Reputation-Driven Asynchronous Federated Learning for Enhanced Trajectory Prediction With Blockchain

Weiliang Chen, Li Jia, Yang Zhou, and Qianqian Ren

***Abstract*—Federated learning (FL), when integrated with blockchain, facilitates secure data sharing in autonomous driving applications. As vehicle-generated data becomes more granular and complex, the absence of data quality audits raises concerns about multiparty mistrust in trajectory prediction tasks. However, most of the existing research on trajectory prediction focuses on how to improve the model to enhance the prediction accuracy, and lacks the consideration of the privacy and security issues of data sharing in real-world scenarios. To address this, we propose an asynchronous FL data-sharing method, incorporating an interpretable reputation quantization mechanism based on graph convolutional networks. Data providers share data structures under differential privacy constraints, ensuring security while minimizing redundancy. We utilize deep reinforcement learning to classify vehicles by reputation level, optimizing FL aggregation efficiency. Experimental results show that the proposed scheme not only strengthens the security of trajectory prediction but also improves prediction accuracy.**

***Index Terms*—Asynchronous federated learning (AFL), data sharing, deep reinforcement learning (DRL), differential privacy (DP), graph convolutional network, trajectory prediction.**

I. Introduction

THE RAPID advancements in computational and communication technologies, particularly within the 5G network, have revolutionized modern vehicular services and applications, significantly enhancing the driving experience [1]. Autonomous vehicles are complex systems that integrate various technologies, including perception algorithms [2], path planning [3], control theory [4], and positioning [5]. Among the key research areas, trajectory prediction [6] is pivotal to the decision-making processes of intelligent vehicles. A precise and reliable trajectory prediction algorithm can anticipate potential traffic accidents, thereby enhancing overall vehicular safety through timely preventive measures.

Received 25 March 2024; revised 18 August 2024, 12 September 2024, and 26 September 2024; accepted 6 November 2024. Date of publication 11 November 2024; date of current version 7 March 2025. This work was supported by the National Natural Science Foundation of China under Grant 62303296 and Grant 61773251. *(Corresponding authors: Yang Zhou; Qianqian Ren.)*

Weiliang Chen and Qianqian Ren are with the Department of Computer Science and Technology, Heilongjiang University, Harbin 150080, China (e-mail: chanweiliang@s.hlju.edu.cn; renqianqian@hlju.edu.cn).

Li Jia is with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China (e-mail: jiali@shu.edu.cn).

Yang Zhou is with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China (e-mail: zhouyang0410@shu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3495693

Trajectory prediction for autonomous vehicles is predominantly driven by neural network approaches, with recurrent neural networks (RNNs) at the forefront. Researchers have developed numerous models based on historical trajectory data and environmental information. For example, GRIP [7] enhances convolutional layers with graph operations, modeling driving scenarios as graphs where vehicle interactions are depicted as node interrelationships. Another method [8] introduces a novel framework leveraging spectral clustering, enabling the simultaneous prediction of vehicle trajectories and driving behaviors. Despite these promising advances, trajectory prediction faces significant real-world challenges, particularly in traditional telematics environments where data silos persist. This challenge underscores the need for innovative solutions, with federated learning (FL) emerging as a promising approach.

FL offers a robust strategy for preserving data privacy in distributed environments [9]. It enables edge intelligence by leveraging knowledge from decentralized data sources while maintaining strict privacy standards. When combined with the inherent features of blockchain technology [10], FL provides a solid foundation for efficient and reliable data sharing among participants. Even in low-trust environments, model parameters are transmitted to a central server for aggregation, facilitating parallel learning to improve the global model while ensuring data integrity, maintaining anonymity, and enabling traceability. However, as autonomous driving technology advances and vehicle-generated data becomes more complex, trajectory prediction still faces several critical challenges.

1) *Limited Trust Among Multiple Parties:* Current data-sharing schemes with central administrators increase the risk of data leakage. Administrators must manage large volumes of aggregated data from various sources, including unverified or raw data.
2) *Uncertain Data Quality From Participants:* Due to vehicle mobility and unreliable communications, the data-sharing environment is highly dynamic, with new unverified data constantly being generated. Malicious data shared by providers can significantly bias the prediction process.
3) *Significant Prediction Delays:* In the context of dynamic vehicle data updates, it is crucial to address aggregation challenges caused by delays in heterogeneous vehicle data. Conventional approaches

relying on FL with blockchain technology require coordination among roadside units (RSUs) at each timestamp before uniformly uploading data for global aggregation.

To systematically address these limitations, we propose a novel trajectory prediction approach, reputation-driven asynchronous federated learning for enhanced trajectory prediction with blockchain (RAFT), built on a graph convolution network (GCN). RAFT incorporates an asynchronous FL (AFL) strategy that evaluates vehicle reputation through their interactions. Data providers share data structures under differential privacy (DP) constraints [11], ensuring data security. Given the effectiveness of deep reinforcement learning (DRL) in managing dynamic decision-making tasks, many studies have applied DRL to enhance FL performance. For example, deep Q-learning networks (DQNs) have been combined with FL to address edge computing challenges, such as task offloading [12], caching, and communication issues [13]. Similarly, deep deterministic policy gradient (DDPG) has been used to select high-quality device nodes for better model aggregation and lower communication costs [14], [15], [16]. In contrast, proximal policy optimization (PPO) has proven effective in optimizing task scheduling strategies [17]. While both DQN and PPO are suitable for discrete action spaces, the stability and efficiency of PPO make it the ideal choice for grouping vehicles in RAFT, reducing FL costs [18]. The contributions of this article are as follows.

1) We propose an interpretable reputation-based framework for secure data sharing in trajectory prediction. Our framework includes a reputation reward mechanism that replaces traditional loss functions with reputation values derived from vehicle trajectory data, fostering cooperation among method components and enabling FL to develop more effective global models.
2) We introduce a reputation-driven AFL scheme tailored for dynamic and heterogeneous vehicle networks. This scheme features a reputation-enhanced PPO algorithm that groups vehicle trajectory models based on reputation values to reduce FL costs. High-reputation vehicle clusters, which exhibit high data quality and similar trajectory graph models, are prioritized for global FL aggregation.
3) We implement DP techniques to protect the privacy and security of shared trajectory graph models and vehicle reputation values while preserving graph structures and vehicle interactions.
4) Extensive experiments on large-scale vehicle trajectory datasets, including NGSIM and ApolloScape, across various real-world scenarios, validate our approach. Results show that our data-sharing scheme effectively ensures data security, improves trajectory prediction accuracy, and adapts well to diverse traffic scenarios. Comprehensive ablation studies further confirm the effectiveness of our method components.

The remainder of this article is organized as follows. Section II presents related work. Section III introduces the data-sharing framework and system workflow. Sections IV and V delve into the details of our proposed model, explaining the methodology and approach. Section VI presents a comprehensive evaluation through extensive experiments, demonstrating the effectiveness of our method. Finally, Section VII concludes this article, summarizing our findings and future directions.

II. Related Work

Trajectory prediction spans multiple domains, including statistics [19], signal processing [20], and control systems engineering [4]. Given the nonstationary nature of vehicle trajectories, RNN-based methods have become prevalent. CS-LSTM [21] identifies maneuvers of surrounding vehicles on highways and predicts their trajectories using a convolutional social pooling layer to capture interactions. TraPHic [22] utilizes a novel LSTM-CNN hybrid network to model interactions among various traffic agents. Subsequently, graph-based models, which represent vehicles as nodes to study spatiotemporal interactions, have gained significant attention for their ability to simulate traffic scenarios effectively [7]. Spectral [8] captures temporal correlations across modeling the whole traffic environment but suffers from high computational costs and sparse adjacency matrices that reduce accuracy. To better adapt to dynamic traffic environments, GRIP++ [23] utilizes both fixed and dynamic graphs to capture the complex interactions among different types of traffic agents, thereby improving trajectory prediction accuracy. However, the aforementioned trajectory prediction methods overlook the issue of data privacy in real-world traffic scenarios. Our proposed RAFT framework addresses this by considering the dynamic traffic environment. Based on a dynamic GCN trajectory prediction model, RAFT leverages a distributed and secure data-sharing framework, combining AFL with blockchain technology to mitigate privacy leakage during vehicle data sharing.

FL [24] provides privacy-preserving edge intelligence in distributed scenarios, allowing users to retain their data locally while sharing model parameters with a central server for federated global model learning. This makes it a popular solution for addressing data silos and has been applied in fields, such as speech recognition, image processing, healthcare, and Internet of Vehicles (IoV) [9]. However, traditional FL relies on a centralized server to aggregate parameters from numerous distributed clients, which can lead to a single point of failure. The decentralized and traceable nature of blockchain makes it a vital complement to FL. Warnat-Herresthal et al. [25] integrated edge computing, blockchain, and FL to propose a decentralized training model, enabling all participants to merge parameters with equal rights while maintaining privacy. To further accommodate the complex and flexible on-chain environment, DRL [26] has emerged as a technology well suited to edge computing [27], offering end-to-end learning capabilities for decision making and understanding across environments. Lu et al. [16] proposed an algorithm combining DDPG and FL through smart contracts, improving node selection in IoV and enhancing data-sharing security among vehicles. However, in dynamic trajectory prediction scenarios, existing blockchain-based FL frameworks are suboptimal for fairly selecting

vehicle nodes, as they lack appropriate evaluation metrics and are vulnerable to malicious node interference. Therefore, we propose an interpretable reputation mechanism enhanced data-sharing framework, where reputation values derived from vehicle trajectory data replace traditional loss functions to improve vehicle node selection. A reputation-based grouping PPO algorithm is also presented to optimize global model aggregation.

Data sharing [28] is crucial for improving driving experiences and Internet of Things (IoT) services, with data quality largely dependent on vehicle reputation [29]. As blockchain-based FL frameworks evolve, integrating reputation-based mechanisms to tackle reliability issues like sensor defects, firmware malfunctions, or selfish behavior has become a promising solution [30]. Centralized trust systems, which rely on quick decision making and a central server, often struggle to meet stringent quality of service requirements. In contrast, decentralized trust systems, where vehicles or RSUs manage trust locally, reduce the need for interaction with network infrastructure. Huang et al. [31] improved reputation updates in distributed vehicular edge computing by employing multiweight subjective logic. Kang et al. [32] proposed a three-weight subjective logic model for more accurate reputation management in blockchain-based vehicular networks to address challenges posed by varying vehicle capabilities and dynamic network conditions. However, integrating the reputation evaluation system into the entire trajectory prediction framework is challenging. To address this gap, we propose RAFT. To better accommodate trajectory prediction scenarios, we calculate multiweighted reputation values based on the similarity of vehicle trajectories and design a reputation-based consensus process to minimize FL costs. Additionally, DP [11] is incorporated to further enhance the privacy protection of shared local trajectory models with reputation data.

III. Proposed RAFT Model

In this article, we address the inefficiencies and security concerns associated with current trajectory prediction methods by proposing a novel AFL model. This model is designed to effectively mitigate issues related to inefficient training and privacy leakage in existing methods.

A. Collaborative Data Sharing and Trajectory Prediction Framework

We propose a practical framework for trajectory prediction within a distributed data-sharing scenario involving multiple participants. Each participant has their own data and is willing to share it to realize the collaborative task, but we cannot exclude the existence of malicious sharers who try to interfere with the trajectory prediction task. In addition, the vehicular network employs vehicle-to-vehicle (V2V) and vehicle-to-RSU communications, with RSUs equipped with mobile-edge computing servers that provide computational and storage capabilities. Fig. 1 illustrates the architecture of the proposed data-sharing system. Vehicles collect trajectory data and train local graph models, which they share with RSUs. The RSUs then communicate with MBSs to upload

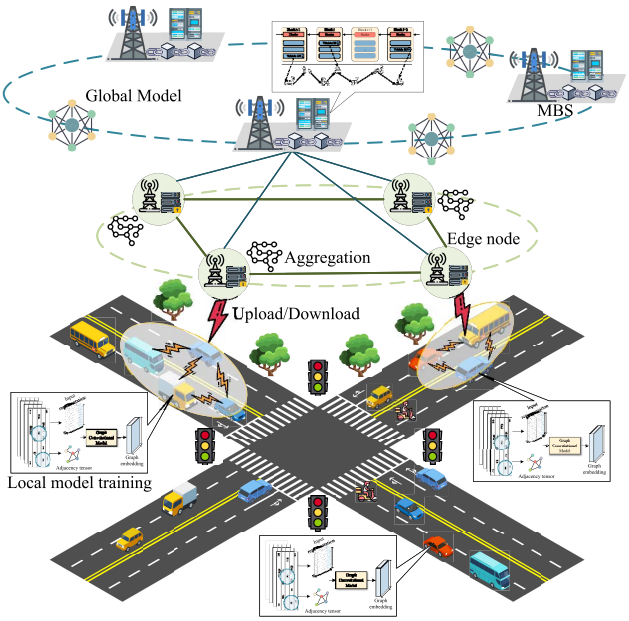

Fig. 1. Architecture of secure data-sharing solution.

these local models. After achieving consensus, the federated global model is downloaded by the RSUs and broadcast to the vehicles. Importantly, the permissioned blockchain facilitates data retrieval and manages data-sharing transactions, ensuring that only authorized participants can access and share relevant data while preserving data privacy and security.

Building upon the framework, we specifically focus on the scenario involving N vehicle participants and a joint dataset $D = D_1, D_2, \ldots, D_n$. For any vehicle $v_{\text{req}} \in v_i$ participating in the data-sharing request, its local dataset $D_i \in D$ contains historical trajectory data $X \in \mathbb{R}^{\tau \times n \times 2}$, representing observations at time step $\mathcal{T}$

$$X = \left[\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \ldots, \mathcal{P}^{(\mathcal{T})}\right] \tag{1}$$

where $\mathcal{P}^{(t)} = [[x_0^t, y_0^t], [x_1^t, y_1^t], \ldots, [x_n^t, y_n^t]] \in \mathbb{R}^{n \times 2}$ denotes the coordinates of all observations at time t, and n represents the number of observations. Following GRIP++ [23], we also use both fixed and dynamic graphs to capture the complex interactions between different types of traffic participants. We output trajectory predictions using the encoder–decoder network, and apply the global coordinate system which defined in [21]. The output $\mathcal{Y}' \in \mathbb{R}^{T \times n \times 2}$ predicts the future positions of all observations from time step $\mathcal{T}+1$ to $\mathcal{T}+\mathcal{T}_f$

$$\mathcal{Y}' = \left[\mathcal{P}^{(\mathcal{T}+1)}, \mathcal{P}^{(\mathcal{T}+2)}, \ldots, \mathcal{P}^{(\mathcal{T}+\mathcal{T}_f)}\right] \tag{2}$$

where $\mathcal{T}_f$ represents the prediction horizon.

B. System Workflow

To enhance computational and storage efficiency under resource constraints and ensure secure vehicle interactions, we introduce a graph-based approach for model training and data sharing. Initially, vehicles must undergo authentication to become legitimate nodes within the permissioned blockchain. After authentication, vehicles obtain certificates that allow

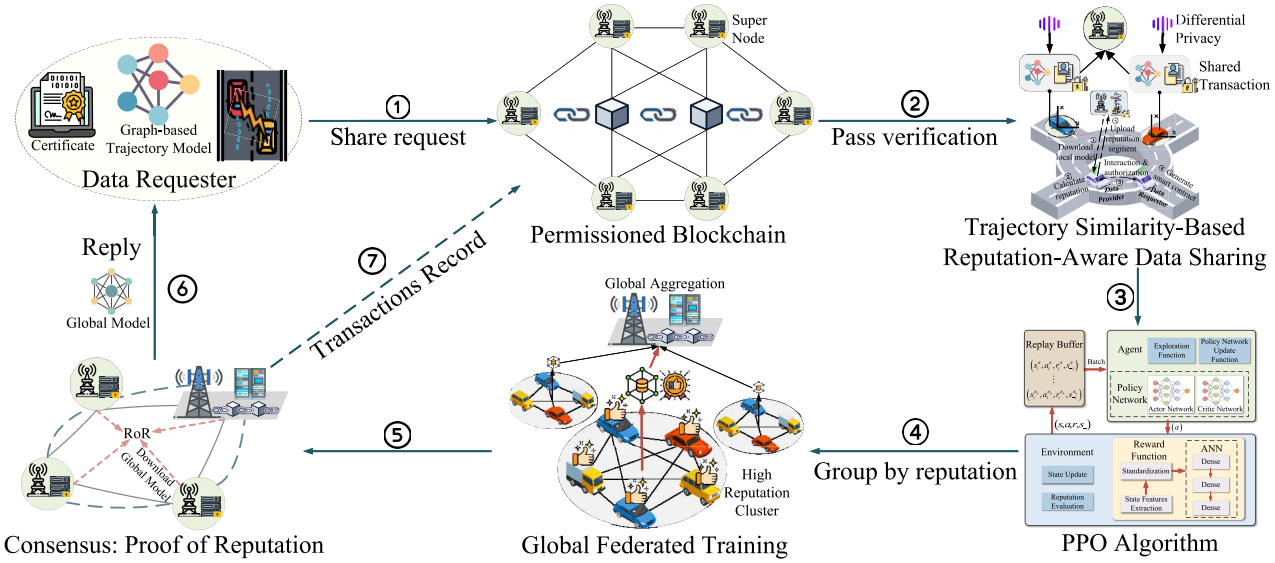

Fig. 2. Working mechanisms for the proposed methodology.

them to participate in V2V data sharing and to download the global model locally for training. Next, vehicles transmit data-sharing requests to nearby super nodes, such as MBS and RSU, which process these requests. Upon verifying the vehicles' public keys [33], the super nodes begin collecting weighted graph models from vehicles within the community. To assess the quality of the trajectory data, participating vehicles compute reputation values based on their original data. These values are recorded as shared transactions and in the local models of other participating vehicles within each vehicle's local directed acyclic graph (DAG) [34]. These models are perturbed with DP noise to ensure privacy during FL training. To optimize the global aggregation process and reduce latency, we employ the PPO algorithm to group vehicles based on their reputation evaluations. High-quality vehicle clusters are prioritized in the deep aggregation process, ensuring that the best data contributes most significantly to the global model. The super nodes then collaborate to train the global model M using an AFL approach. Once trained, the global model is broadcasted to all committee nodes, which are responsible for driving the consensus process in the permissioned blockchain. The committee nodes collect transactions into blocks, verify them, and add them to the blockchain. Finally, vehicles optimize their local models by integrating the updated global model, thereby improving the accuracy of their trajectory predictions. Fig. 2 illustrates the working mechanism of our proposed scheme.

IV. Reputation-Based Hybrid Blockchain for Data Sharing

Considering resource constraints and privacy concerns of vehicle users, we propose a data-sharing scheme that involves the collaboration of a federated GCN model among decentralized parties to share well-trained model parameters instead of raw data.

A. Graph-Based Trajectory Modeling

Our approach models object trajectories as weighted graphs to capture interactions. Referring to the trajectory prediction model GRIP++ [23], each object trajectory is organized as a 3-D array, computed for improved speed to enhance prediction accuracy. We represent the interactions between objects using undirected graphs $G = \{V, E\}$. The input data is processed by a GCN, where graph operations handle spatial interactions and temporal convolutions capture time-based features. The trajectory prediction model utilizes a two-layer gated recurrent unit (GRU) network as both the encoder and decoder. Based on this structure, $G = \{V, E\}$ is converted into a weighted graph of vehicle trajectories. The set of nodes V is defined as $V = \{v_{it} \mid i = 1, \ldots, n, t = 1, \ldots, \mathcal{T}\}$, with n denoting the total number of objects observed in the scene. Each node v_{it} contains its own weights w_i^n, determined by the similarity of trajectories between vehicles, considering factors like direction, speed, and tilt angle, as described in the following section. G is then serialized into ordered vectors and mapped onto linear vectors, which are combined to form a global vehicle network graph $\mathcal{G} = \{G_1 \cup G_2, \ldots, \cup G_n\}$. For the overall graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the number of representative vertices is denoted as k. Afterward, the normalization properties of the vertices and edges are then calculated, with the vertices normalized to size k and the edges to size $k \times (k-1)/2$. Eventually, the normalized vector $Seq = \mathcal{V} \cup \mathcal{E} = \{\mathcal{V}_1, \ldots, \mathcal{V}_k\} \cup \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_{k(k-1)/2}\}$.

After that, we use cosine similarity as a distance function to compare vehicle matrices sequentially in temporal order, and confidence levels above 80% will fail validation. New weighted graphs added in future moments with confidence levels above a preset threshold will fail validation. By using weighted graphs and the PPO algorithm, the set of vehicles $\{V_1, \ldots, V_n\}$ is divided into high and low reputation groups. Through this enhanced FL mechanism, instead of sharing local vehicle model parameters directly, the shared GCN model retrieves and compares confidence levels. This filtering process

excludes graphs with high trustworthiness but low similarity, preventing the leakage of vehicle trajectory information and improving the efficiency of the next global aggregation.

B. Differential Privacy-Enhanced Data Sharing

Our objective is to develop a secure data-sharing mechanism for trajectory prediction scenarios that intelligently facilitates data exchange among distributed multiuser environments while effectively safeguarding data privacy. We consider a scenario involving $\mathcal{N}$ parties (or data holders) and a joint dataset $\mathbb{D}$. Each party P_i possesses a local dataset $\mathbb{D}_i$ in $\mathbb{D}$. All $\mathcal{N}$ parties unanimously agree to share data without compromising sensitive information. Let $R = \{r_1, r_2, \ldots, r_m\}$ represent the set of data-sharing requests. When a requester submits a query r_i, our approach returns computed results instead of raw data to fulfill the sharing requirements. After processing, the trained global model M is sent back to the committee node. Recipients can then use the received global aggregation model to respond to data-sharing requests locally.

Given the large size and sensitivity of the data, we draw inspiration from previous work [35] and utilize blockchain for data retrieval, ensuring that the actual data remains stored on local vehicles. We integrate GCNs with DP for quality verification. Using a graph representation of raw data enhances computational and storage efficiency under resource constraints, preserving more structural and contextual information for validation while mitigating privacy risks. When new data providers join, their unique identifiers (vehicle IDs) are recorded on the blockchain along with an overview of their data, including trajectory data, vehicle types, and data sizes. Data profiles from multiple participants are logged as transactions and verified by blockchain nodes using Merkle trees [36]. Each data-sharing event is also recorded as a transaction on the blockchain. Participants ($\{P_1, P_2, \ldots, P_n\}$) are selected through multiparty searches in the blockchain and grouped into communities based on their reputation values. These communities consist of members with similar data quality. Given the limited communication resources of IoT devices, the retrieval process should also consider the matrix cosine similarity between the two participants. When a user submits a data-sharing request to a nearby node P_i, all nodes within the identical community as P_i broadcast this request to other vehicles observing the target during that timestamp to initiate the retrieval process. The process is recursively executed until all trajectory graphs for that timestamp have been retrieved. The result is a subset of vehicles $P_s \subseteq \mathbb{P}$ that fulfill the request, characterized by structural stability, the absence of anomalous data, and minimal redundancy in trajectory graphs, addressing the constraints posed by limited communication resources. To ensure privacy protection in the shared model, we apply DP to each local model m_i using the Laplace mechanism. Calibrated noise with sensitivity s is added to the local data to train $\hat{m}_i$, which is expressed as

$$\hat{m}_i = m_i + \text{Laplace}(s/\epsilon) \tag{3}$$

where ϵ denotes the privacy budget. Participant P_i then transmits the model $\hat{m}_i$ as a blockchain transaction, broadcasting it to other participants for FL. Upon receiving $\hat{m}_i$, participant P_{i+1} trains a new local data model $\hat{m}_{i+1}$ based on the received $\hat{m}_i$ and its local data, then broadcasts $\hat{m}_{i+1}$ to other participants. This iterative process continues among participants until a global model M is generated, represented as $M = \{\hat{m}_1 \cup \hat{m}_2 \ldots \cup \hat{m}_n\}$.

C. Trajectory Similarity-Based Reputation-Aware Data Sharing

The mean absolute error may not be practical for real-world trajectory prediction scenarios. Instead, we introduce a novel evaluation metric: a reputation value derived from the trajectory similarity among vehicles. Given the spatial correlation inherent in locally collected vehicle data, incorporating trajectory similarity into reputation calculations enhances location awareness and data relevance. The more similar the trajectories, the higher the relevance of the shared data from the provider, leading to improved data quality, accuracy, and reliability. The trajectory coefficients for a vehicle are represented as $v = \{\text{velocity, position, orientation}\}$, with corresponding weights ρ_1, ρ_2, and ρ_3, and $\rho_1 + \rho_2 + \rho_3 = 1$. The similarity between two trajectory segments, $\mathcal{J}_i$ and $\mathcal{J}_j$, of vehicles i and j, can be expressed as $\mathcal{SIM}(\mathcal{J}_i, \mathcal{J}_j)$, is defined by

$$\mathcal{SIM}(\mathcal{J}_i, \mathcal{J}_j) = 1 - \mathcal{DISS}(\mathcal{J}_i, \mathcal{J}_j). \tag{4}$$

Here, $\mathcal{DISS}(\mathcal{J}_i, \mathcal{J}_j)$ represents the normalized dissimilarity between the two trajectory segments, calculated using

$$\begin{aligned}\mathcal{DISS}(\mathcal{J}_i, \mathcal{J}_j) = \rho_1 \,\text{velocity}(\mathcal{J}_i, \mathcal{J}_j) + \rho_2 \,\text{position}(\mathcal{J}_i, \mathcal{J}_j) \\ + \rho_3 \,\text{orientation}(\mathcal{J}_i, \mathcal{J}_j).\end{aligned} \tag{5}$$

We posit that $\mathcal{DISS}(\mathcal{J}_i, \mathcal{J}_j)$ depends on the disparities in velocity, position, and orientation between the two trajectory segments. The velocity difference is given by

$$\text{velocity}(\mathcal{J}_i, \mathcal{J}_j) = \frac{\left|W_{\text{ave}}(\mathcal{J}_i) - W_{\text{ave}}(\mathcal{J}_j)\right|}{\max\left[W(\mathcal{J}_i), W(\mathcal{J}_j)\right]} \tag{6}$$

where $W(\mathcal{J}_i)$ and $W(\mathcal{J}j)$ represent the velocities of vehicles i and j during their respective trajectory segments. $W_{\text{ave}}\,(\mathcal{J}i)$ and $W_{\text{ave}}\,(\mathcal{J}_j)$ are the average velocities. The position$(\mathcal{J}_i, \mathcal{J}_j)$ shows the variance in the position between the trajectory segments. The number of sampled points for $\mathcal{J}_i$ and $\mathcal{J}_j$ during the time window T are denoted as e and k, respectively. The set of sampled points in temporal order are $\{P_{i1}, P_{i2}, \ldots, P_{ie}\}$ and $\{P_{j1}, P_{j2}, \ldots, P_{jk}\}$. We employ the longest common sequence (LCS) method to measure the similarity of trajectory segments. For trajectory segments $\mathcal{J}_i$ and $\mathcal{J}_j$, the LCS is described as $\text{LCS}(\mathcal{J}_i, \mathcal{J}_j) = \{P_{ie} = P_{jk} \mid e = k\}$, where $e \in \{1, 2, \ldots, E\}, k \in \{1, 2, \ldots, K\}$. Therefore, the equation for the difference in position between the trajectory segment position$(\mathcal{J}_i, \mathcal{J}_j)$ is provided as follows:

$$\text{position}(\mathcal{J}_i, \mathcal{J}_j) = \frac{\max(e, k) - \text{num}\left[\text{LCS}(\mathcal{J}_i, \mathcal{J}_j)\right]}{\max(e, k)} \tag{7}$$

where num [LCS$(\mathcal{J}_i, \mathcal{J}_j)$] is the number of points in the LCS of the trajectory segments $\mathcal{J}_i$ and $\mathcal{J}_j$. The directory difference between the two trajectory segments is the angle between the

two trajectory segments. Here, we employ φ as the angle between the trajectories $\mathcal{J}_i$ and $\mathcal{J}_j$ to represent the direction. More precisely

$$\text{orientation}\left(\mathcal{J}_i, \mathcal{J}_j\right) = \begin{cases} \frac{\sin\varphi}{2}, & 0 < \varphi \leq \frac{\pi}{2} \\ \frac{1}{2} + \frac{|\sin(\varphi+\frac{\pi}{2})|}{2}, & \frac{\pi}{2} < \varphi \leq \pi. \end{cases} \tag{8}$$

1) Local Opinions for Subjective Logic: In the case of two vehicles, the reputation between v_i and v_j can be formally described as a local opinion vector $\omega_{i\to j} := \{r_{i\to j}, d_{i\to j}, u_{i\to j}\}$, where $r_{i\to j}$, $d_{i\to j}$, and $u_{i\to j}$ denote trust, distrust, and uncertainty, respectively. These components satisfy the condition $r_{i\to j}+d_{i\to j}+u_{i\to j} = 1$, with $r_{i\to j}$, $d_{i\to j}$, and $u_{i\to j}$ each confined within the range $[0, 1]$. The definitions of these components are given by

$$\begin{cases} r_{i\to j} = \left(1 - u_{i\to j}\right)\frac{\alpha}{\alpha+\beta} \\ d_{i\to j} = \left(1 - u_{i\to j}\right)\frac{\beta}{\alpha+\beta} \\ u_{i\to j} = 1 - s_{i\to j} \end{cases} \tag{9}$$

where β represents the number of negative events, and α represents the number of positive events. The uncertainty $u_{i\to j}$ in the local opinion vector depends on the communication quality $s_{i\to j}$ between vehicles i and j. Here, $s_{i\to j}$ denotes the probability of successfully transmitting data-sharing request packets during communication. Based on the local opinion vector $\omega_{i\to j}$, the reputation value $\mathcal{R}_{i\to j}$, representing the expected belief that v_i provides true and relevant data to v_j, is given by

$$\mathcal{R}_{i\to j} = r_{i\to j} + \gamma u_{i\to j} \tag{10}$$

where the constant γ, assigned by the vehicle, measures the impact of uncertainty on the vehicle's reputation. It is recommended to initially set γ to 0.5.

2) Combine Local Opinions With Suggested Opinions: When evaluating the reputation of a vehicle, the subjective opinions of neighboring vehicles should be considered alongside the opinion of the requesting vehicle. We combine the subjective opinions of different recommenders into a unified opinion, denoted as $\omega^{\text{rec}}_{x\to j} := \{r^{\text{rec}}_{x\to j}, d^{\text{rec}}_{x\to j}, u^{\text{rec}}_{x\to j}\}$. The aggregation process is as follows:

$$\begin{cases} r^{\text{rec}}_{x\to j} = \frac{1}{\sum_{x\in X}\delta_{x\to j}} \sum_{x\in X} \delta_{x\to j} b_{x\to j} \\ d^{\text{rec}}_{x\to j} = \frac{1}{\sum_{x\in X}\delta_{x\to j}} \sum_{x\in X} \delta_{x\to j} d_{x\to j} \\ u^{\text{rec}}_{x\to j} = \frac{1}{\sum_{x\in X}\delta_{x\to j}} \sum_{x\in X} \delta_{x\to j} u_{x\to j} \end{cases} \tag{11}$$

where $x \in X$ represents the set of neighboring vehicles that interact with v_j. After receiving data from the data provider, the data requestor forms a local opinion. To avoid spoofing, this local opinion is also considered in the final reputation evaluation. The final opinion vector $\omega^{\text{final}}_{x\to j} := \{r^{\text{final}}_{x\to j}, d^{\text{final}}_{x\to j}, u^{\text{final}}_{x\to j}\}$, where $r^{\text{final}}_{x\to j}$, $d^{\text{final}}_{x\to j}$, and $u^{\text{final}}_{x\to j}$ are calculated as follows:

$$\begin{cases} r^{\text{final}}_{i\to j} = \frac{r_{i\to j}u^{\text{rec}}_{x\to j} + r^{\text{rec}}_{x\to j}u_{i\to j}}{u_{i\to j} + u^{\text{rec}}_{x\to j} - u^{\text{rec}}_{x\to j}u_{i\to j}} \\ d^{\text{final}}_{i\to j} = \frac{d_{i\to j}u^{\text{rec}}_{x\to j} + d^{\text{rec}}_{x\to j}u_{i\to j}}{u_{i\to j} + u^{\text{rec}}_{x\to j} - u^{\text{rec}}_{x\to j}u_{i\to j}} \\ u^{\text{final}}_{i\to j} = \frac{u^{\text{rec}}_{x\to j}u_{i\to j}}{u_{i\to j} + u^{\text{rec}}_{x\to j} - u^{\text{rec}}_{x\to j}u_{i\to j}}. \end{cases} \tag{12}$$

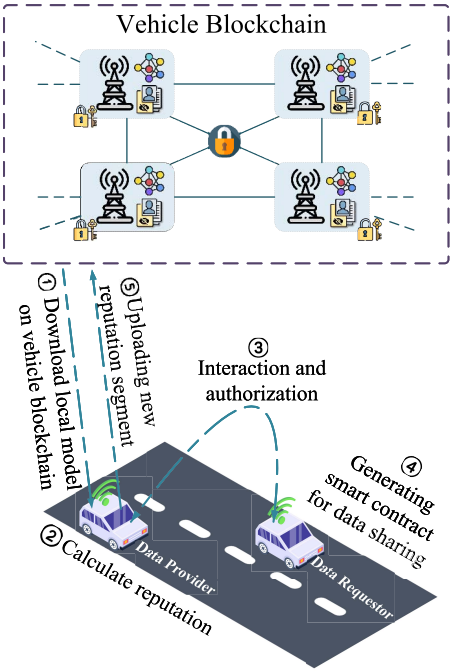

Fig. 3. Reputation-aware data-sharing mechanism.

The final reputation of v_i toward v_j is then calculated as

$$\mathcal{R}^{\text{final}}_{i\to j} = r^{\text{final}}_{i\to j} + \gamma u^{\text{final}}_{i\to j}. \tag{13}$$

These reputation values are incorporated into the DAG as parameters for training models. The DAG consists of nodes representing both data-sharing events and training model parameters, with edge connections established through approval relationships between transactions. These connections allow nodes to interact and connect with each other. Vehicles maintain and update DAGs locally, asynchronously achieving consistency with other vehicles. Each vehicle disseminates its latest DAG, including transactions and approval relationships, to neighboring vehicles. This approach propagates updated DAGs, maintains loose consistency among vehicles, reduces computational intensity, and enhances global robustness. The specific process is illustrated in Fig. 3.

D. Consensus: Proof of Reputation

Transforming the data-sharing problem into a model-sharing problem not only enhances the privacy of data holders but also allows the GCN model to supply essential information for new sharing requests effectively. This transformation preserves vehicle trajectory features more effectively during data fusion. However, prevailing consensus mechanisms like Proof of Work in data sharing often result in high computational and communication costs. To address these challenges, we propose implementing the FL authorized consensus Proof-of-Reputation (PoR) protocol. PoR integrates vehicle reputation values into the consensus process, optimizing node computational resources. When data-sharing requests occur, consensus committee members are chosen by retrieving the relevant blockchain nodes. This committee supervises the consensus process and gains insights into the data model associated

with the requested data. AFL seeks to develop a worldwide information model M, which delivers an effective response M(Req) to inquiries concerning data sharing, thereby achieving the intended objective.

In particular, committee leaders are selected based on their reputation scores, ensuring that the most reliable nodes are responsible for guiding the consensus process. As each committee node trains a local data model, it is critical to verify and measure the quality of these models during consensus. We evaluate the performance of the trained local models by assessing their prediction accuracy, which is crucial in trajectory prediction tasks. Specifically, the model is trained as a regression task, and the total loss for each prediction task is calculated as follows:

$$\text{Loss} = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \left\| Y_{\text{pred}}^{t} - Y_{GT}^{t} \right\| \tag{14}$$

where $\mathcal{T}$ represents the number of future time steps, and Y_{pred}^{t} and Y_{GT}^{t} denote the predicted position and ground truth at time step t, respectively. For each committee node, a trained global model M and a local model $\hat{m}_i$ are obtained after asynchronous training. During the consensus process, committee node P_i sends its trained model $\hat{m}_i$ to the next committee node when responding to a data-sharing request. The transmission is recorded as a model transaction $t_{\hat{m}_i}$ along with its corresponding loss $(\hat{m}_i)$. P_i has a pair of public and private keys (PK_i, SK_i) to encrypt and sign the message, sending the encrypted message $E(\text{SK}_i(t_m), \text{PK}_i)$ to the other committee nodes. All model transactions are then collected and stored locally as potential blocks by committee node P_j. As part of the training procedure, P_j validates all transactions it receives by computing the loss defined in (14). The validation loss for P_j, denoted as $\text{Loss}^u(P_j)$, is calculated using the following equation:

$$\text{Loss}^u(P_j) = \gamma \cdot \text{Loss}(M_j) + \frac{1}{n} \sum_i \text{Loss}(\hat{m}_i) \tag{15}$$

where γ represents a weighting parameter that indicates the contribution of P_j to the global model, and $\text{Loss}(M_j)$ represents the loss of the local training model $\hat{m}_j$. The value of γ is determined by the size of the training data of P_j relative to that of other participants, calculated as $\gamma = 1 + |d_j| / \sum_i |d_i|$.

Leveraging the aforementioned approach, we achieve effective and interpretable FL for trajectory prediction tasks. By representing vehicle trajectory data as a normalized weighted graph, we ensure the preservation of critical private parameters. To prevent data redundancy, we employ a multiparty query approach that eliminates vehicles with high similarity among neighboring nodes while maintaining privacy protection. Recognizing that the quality of vehicle data directly influences the parameter fusion outcomes in the FL training, we introduce a reputation mechanism to address this issue. This mechanism utilizes reputation values calculated from the private trajectory data of noneliminated vehicles, which are then submitted to reviewers for further grouping. The detailed grouping methodology will be discussed in Section V.

The complete process of our proposed reputation-driven hybrid blockchain FL framework is provided in Algorithm 1.

Algorithm 1 Reputation-Driven Hybrid Blockchain FL Framework

Input: The registering vehicles as participating nodes $V = \{v_{it} \mid i = 1, \cdots, n, t = 1, \cdots, \mathcal{T}\}$, the local model of vehicle i, $m_i \in M$. Vote the delegates $r_1, r_2, \ldots, r_n$.
Input: Initialize the permissioned blockchain B and DAG. Initialize the initial global model M_0.
Output: data model M
for each episode e **do**
  Select a leader r_0 from delegates
  for each time slot t **do**
    for each vehicle $v_{it} \in V$ **do**
      v_{it} retrieves global model M_{t-1} from permissioned blockchain B
      v_{it} executes the local training on its local trajectory data
      v_{it} retrieves local model updates from DAG
      v_{it} calculates reputation based on Eq. (13), and upload new reputation segment.
      v_{it} executes local aggregation and obtains updated local model $m_i(t)$
      Construct differential private data model $\hat{m}_i$ with previously received models
      Broadcast $\hat{m}_i$ to the other participants who are engaged in the data sharing process
      v_{it} adds the parameters of model $m_i(t)$ as a transaction to the DAG
    end for
  end for
  The leader r_0 retrieves the current verified updated models from DAG
  r_0 executes the reputation-based PPO algorithm, assigning higher weights to grouped high-reputation vehicle trajectory models, which are then aggregated into $M(e)$ after global federated learning
  r_0 broadcasts $M(e)$ to other delegates for verification, and collects all transactions into a new block
  r_0 appends the block including the global model $M(e)$ to the permissioned blockchain
end for
return The final global model M the requester

V. Enhanced Asynchronous Federated Learning With PPO Clustering Algorithm Based on Vehicle Reputation

In this section, we tackle the challenges of insufficient heterogeneous model aggregation and significant delays in global model updates in FL. We achieve this by integrating the proposed reputation evaluation mechanism into an interpretable AFL framework which employs the reputation-enhanced PPO algorithm.

A. Graph Asynchronous Federated Learning for Trajectory Prediction

In the IoV context, varying computational power and dynamic communication conditions among vehicles result

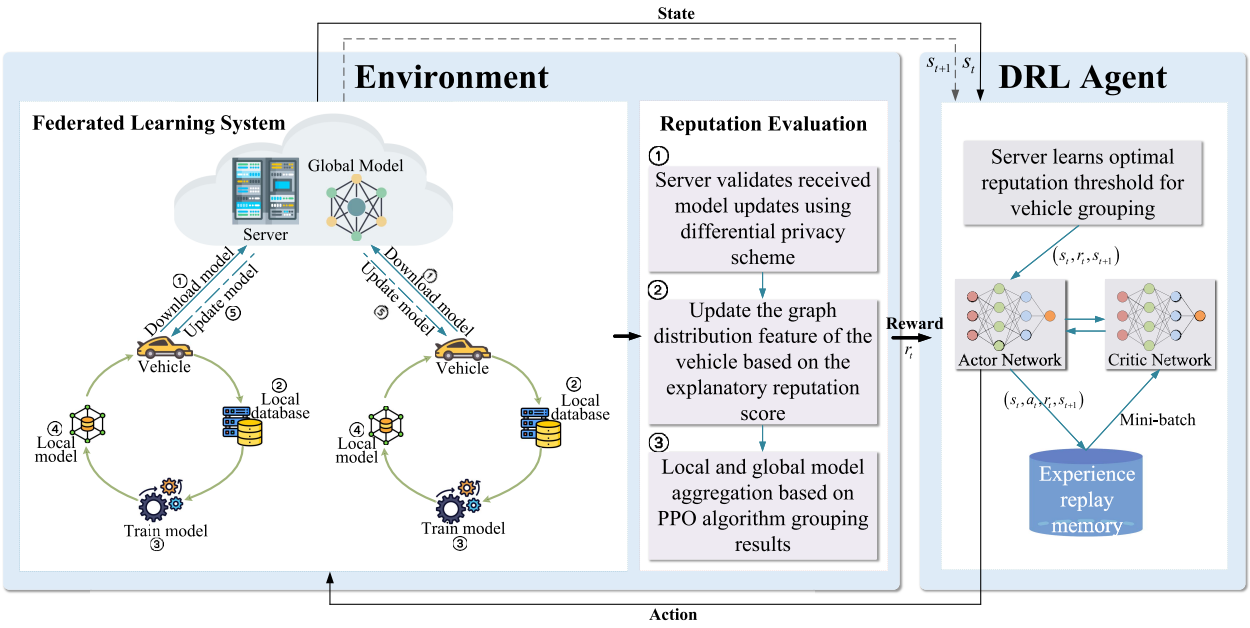

Fig. 4. Reputation-driven PPO vehicle clustering algorithm optimizes AFL to complete global model aggregation and improve vehicle local models.

in disparate learning times. Consequently, the slowest participant dictates the duration of each learning iteration, causing other participants to wait to maintain synchronization. Currently, most applications of FL across various fields use synchronous federated algorithms, such as FedAvg [37], which enhances data security in multiclient model training. However, synchronous FL (SFL) suffers from high communication costs and extended waiting times for idle nodes, complicating training for tasks with spatial and temporal dependencies and delaying global aggregation. Several studies have explored asynchronous learning mechanisms to enhance learning performance [38], [39], but applying these mechanisms to trajectory prediction may result in suboptimal node selection, as their lack of interpretability can introduce biases in vehicle evaluations. To overcome these challenges, this article introduces an AFL approach for trajectory prediction. By incorporating the proposed reputation evaluation mechanism, this method groups nodes using the enhanced PPO algorithm. This approach facilitates asynchronous training among trusted vehicle clusters, improving the efficiency of model updates and aggregation.

With the filtering and reputation value calculations from the previous section, we obtained a set of vehicles with distinct features. Each vehicle, v_i, performs local aggregation asynchronously within its operational range to enhance the quality of its locally trained model. Vehicles preserve and update their DAGs locally to maintain asynchronous consistency with other nodes. In asynchronous consensus, vehicles reach agreement based on historical states rather than the current state. Vehicle i sends updates to its local, DAG_i, and then transmits these updates to nearby vehicles using a gossip scheme to achieve synchronization. This scheme propagates DAG updates across neighboring vehicles, ensuring loose consistency and reducing computational intensity. Although AFL improves aggregation efficiency, the challenge of vehicle grouping remains. Limitations arise from exclusively conducting asynchronous local aggregation on nearby local vehicles to obtain a global model, as it fails to adapt to dynamic traffic conditions and thus cannot effectively optimize local prediction models. To address this, we propose the PPO algorithm with an innovative reward feedback mechanism driven by reputation evaluation. As illustrated in Fig. 4, this mechanism prioritizes the aggregation of high-reputation vehicles, ensuring that more reliable vehicles contribute more significantly to the global model. This method not only improves the quality of local models but also enhances the global aggregation model by incorporating deep aggregation from vehicles with high reputational trust.

B. Reputation-Based PPO Vehicle Clustering Algorithm

To minimize execution time and enhance model accuracy, we propose a reputation-based vehicle clustering approach that categorizes vehicle nodes based on their reputation values. During the global aggregation phase, the disparate computational resources and fluctuating communication conditions among vehicles can hinder efficient execution. To address this, we propose selecting participating vehicles at each timestamp based on their reputation values, ensuring that each cluster contains no more than n vehicles. This method prioritizes the aggregation of trajectory graphs from high-reputation vehicles, which are indicative of high data quality, thereby improving the global aggregation process.

Unlike conventional reinforcement learning, which typically evaluates an agent's performance-based solely on accuracy loss, our approach incorporates data characteristics into node clustering. We introduce the Reputation of Learning (RoL) to represent the reputation value of vehicle i during the

Algorithm 2 PPO-Based Node Clustering Algorithm

INPUT: initial policy parameters θ_0, initial value function parameters ϕ_0

for $k = 0, 1, 2, \ldots$ *iterations* **do**

Collect set of local model $M_k = \{\hat{m}_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.

Compute rewards-to-go $\hat{R}_t$

Compute advantage estimates $\hat{A}_t$, (using any method of advantage estimation) based on the current value function V_{ϕ_k}

Update the policy by maximizing the PPO objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in M_k} \sum_{t=0}^{T} \min \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_k}(a_t \mid s_t)} A^{\pi_{\theta_k}}(s_t, a_t), g\left(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)\right) \right)$$

typically via stochastic gradient ascent with Adam.

Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in M_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2$$

typically via some gradient descent algorithm.

end for

aggregation process in time slot t, defined as follows:

$$\mathcal{C}_r^t = \sum_{i \in V_P} \sigma_i^t(w^t, \mathfrak{d}_i) = \sum_{i \in V_P} \sum_j L\left(y_j - \frac{1}{\mathcal{R}^t(x_j)} \right) \tag{16}$$

where w^t represents the completed combined model in slot t, and $\mathfrak{d}_i = \{(x_j, y_j)\}$ represents the training data of vehicle i. We define the local learning time cost and communication cost of vehicle i as follows:

$$\mathcal{C}_a^t(i) = \frac{\mathfrak{d}_i \cdot \beta_m}{\xi_i(t)}, \mathcal{C}_u^t(i) = \frac{|w_i|}{\tau_i} \tag{17}$$

where β_m denotes the CPU cycles required to train model m per iteration. The total time consumption function is given by

$$\mathcal{C}_e^t = \frac{1}{|V_P|} \sum_{i=1}^{|V_P|} \left(\mathcal{C}_a^t(i) + \mathcal{C}_u^t(i) \right). \tag{18}$$

In this way, the total cost of FL in time slot t can be expressed as

$$\mathcal{C}_f^t(\lambda^t) = \mathcal{C}_r^t + \mathcal{C}_{te}^t \tag{19}$$

where $\lambda^t = [\lambda_i^t]$ in the time step t is an indicator vector for vehicle selection, with $\lambda_i^t = 1$ indicating activation and $\lambda_i^t = 0$ indicating the opposite. We formulate the problem as a combinatorial optimization problem using a Markov decision process, denoted as $\mathcal{M} = (S, V, P_v, C_v)$. The node selection problem can be expressed as

$$\min_{\lambda^t} \ \mathcal{C}_f^t(\lambda^t) \tag{20a}$$

$$\text{s.t.} \quad \lambda_i^t \in \{0, 1\} \quad \forall i \tag{20b}$$

$$\left| p_{i|\lambda_i=1}(t) - p_c(t) \right| \leq r_0^2 \tag{20c}$$

where (20c) ensures that the distance between the selected participating vehicles and the computed centroid cannot exceed a finite distance r_0. We use PPO to solve the problem (20a), which involves updating the system policy based on a value function.

We utilize PPO to determine the optimal resolution for vehicle reputation categorization in AFL. At each time slot t in FL, the system state is represented by $s_t = \{\boldsymbol{\tau}(t), \boldsymbol{\xi}(t), \boldsymbol{\gamma}(t), \boldsymbol{\lambda}(t-1)\}$, where $\boldsymbol{\tau}(t)$ denotes the wireless data rate between vehicles, $\boldsymbol{\xi}(t)$ represents the available computing resources of the vehicles, $\boldsymbol{\gamma}(t)$ indicates the reputation of the vehicles, and $\boldsymbol{\lambda}(t-1)$ signifies the selection state of the vehicles from the previous time slot. The action taken at time slot t, denoted by $\lambda^t = (\lambda_1^t, \lambda_2^t, \ldots, \lambda_n^t)$, corresponds to a vehicle selection decision and can be framed as a 0-1 problem. Specifically, $\lambda_i^t = 1$ when vehicle i is selected as a node with a high reputation value, and $\lambda_i^t = 0$ otherwise. Our primary contribution lies in the customization of the reward function. This function is designed as follows.

1) Add 1 to the reward for selecting a vehicle with a high reputation value.
2) Subtract 1 from the reward for selecting a vehicle with a low reputation value.
3) Subtract 10 from the reward for selecting a vehicle with a low reputation value for five consecutive times and end the round.
4) Reward 20 points if the number of selected high-reputation vehicles reaches a predefined threshold, and ends the round.

The reward function $R(s_t, \lambda_t)$ is used to assess the impact of an action λ_t taken while in state s_t. The reward function is formulated as

$$\begin{aligned} R(s_t, \lambda_t) &= -\frac{1}{\left|\sum_{i=1}^n \lambda_i\right|} \sum_{i=1}^n \mathcal{C}_i^t \cdot \lambda_i^t \\ &= -\frac{1}{\left|\sum_{i=1}^n \lambda_i\right|} \left(\sum_{i=1}^n \lambda_i \left(\frac{\mathfrak{d}_i \cdot \beta_m}{\xi_i(t)} + \frac{|w_i|}{\tau_i} \right) + \sum_{i=1}^n \lambda_i \sigma_i^t(w^t, \mathfrak{d}_i) \right). \end{aligned} \tag{21}$$

The overall cumulative reward is given by

$$\mathbb{E}\left[\sum_{t=0}^{T-1} \gamma R(s_t, \lambda_t) \right] \tag{22}$$

where $\gamma \in (0, 1]$ is the reward discount factor. The PPO algorithm seeks to maximize the reputation accumulation by solving the following optimization problem:

$$\lambda = \arg\max \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma R(s_t, \lambda_t) \right]. \tag{23}$$

Through Algorithm 2, we can identify the groups of vehicles with high and low reputations. The vehicle feature parameters from each group are then combined, followed by a global fusion process, prioritizing the high-reputation group for deep aggregation.

(a) (b) (c)

Fig. 5. Datasets used in the experiments: the NGSIM dataset features vehicle trajectory data collected by multiple digital cameras under various traffic conditions, including light, moderate, and heavy congestion on real highways. The overhead views of the two study areas, US-101 highway [41] and I-80 highway [40], are shown in (a) and (b), respectively, as obtained from Google Maps. The ApolloScape trajectory dataset [42] comprises data collected by a vehicle named "Apollo acquisition car" during rush hours in urban environments. Our proposed approach is validated using these datasets across different traffic scenarios. (c) ApolloScape trajectory.

VI. Experiments

In this section, we first present the well-established datasets utilized in our experiments, followed by an overview of the evaluation metrics employed. We then explore the impact of various model components on prediction performance and compare our approach with previous methods. Additionally, we examine how different iterations of the model affect convergence and analyze the influence of varying numbers of bad nodes on model predictions. All experiments are conducted on a desktop running Ubuntu 20.04, equipped with a 2.30-GHz Intel Xeon E5-2686 v4 CPU, 64 GB of memory, and an NVIDIA 3090 Ti Graphics Card. Our code will be available at https://github.com/lucasc928/TPFL.

A. Datasets

We evaluated our proposed method RAFT on well-known real-world datasets: NGSIM [40], [41], and ApolloScape Trajectory dataset [42], as shown in Fig. 5. The basic info is listed in Table I. NGSIM contains two road segment datasets, US-101 and I-80. Referring to the method [21], [43], [44] for data segmentation of the training set and testing set, a quarter of each of the three road condition data is selected as the test set. Each trajectory is segmented into 8 s, the first 3 s are used as training data, and the last 5 s are used as training labels. The ApolloScape Trajectory dataset was gathered in an urban area during rush hour by a vehicle called the Apollo acquisition car. The data primarily comprises vehicle trajectories that are based on object detection and tracking algorithms. During the initial phase, we adopt GRIP++ [23] to select 20% of the sequences for validation purposes, while the remaining 80% is utilized as the training set.

TABLE I
Dataset Profile

Attributes	ApolloScape	NGSIM
Frequency of sample	2 Hz (downsampled)	5 Hz
Total number of vehicles	5156	11 779
Total frames	5593 (downsampled)	30 476
Train len/pred len	3 s/3 s	3 s/5 s

B. Model Configuration

In this part, we provide the detailed configuration of our model. Our model is built upon GRIP++ [23], where objects are considered neighbors if they are within 25 feet of each other. We employ a 3-layer GCN, with 64 units per layer, followed by temporal convolution layers using a convolutional kernel of size (1×3). To prevent overfitting, we apply dropout with a probability of 0.5 after each GCN layer. The encoder and decoder of the prediction model are both 2-layer GRU networks. We train the model using the Adam optimizer with default settings from the PyTorch Library, a batch size of 64, and 50 epochs.

To achieve a balance between stability and time efficiency, we set the global aggregation frequency to 5 epochs (see Section VI-D4 for further discussion). For blockchain integration, we use the default parameter settings in Hyperledger Fabric [45]. A new block is generated when any of the following conditions are met: the waiting time for the next invoke reaches 2 s, the number of invokes reaches 10, or the block size reaches 10 MB.

For the PPO algorithm, a multilayer perceptron policy network is used with two hidden layers of 64 units each. Key hyperparameters include a learning rate of 0.00025, 2048 steps per update, a batch size of 64, and 10 training epochs. We set the discount factor at 0.99 and the generalized advantage estimation λ at 0.95, with a clipping range of 0.2. The entropy coefficient is 0.01, and the value function coefficient is 0.5. To regulate policy updates, we use a maximum gradient norm of 0.5 and a target KL divergence of 0.01, training for 10 000 timesteps.

C. Metrics

To rigorously evaluate the proposed methodology, we selected several well-established metrics regarding the state of the art in the field [46], each of which captures key aspects of performance in our specific application domain. The root-mean-square error (RMSE) is adopted for its sensitivity to larger errors, thus reflecting the overall prediction accuracy. The average displacement error (ADE) and final displacement error (FDE) are selected to measure the trajectory prediction accuracy over time and at the final timestep, respectively, which are critical for understanding the precision and reliability of the proposed model.

D. Ablation Study

To demonstrate the advantages of our proposed approach, we performed several ablation experiments. This section presents the results and analysis of these experiments.

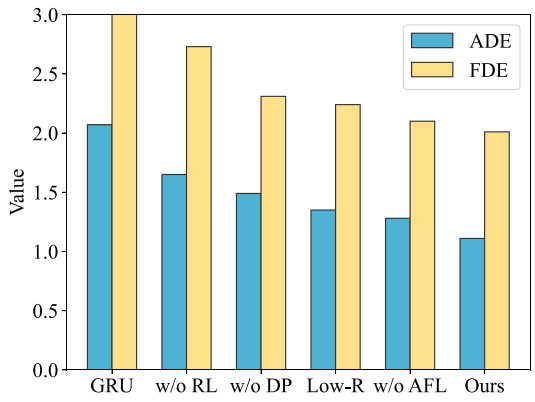

Fig. 6. Influence of different components.

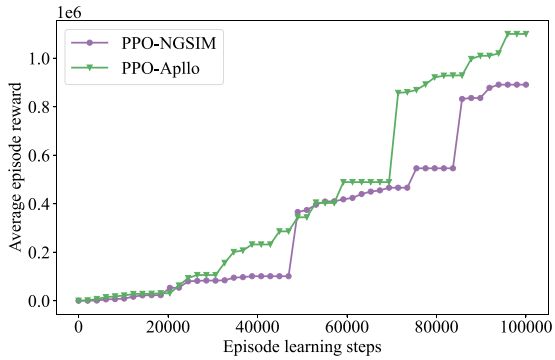

Fig. 7. Average accumulated episode reward of the PPO algorithm during training on different datasets.

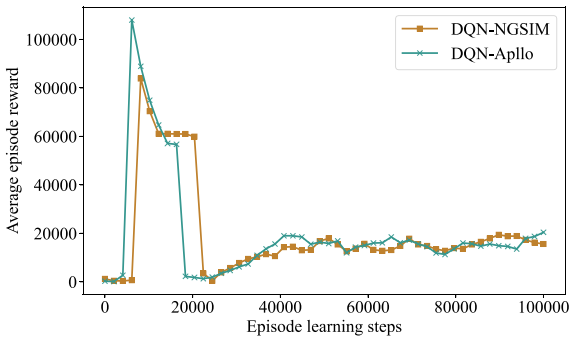

Fig. 8. Average accumulated episode reward of the DQN algorithm during training on different datasets.

1) Influence of Different Components: To evaluate the effectiveness of various components in our framework, we conducted ablation experiments on the ApolloScape Trajectory dataset. We considered five variants: basic encoder–decoder network (GRU-based), w/o reinforcement learning, w/o DP, w/o AFL, priority aggregation of low reputation vehicles (Low-R), and our complete model. As shown in Fig. 6, the basic encoder–decoder network produces the highest error. This is mainly because it does not consider how vehicles interact with each other. GRU models are designed to focus on sequential data but fail to capture the complex relationships between vehicles that affect their trajectories. This lack of interaction modeling leads to lower prediction accuracy. When we remove the reputation-based PPO module, ADE increases noticeably. This underscores the critical role of our PPO-based node grouping approach in enhancing trajectory prediction. By focusing on data from high-reputation vehicles and giving them more weight, we reduce the impact of bad or noisy data. Without this mechanism, the model struggles to handle unreliable data, which harms the predictions. The DP also plays an important role. It enforces constraints to filter out abnormal data, resulting in more robust and secure model aggregation. Without DP, the model becomes more vulnerable to bad data, which could come from attacks or mistakes. This balance between protecting data and keeping quality high is important for practical use. It is noteworthy that the absence of FL leads to lower prediction accuracy compared to the full model. This emphasizes the effectiveness of reputation-based grouping aggregation in alleviating the challenges associated with heterogeneous data fusion and enhancing trajectory prediction accuracy. Finally, giving priority to low-reputation vehicles during aggregation leads to worse prediction results. This confirms that high-reputation vehicles provide higher-quality data. Low-reputation vehicles are more likely to contribute noisy or incorrect data, which disrupts the model's learning process. By focusing on high-reputation vehicles, we can maintain the quality of the aggregated data, ensuring more accurate predictions.

2) Influence of DRL Methods: We compared the learning results of the PPO algorithm and the DQN algorithm using the NGSIM and ApolloScape datasets. Both algorithms were configured with identical network layers, neurons, and training parameters. Figs. 7 and 8 illustrate that the PPO algorithm demonstrates superior stability and adaptability across both datasets. PPO converges more quickly and reliably following positive feedback compared to DQN. Furthermore, our experiments revealed that the PPO algorithm achieves a 34.5% reduction in training time relative to the DQN algorithm. This comparison highlights the advantages of PPO in terms of both convergence speed and training efficiency, making it a more effective choice for the trajectory prediction tasks in our framework.

3) Influence of Bad Nodes: To evaluate the impact of bad nodes on different modules within our framework, we assessed how varying amounts of bad nodes affect the ADE. We randomly selected a certain number of data providers at each iteration and manipulated their data to introduce anomalies. Fig. 9 shows that without DP, the ADE varies with the number of bad nodes. Surprisingly, an increased number of bad nodes can result in a lower ADE. This occurs because the PPO-based vehicle grouping, which assigns low-quality data providers to less critical aggregation groups, reduces their negative impact on the model.

In contrast, Fig. 10 shows the effect of bad nodes when the DP module is active. In this case, the number of bad nodes has no significant impact on the ADE, demonstrating the effectiveness of DP in protecting the model from the influence of anomalous data. The DP module enforces constraints on the data shared by each provider, which prevents the propagation

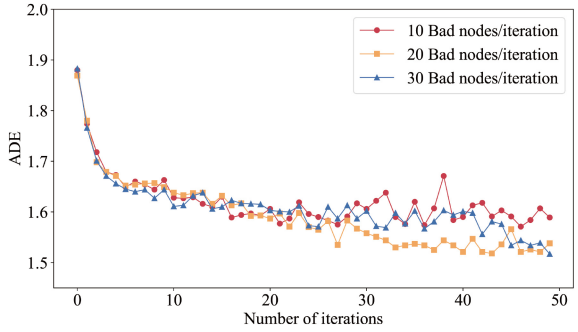

Fig. 9. Impact of bad nodes on ADE w/o DP.

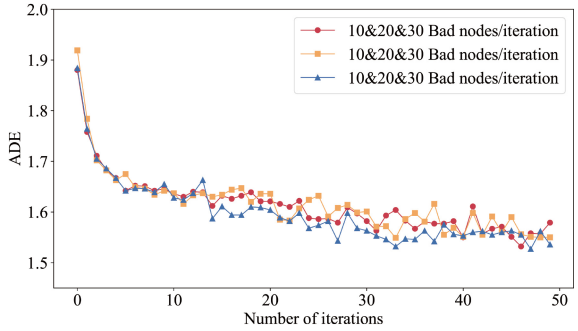

Fig. 10. Impact of bad nodes on ADE: our model (blue solid line) versus w/o reinforcement learning (orange solid line) versus w/o FL module (red solid line).

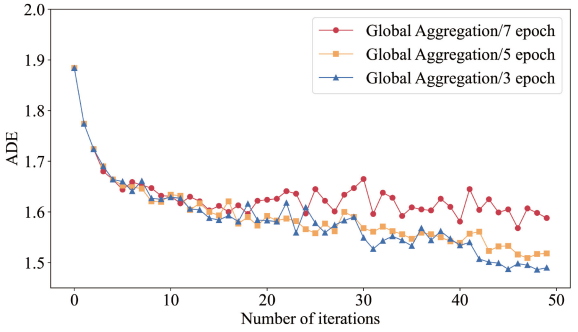

Fig. 11. Effect of global aggregation frequency on ADE.

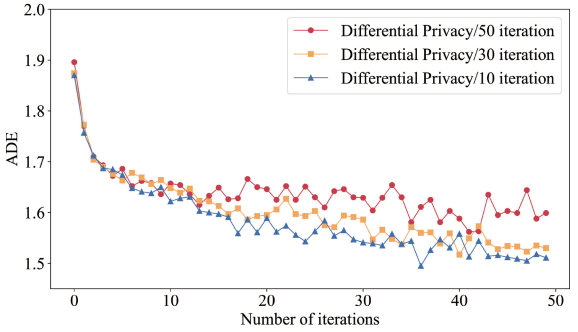

Fig. 12. Effect of DP module invocation frequency on ADE.

of harmful or noisy data throughout the aggregation process. As a result, even with an increasing number of bad nodes, the model maintains high accuracy and stability.

These results emphasize the complementary strengths of both the PPO-based grouping and the DP module in handling bad nodes. While PPO helps to manage the impact of unreliable data by dynamically adjusting the aggregation process, DP provides an additional layer of protection by filtering out anomalies at the data-sharing level. In combination, these mechanisms ensure that the presence of bad nodes does not significantly degrade the performance of the trajectory prediction model. In real-world scenarios, where data quality can vary widely, these features are crucial for maintaining model robustness and reliability.

4) Frequency of Aggregation: We investigated how different local and global aggregation frequencies impact prediction accuracy and explored the effect of varying the invocation frequency of the DP module. As illustrated in Fig. 11, increasing both local and global aggregation frequencies results in faster convergence and enhanced prediction accuracy. The improvement can be attributed to the fact that more frequent updates allow the model to refine its parameters more often, enabling better incorporation of new information from multiple vehicles and leading to more accurate trajectory predictions. This highlights the importance of regular updates in dynamic environments like vehicular networks, where traffic patterns can change rapidly. After testing several combinations, we selected a global aggregation frequency of 5 epochs. While a higher frequency of 3 epochs provides faster convergence, it introduces greater variability in accuracy across iterations, compromising stability. On the other hand, a frequency of 7 epochs, although stable, slows down convergence and reduces efficiency. By setting the global aggregation frequency to 5 epochs, we achieve a balance between convergence speed and stability, which is essential in dynamic vehicular networks where both performance and time efficiency are critical.

We also conducted experiments to analyze how different frequencies of invoking the DP module affect model performance. As shown in Fig. 12, invoking DP more frequently, such as every ten iterations, improves prediction accuracy by managing privacy without significantly affecting model performance. However, too frequent invocations can lead to increased computational overhead, slowing down the training process. Conversely, invoking DP less frequently, say every 50 iterations, results in lower accuracy and makes the model more vulnerable to privacy risks. Considering the tradeoffs between accuracy, computational efficiency, and privacy management, we chose an invocation frequency of 30 iterations. This balance helps the model maintain high predictive accuracy while effectively protecting privacy.

The interaction between aggregation frequency and DP invocation is particularly noteworthy. Higher aggregation frequencies allow for faster learning, but without proper privacy controls, this can lead to vulnerabilities, especially in scenarios where data from different vehicles vary significantly in quality or sensitivity. By balancing frequent aggregation with frequent DP invocation, our framework ensures that the benefits of rapid model updates are not compromised by security risks. This balance is crucial for real-world applications, where both performance and privacy are important.

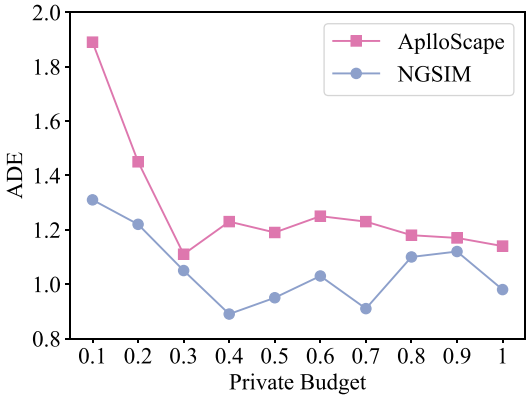

Fig. 13. Impact of private budget ϵ on ADE in different datasets.

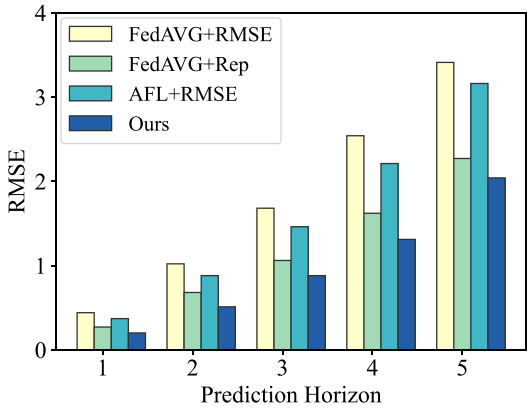

Fig. 14. Impact of reputation quantization mechanism.

5) Impact of Privacy Budget: To assess the effect of DP hyperparameters on prediction accuracy and to balance data utility with privacy, we examine the privacy budget ϵ ranging from 0.1 to 1.0 across different datasets. As depicted in Fig. 13, in the ApolloScape dataset, setting $\epsilon < 0.3$ significantly diminishes data utility due to excessive noise, while stability is observed with $\epsilon \geq 0.3$. However, very high values of ϵ undermine privacy protection. Consequently, $\epsilon = 0.3$ offers a good compromise between privacy and data utility. In contrast, for the NGSIM dataset, which contains a larger number of vehicles and more complex data, we observed that a slightly higher privacy budget of $\epsilon = 0.4$ is needed to maintain a similar balance. This difference can be attributed to the scale and complexity of the dataset. With more vehicles and interactions to account for, the model requires a slightly higher privacy budget to prevent overnoising, ensuring that the essential patterns within the data remain intact. Therefore, $\epsilon = 0.4$ provides the best tradeoff for this dataset, allowing the model to achieve accurate predictions while still offering a reasonable level of privacy protection.

6) Impact of Reputation Quantization Mechanism: To demonstrate the superiority of our proposed interpretable reputation reward mechanism as a quantitative metric for optimizing FL, we conducted ablation experiments on the NGSIM dataset. Specifically, we replaced our AFL algorithm with the traditional SFL scheme FedAVG [37] and used RMSE as the trajectory prediction loss for uploading local parameters to the server for vehicle grouping. As depicted in Fig. 14, the traditional SFL scheme yielded inferior prediction results compared to the AFL scheme. In dynamic trajectory prediction scenarios, waiting for inefficient vehicle nodes to complete local model aggregation inevitably leads to suboptimal global model outcomes. Additionally, substituting RMSE for reputation values for reinforcement learning-based vehicle grouping resulted in a reduction in trajectory prediction accuracy. This is because loss functions provide an incomplete assessment of the impact of vehicle nodes on both local and global aspects. In dynamic traffic environments, vehicles continuously influence each other, leading to unavoidable biases in trajectory prediction. Discarding vehicle nodes solely based on significant biases and rewarding vehicles that consistently exhibit smooth and straight trajectories would be unfair. In contrast, our proposed reputation quantification mechanism, which focuses on vehicle data quality and local trajectory similarity, overcomes these limitations. Prioritizing high-quality data and similar trajectory models for deep aggregation enhances the global model's accuracy and guidance value.

E. Comparison Analysis

With regard to the latest findings in DGInet [46], we have compared our model with seven baseline methods consisting of state-of-the-art trajectory prediction methods.

1) *Encoder–Decoder [8]* employs an enc–dec architecture based on LSTM.
2) *CS-LSTM [21]* combines CNN and LSTM to extract spatiotemporal features.
3) *TraPHic [22]* uses spatial attention pooling combined with CNNs and LSTMs for trajectory prediction.
4) *Social-GAN [47]* utilizes an encoder–decoder architecture as a generator and trains an additional encoder as a discriminator for trajectory prediction.
5) *GRIP [7]* employs graph convolution for trajectory prediction and integrates the results with an encoder–decoder network.
6) *Spectral [8]* forecasts paths and actions using an LSTM-based encoder–decoder network enhanced with spectral clustering.
7) *DGInet [46]* combines a semi-global graph mechanism with a convolutional graph network based on M-products for trajectory prediction.

Table II presents a comparison of ADE and FDE for our method against the aforementioned baseline methods. Although our model did not achieve state-of-the-art results overall, it demonstrates notable improvements compared to specific methods. Specifically, on the ApolloScape dataset, our approach reduces ADE by 11.2% and FDE by 10.2% compared to GRIP and surpasses Spectral's performance. Unlike Spectral, which predicts trajectories for a single vehicle, our method, along with GRIP and DGInet, predicts trajectories for all vehicles in a traffic scenario. This aligns our approach

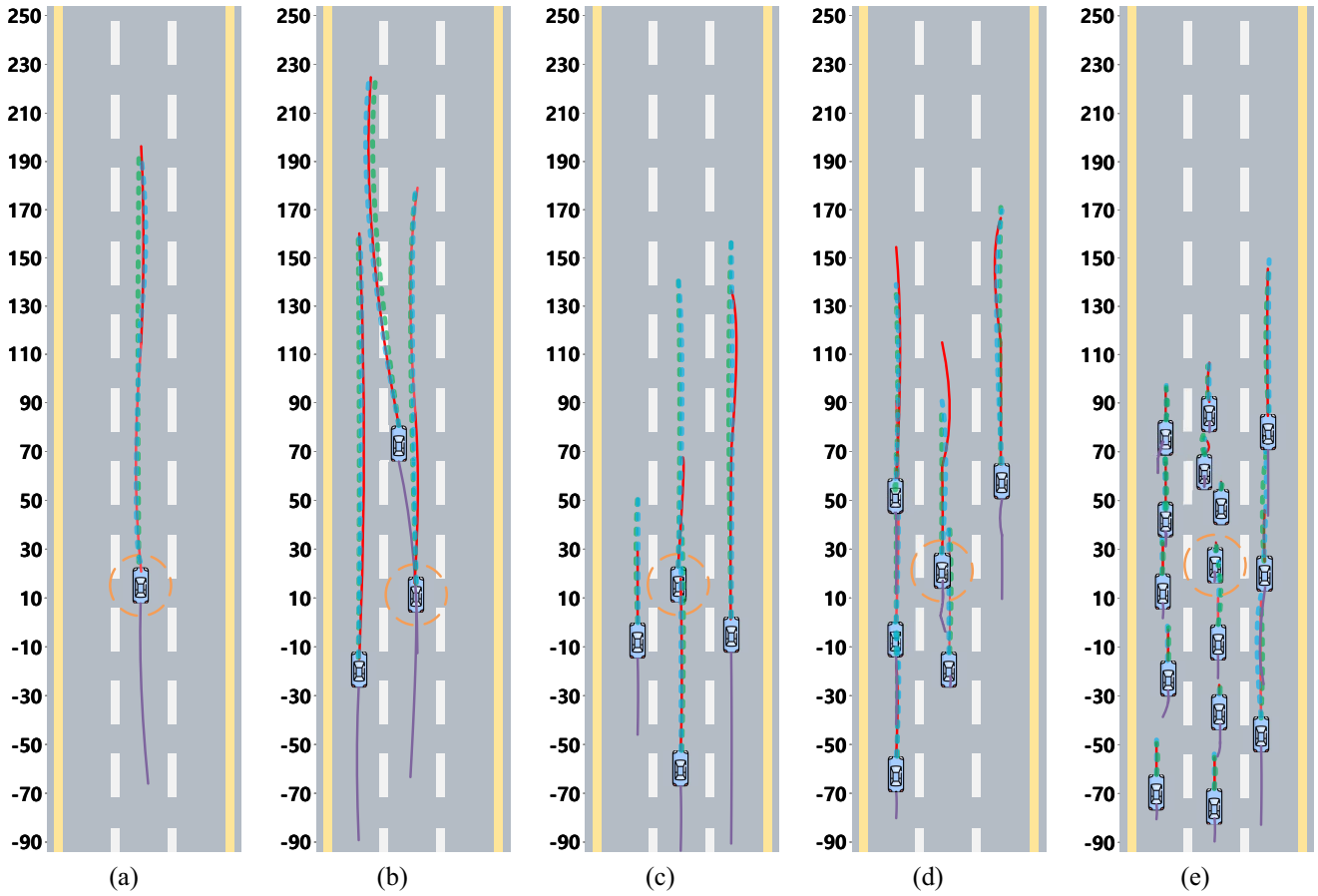

Fig. 15. Visualization of predicted trajectories under mild, moderate, and congested traffic scenarios, where the circled vehicles are those that GRIP++ tries to predict, the purple solid line is the observed history, the red solid line is the future ground truth, the blue dashed line is the prediction result of our model (5 s), and the green dashed line is the GRIP++ prediction (5 s). (a) Only one vehicle; (b) two vehicles with few interactions; (c) three vehicles with more interactions; (d) five vehicles with some interactions in which a vehicle would conduct lane changing; and (e) heavy traffic.

TABLE II
AVERAGE PERFORMANCE COMPARISON

Method	ApolloScape (3s)		NGSIM (5s)	
	ADE	FDE	ADE	FDE
Enc-Dec(LSTM)	2.24	8.25	6.86	10.02
CS-LSTM	2.14	11.69	7.25	10.05
TraPHic	1.28	11.67	5.63	9.91
Social-GAN	3.98	6.75	5.65	10.29
GRIP	1.25	2.34	1.61	3.16
Spectral	1.12	2.05	0.40	1.08
RAFT (Ours)	1.11	2.01	0.98	2.04
DGInet	0.99	1.74	0.37	1.01

more closely with real-world traffic conditions, enhancing the security and robustness of data sharing in distributed trajectory prediction. Thus, while our model may not lead in all metrics, its practical alignment with real-world scenarios and unique contributions to data-sharing security make it a valuable approach.

F. Visualization of Prediction Results

In Fig. 15, we use the NGSIM datasets to show several prediction results under different traffic conditions, where the circled vehicles are those that GRIP++ tries to predict, the purple solid line is the observed history, the red solid line is the future ground truth, the blue dashed line is the prediction result of our model (5 s), and the green dashed line is the GRIP++ prediction (5 s). The observation range is from −90 to 90 feet. Our model predicts trajectories 5 s into the future based on 3 s of historical data. As depicted in Fig. 15, across different traffic scenarios, our model's predictions (blue dashed line) are consistently closer to the actual future trajectories (red solid line) compared to GRIP++ (green dashed line), particularly at the endpoints of the prediction horizon. This highlights the superior accuracy of our approach over GRIP++ in trajectory prediction.

The improved performance of our model is due to its ability to capture complex vehicle interactions through a reputation-based data-sharing mechanism, which filters out low-quality data and focuses on high-reputation sources, ensuring more accurate predictions. The inclusion of DP further enhances robustness by protecting against noisy or anomalous data, maintaining stable performance even in dynamic traffic conditions. Additionally, our model excels in dense traffic scenarios, where GRIP++ struggles due to its reliance on graph convolution alone. By combining graph-based representation with PPO for reputation-based grouping, our model adapts better to the diverse behaviors of vehicles, leading to more precise trajectory predictions.

TABLE III
COMPUTATION TIME

Scheme	Predicted #	Times (s) batch size 128	Times (s) batch size 1
CS-LSTM	1000	0.29	35.13
GRIP	1000	0.05	6.33
GRIP++	1000	0.02	1.62
Spectral	1000	1.10	120.20
DGInet	1000	0.07	6.42
RAFT (Ours)	1000	0.38	5.278

G. Computation Time

Computational efficiency is a crucial performance metric for autonomous vehicles. Table III presents the computation time of our proposed approach, implemented using PyTorch. We evaluated the prediction time for 1000 vehicles with batch sizes of 128 and 1, respectively. Our trajectory prediction model builds upon the GRIP++ framework. Although our approach incurs a slightly longer computational time compared to GRIP++, it remains competitive relative to other methods. The additional latency is offset by the benefits of enhanced data privacy protection and improved prediction accuracy. Importantly, the PPO-based vehicle grouping method operates on the server, where the pretrained PPO model is deployed. This approach requires only periodic experience replay [48] and asynchronous updates, thus not affecting the prediction efficiency of the vehicles themselves.

VII. CONCLUSION

In this article, we introduced a novel framework for distributed trajectory prediction that leverages advanced graph neural network techniques and AFL. Our approach integrates an interpretable reputation quantization mechanism and employs DP to secure data sharing. By applying PPO for vehicle categorization based on reputation, we optimized the efficiency of FL aggregation. Our experimental results validate that this framework effectively handles bad nodes, enhances the security of distributed trajectory prediction tasks, and improves prediction accuracy. The proposed method offers a robust solution to the challenges of data privacy and quality in dynamic traffic environments. Future work will focus on integrating this approach with 6G networks to further enhance efficiency and scalability.

REFERENCES

[1] K. Yu, L. Lin, M. Alazab, L. Tan, and B. Gu, "Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system," *IEEE Trans. Intell. Transport. Syst.*, vol. 22, no. 7, pp. 4337–4347, Jul. 2021.

[2] Z. Xiao, J. Shu, H. Jiang, G. Min, H. Chen, and Z. Han, "Perception task offloading with collaborative computation for autonomous driving," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 457–473, Feb. 2023.

[3] J. Wang, X. Yuan, Z. Liu, W. Tan, X. Zhang, and Y. Wang, "Adaptive dynamic path planning method for autonomous vehicle under various road friction and speeds," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10977–10987, Oct. 2023.

[4] Z. Peng, Y. Jiang, and J. Wang, "Event-triggered dynamic surface control of an underactuated autonomous surface vehicle for target enclosing," *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3402–3412, Apr. 2021.

[5] H. Xu, W. Liu, M. Jin, and Y. Tian, "Positioning and contour extraction of autonomous vehicles based on enhanced DOA estimation by large-scale arrays," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11792–11803, Jul. 2023.

[6] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 652–674, Sep. 2022.

[7] X. Li, X. Ying, and M. C. Chuah, "GRIP: Graph-based interaction-aware trajectory prediction," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3960–3966.

[8] R. Chandra et al., "Forecasting trajectory and Behavior of road-agents using spectral clustering in graph-LSTMs," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4882–4890, Jul. 2020.

[9] Q. Li et al., "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, Apr. 2023.

[10] Z. Peng et al., "VFChain: Enabling verifiable and auditable federated learning via blockchain systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 173–186, Jan./Feb. 2022.

[11] C. Dwork, "Differential privacy," in *Proc. 33rd Int. Conf. Automata, Languages Programming*, 2006, pp. 1–12. [Online]. Available: https://doi.org/10.1007/11787006_1

[12] S. Shen, Y. Han, X. Wang, and Y. Wang, "Computation offloading with multiple agents in edge-computing–supported IoT," *ACM Trans. Sensor Netw.*, vol. 16, no. 1, p. 8, Dec. 2019. [Online]. Available: https://doi.org/10.1145/3372025

[13] X. Wang, R. Li, C. Wang, X. Li, T. Taleb, and V. C. M. Leung, "Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 154–169, Jan. 2021.

[14] P. Zhang, C. Wang, C. Jiang, and Z. Han, "Deep reinforcement learning assisted federated learning algorithm for data management of IIoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8475–8484, Dec. 2021.

[15] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 433–443, Jan. 2020.

[16] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4298–4311, Apr. 2020.

[17] T. M. Ho, K.-K. Nguyen, and M. Cheriet, "Federated deep reinforcement learning for task scheduling in heterogeneous autonomous robotic system," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 1, pp. 528–540, Jan. 2024.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[19] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 976–985, 2021.

[20] Y. Wang, W. Yang, D. Li, and J. Q. Zhang, "An RFS time-frequency model for decomposing chirp modes with dynamic cross, appearance, and disappearance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 4525–4539, Aug. 2023.

[21] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2018, pp. 1549–15498.

[22] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "TraPHic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8475–8484.

[23] X. Li, X. Ying, and M. C. Chuah, "GRIP++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving," 2020, *arXiv:1907.07792*.

[24] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2017, *arXiv:1610.0549*.

[25] S. Warnat-Herresthal et al., "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[26] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[27] Q. Yang and P. Li, "Deep reinforcement learning based energy scheduling for edge computing," in *Proc. IEEE Int. Conf. Smart Cloud (SmartCloud)*, 2020, pp. 175–180.

[28] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, Oct. 2018.
[29] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based Decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.
[30] E. M. Ghourab, M. Azab, and N. Ezzeldin, "Blockchain-guided dynamic best-relay selection for trustworthy vehicular communication," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13678–13693, Aug. 2022.
[31] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25408–25420, 2017.
[32] J. Kang et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.
[33] Y. Liu, Y. Wang, and G. Chang, "Efficient privacy-preserving dual authentication and key agreement scheme for secure V2V communications in an IoV paradigm," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2740–2749, Oct. 2017.
[34] K. Karlsson et al., "Vegvisir: A partition-tolerant blockchain for the Internet-of-Things," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 1150–1158.
[35] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Peer-to-Peer Systems*, G. Goos, J. Hartmanis, J. Van Leeuwen, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Heidelberg, Germany: Springer, 2002, vol. 2429, pp. 53–65.
[36] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Security Privacy (SP)*, 2016, pp. 839–858.
[37] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2016, pp. 1–10. [Online]. Available: https://api.semanticscholar.org/CorpusID:14955348
[38] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "An asynchronous mini-batch algorithm for regularized stochastic optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3740–3754, Dec. 2016.
[39] S. Ko, K. Lee, H. Cho, Y. Hwang, and H. Jang, "Asynchronous federated learning with directed acyclic graph-based blockchain in edge computing: Overview, design, and challenges," *Expert Syst. Appl.*, vol. 223, Aug. 2023, Art. no. 119896.
[40] J. Halkias and J. Colyar, "U.S. highway 80 dataset," Federal Highway Admin., Washington, DC, USA, Rep. FHWA-HRT-07-030, 2007.
[41] J. Halkias and J. Colyar, "U.S. highway 101 dataset," Federal Highway Admin., Washington, DC, USA, Rep. FHWA-HRT-07-030, 2007.
[42] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.
[43] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 129–140, Jun. 2018.
[44] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2018, pp. 1179–1184.
[45] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–15. [Online]. Available: https://doi.org/10.1145/3190508.3190538
[46] J. An, W. Liu, Q. Liu, L. Guo, P. Ren, and T. Li, "DGInet: Dynamic graph and interaction-aware convolutional network for vehicle trajectory prediction," *Neural Netw.*, vol. 151, pp. 336–348, Jul. 2022.
[47] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.
[48] X. Liang, Y. Ma, Y. Feng, and Z. Liu, "PTR-PPO: Proximal policy optimization with prioritized trajectory replay," 2021, *arXiv:2112.03798*.

Weiliang Chen received the B.S. degree in software engineering from Heilongjiang University, Harbin, China, in 2022, where he is currently pursuing the M.S. degree in computer science and technology.

His current research interests include deep learning, urban data mining, and trustworthy artificial intelligence.

Li Jia received the Ph.D. degree from the East China University of Science and Technology, Shanghai, China, in 2003.

She is currently with the Faculty of Mechanical and Electrical Engineering and Automation, Shanghai University, Shanghai. Her research interests include industrial process control, artificial intelligence, and energy data modeling.

Yang Zhou received the Ph.D. degree from the East China University of Science and Technology, Shanghai, China, in 2021.

She is currently with the Faculty of Mechanical and Electrical Engineering and Automation, Shanghai University, Shanghai. Her research interests include artificial intelligence, Internet of Vehicles, Internet of Things, and modeling.

Qianqian Ren received the Ph.D. degree from the Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, in 2012.

She is currently a Professor of Computer Science with Heilongjiang University, Harbin. She visited the University of Alberta, Edmonton, AB, Canada, as a Visiting Scholar from 2018 to 2019. Her research interests include wireless sensor networks, data analytics, intelligent transportation, and deep learning.