resources, and efficient algorithms can significantly improve user experience and system response speed.

In order to analyze the model effects of different levels of malicious nodes, Figure 10 gives the robustness of the three algorithms when facing different proportions of malicious node attacks. We simulated the malicious behaviors that may occur in the real environment and set four different attack intensities: normal (no malicious nodes), 10% malicious nodes, 30% malicious nodes, and 50% malicious nodes. The algorithms involved in the comparison include:

- Classic AFL (blue area): Classic asynchronous federated learning algorithm, using traditional parameter aggregation method;
- DBAFL (orange area): Decentralized blockchain-assisted federated learning algorithm, using blockchain technology to enhance the decentralized characteristics and security of the system;
- Proposed (red area): Our proposed federated learning algorithm with enhanced robustness.



(**a**) Classic AFL (**b**) DBAFL (**c**) Proposed

Figure 10. Comparison of model accuracy of different methods and different numbers of malicious nodes.

The experimental results clearly demonstrate the superior robustness of our proposed algorithm. Under normal circumstances without attacks, all three algorithms can achieve high performance levels: our algorithm achieves an accuracy of about 99%, DBAFL achieves about 97%, and Classic AFL achieves about 95%. This establishes a baseline performance for subsequent attack tests.

When 10% malicious nodes are introduced, the difference in algorithm robustness begins to emerge. The performance of Classic AFL shows a significant decline, with the accuracy dropping from about 95% to about 85%, a decrease of 10%. Due to the integrated blockchain verification mechanism, the accuracy of DBAFL only drops by about 3% to 94%. Our algorithm performs best, dropping only to 97%, a decrease of only 2%.

Under a medium attack intensity of 30% malicious nodes, the accuracy of Classic AFL drops to about 78%, a decrease of 17%. DBAFL drops to about 90%, with a performance loss of 7%. Our algorithm still maintains an accuracy rate of about 96%, only decreasing by 3%, demonstrating excellent anti-interference ability.

In the extreme scenario of 50% malicious node attacks, the accuracy rate of Classic AFL drops to about 75%, losing its practical value. DBAFL drops to about 85%, while our algorithm still maintains an accuracy rate of about 95%, only losing 4%, reflecting extremely strong robustness.

This strong robustness is due to the multi-layer protection mechanism of our algorithm: the anomaly detection module identifies suspicious updates, the adaptive weight strategy reduces the impact of anomalies, and the robust aggregation mechanism suppresses outlier interference.

In terms of recovery ability, our algorithm training curve fluctuates little and converges quickly. In contrast, Classic AFL oscillates significantly under attack, and DBAFL has a recovery delay. We also tested it in scenarios such as model poisoning and Byzantine attacks, and the algorithm still maintains high performance.

From the perspective of actual deployment, our algorithm is of great significance to security-sensitive scenarios such as the IoV, ensuring that reliable services and prediction capabilities are still provided under malicious attack conditions.

As shown in Figure 11, the ApolloScape trajectory dataset is used to visualize typical scenes under light, moderate, and congested traffic conditions [49]. Each group of sub-

figures shows the predicted 5-s trajectory of the vehicle after observing the 3-s historical trajectory. In the figure, the solid line part represents the observed historical trajectory, the dotted line part is the model predicted trajectory, and different colors represent different predicted paths. It can be clearly seen that as the traffic density changes from sparse (a) to congested (e), the model can still accurately predict the vehicle behavior in complex traffic environments within multiple lanes, reflecting good generalization ability and trajectory fitting accuracy.

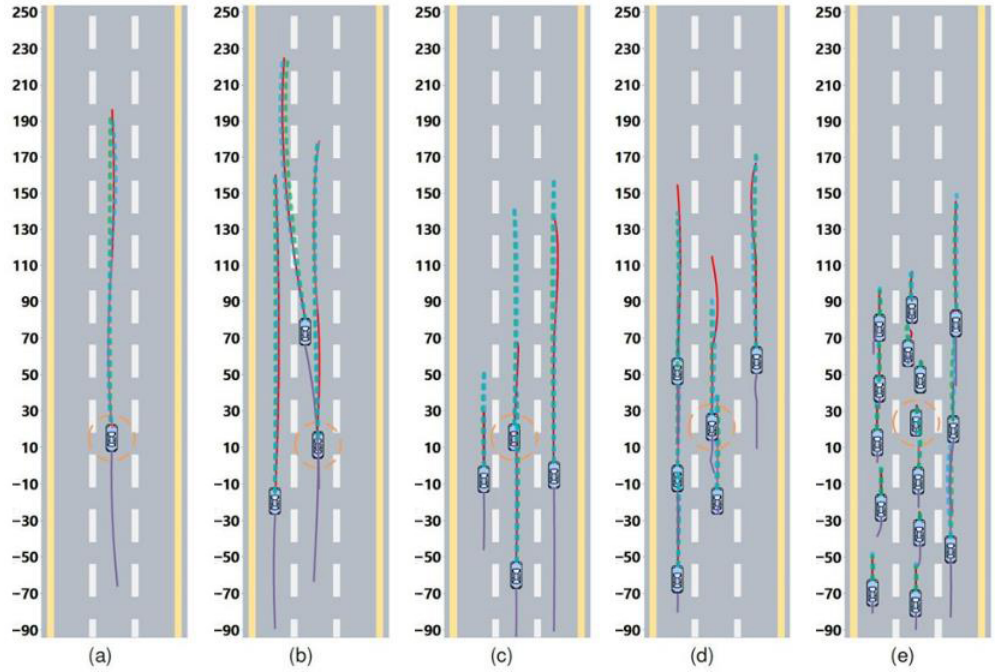


Figure 11. Visualization of vehicle road cooperation system. For different road situations: (**a**) Initial state: Only one vehicle (blue) at the bottom of the road; (**b**) Lane change begins: Blue vehicle moves to the left lane, trajectory shown with green dashed line; (**c**) Multi-vehicle scenario: Multiple vehicles appear, distributed across different lanes and positions; (**d**) Complex traffic: More vehicles join, some accelerating forward, others changing lanes; (**e**) Dense traffic: Maximum number of vehicles, demonstrating how the system handles crowded multi-vehicle cooperation scenarios.

In particular, in medium and high-density traffic scenarios, the model successfully predicted the lane change and following behavior of the vehicle in front, and the predicted trajectory was consistent with the actual driving path, demonstrating effective modeling of the impact of vehicle-to-vehicle interactions. This visualization result further verifies the effectiveness of the structural reputation mechanism proposed in this paper for modeling multi-vehicle interaction relationships, and it provides a theoretical basis and experimental support for trajectory prediction modules deployed in real IoV systems in the future.

6. Conclusions

This paper proposes a trust assessment framework for federated learning based on graph neural network edge weight similarity to address data sharing challenges in connected vehicle environments. We first design a graph-based reputation mechanism, consisting of an interpretable graph similarity assessment mechanism and a local directed acyclic graph (DAG). Based on this reputation mechanism, we propose an asynchronous federated learning scheme and further utilize deep reinforcement learning (DRL) to select optimal participating nodes to improve learning efficiency. By integrating learning parameters into the blockchain, the quality of the learned model can be further ensured through structured reputation-driven verification. Our core contribution lies in a novel

interpretable trust assessment mechanism that combines graph neural network (GNN) edge weight similarity analysis with distribution comparison based on maximum mean divergence (MMD) to provide transparent and mathematically sound trust assessment for vehicular federated learning. Experimental results on the ApolloScape dataset demonstrate the effectiveness of our proposed solution in terms of efficiency and accuracy, achieving 99% accuracy and maintaining 95% accuracy even under 50% malicious node attacks, significantly outperforming existing methods. Future work will focus on scaling this framework to larger-scale networks and integrating emerging communication technologies to enhance performance optimization.

Author Contributions: Conceptualization, W.L.; Methodology, W.L.; Software, W.L.; Validation, W.L. and Y.Z.; Formal analysis, W.L.; Investigation, W.L.; Resources, W.L.; Data curation, W.L.; Writing—original draft, W.L.; Writing—review and editing, Y.Z.; Visualization, W.L.; Supervision, Y.Z.; Project administration, Y.Z.; Funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (grant number 62303296).

Data Availability Statement: The original data presented in the study are openly available in Apolloscape at https://apolloscape.auto/trajectory.html, accessed on 12 October 2025.

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

References

1. Andrews, J.G.; Buzzi, S.; Choi, W.; Hanly, S.V.; Lozano, A.; Soong, A.C.; Zhang, J.C. What will 5G be? *IEEE J. Sel. Areas Commun.* **2014**, *32*, 1065–1082. [CrossRef]
2. Boban, M.; Kousaridas, A.; Manolakis, K.; Eichinger, J.; Xu, W. Connected roads of the future: Use cases, requirements, and design considerations for vehicle-to-everything communications. *IEEE Veh. Technol. Mag.* **2018**, *13*, 110–123. [CrossRef]
3. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
4. Wang, Y.; Wu, H.; Li, R. Deep graph reinforcement learning for mobile edge computing: Challenges and solutions. *IEEE Netw.* **2024**, *38*, 314–323. [CrossRef]
5. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Adaptive federated learning in resource constrained edge computing systems. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1205–1221. [CrossRef]
6. Nakamoto, S.; Bitcoin, A. A peer-to-peer electronic cash system. *Bitcoin* **2008**, *4*, 15.
7. Yuan, Y.; Wang, F.Y. Towards blockchain-based intelligent transportation systems. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 2663–2668.
8. Buterin, V. A next-generation smart contract and decentralized application platform. *White Pap.* **2014**, *3*, 1–36.
9. Zhang, Y.; Zheng, D.; Deng, R.H. Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet Things J.* **2018**, *5*, 2130–2145. [CrossRef]
10. Popov, S. The tangle. *White Pap.* **2018**, *1*, 30.
11. Silvano, W.F.; Marcelino, R. Iota Tangle: A cryptocurrency to communicate Internet-of-Things data. *Future Gener. Comput. Syst.* **2020**, *112*, 307–319. [CrossRef]
12. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
13. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
14. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
15. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Communication-efficient federated learning for digital twin edge networks in industrial IoT. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5709–5718. [CrossRef]
16. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef]

17. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
18. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
19. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
20. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
21. Contreras-Castillo, J.; Zeadally, S.; Guerrero-Ibañez, J.A. Internet of vehicles: Architecture, protocols, and security. *IEEE Internet Things J.* **2017**, *5*, 3701–3709. [CrossRef]
22. Liu, J.; Kato, N.; Ma, J.; Kadowaki, N. Device-to-device communication in LTE-advanced networks: A survey. *IEEE Commun. Surv. Tutor.* **2014**, *17*, 1923–1940. [CrossRef]
23. Zhang, K.; Leng, S.; He, Y.; Maharjan, S.; Zhang, Y. Cooperative content caching in 5G networks with mobile edge computing. *IEEE Wirel. Commun.* **2018**, *25*, 80–87. [CrossRef]
24. Wang, C.; Liang, C.; Yu, F.R.; Chen, Q.; Tang, L. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 4924–4938. [CrossRef]
25. Zhang, K.; Leng, S.; Peng, X.; Pan, L.; Maharjan, S.; Zhang, Y. Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks. *IEEE Internet Things J.* **2018**, *6*, 1987–1997. [CrossRef]
26. Dai, H.N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A survey. *IEEE Internet Things J.* **2019**, *6*, 8076–8094. [CrossRef]
27. Zhang, K.; Zhu, Y.; Maharjan, S.; Zhang, Y. Edge intelligence and blockchain empowered 5G beyond for the industrial Internet of Things. *IEEE Netw.* **2019**, *33*, 12–19. [CrossRef]
28. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.
29. Szabo, N. Smart contracts: building blocks for digital markets. *Extropy J. Transhumanist Thought* **1996**, *18*, 28.
30. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM SIGMETRICS Perform. Eval. Rev.* **2014**, *42*, 34–37. [CrossRef]
31. Kshetri, N. 1 Blockchain's roles in meeting key supply chain management objectives. *Int. J. Inf. Manag.* **2018**, *39*, 80–89. [CrossRef]
32. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
33. Xie, C.; Koyejo, S.; Gupta, I. Asynchronous federated optimization. *arXiv* **2019**, arXiv:1903.03934.
34. Dwork, C. Differential privacy. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Venice, Italy, 10 July 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–12.
35. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
36. Qi, J.; Lin, F.; Chen, Z.; Tang, C.; Jia, R.; Li, M. High-quality model aggregation for blockchain-based federated learning via reputation-motivated task participation. *IEEE Internet Things J.* **2022**, *9*, 18378–18391. [CrossRef]
37. Lu, Y.; Huang, X.; Dai, Y.; Maharjan, S.; Zhang, Y. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4177–4186. [CrossRef]
38. Cao, X.; Fang, M.; Liu, J.; Gong, N.Z. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv* **2020**, arXiv:2012.13995.
39. Ji, S.; Zhang, J.; Zhang, Y.; Han, Z.; Ma, C. LAFED: A lightweight authentication mechanism for blockchain-enabled federated learning system. *Future Gener. Comput. Syst.* **2023**, *145*, 56–67. [CrossRef]
40. Konečnỳ, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
41. He, C.; Balasubramanian, K.; Ceyani, E.; Yang, C.; Xie, H.; Sun, L.; He, L.; Yang, L.; Yu, P.S.; Rong, Y.; et al. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv* **2021**, arXiv:2104.07145.
42. Cho, J.H.; Swami, A.; Chen, R. A survey on trust management for mobile ad hoc networks. *IEEE Commun. Surv. Tutor.* **2010**, *13*, 562–583. [CrossRef]
43. Yu, W.; Liang, F.; He, X.; Hatcher, W.G.; Lu, C.; Lin, J.; Yang, X. A survey on the edge computing for the Internet of Things. *IEEE Access* **2017**, *6*, 6900–6919. [CrossRef]
44. Xiao, H.; Zhao, J.; Pei, Q.; Feng, J.; Liu, L.; Shi, W. Vehicle selection and resource optimization for federated learning in vehicular edge computing. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 11073–11087. [CrossRef]
45. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
46. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.

47. Li, J.; Ma, G.; Yang, W.; Li, R.; Wang, H.; Gu, Z. FedDDPG: A reinforcement learning method for federated learning-based vehicle trajectory prediction. *Array* **2025**, *27*, 100450. [CrossRef]
48. Wang, P.; Huang, X.; Cheng, X.; Zhou, D.; Geng, Q.; Yang, R. The apolloscape open dataset for autonomous driving and its application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2702–2719. [CrossRef]
49. Li, X.; Ying, X.; Chuah, M.C. Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving. *arXiv* **2019**, arXiv:1907.07792.

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.