Computers and Electrical Engineering 120 (2024) 109678

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Road network traffic flow prediction: A personalized federated learning method based on client reputation

Guowen Dai, Jinjun Tang *, Jie Zeng, Chen Hu, Chuyun Zhao

Smart Transport Key Laboratory of Hunan Province, School of Transport and Transportation Engineering, Central South University, Changsha, 410075, China

ARTICLE INFO

Keywords:
Traffic flow prediction
Data privacy
Federated learning
Personalization

ABSTRACT

Accurate traffic flow prediction can provide effective decision-making support for traffic management, alleviate traffic congestion, and improve road traffic efficiency. Traffic flow data contains personal privacy information, such as vehicle trajectories, driving speed, etc. However, most existing research focuses on using all local data to jointly construct prediction models, facing data security and privacy issues. In response to these challenges, this paper presents a Personalized Federated Learning method based on Client Reputation for traffic flow prediction. This method avoids direct sharing of original data by performing model training and information aggregation locally, thereby better protecting user privacy. We initially constructed the foundational model by integrating Graph Neural Networks (GCNs) and Gated Recurrent Units (GRUs) for federated learning. GCNs are capable of capturing the spatial relationships between nodes in a road network, thereby extracting the spatial features of traffic flow. On the other hand, GRUs has advantages at handling sequential data, effectively capturing the dynamic temporal characteristics of traffic flow. Consequently, by combining GCNs and GRUs, we are able to build a model that captures both the spatial and temporal features of traffic flow. Subsequently, we employed a federated learning framework to train the model. In the training process, we propose two evaluation algorithms to evaluate the reputation of all clients participating in federated learning. Then, different weights are customized for each client through a personalized algorithm based on the client's reputation to improve its personalization. Finally, we validated the method on the License Plate Recognition (LPR) datasets collected in Changsha city, China. The experimental results indicate that the method can achieve good prediction performance and stability while protecting the privacy of all participants.

1. Introduction

Traffic flow prediction is a pivotal ingredient in intelligent transportation systems, which aims to accurately predict future traffic flow distribution, congestion level, and travel patterns on the road network, as well as provide decision support for traffic management and planning [1-3]. Traffic flow prediction necessitates the processing of a substantial amount of spatiotemporal data, encompassing historical traffic flow, weather information, road events, and more [4]. Traditional traffic flow prediction methods are typically based on statistical or physical models, but these methods fail to capture the spatiotemporal characteristics and complex nonlinear

* Corresponding author.
E-mail address: jinjuntang@csu.edu.cn (J. Tang).

https://doi.org/10.1016/j.compeleceng.2024.109678
Received 28 June 2024; Received in revised form 20 August 2024; Accepted 7 September 2024
Available online 19 September 2024

relationships of the data. In recent years, machine learning methods have made significant progress in traffic flow prediction, such as neural network-based methods that can effectively capture the spatiotemporal relationships of data [5]. However, these methods usually require a large amount of training data, and privacy issues are usually involved as the data is often distributed in different areas. The traffic data between cities or regions are generally not shared, while the data collected in many areas suffer from quality problems. In this condition, the performance of traditional centralized training models will be greatly limited. The quality of training data and the prediction ability of the model can be effectively improved if different cities or regions can jointly train the traffic flow prediction model without leaking their data, especially for cities or regions with finite data. However, all cities and regions currently have high confidentiality requirements, which limits data-sharing training. Thus, the data privacy and confidentiality issues related to road network traffic flow prediction need to be resolved urgently.

In this regard, federated learning was proposed to address the above challenges. Federated learning allows model training to be distributed across data holders without sharing raw data [6], which has been widely applied in the fields of healthcare and finance. Federated learning can bring manifold benefits in traffic flow prediction such as data privacy protection, data diversity utilization, and cross-border cooperation. It provides a safe, efficient, and scalable method for traffic management and travel decision-making. Traffic flow prediction typically requires collaboration across different administrative jurisdictions and traffic management agencies. Traditional centralized methods often require the establishment of complex data-sharing and cooperation mechanisms. By conducting model training and information aggregation locally, federated learning enables participants to collaborate more flexibly across boundaries, promoting the accuracy and scalability of traffic flow prediction. In certain regions or time periods, traffic data may be insufficient or unbalanced, which can harm the accuracy of prediction models. Through federated learning, participants can share model parameters instead of raw data and use global model parameters to predict data-sparse regions or time periods, thereby offsetting data deficiencies and imbalances.

Nevertheless, federated learning encounters challenges in traffic flow prediction. In real-world scenarios, the disparity in data volume among participating clients in federated learning leads to varying impacts on model parameters during local training for each client. Clients with large amounts of data will have an excessive impact on the final global model, while other clients may lose personalization. In addition, some clients may lie about their data contribution value during the federated learning communication process to increase their profits, which is unfair to other clients. Federated learning needs to solve the problem of model aggregation to ensure that the models uploaded by each client can be effectively integrated and the weight ratio of each client can be reasonably distributed while protecting privacy.

To address the difficulties and challenges in the intersection of traffic flow prediction and federated learning, we propose a personalized federated learning method for traffic flow prediction based on client reputation (PFGCN-GRU). The primary contributions are outlined as follows:

(1) We propose a federated learning method based on GCN-GRU for traffic flow prediction in road networks. This method ensures the protection of each client's data privacy while fully utilizing the data from all parties to train a global prediction model.
(2) We introduce a personalized federated learning method based on client reputation. By evaluating and calculating the reputation of models uploaded by each client, then ranking them accordingly, we customize the weight parameters of the aggregated models based on the clients' reputation scores to achieve personalization for each client.
(3) We validate the proposed reputation-based personalized federated learning framework for short-term traffic flow prediction on the license plate recognition (LPR) datasets in Changsha City, China. The experimental results show that the framework can achieve better prediction accuracy when the data of all parties are not interoperable, and the method is highly portable.

The organization of this paper is structured as follows: Section 2 presents the related work, Section 3 outlines the methodology, Section 4 provides a description of the data, Section 5 discusses the experimental analysis, and Section 6 offers the conclusion.

2. Related works

2.1. Traffic flow prediction

Over the past few decades, many methods related to traffic flow prediction have been developed, which can be roughly divided into two categories, i.e., model-driven and data-driven. Model-driven methods describe and predict complex and changing traffic operating conditions by establishing a mathematical model to characterize dynamic traffic flow [7-9] The data-driven method mainly focuses on data when the model is uncertain, combining the research objectives with current and massive historical data, and utilizing massive data to continuously learn and tune the model, thereby achieving estimation of the operating state of traffic flow [10-12]. With the development of technology, data-driven traffic flow prediction models have received extensive attention [13]. For example, Tang et al. used clustering methods at the data level to aggregate traffic data and mine their intrinsic relationships, and achieved more accurate prediction results [14]. Although this type of traffic flow prediction model can achieve good prediction results on individual road segments, it cannot capture the spatial correlation between traffic counting points. Therefore, it is difficult for these models to play a role in network-level traffic flow prediction.

In recent years, predicting traffic flow at the road network level has emerged as a prominent trend within the field of traffic flow prediction [15-18]. The network-level traffic flow prediction model is designed to forecast the future traffic conditions across all traffic stations simultaneously. According to whether the spatial arrangement of these traffic stations is regular or not, this type of spatiotemporal data can be further divided into two categories: spatiotemporal grid data and spatiotemporal network data. For spatiotemporal grid data, urban areas are generally divided into grids of equal sizes through data preprocessing, and then neural networks are used to model the spatiotemporal grid data. For example, the DMVST-Net (Deep Multi-View Spatial-Temporal Networks) model proposed by Yao et al. [19] uses the DTW (Dynamic Time Warping) algorithm to calculate the traffic similarity between any two traffic objects to assist in modeling the global meaning spatial correlation. The model was successfully applied to taxi demand forecasting tasks and achieved good forecasting results. Guo et al. [20] introduced three-dimensional convolution into the field of spatiotemporal flow prediction, and the experimental results proved that it can effectively extract spatiotemporal features from two dimensions of space and time. Although these models can achieve good prediction results on spatiotemporal grid data, most of the traffic data in actual traffic scenarios is usually collected from irregular topological traffic systems, such as urban road networks, subway networks, highway station networks, etc. It is a type of irregular spatiotemporal network data. These models are unable to model such spatiotemporal network data, and their application scope is limited.

To address this challenge, scholars have begun to attempt to use Graph Convolutional Networks (GCN) [21] to model spatial correlations on spatiotemporal network data. For example, the DCRNN (Diffusion Convolutional Recurrent Neural Networks) model proposed by Li et al. [22] applies diffusion graph convolution to the field of spatiotemporal flow prediction to simulate the propagation and diffusion process of traffic signals in spatial road networks. It can effectively capture spatial dimensional features. After capturing spatial features, combined with GRU to model temporal correlation. The STGCN model proposed by Yu et al. [23] extracts feature in both temporal and spatial dimensions by combining CNN and GCN. Different from previous methods, it uses CNN instead of RNN-type models to effectively model temporal correlation, which is somewhat novel. Zhao et al. proposed a traffic congestion calculation method based on image processing methods and trajectory feature learning using time series prediction [24]. The MS-HAIE module designed by Xu et al. aims to capture spatial interaction information at different time scales. In addition, the CA-STIM module utilizes cross-attention to comprehensively capture the interaction of intelligent agents, integrating the spatial and temporal features of agent trajectories, thereby enhancing prediction accuracy [25].

2.2. Federated learning and personalization

With the advancement of intelligent transportation systems, the volume and intricacy of traffic data are consistently expanding, simultaneously raising significant concerns about privacy and security. For example, LPR data contains a large amount of license plate information and vehicle trajectory information. In this era where data privacy is highly valued, it is difficult for people to tolerate the leakage of their travel trajectory and personal information. In this context, federated learning, as a distributed machine learning paradigm, can effectively solve data privacy and security issues in the transportation field and improve the transplantation and generalization capabilities of traffic prediction models [26]. In 2016, Google proposed the concept of federated learning [27]. In 2017, the classic federated averaging algorithm, a deep network federated learning method underpinned by iterative model averaging, was born [28]. This algorithm is capable of training high-quality models with a relatively small number of communication rounds. In 2019, in order to fit the data of each client corresponding to different task distributions, Mikhail et al. [29] applied the meta-learning (Model-Agnostic Meta-Learning, MAML) of the unknown model to the FL scenario, and used MAML to obtain a global model, and then perform fine-tuning on each client to help learn on new tasks by mining this meta-distribution information. In 2020, Zhou et al. [30] introduced a cost-effective optimization framework designed to synchronize edge devices. Simulation results validated that the proposed algorithm effectively reduces both delays and communication rounds. In 2022, Shashi et al. [31] proposed a method to select devices that provide newer ones to achieve improved generalization, fast convergence, and better device performance, using an improved truncated Monte Carlo method to estimate the device contribution and reduce communication overhead. In 2024, Wang et al. [32] propose a federated learning framework that incorporates layered protection and multiple aggregation methods to enhance data security while simultaneously addressing the challenges of accuracy, training time, and communication overhead.

However, during the training process of federated learning models, by reason of the large differences in data distribution among the parties participating in federated learning training, it can result in poor performance between the client model and the global model. The idea of personalized federated learning is different from training a single global model. It seeks to correct the client model using local data on different clients, making the client model more suitable for local clients. According to the solution approach of personalized federated learning, it can be divided into two types: model partitioning based and regularization based.

Method based on model partitioning: Divide the neural network model into public and private parts, and use local data from the client to train the private part. Fedper [33] divided the model into basic layer and personalized layer parts. During the training process, each client only shares the base layer of the model for federated averaging. The personalized part of the model is always retained for training on the client side using local data.The implementation approach of FedBABU [34] is similar to FedPer, which divides the neural network model into three parts: I/O, head, and body. During the communication process, only the model parameters of the head section are uploaded to participate in the aggregation update of the common model. LG Fedavg [35] uses the client model as a feature extractor and employs supervised, semi supervised, unsupervised, and adversarial training for feature extraction. By conducting multitasking training on the client model, the feature extraction ability of the client model is improved. The common model is used to combine the features extracted by the client model and complete the corresponding classification or regression tasks.The method based on model partitioning essentially involves using different parts of the model for feature extraction and personalized learning, but this type of method does not completely solve the problem of slow convergence of the common model due to the large differences between the common model and the client model.

Based on regularization methods: Divergent data distributions on the client side can give rise to substantial disparities among the trained client models, thereby considerably impeding the convergence speed of the global model. To address the non-convergence issue arising from substantial parameter differences between models, Fedmmd [36] introduced a “dual stream model” during training to regulate the client model. The intention is to mitigate distinctions between the client model and the global model using this approach. FedProx [37] constrains the parameters of the client model to bring the model into close proximity to the shared model. The approach of regularizing the parameters of the client model to enhance its closeness to the common model for accelerating the convergence speed is achieved at the expense of sacrificing model accuracy. This method not only results in a reduction in the efficacy of the common model but also diminishes its applicability on the client side. Yi et al [38]. introduces pFedKT, an innovative personalized federated learning method that supports dual knowledge transfer—leveraging local supernets for historical local knowledge and contrastive learning for global knowledge—thereby enabling language models (LMs) to better balance personalization and generalization. But it has the drawback of high computational and communication costs. Zhao et al [39]. presents a personalized robust federated learning (PRFL) approach for ultra-short-term wind power forecasting that leverages spatiotemporal correlations in a privacy-preserving manner to enhance collaboration among wind farms, employing a Bi-LSTM-based standard federated learning framework, a geometric median-based aggregation scheme to improve robustness against anomalies, and a personalized federated learning strategy incorporating transfer learning to address the challenges of model applicability due to data heterogeneity. However, as the data size increases, the experimental results will deteriorate.

Therefore, in order to obtain a relatively comprehensive personalized federated learning method, it is necessary to ensure that its computational complexity is not too high and can adapt to different data volumes.

Personalized federated learning methods focus on customizing models for each client, with commonly used methods including meta learning, multi task learning, and model distillation. These methods focus more on how to utilize local information of the client to enhance personalized effects, rather than adjusting the global model through reputation. However, introducing client reputation can significantly enhance the understanding of the model aggregation process and the reliability of the results. Client reputation refers to the reliability and honesty demonstrated by the client during the federated learning process. High reputation clients are typically able to provide high-quality local model updates, while low reputation clients may provide low-quality or malicious model updates. By introducing client reputation, the server can identify and reduce the impact of low reputation clients on the global model, thereby improving the overall reliability and stability of the model. This is particularly important in situations where there are malicious clients or significant differences in data quality. High reputation clients usually provide better local updates, and personalized weights are adjusted based on their reputation, which can better adapt the global model to the data distribution of these high reputation clients and improve personalized effects.

In summary, the rapid development of computer technology and big data technology in the information age poses significant challenges to the security of traffic data. Although there have been some studies on traffic data security recently, there are still relatively few specialized studies on traffic data privacy overall. Federated learning, as an emerging distributed machine learning method, has data privacy protection characteristics that are very suitable for current research. Therefore, we introduce federated learning into the research of traffic flow prediction. And in order to better adapt the global model to the characteristics of each client, we propose a personalized federated learning method based on client reputation.

3. Methodology

3.1. Problem formulation

A spatial traffic network is commonly defined as an undirected graph $G = (V, E, A)$, where V represents the set of all stations, E is the set of edges, and N is the total number of nodes, denoted as $N = |V|$. Each traffic station, indexed as i, is represented by vi. The adjacency matrix $A \in R^{N \times N}$ of the spatial traffic network directly captures the adjacency relationships between stations.

The objective of traffic flow prediction is to forecast the flow value for a specific period in the future, relying on the historical traffic flow values from several stations. The traffic flow value observed on the spatial traffic network G at the $t-th$ time step is defined as a graph signal matrix $X(t) \in R^{N \times F}$, where F refers to the number of features collected by each station. Define $X = (X^{(t-T+1)}, X^{(t-T+2)}, \cdots, X^{(t)}) \in R^{T \times N \times F}$ is the input graph signal matrix. $Y = (X^{(t+1)}, X^{(t+2)}, \cdots, X^{(t+P)}) \in R^{P \times N \times F}$ is the graph to be predicted Signal matrix. The objective of the traffic flow prediction task is to learn a functional mapping relationship, as depicted in the formula:

$$[X, G] \xrightarrow{f(.)} Y \tag{1}$$

In order to address common issues in traffic flow prediction, such as low data quality, poor data privacy, and weak model generalization, this paper introduces federated learning into the research on traffic flow prediction. Federated learning is an emerging distributed machine learning method, that regards each participant (such as a government or institution) as a "client" and provides a fair third party (server) for model aggregation. Federated learning can conduct joint training through its unique "client-server" training mechanism without touching the data of all parties. The global model trained in this way can be applied to individual clients. This paper improves the basic federated learning and adds a personalized mechanism based on client reputation. This mechanism can greatly guarantee the interests of each client and maintain the fairness of the entire federated learning mechanism. The details of the federated learning framework will be described in Section 3.3. In Section 3.2 we propose a GCN-GRU model as a local model for each client to explore the dynamic spatiotemporal patterns inherent in traffic data and discover their intrinsic spatiotemporal dependencies.

3.2. Traffic prediction model

The overall framework of the prediction model is shown in Fig. 1. As illustrated in Fig. 1, the process begins with the preprocessing

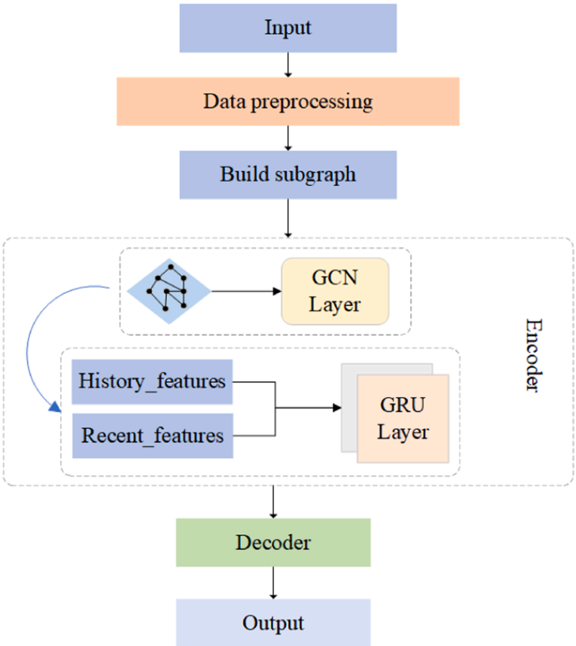

Fig. 1. Prediction model construction.

of historical data, followed by the construction of a traffic flow spatial node graph based on the spatial distances between collection nodes. This graph is employed to model the complex dependencies among the nodes. Subsequently, at each sampling time point, the traffic flow and distance weight parameters between nodes are utilized as inputs. Once the spatial node graph structure is established, spatial feature adaptive learning and graph convolution operations are conducted using a GCN to extract the spatial characteristics of traffic flow. The obtained spatial feature information is then converted into sequence data and fed into a GRU. Finally, the extracted node information is encoded by a GRU-based encoder, and the resulting encoded output is processed by a decoder, which employs the same parameter matrix as the encoder to produce the final predicted output after decoding.

After completing the construction of the spatial node graph, the information interaction between nodes is realized through GCN to model the spatial dependencies between nodes.

$$H^{(l+1)} = \sigma\left(\widetilde{D} - \frac{1}{2}\widetilde{A}\widetilde{D} - \frac{1}{2}H^{l}\theta^{l}\right) \tag{2}$$

In the formula: $H^{(l+1)}$ and $H^{l}$are the output of layer $l+1$ and layer l respectively; σ represents the Sigmoid activation function of the model; $\widetilde{D}$ is the normalized degree matrix; $\widetilde{A}$ is the feature matrix; θ^{l} is the parameter set of the $l-th$ layer. The overall training is performed by backpropagation, and a 2-layer GCN network is selected for forward propagation. After forward calculation, the data represented by the graph is projected to the spectral domain for spatial feature learning.

3.2.1. Sequence-to-sequence model

This paper employs GRU to capture the temporal features in traffic flow data. The GRU model primarily comprises a reset gate and an update gate. Where i_t is the output of the update gate, O_t is the output of the reset gate, h_t is the hidden state at time t, and h_{t-1} is the hidden state at time $t-1$. The reset gate stitches the hidden state of the previous moment with the input data of the current moment and then scales the data through the tanh activation function to achieve selective memory of the hidden state of the previous moment. The update gate determines the inflow of information by calculating the previous moment and the current moment.

The specific calculation process is as follows:

$$i_t = \sigma\left(W^{i}X_{*}^{(t)} + U^{i}h_{t-1}\right) \tag{3}$$

$$O_t = \sigma\left(W^{O}X_{*}^{(t)} + U^{O}h_{t-1}\right) \tag{4}$$

$$h_t^{*} = \tan\left(WX_{*}^{(t)} + O_t \odot Uh_{t-1}\right) \tag{5}$$

$$h_t = h_t^{*} \odot (1 - i_t) + i_t \odot h_{t-1} \tag{6}$$

In the above formula, $X_{*}^{(t)}$ is the image signal calculated by multi-layer convolution, h_t^{*} is the hidden state calculated based on the reset gate, W^{i}, W^{O},W, U^{i}, U^{O}, U is a learnable parameter, which participates in the calculation in the form of a weight matrix. $\odot$ is

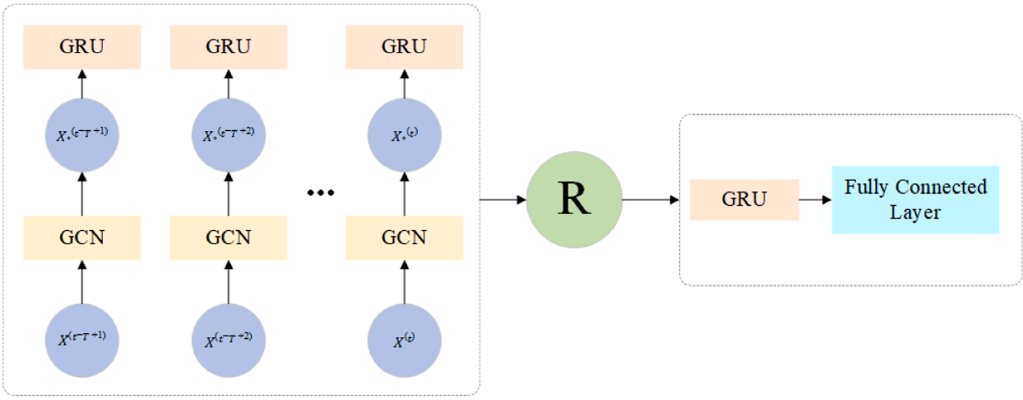

Fig. 2. The structure of the sequence model.

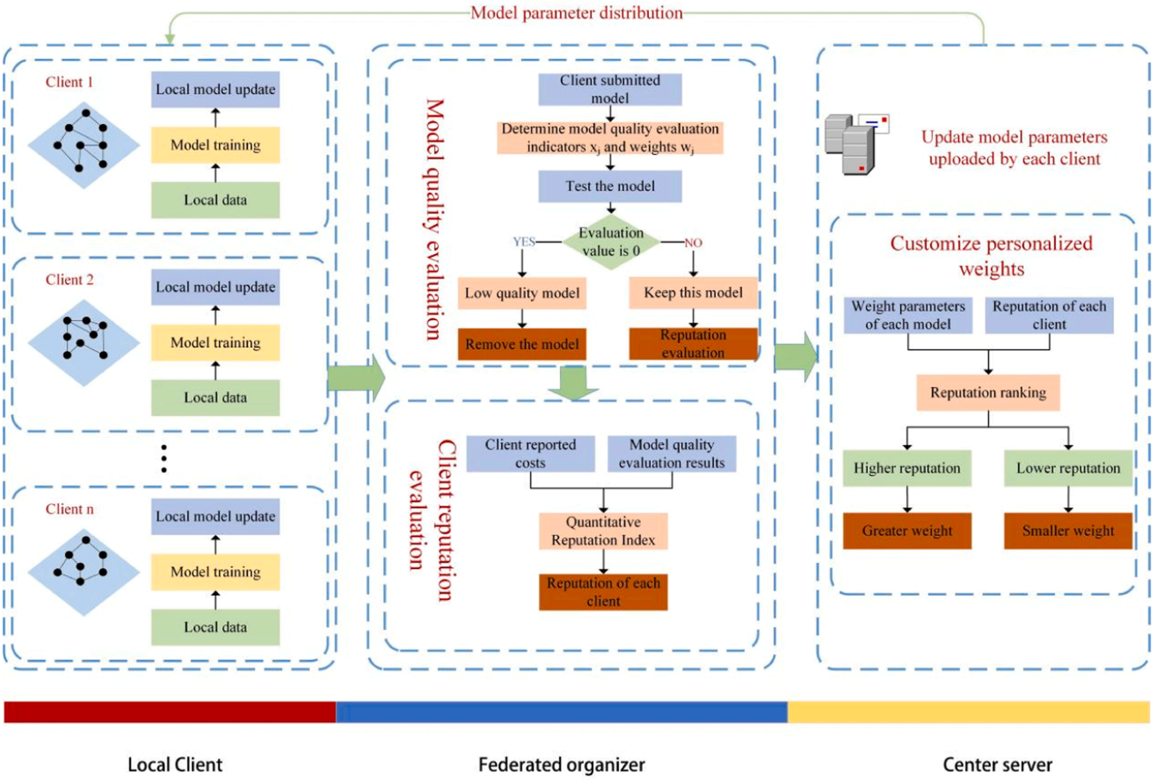

Fig. 3. The framework of personalized federated learning based on client reputation (PFLR).

Hadamard product.

3.2.2. Sequence-to-sequence decoding

Fig. 2 depicts a sequence-to-sequence model. In this paper, a sequence-to-sequence model is used for prediction, and an autoencoder is introduced to generate prediction results. The internal structure of the sequence-to-sequence model is shown in Fig. 2. The inside of the encoder is mainly composed of two kinds of network structural units, GCN and GRU. The traffic flow data is mapped to the hidden space representation R through the autoencoder, and R is input to the decoder for data reconstruction to obtain the final spatiotemporal prediction output.

Assume that the graph signal input to the model from the initial time t to the final time T is $(X^{(t-T+1)}, X^{(t-T+2)}, \cdots, X^{(t)})$. The graph signal is passed through a 2-layer neural network to obtain the graph signal calculated by graph convolution and then passed through the GRU unit to obtain the time dependence of the traffic flow sequence.

Algorithm 1
Model quality evaluation.

Input: Models trained and submitted by each client
Output: Model quality evaluation results
1: Determine model quality evaluation indicators
2: Determine the weight of evaluation indicators
3: Test the model submitted by the clients using the same test set and obtain y_{ij}
4: for $i = 1, 2, 3, \ldots, n$; $j = 1, 2, 3, \ldots, m$
5: Normalize the model quality evaluation values to obtain a sequence of evaluation values for model i under each indicator
6: $G_{ij} = \begin{cases} 0, & y_{ij} < x_j^{min} \\ \dfrac{y_{ij} - x_j^{min}}{x_j^{max} - x_j^{min}}, & x_j^{min} \le x_{ij} \le x_j^{max} \end{cases}$
7: Calculate the minimum evaluation value for each evaluation indicator
8: $G_i^{min} = \min\{G_j\}$
9: Determine whether the model is a low-quality model based on the minimum value of the indicator and screen it.
10: if $G_i^{min} = 0$
11: The model submitted by this client is considered a low-quality model and needs to be removed.
12: End if
13: Evaluate the quality of submitted data models
14: $Z_i^k = \sqrt{\sum_{j=1}^{m} (\omega_j \times G_{ij})^2}$
15: End for

$$Y_t = f_{GRU}\left(h_{t-1}, X_*^{(t)}\right) \tag{7}$$

In the formula: Y_t is the output of the encoder, that is, the time dependence of the traffic flow sequence; $f_{GRU}()$ is the calculation process inside the GRU unit. The hidden space representation R is calculated through the encoder structure, and the spatial representation sequence is used as the site encoding vector to initialize the decoding GRU. In the decoder, the hidden state obtained at each time step is fed into the GRU unit, which in turn inputs it to the fully connected layer. The final output is then obtained, representing the prediction result for each step [40].

3.3. Personalized federated learning based on client reputation

The framework of the personalized federated learning method based on client reputation (PFLR) proposed in this paper is shown in Fig. 3. It is mainly divided into three parts, namely the local client, the federation organizer, and the central server. Among them, the local client is mainly responsible for the training of the local model and the receiving and uploading of model parameters. The federated organizer is an "intermediate organizer" with a certain computing power, and its main function is to perform quality screening and reputation calculation on the models uploaded by each client. The main function of the central server is to accept and sort the reputation of each client passed from the federation organizer, and customize the weight parameters of each client model after aggregation according to the level of reputation to realize client personalization. Finally, the model is sent to each local client for the next round of communication.

In federated learning, each participant (such as a government or institution) is called a client, which provides the data resources needed for federated learning. Each client can be regarded as an independent working group, which can manipulate local data completely autonomously to decide when to join the federated learning system and how much to contribute. Due to the unique architecture of federated learning, the withdrawal and joining of a certain client will not have a large impact on the final effect of the prediction model. This is of great significance to all parties involved, who can have full freedom. In this paper, the primary role of the client is to receive the model and weight parameters sent from the server and subsequently train the model with its own data. Upon the completion of each training round, the trained model is uploaded to the server.

As the core of federated learning, the server side is also called a “third party”, which is independent of each data owner and does not directly contact the data of each client. Its role is mainly responsible for initializing the prediction model parameters at the beginning of training and delivering the model to each client. In the training phase, it is responsible for aggregating the model parameters uploaded by each client and delivering them again. It is worth mentioning that the server plays an important role as the "central brain" during the entire federated learning training period, but it fully guarantees the data privacy of each data owner.

3.3.1. Model quality evaluation

This section evaluates the quality of models submitted by federated participants. When communication starts, the client will download the model from the server and use local data for training. The model parameters will be uploaded after the training is complete. Submit the trained model as $M = \{m_1, m_2, \ldots, m_n\}$.

Establish two metrics to assess the model’s quality: accuracy and recall: $\{x_1, x_2\}$. The value of model m_i on two indicators is y_{ij}.

The evaluation value sequence of the submitted model m_i under each indicator is $\{G_{i1}, G_{i2}\}$.

Since the above evaluation indicators are positively correlated with model quality, the indicators can be normalized using Eq. (8) and used as a basis to remove low-quality models.

Algorithm 2
Reputation evaluation.

Require: E_i^k: Client A_i reputation after submitting the model for the m-th time
Require: e_i^k: Input parameter
Require: H_i^k: Client A_i false cost report for the *k-th* submission of the model
Require: $\overline{H^k}$: The average cost falsely reported by client A_i for the k-th submission of the model
Require: S_i^k: Client A_i reported the cost for the k-th time
Require: Z_i^k: The true cost of client A_i for the kth time
Input: The cost and model quality evaluation results reported by clients
Output: Client Reputation
1: Evaluate the client's reputation based on the cost reported by the client and the model quality evaluation results
2: For $k = 1, 2, 3 \ldots; i = 1, 2, \ldots, n$
3: Calculate the cost false report quantity and overall average false report quantity submitted by client A_i for the *k-th* time
4: $H_i^k = S_i^k - Z_i^k$
5: $\overline{H^k} = \frac{\sum_{i=1}^n H_i^k}{n}$
6: Quantitative Reputation Index
7: $e_i^k = \begin{cases} 1, & H_i^k = 0 \\ \frac{1}{1+\frac{H_i^k}{\overline{H^k}}}, & H_i^k \neq 0 \end{cases}$
8: Calculate the credibility of client A_i after m submissions
9: $E_i^k = \frac{\sum_{k=1}^m e_i^k}{m}$
10: End for

$$G_{ij} = \begin{cases} 0, & y_{ij} < x_j^{min} \\ \frac{y_{ij} - x_j^{min}}{x_j^{max} - x_j^{min}}, & x_j^{min} \leq x_{ij} \leq x_j^{max} \end{cases} \tag{8}$$

$x_j^{min}$and x_j^{max} B represent the lower and upper limits of the evaluation indicators.

Calculate the minimum estimated value of model m_i: $G_i^{min} = \min\{G_{ij}\}$.

If $G_i^{min} = 0$, the model submitted by the client is regarded as a low-quality model and needs to be removed.

To meet the performance requirements of different models, weight ω_j is introduced to make the results of model evaluation quality more in line with the actual situation. Use Eq. (9) to evaluate the model quality of the k-th federated participant for the subsequent client reputation assessment.

$$Z_i^k = \sqrt{\sum_{j=1}^m \left(\omega_j \times G_{ij}\right)^2} \tag{9}$$

The specific algorithm of model quality evaluation is shown in Algorithm 1.The flowchart is shown in Fig. A1 of the appendix.

3.3.2. Client reputation evaluation

In federated learning, each client will pursue the maximization of individual interests, so there will be false reporting of training costs, that is, the reported cost does not match the quality of the submitted model. This will lead to the damage of collective interests and even the failure of cooperation. Therefore, after evaluating the quality of the model submitted by the client, it is necessary to evaluate the reputation of the phenomenon of false cost reporting.

In general, the training cost of the client in federated learning is directly proportional to the quality of the model it submits, so this paper sets its true training cost and model quality to be numerically equal. In federated learning, the more the cost reported by the client exceeds the real cost, the lower its reputation is, and vice versa.

Suppose the reputation of federated participant A_i after submitting the model for the *m-th* time is:$E_i^k = \frac{\sum_{k=1}^m e_i^k}{m}$.

$$e_i^k = \begin{cases} 1, & H_i^k = 0 \\ \frac{1}{1+\frac{H_i^k}{\overline{H^k}}}, & H_i^k \neq 0 \end{cases} \tag{10}$$

H_i^k is the false cost amount of participant A_i submitting the model for the *k-th* time, and $\overline{H^k}$ is the average cost false state amount of the model submitted by participant A_i for the *k-th* time.

$$H_i^k = S_i^k - Z_i^k \tag{11}$$

Algorithm 3
Personalized federated algorithm based on client reputation.

Require: W: Initial model weight parameters
Require: W_{random}: Random parameters
Require: D_{data}: Local traffic flow data
Input: Execution rounds T, client A participating in training, model quality, client reputation
Output: Personalized prediction model
1: initialize the server
2: W is obtained by pre-training of the client model.
3: Server-side: $W_{server} \leftarrow W_{random}$
4: initialize A
5: set the communication address between the local client and the central server
6: start communication
7: all the A_i $(i = 1,2, \ldots, n)$ participating in the training, $G_i \in G$
8: For $t = 1, 2, \ldots, T$
9: For $A_i \in A$, do:
10: A_i downloads W from the server
11: A_i update $W_i = W$
12: A_i uses D_{data} to train the GCN-GRU model and update W_i
13: Upload W_i to the Federated organizer
14: Federated organizers use Algorithm 2 to evaluate the reputation of Client A_i
15: Federal organizers send evaluation results to the server
16: end for
17: The server sorts the reputation of each client
18: Distribute the impact factors of each client proportionally based on the sorting results $\mathcal{B}i$
19: Updating global model parameters, the larger the impact factor $\mathcal{B}i$, the greater the impact of clients on the global model
20: End for

$$\overline{H^k} = \frac{\sum_{i=1}^{n} H_i^k}{n} \tag{12}$$

S_i^k is the reporting cost of participant A_i for the kth time. Z_i^k is the real cost of participant A_i at the kth time.

It can be known from the above formula: when $H_i^k = 0$, $E_i^k = \frac{1}{m}$.

When $H_i^k \neq 0$, the formula is derived as follows:

$$E_i^k = \frac{1}{m} \cdot \sum_{k=1}^{m} \frac{\overline{H^k}}{\overline{H^k} + H_i^k} = \frac{1}{m} \cdot \sum_{k=1}^{m} \frac{\frac{1}{n} \cdot \sum_{i=1}^{n} H_i^k}{\frac{1}{n} \cdot \sum_{i=1}^{n} H_i^k + H_i^k} = \frac{1}{m} \cdot \sum_{k=1}^{m} \frac{\sum_{i=1}^{n} H_i^k}{\sum_{i=1}^{n} H_i^k + n \cdot H_i^k} = \frac{1}{m} \cdot \sum_{k=1}^{m} \frac{\sum_{i=1}^{n} \left(S_i^k - Z_i^k\right)}{\sum_{i=1}^{n} \left(S_i^k - Z_i^k\right) + n \cdot \left(S_i^k - Z_i^k\right)}$$
$$= \frac{1}{m} \cdot \sum_{k=1}^{m} \frac{\sum_{i=1}^{n} \left(S_i^k - \sqrt{\sum_{j=1}^{m} \left(\omega_j \times G_{ij}\right)^2}\right)}{\sum_{i=1}^{n} \left(S_i^k - \sqrt{\sum_{j=1}^{m} \left(\omega_j \times G_{ij}\right)^2}\right) + n \cdot \left(S_i^k - \sqrt{\sum_{j=1}^{m} \left(\omega_j \times G_{ij}\right)^2}\right)} \tag{13}$$

The specific client reputation evaluation algorithm is shown in Algorithm 2. The flowchart is shown in Fig. A2 of the appendix.

3.3.3. Federated personalization algorithm based on client reputation

In the model aggregation stage of federated communication, the server formulates personalized weights for each local model by calculating the model quality and reputation provided by each client during aggregation. Based on this, the global model is obtained. First, the server sorts the reputation of each client sent by the federated organizer. Then we introduce the concept of an impact factor $\mathcal{B}i$. The server assigns the impact factor $\mathcal{B}i$ to each client according to the ranking results of reputation and updates the global model parameters. A client with a larger impact factor $\mathcal{B}i$ has a greater influence on the global model. The final global model will favor clients with high reputations.

The server initially collects the reputation ratings of each client and sorts these ratings. Assuming there are N clients, the server obtains the reputation rating E for each client, where $i = 1,2,\ldots,$N. The impact factor $\mathcal{B}i$ is a weight allocated proportionally based on the client's reputation rating. The calculation of the influence factor can be carried out through the following steps:

Calculate total reputation:$E_{total} = \sum_{j=1}^{N} E_j$

Calculate the impact factor for each client:$\mathscr{B}_{\mathrm{i}} = \frac{E_i}{E_{total}}$

In the model aggregation stage, the server uses the impact factor $\mathscr{B}_{\mathrm{i}}$ of each client to weight and average the local model parameters provided by each client. Assuming that each client i submits a local model parameter of θ_i, the update of the global model parameter θ global can be expressed as:$\theta_{global} = \sum_{i=1}^{N} \mathscr{B}_{\mathrm{i}} \cdot \theta_i$

The specific algorithm is shown in Algorithm 3. The flowchart is shown in Fig. A3 of the appendix.

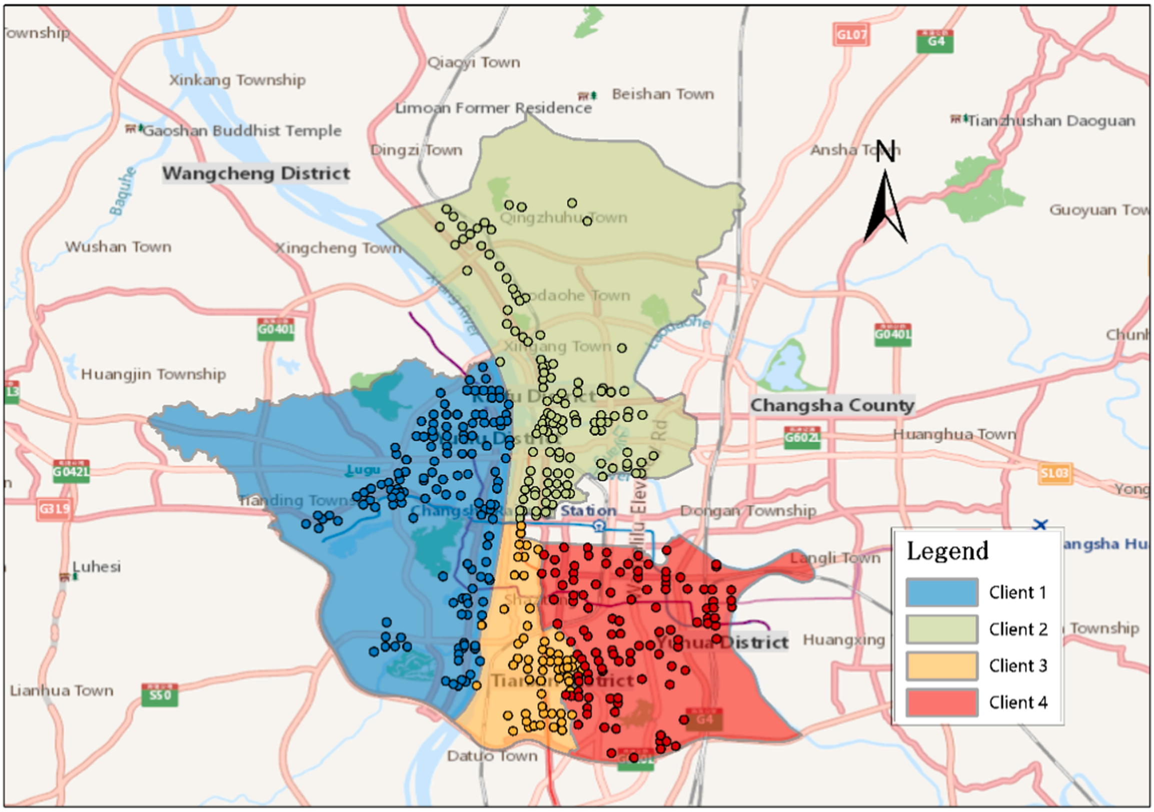

Fig. 4. Data source.

4. Data description

We selected LPR data collected in Changsha City as the data source in the experiment. LPR data includes a total of 8 fields, namely license plate number, license plate type, elapsed time, collection point number, collection point name, equipment number, direction number, and lane number. The Data were collected from September 1st to September 30th, 2019. We first performed data cleaning on the raw data to ensure its quality and completeness. Subsequently, we standardized the timestamps to ensure a consistent format, converting them into a standard time representation. After that, the timestamps were converted into 5-minute intervals, and the data was aggregated accordingly. Finally, we counted the records within each 5-minute interval to calculate the traffic flow for those intervals.

After data processing, we obtained traffic volume data accumulated in five minutes. Fig. 4 shows the locations of LPR detectors in four administrative districts of Changsha City: Yuelu District, Kaifu District, Tianxin District, and Yuhua District. In the experiment, we assume the data source collected in different districts were not shared with each other. Then, we set each zone as an independent client, namely client 1, client 2, client 3, and client 4. Client 1 contains 100 LPR detectors, Client 2 contains 85 LPR detectors, Client 3 contains 110 LPR detectors, and Client 4 contains 78 LPR detectors.

5. Experimental analysis

5.1. Experimental setup and evaluation function

All experiments were conducted on the AMD Ryzen 7 5800H processor, NVIDIA GeForce RTX 3070 graphics card, 16GB memory, and Python 3.9. We utilized LPR data for the training and prediction of all comparative methods, maintaining consistent hyper-parameter settings across the board. Specifically, we set the learning rate to 0.001, the batch size to 64, and conducted training over 100 epochs. We set the number of communication rounds for federated learning to 50.

RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) are frequently employed evaluation metrics in traffic flow prediction, serving to quantify the magnitude of errors between the predicted values of the model and the actual values. The calculation formulas are as follows:

Table 1
Performance of the different methods. ↓ better results for smaller values.

Client1	1-step		2-step		3-step		4-step	
	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓
LSTM	7.56	4.08	8.11	4.29	8.63	4.35	9.69	5.28
GRU	7.32	3.57	7.83	4.16	8.15	4.31	8.55	4.39
GCN	6.51	3.27	6.83	3.39	7.05	3.48	7.62	4.06
GCN-GRU	5.79	3.05	5.95	3.09	6.27	3.18	6.68	3.29
PFGCN-GRU	6.02	3.11	6.08	3.11	5.89	3.05	6.19	3.17
Client 2	1-step		2-step		3-step		4-step	
	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓
LSTM	8.71	6.05	9.72	6.88	11.25	8.01	12.09	8.37
GRU	8.58	6.01	9.49	6.62	10.22	7.13	12.23	8.53
GCN	6.91	3.83	8.03	4.72	9.62	6.79	11.87	7.76
GCN-GRU	6.20	3.57	6.81	3.77	7.59	4.43	9.78	6.75
PFGCN-GRU	6.39	3.61	7.12	4.06	7.55	4.43	10.25	7.11
Client 3	1-step		2-step		3-step		4-step	
	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓
LSTM	10.77	7.82	13.15	9.47	16.32	11.59	20.19	14.98
GRU	10.56	7.76	12.89	8.75	16.01	11.43	19.23	14.62
GCN	8.16	5.63	9.42	6.08	11.69	8.34	15.19	10.95
GCN-GRU	7.26	4.38	8.09	5.11	10.66	7.12	13.21	8.92
PFGCN-GRU	7.56	4.99	7.96	5.09	9.85	6.28	13.81	9.73
Client 4	1-step		2-step		3-step		4-step	
	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓
LSTM	6.21	3.54	6.92	3.69	8.11	4.27	9.55	5.05
GRU	6.02	3.47	6.56	3.61	7.63	3.96	8.89	4.71
GCN	6.13	3.27	6.41	3.40	7.12	3.71	9.03	4.83
GCN-GRU	5.15	2.79	5.43	2.93	6.39	3.56	7.16	3.75
PFGCN-GRU	6.19	3.29	6.35	3.33	6.99	3.64	7.55	3.91

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}}\left(\widehat{\mathrm{Q}}_{\mathrm{i}} - \mathrm{Q}_{\mathrm{i}}\right)^2} \tag{14}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{Q}_i - Q_i\right| \tag{15}$$

Where n represents the number of samples in the dataset, Q_i represents the actual value of the i *th* sample, $\widehat{Q}_i$ represents the predicted value of the i *th* sample.

5.2. Performance of the different methods

We selected several commonly used traffic flow prediction models as baseline models for comparative analysis with the proposed PFGCN-GRU model in this paper.

LSTM [4]: Long Short-Term Memory (LSTM) is a specialized type of recurrent neural network (RNN) that introduces gating mechanisms—including input, forget, and output gates—to regulate the flow of information, thereby effectively capturing long-term dependencies in time series data. In traffic flow prediction, LSTM can model the temporal dynamics of historical traffic data.

GRU [12]: The Gated Recurrent Unit (GRU) is a simplified version of LSTM that reduces the number of parameters by merging the input and forget gates while maintaining the capability to capture long-term dependencies. Due to its relatively simple architecture and higher computational efficiency, GRU is more commonly used in resource-constrained environments or scenarios requiring rapid training. It can also be employed to analyze and predict temporal patterns in traffic flow.

GCN [21]: Graph Convolutional Networks (GCN) are deep learning models designed to handle graph-structured data. By performing convolution operations on the graph, GCN extracts features from nodes that capture relationships between nodes and their neighbors. In traffic flow prediction, GCN can model the topological structure of transportation networks, such as the connections between roads, thereby enhancing the understanding and prediction of the spatial distribution of traffic flow.

GCN-GRU [15]: The GCN-GRU model combines the strengths of both GCN and GRU. In this hybrid model, the GCN layer first processes graph-structured data to extract spatial features, which are then fed into the GRU layer to capture dynamic changes over time. This integration allows the model to comprehend the spatial structure of the traffic network while also capturing the temporal variations in traffic flow. GCN-GRU can provide more comprehensive and accurate predictions in traffic flow prediction.

Fig. 5. Performance of two methods in Mid-to-Long-Term traffic flow prediction.

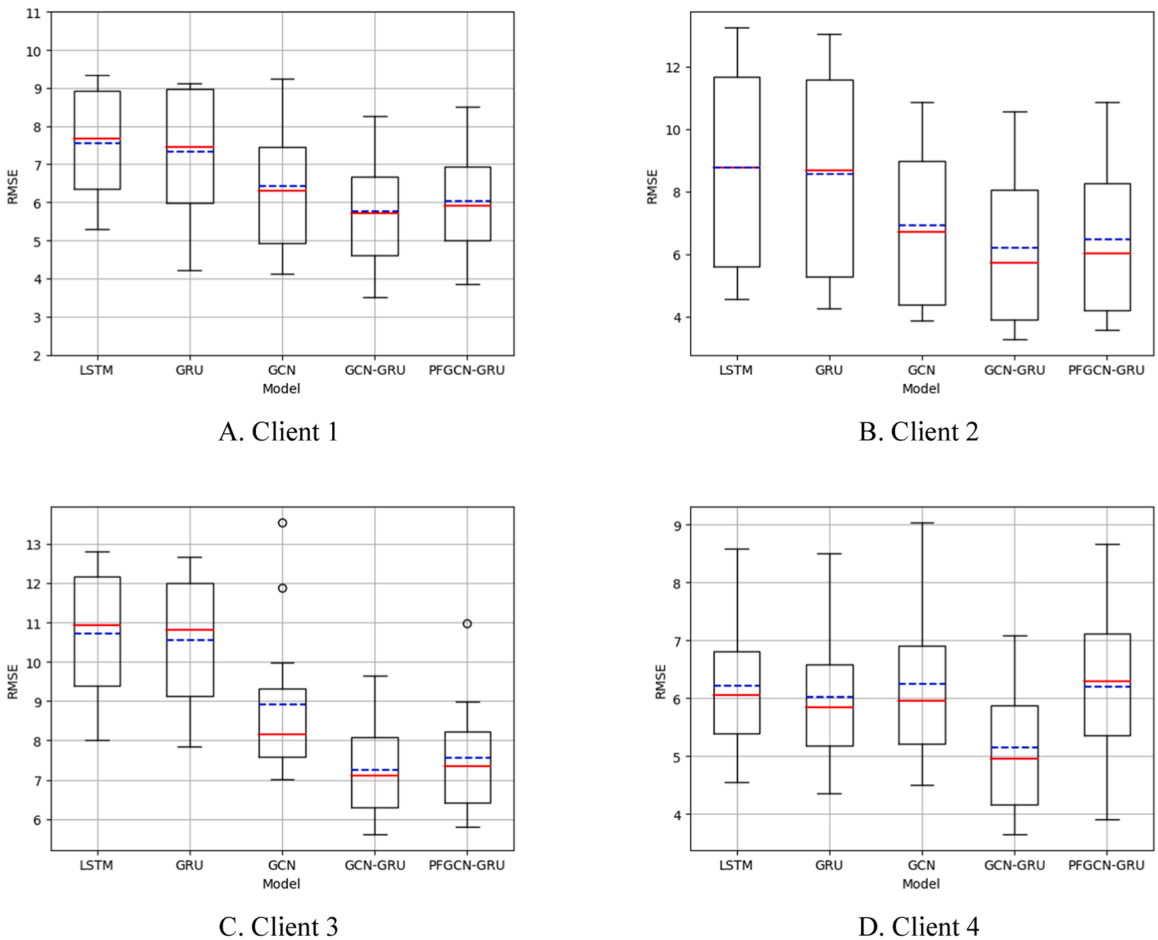

Fig. 6. Box plots of PFGCN-GRU and each baseline model on four clients.

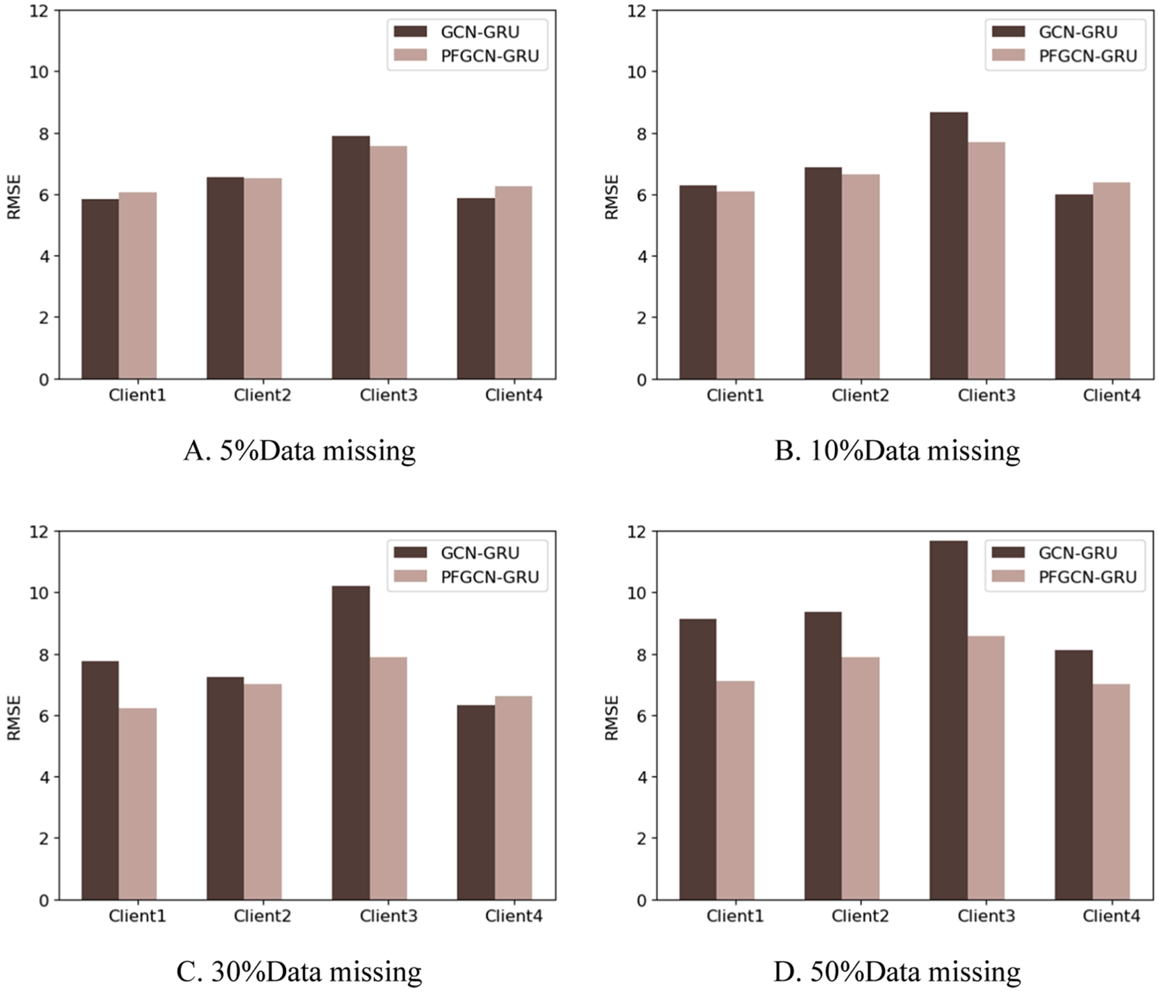

Fig. 7. Comparison of the prediction performance of PFGCN-GRU and GCN-GRU when the amount of data is missing.

Table 1 shows the prediction performance of the personalized federated learning short-term traffic flow prediction model PFGCN-GRU based on client reputation, the local model GCN-GRU and other machine learning models on the Changsha LPR data set. Set five minutes as a time step.

As can be seen from Table 1, in most cases, the RMSE values and MAE values of the PFGCN-GRU model and the GCN-BIGRU model are relatively close. This indicates that the PFGCN-GRU model, which incorporates personalized federated learning, can achieve good prediction performance while protecting the data privacy of each client in short-term traffic flow prediction experiments. At a 5-minute time step, the PFGCN-GRU model's prediction performance on some clients is slightly inferior to that of the centrally trained GCN-GRU model. However, as the time step increases, the PFGCN-GRU model consistently maintains a good level of prediction performance without significant fluctuations. In contrast, the GCN-GRU model's prediction performance declines more noticeably with increasing time steps. This suggests that the personalized federated learning short-term traffic flow prediction model, PFGCN-GRU, based on client reputation, can achieve prediction accuracy that is on par with, or even surpasses, that of ordinary machine learning models in multi-step predictions. Moreover, as the time step increases, the prediction stability of PFGCN-GRU is superior to that of centrally trained machine learning models. We hypothesize that this is due to the distributed training structure of PFGCN-GRU, which effectively leverages the data characteristics from different clients.

To explore whether the personalized federated learning short-term traffic flow prediction model PFGCN-GRU, based on client reputation, can achieve better prediction results and stability with increasing time steps, we designed a mid-to-long-term traffic flow prediction experiment. We set up comparative experiments for PFGCN-GRU and centrally trained GCN-GRU models at 15-minute, 30-minute, 45-minute, and 60-minute intervals. The experimental results are shown in Fig. 5.

As illustrated in Fig. 5, with increasing time steps, the personalized federated learning short-term traffic flow prediction model PFGCN-GRU, based on client reputation, demonstrates superior and more stable prediction performance compared to the centrally trained GCN-GRU model. This enhanced performance and stability can be attributed to PFGCN-GRU's ability to personalize model adjustments based on each client's historical data and environmental changes. This personalization allows the model to better adapt to

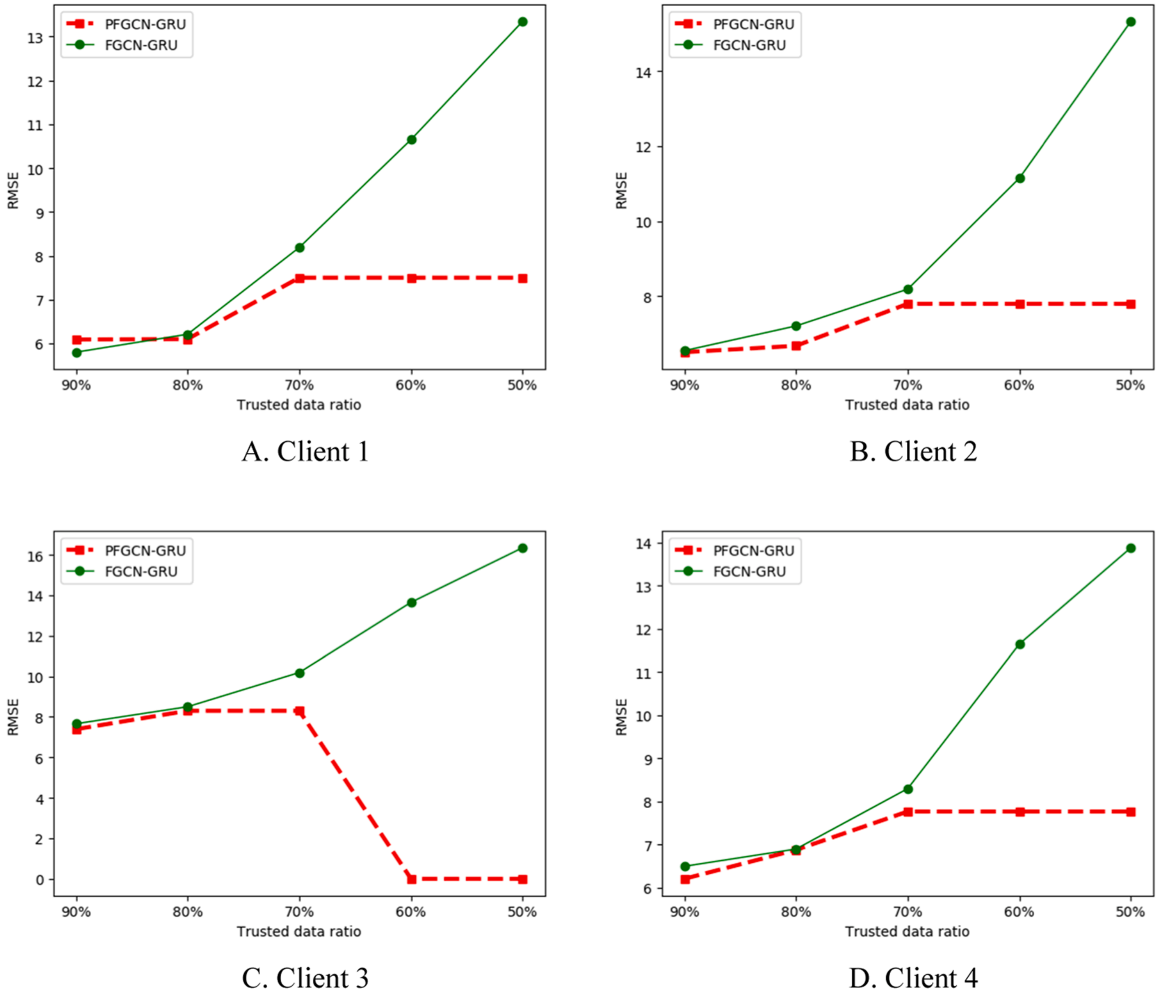

Fig. 8. The prediction performance of PFGCN-GRU on each client when client 3 provides untrusted data.

specific users' traffic patterns and behaviors, thereby improving prediction accuracy. Federated learning enables multiple clients to train the model simultaneously, leveraging their local data for parallel computation, thus enhancing training efficiency. This flexibility allows the model to quickly adapt to new data changes, especially with increasing time steps, facilitating timely updates to the model parameters. As the time step increases, the dynamic changes in traffic flow predictions may become more pronounced. The PFGCN-GRU model can rapidly adapt to these changes through real-time client feedback, whereas the GCN-GRU model may experience delays in dynamic adjustments.

With the increase in time steps, the personalized federated learning short-term traffic flow prediction model PFGCN-GRU exhibits superior performance and stability in terms of data adaptability, personalized training, privacy protection, and dynamic adjustments compared to the centrally trained GCN-GRU model. These combined factors enable PFGCN-GRU to more effectively handle complex and variable traffic flow prediction tasks in practical applications.

Fig. 6 is the 5 min prediction error distribution of PFGCN-GRU, GCN-GRU and other baseline models on four clients. In Fig. 6, the blue dashed line of the box plot is the mean value, and the red solid line is the median value. As can be seen from Fig. 6, the prediction performance of the centralized training machine learning model GCN-GRU selected in this sutdy is significantly better than other baseline models, and it has less abnormal data and a relatively even error distribution. Therefore, as the initial model of federated learning, the effect is better than other baseline models. As a global prediction model trained under a reputation-based personalized federated learning framework, PFGCN-GRU's prediction performance on the four clients is basically the same as GCN-GRU, and the prediction stability is good and the overall error is small. This shows that the prediction model obtained through the federated learning collaborative training framework can meet the needs of high-quality traffic flow prediction.

5.3. Prediction stability experiment

To further validate the stability of the proposed model, the experiments with missing data is conducted. We set all client data to missing 5 %, 10 %, 30 %, and 50 %, respectively, to explore the prediction performance of GCN-GRU and PFGCN-GRU in the context of missing data. In this paper, the setting of missing values is based on random selection to ensure minimal human intervention, thereby more closely reflecting real-world scenarios. The results are shown in Fig. 7.

Fig. 7 shows the comparison of the prediction performance between the PFGCN-GRU model trained under the personalized federated learning framework based on client reputation and the centralized training model GCN-GRU when the data volume of all clients is missing by 5 %, 10 %, 30 %, and 50 %. It can be seen that when the missing rate of each client's data volume is 5 %, the prediction performance of the centralized training model GCN-GRU is good. We speculate that this is because the amount of missing data is small, and the overall data amount can still meet the needs of centralized training. When the data missing rate reaches 10 %, the prediction performance of the centrally trained model GCN-GRU starts to deteriorate. In contrast, the prediction performance of the PFGCN-GRU model, trained within the personalized federated learning framework based on client reputation, surpasses that of GCN-GRU across all four clients. When the data missing rate reaches 30 %, the prediction performance of GCN-GRU drops significantly, while the prediction performance of PFGCN-GRU on the four clients is basically the same as the prediction performance when there is complete data. Finally, when the missing rate of data volume reaches 50 %, the prediction performance of GCN-GRU is significantly different from that when complete data is available.

From Fig. 7, it can be observed that the prediction performance of the centralized training model experiences a significant decline in the presence of missing data. Moreover, when the proportion of missing values exceeds half, its final prediction performance undergoes a catastrophic drop, rendering it incapable of fulfilling high-quality traffic flow prediction tasks. Conversely, when the data missing rate is within 30 %, the prediction performance of the PFGCN-GRU model trained under the personalized federated learning framework based on client reputation exhibits minimal fluctuation. Even when the data missing rate reaches 50 % across all four clients, PFGCN-GRU maintains a relatively stable and satisfactory prediction performance. This underscores that federated learning not only safeguards the privacy of each participant's data but also, in scenarios of data scarcity or low data quality, leverages collaborative training across clients within the federated learning framework. This significantly enhances the prediction stability of the global model and, to a certain extent, improves the model's prediction capabilities.

5.4. Impact of reputation-based client personalization mechanisms

To investigate the influence of the reputation-based client personalization mechanism on the ultimate global prediction model, we conducted a controlled experiment. We set client 3 as a "dishonest client", which provides 90 %, 80 %, 70 %, 60 %, and 50 % of trusted data respectively, but all of them lied about providing 100 % true data. For instance, when Client 3 asserted that it had provided 100 % authentic data, in reality, only 70 % of the data was trustworthy. To simulate this dishonest behavior, we randomly selected the remaining 30 % of the data and scrambled it. This scrambling operation rendered these data devoid of their original temporal characteristics, thereby making them untrustworthy. Consequently, we employed this method to simulate the behavior of Client 3 supplying inaccurate data under different scenarios. In this scenario, we executed a controlled experiment comparing the model FGCN-GRU based on conventional federated learning with the model PFGCN-GRU based on personalized federated learning incorporating client reputation. The outcomes are depicted in Fig. 8, where A, B, C, and D represent the prediction performances of the two global prediction models on client 1, client 2, client 3, and client 4, respectively.

It can be seen from Fig. 8 that when client 3 provides 90 % trusted data, the prediction performances of PFGCN-GRU, FGCN-GRU and GCN-GRU are basically close to each other. We speculate that this is because trusted data occupies the vast majority and the data is relatively complete, so it basically has a limited impact on the two prediction methods. When client 3 provides less than 70 % of the trusted data, it can be clearly seen that the prediction performance of PFGCN-GRU and FGCN-GRU is better than the GCN-GRU model. And with the reduction of trusted data, this phenomenon becomes more obvious on FGCN-GRU. As the number of trusted data provided by "dishonest clients" continues to decrease, personalization mechanisms based on client reputation come into play. It can be clearly seen from Figs. 8(A), 8(B) and 8(D) that when the proportion of trusted data provided by client 3 drops to 70 %. The prediction performance of the global model on client 1, client 2, and client 4 basically does not change. This is because at this time, the personalized mechanism of client reputation determines that "dishonest clients" have appeared in the federated learning collaborative training framework, and evaluates the credibility of all clients participating in collaborative training. Finally, client 3 was identified as a "dishonest client" and removed from the federated learning collaborative training framework. This shows that the federated learning personalization mechanism based on client reputation can quickly determine "dishonest clients" and can respond quickly to remove them from the federated learning collaborative training framework.

Experimental results show that the federated learning personalization mechanism based on client credibility proposed in this article can quickly and accurately identify "dishonest clients". This eliminates the situation where some clients only provide a small amount of data but still enjoy the same treatment as clients that provide a large amount of data. The interests of other clients are safeguarded to a greater extent, making the federated learning framework reach a relatively fair state.

5.5. Comparative experiment

In this subsection, we further compare the personalized federated learning method proposed in this paper, named PFGCN-GRU, based on client reputation, with two existing popular personalized federated learning methods. The purpose is to demonstrate the effectiveness of the personalized approach presented in this study. These two personalized methods of federated learning are FT FedAvg [41] and pFedme [42].

FT FedAvg: Adding a local fine-tuning method for the client part based on the FedAvg algorithm. Perform additional local training

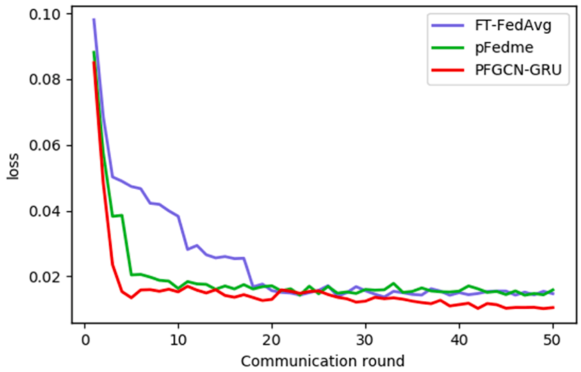

Fig. 9. The average loss of three methods across four clients.

Fig. 10. The average MAE of three methods across four clients.

on the aggregated global model using client local data to adapt to the specific data distribution of the client.

PFedme: Personalize federated learning by regularizing the loss function of the client. The distance between the global model parameters and the local model parameters of the regularization term controls the degree to which the client model deviates from the global model. We set the communication rounds to 50 for the experiments comparing these three personalized methods. Fig. 9 displays the average loss values across all clients for the three personalized methods. Fig. 10 presents the prediction accuracy of individual clients under different personalized methods, measured in terms of MAE.

Observing Fig. 9, it is evident that the method proposed in this paper demonstrates superior convergence speed in the initial stages compared to the other two personalized methods. Furthermore, it demonstrates overall better convergence rates. Fig. 10 illustrates that in terms of prediction performance across various clients, the proposed PFGCN-GRU outperforms both FT-FedAvg and pFedme. FT-FedAvg involves simple fine-tuning on the basis of FedAvg, and while it improves the prediction performance of the global model across clients compared to ordinary federated learning models, the improvement is relatively modest. While the prediction performance of pFedme is superior to that of FT-FedAvg, its convergence speed is relatively slow, indicating that it necessitates a greater number of communication rounds to attain satisfactory performance. In summary, the personalized federated learning method PFGCN-GRU proposed in this paper, based on client reputation, demonstrates favorable results in both convergence speed and prediction performance. It is capable of achieving satisfactory prediction outcomes with reduced communication rounds.

6. Conclusion

This paper introduces a short-term traffic flow prediction method that integrates personalized federated learning with GCN-GRU, allowing for distributed model training among data holders without sharing raw data. Recognizing the varying contributions of clients with different data volumes to the global model, we propose a personalized federated learning approach based on model quality and client reputation. We filter the quality of models uploaded by each client, calculate their reputation, and then rank them according to their reputation scores. Customized weight parameters for each client's model are adjusted based on their reputation to achieve client-specific personalization.

However, there are limitations to this study. The computational capabilities of edge devices in federated learning were not considered, and in practical applications, the computational capacities of clients vary significantly. The impact of the initial model on the overall prediction framework was not extensively discussed, which undoubtedly affects the final prediction results. Additionally, external factors affecting traffic flow, such as severe weather and traffic accidents, were not taken into account. In light of these shortcomings, we suggest the following future research directions:

(1) Employing different initial models for clients, as various machine learning models exhibit different performances in traffic flow prediction. We believe that selecting better initial models can enhance the communication efficiency and accuracy of federated learning, thereby reducing computational burdens.
(2) Considering the influence of external factors during model input and exploring the extent to which these factors impact the final prediction accuracy.

Declaration of competing interest

No conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

Data availability

The authors do not have permission to share data.

Acknowledgments

This research was funded in part by the Key R&D Program of Hunan Province (No. 2023GK2014), National Natural Science Foundation of China (No.52172310).

Appendix

Fig. A1, Fig. A2, Fig. A3

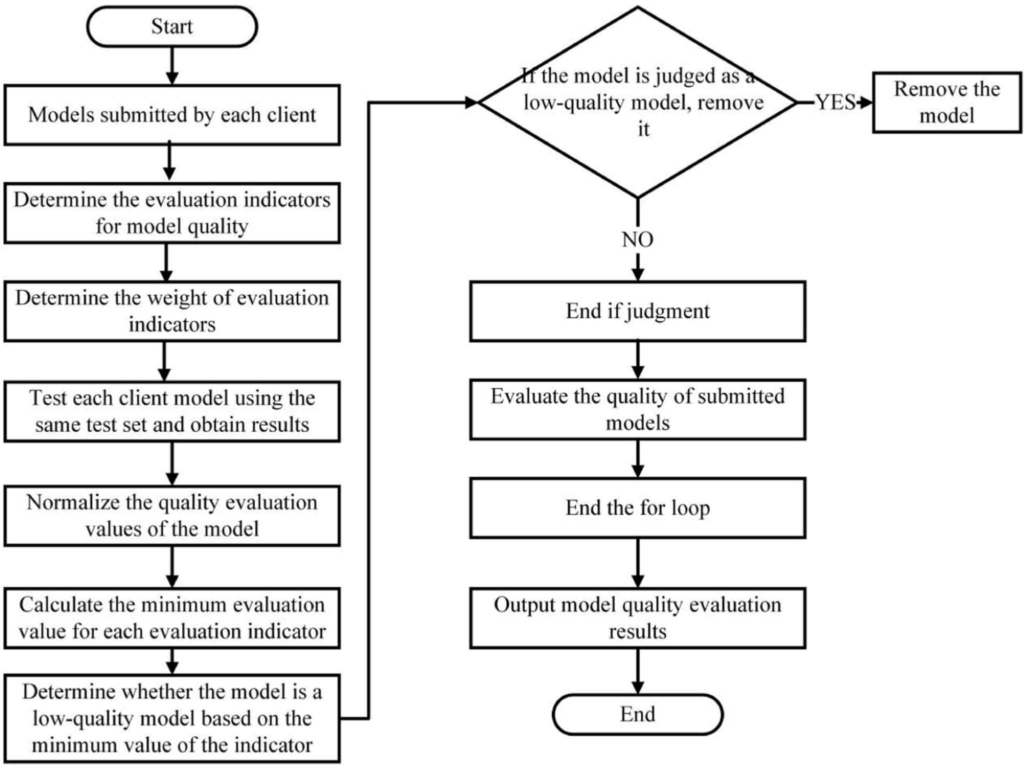

Fig. A1. Flowchart of Algorithm 1.

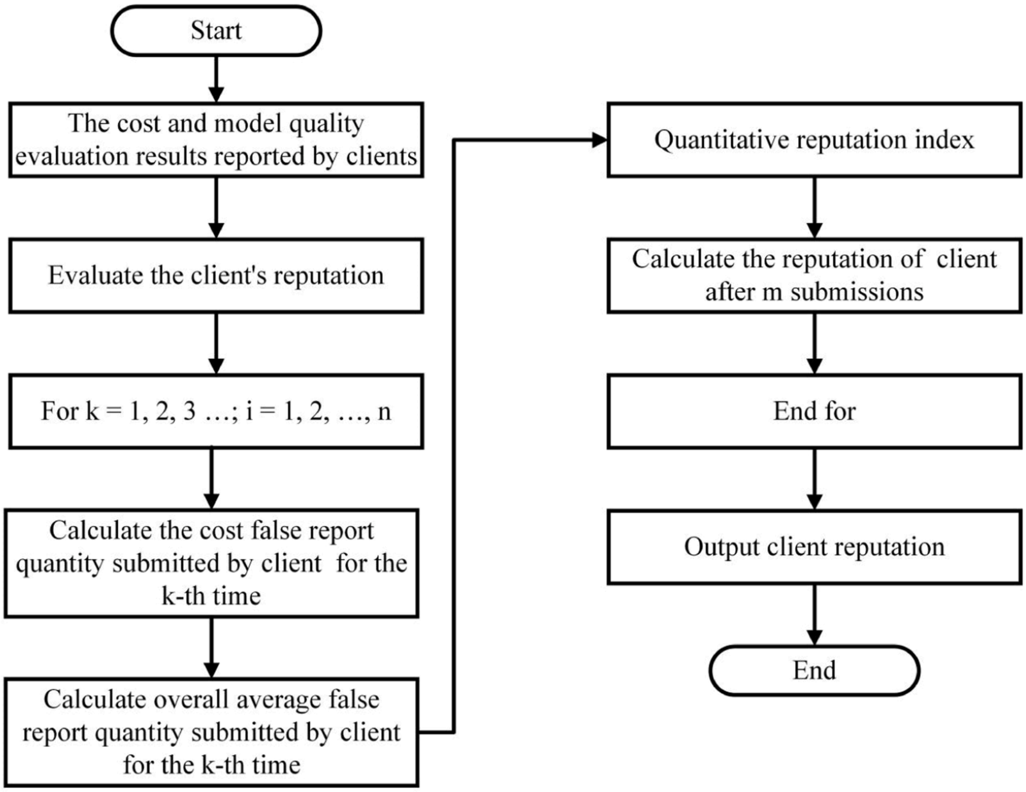

Fig. A2. Flowchart of Algorithm 2.

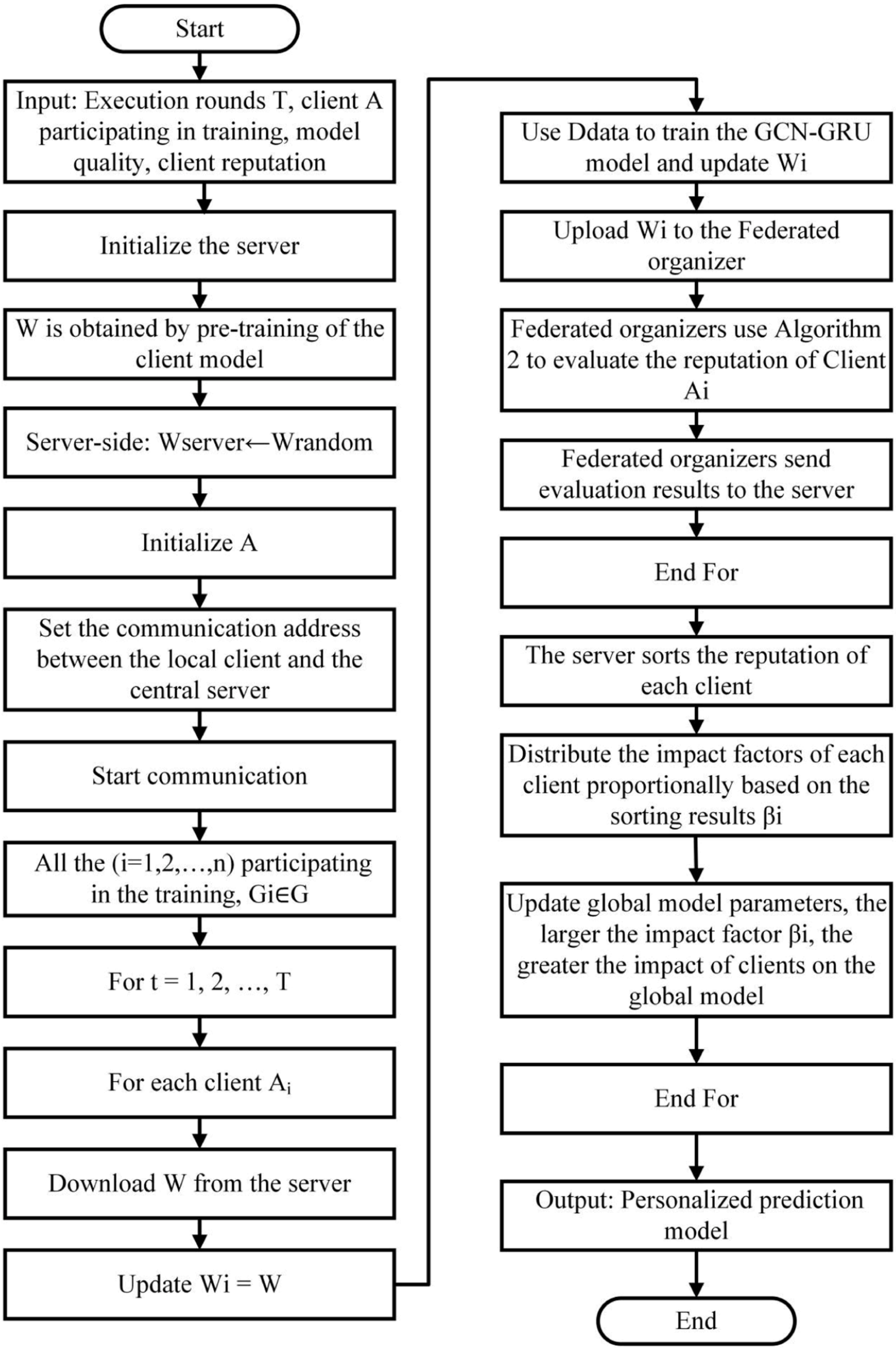

Fig. A3. Flowchart of Algorithm 3.

Table A1
Comparison of computational complexity.

Method	Computational complexity	advantage
LSTM	$O(T \cdot (F' + H))$	Suitable for capturing time series dependencies
GRU	$O(T \cdot (F' + H))O(T \cdot (F' + H))$	Short term dependence has a good effect
GCN	$O(L \cdot (E + N) \cdot F)$	Effectively capturing spatial structures
GCN-GRU	$O(L \cdot (E + N) \cdot F) + O(T \cdot (F' + H))$	Balancing spatial and temporal information
PFGCN-GRU	$C \cdot (O(L \cdot (E + N) \cdot F) + O(T \cdot (F' + H))) + O(C \cdot P)$	Strong generalization ability and privacy protection capability

Table A1 compares the computational complexity of several prediction methods. Input feature dimension: F', hidden state dimension: H, number of time steps: T, number of nodes: N, number of edges: E, feature dimension: F, number of layers: L, number of clients: C, number of model parameters: P.

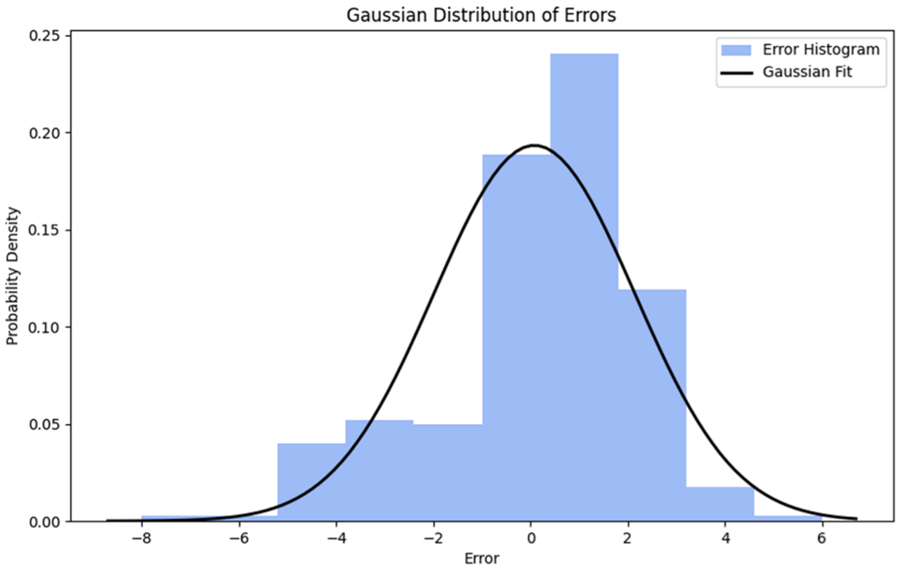

Fig. A4. Gaussian distribution of PFGCN-GRU model predicted values and true values.

Fig. A4 shows the error distribution between the 5-minute time step predicted values and the true values of the PFGCN-GRU model. From Fig. A4, we can observe that the histogram of errors shows that the errors are predominantly clustered around zero and are relatively symmetrical, indicating that the model's predictions are quite accurate and there is no systematic overestimation or underestimation. The mean of the Gaussian distribution curve is also close to zero, suggesting that the model's average error is small, and the prediction accuracy is high on an average level. Moreover, the small standard deviation indicates that the range of error fluctuations is narrow, signifying that the model's predictions are stable.

Table A2
Detailed information of LPR data.

Field name	Meaning	Data type	Example
LPR	License plate recognition (encrypted))	str	02b607cfa745a32141b07ab09cafbbfa
TLR	Type of license plate	int	41
ET	elapsed time	str	20,220,801,210,449
CPN	Collection point number	str	690,051,051,000
CPNA	Collection point name	str	Intersection of Fenglin Road and Qilong Road
EN	Equipment number	str	430,106,000,000,012,000
DN	Direction number	int	1
LN	Lane number	int	2

Table A2 shows the detailed information of LPR data. Among them, LPR is the license plate recognition (encrypted), TLP is the type of license plate, ET is the elapsed time, CPN is the collection point number, CPNA is the collection point name, EN is the equipment number, DN is the direction number (1: east 2: west 3: south 4: north), and LN is the lane number.

References

[1] Abadi A, Rajabioun T, Ioannou PA. Traffic flow prediction for road transportation networks with limited traffic data. IEEE Trans Intell Transp Syst April 2015;16 (2):653–62. https://doi.org/10.1109/TITS.2014.2337238.
[2] Hou Z, Li X. Repeatability and similarity of freeway traffic flow and long-term prediction under big data. IEEE Trans Intell Transp Syst June 2016;17(6): 1786–96. https://doi.org/10.1109/TITS.2015.2511156.
[3] Zheng C, Fan X, Wen C, Chen L, Wang C, Li J. DeepSTD: mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction. IEEE Trans Intell Transp Syst Sept. 2020;21(9):3744–55. https://doi.org/10.1109/TITS.2019.2932785.
[4] Mackenzie J, Roddick JF, Zito R. An evaluation of HTM and LSTM for short-term arterial traffic flow prediction. IEEE Trans Intell Transp Syst May 2019;20(5): 1847–57. https://doi.org/10.1109/TITS.2018.2843349.
[5] Chen C, Liu Z, Wan S, Luan J, Pei Q. Traffic flow prediction based on deep learning in internet of vehicles. IEEE Trans Intell Transp Syst June 2021;22(6): 3776–89. https://doi.org/10.1109/TITS.2020.3025856.
[6] Hu X, Li R, Wang L, Ning Y, Ota K. A data sharing scheme based on federated learning in IoV. IEEE Trans Veh Technol Sept. 2023;72(9):11644–56. https://doi.org/10.1109/TVT.2023.3266100.
[7] Dabiri Sina, Heaslip Kevin. Developing a Twitter-based traffic event detection model using deep learning architectures. Expert Syst Appl 2019;118:425–39. ISSN 0957-4174.
[8] Zhao J, Gao Y, Yang Z. Truck traffic speed prediction under non-recurrent congestion: based on optimized deep learning algorithms and GPS data. IEEE Access 2019;7:9116–27.
[9] Miglani A, Kumar N. Deep learning models for traffic flow prediction in autonomous vehicles: a review, solutions, and challenges. Vehicular Communications 2019. https://doi.org/10.1016/j.vehcom. ISSN 2214-2096.
[10] Cheng RJ, Ge HX, Wang JF. An extended continuum model accounting for the driver's timid and aggressive attributions. Phys Lett A 2017;381(15):1302–12.
[11] Jiang CT, Ge HX, Cheng RJ. Mean-field flow difference model with consideration of on-ramp and off-ramp. Physica A 2019;513:465–7.
[12] Sun P, Boukerche A, Tao Y. SSGRU: a novel hybrid stacked GRU-based traffic volume prediction approach in a road network. Comput Commun 2020;160: 502–11.
[13] Polson N, Sokolov V. Deep learning for short-term traffic flow prediction. Transportation Res Part C: Emerg Technol 2017;79:1–17. https://doi.org/10.1016/j.trc. ISSN 0968-090X.
[14] Tang J, Zhang G, Wang Y. A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation. Transportation Res Part C 2015;51(feb):29–40.
[15] Liu Z, Ding F, Dai Y, Li L, Chen T, Tan H. Spatial-temporal graph convolution network model with traffic fundamental diagram information informed for network traffic flow prediction. Expert Syst Appl 2024;249(A):123543.
[16] Xu K, Yang Z. Document-level attention-based BiLSTM-CRF incorporating disease dictionary for disease named entity recognition. Comput Biol Med 2019;108: 122–32. https://doi.org/10.1016/j.compbiomed. ISSN 0010-4825.
[17] Derry G, Mullen E, Marcelino K. Effects of dynamical time scale mismatch on time series analysis using event intervals. Commun Nonlinear Sci Numer Simul 2019;80:1007–5704.
[18] Farg E, Ramadan M, Arafat S. Classification of some strategic crops in Egypt using multi remotely sensing sensors and time series analysis. The Egyptian J Remote Sens Space Sci 2019.
[19] Yao H, Wu F, Ke J. Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of 32nd AAAI Conference on Artificial Intelligence; 2018. p. 2588–95.
[20] Guo S, Lin Y, Li S. Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting. IEEE Trans Intell Transp Syst 2019;20(10):3913–26.
[21] T. Kipf, M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," ar Xiv preprint ar Xiv:1609.02907, 2016.
[22] Li Y, Yu R, Shahabi C. Diffusion Convolutional Recurrent Reural Network: data-DrivenTraffic Forecasting. In: Proceedings of 6th International Conference on Learning Representa-tions; 2018.
[23] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: a deep learning frame-work for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence; 2018. p. 3634–40.
[24] Sheng Z, Lv Z, Li J, Xu Z. Deep spatial-temporal travel time prediction model based on trajectory feature. Comput Electr Eng 2023;110:108868.
[25] Xu Y, Jia Q, Wang H, Ji C, Li X, Li Y, Chen F. STI-TP: a spatio-temporal interleaved model for multi-modal trajectory prediction of heterogeneous traffic agents. Comput Electr Eng 2024;118(Part A):109361.
[26] Li Z, Chen J, Ma X, Du M, Zhang Z, Liu Q. A robust and accurate encrypted video traffic identification method via graph neural network. In: 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD); 2023. p. 867–72. https://doi.org/10.1109/CSCWD57460.2023.10152581.
[27] Virginia S, Chao-Kai C, Maziar S. Ameet S. talwalkar: federated multi-task learning. NIPS; 2017. p. 4424–34. Federated Multi-Task Learning.
[28] H Mcmahah, E Moore, Dramage, Federated learning of deep networks using model averaging. 2016. arxiv preprint arxiv:1602.05629, 2(2).
[29] Mikhail K, Maria-Florina B, Ameet S. Talwalkar: adaptive Gradient-Based Meta-Learning Methods. Neur IPS 2019:5915–26.
[30] Zhou Z, Yang S, Pu L. CEFL: online admission control, data scheduling, and accuracy tuning for cost-efficient federated learning across edge nodes. IEEE Internet Things J 2020;7(10):9341–56.
[31] Pandey SR, Nguyen LD, Popovski P. A contribution-based device selection scheme in federated learning. IEEE Communications Letters 2022;26(9):2057–61.
[32] Wang Z, Yu X, Wang H, Xue P. A federated learning scheme for hierarchical protection and multiple aggregation. Comput Electr Eng 2024;117. https://doi.org/10.1016/j.compeleceng.2024.109240.
[33] M. Arivazhagan, V. Aggarwal, et al., Federated learning with personalization layers. arXiv preprint arXiv:1912.00818, 2019.
[34] J. Oh, S. Kim, S. Yun, "FedBABU: towards Enhanced Representation for Federated Image Classification," 2021.
[35] P. Liang, T. Liu, et al., "Think locally, act globally:federated learning with local and global representations," arXiv preprint arXiv: 2001.01523,2020.
[36] X. Yao, T. Huang, et al., "Federated learning with additional mechanisms onclients to reduce communication costs," arXiv preprint arXiv:1908.05891,2019.
[37] Li T, Sahu A, et al. Federated optimization in heterogeneous networks. In: Proceedings of Machine learning and systems. 2; 2020. p. 429–50.
[38] Yi L, Shi X, Wang N, Wang G, Liu X, Shi Z, Yu H. pFedKT: personalized federated learning with dual knowledge transfer. Knowl. Based Syst. 2024;292:111633.
[39] Zhao Y, Pan S, Zhao Y, Liao H, Ye L, Zheng Y. Ultra-short-term wind power forecasting based on personalized robust federated learning with spatial collaboration. Energy 2024;288:129847.
[40] Liang M, Liu RW, Zhan Y, Li H, Zhu F, Wang F-Y. Fine-grained vessel traffic flow prediction with a spatio-temporal multigraph convolutional network. IEEE Trans Intell Transp Syst 2022;23(12):23694–707. https://doi.org/10.1109/TITS.2022.3199160.
[41] Deng T, Hamdan H, Yaakob R, Kasmiran KA. Personalized federated learning for in-hospital mortality prediction of multi-center ICU. IEEE Access 2023;11: 11652–63. https://doi.org/10.1109/ACCESS.2023.3241488.
[42] Dinh CT, Tran NH, Nguyen TD. Personalized federated learning with Moreau envelopes. Proc Adv Neural Inf Process Syst Jun. 2020:1–12.