Traffic Prediction Based on Formal Concept-Enhanced Federated Graph Learning

Kai Wu, Fei Hao, Ruoxia Yao, Jinhai Li, Geyong Min, *Member, IEEE*, and Sergei O. Kuznetsov

***Abstract*— Aiming to improve the efficiency of urban traffic management, previous studies have achieved considerable traffic prediction accuracy. For example, methods based on time series analysis perform well in short-term traffic prediction, and neural networks show strong capabilities in processing complex nonlinear relationships within traffic data. However, previous studies also have the following two limitations: 1) a large amount of complex traffic data will increase the complexity of the model during training and further reduce the accuracy of the training results; 2) the large-scale distribution of traffic data leads to incomplete model training and data security issues. To address these issues, we propose a Formal Concept-enhanced Federated Graph Convolutional Network (FC-FedGCN), which adopts formal concept analysis to fully mine graph data and improve the training accuracy of the GCNs model. Under federated learning, the GCNs model can be trained independently on different clients, and the local model is optimized by sharing model parameters. Coupled with the premise of protecting data privacy, the integrity of the data is guaranteed and the training accuracy of the GCNs model is improved. We compare our model with various baseline models based on the PEMS datasets, and the results demonstrate that FC-FedGCN has significant advantages in traffic prediction, outperforming the comparison methods in multiple indicators.**

***Index Terms*— Formal concept analysis, federated graph learning, traffic prediction.**

I. Introduction

WITH the acceleration of urbanization, traffic prediction has become a key issue in urban management and intelligent transportation systems. Traffic data includes information such as vehicle flow, speed, density, etc., and is highly dependent on time and space [1]. Usually, traffic data sets are abstracted into graph data with intersections as nodes and roads as edges [2], for data analysis and prediction. Accurate traffic flow prediction can effectively alleviate traffic congestion, optimize traffic management and improve travel efficiency. However, traditional prediction methods are difficult to fully exploit the spatio-temporal relationships in traffic data [1], [2].

For example, traffic forecast and time series analysis based on historical averages usually focus on pattern recognition in the time dimension [3]. These methods may be effective in dealing with time dependence, but they often ignore the spatial relationships and interactions between different locations (such as intersections and road sections) in traffic data [4]. Since traffic flow is not only affected by time changes, but also by complex interactions and dependencies between different locations in the transportation network, it is difficult for traditional methods to fully capture and utilize these spatio-temporal relationships, resulting in limited prediction accuracy and effectiveness [5]. This makes them perform poorly when dealing with complex traffic pattern changes and spatial heterogeneity, and cannot fully exploit the potential value in traffic data.

Graph Convolutional Networks (GCNs) are deep learning methods that effectively process graph-structured data and can capture the complex relationships between different nodes (such as intersections and road sections) in the transportation networks [6], [7]. However, there are two main problems when GCNs are applied to traffic data: 1) GCNs analyze and learn the characteristics of traffic data, but a large amount of complex data will increase the complexity of the model during training and reduce the accuracy of the training results [8], [9]; 2) GCNs usually rely on a centralized data collection and processing model, which will lead to data privacy and security issues and the large distribution and scale of traffic data, centralized processing is difficult to efficiently handle such a huge amount of data [10], [11].

Formal Concept Analysis (FCA), as a powerful tool for data analysis and rule extraction for information systems, can clearly express the hierarchical relationship between concepts, as well as the relationship between objects and attributes [12], thereby discovering the structures and relations in the data, solving the problems of high complexity and low result accuracy in GCNs model training effectively. This theory is based on mathematics and uses a formal context to represent the concepts, attributes, and relationships that make up the ontology [13], [14]. Then according to the formal context, its corresponding concept lattice is constructed to clearly express

Received 13 July 2024; revised 24 November 2024; accepted 31 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62477029, Grant 12271324, Grant 62476114, and Grant 61702317; in part by the Ministry of Education Humanities and Social Sciences Research Youth Fund Project under Grant 22YJCZH046; and in part by Yunnan Fundamental Research Projects under Grant 202401AV070009. The Associate Editor for this article was Y. Lv. *(Corresponding author: Fei Hao.)*

Kai Wu, Fei Hao, and Ruoxia Yao are with the School of Computer Science, Shaanxi Normal University, Xi'an 710119, China (e-mail: wukai@snnu.edu.cn; feehao@gmail.com; rxyao@snnu.edu.cn).

Jinhai Li is with the Faculty of Science, Kunming University of Science and Technology, Kunming 650500, China (e-mail: jhlixjtu@163.com).

Geyong Min is with the Department of Computer Science, University of Exeter, EX4 4PY Exeter, U.K. (e-mail: g.min@exeter.ac.uk).

Sergei O. Kuznetsov is with the School of Data Analysis and Artificial Intelligence, National Research University Higher School of Economics, 101000 Moscow, Russia (e-mail: skuznetsov@hse.ru).

Digital Object Identifier 10.1109/TITS.2025.3531108

the structure of the ontology [14]. Among these structures, we can easily explore the maximal clique structure to update the feature representation of nodes in the original graph data, thereby mining the connections between different intersections in the traffic network [15]. The spatial relationship can achieve higher model accuracy in model training. In traffic prediction, GCNs based methods often encounter the problem of insufficient feature expression due to the complexity and diversity of data. Especially in large-scale traffic networks, traditional feature extraction methods are difficult to fully capture the intrinsic structure of traffic data. The introduction of FCA can not only extract the hierarchical relationship and structure between concepts from the data, but also effectively identify the implicit associations between nodes, which is very critical for dealing with the potential complex topological relationships in traffic networks.

Federated Learning (FL) is an emerging distributed machine learning technology that allows models to be trained on multiple distributed devices without the need to centralize data on a central server [16], thereby improving data privacy and security, to solve the data privacy and security issues during GCNs model training. When processing traffic data sets, federated learning can effectively integrate data characteristics from different regions and build a global model with more generalization capabilities [17], [18]. This helps with differences in traffic patterns in different regions and improves the accuracy and robustness of predictions. In addition, the applications of FCA in the framework of FL further alleviates the privacy issues in cross-regional data fusion. By extracting the concept lattice structure locally and sharing parameters instead of raw data, we can improve the generalization performance of the model while ensuring data privacy. Therefore, the integration of FCA in FL not only enriches the structural expression ability of the data, but also alleviates the risk of data leakage through the parameter sharing mechanism, so that the traffic flow prediction model can better handle the heterogeneity and spatio-temporal correlation in traffic data while protecting privacy.

Therefore, we propose a traffic prediction method based on Formal Concept-enhanced Federated Graph Learning. To address the above shortcomings, we combine GCN with FCA and FL. Through this method, it is possible to make full use of the spatio-temporal characteristics of traffic data while protecting data privacy, while solving the challenges of traditional methods in data privacy protection and cross-regional data fusion, thereby significantly improving the effect of traffic flow prediction.

The major contributions of this paper are summarized as follows:

- In order to better extract spatial information from traffic data, we propose a novel Formal Concept-enhanced Graph Convolutional Network (FC-GCN) model. This integration enhances the training accuracy of the models by leveraging FCA to distill structural features from traffic data and subsequently refining the feature representation of the traffic data.
- We create the FC-FedGCN model, a groundbreaking federated graph learning framework that mitigates the risk of privacy breaches during the GCN training for traffic data from distinct regions. This model also effectively addresses the challenges of data dispersion and privacy preservation when predicting traffic flows across different areas.
- We assess the performance of our method using the datasets from the PEMS. The outcomes indicate that our method achieves a reduction in forecasting error by roughly 0.57% to 8.27% when compared to various benchmark techniques. This suggests that the FC-FedGCN model outperforms others in the domain of traffic flow prediction.

The remainder of this article is organized as follows. Section II introduces the related research work of traffic prediction and Federated Graph Learning. The key formulas and examples are used to explain the preliminary knowledge of FCA, GCN and FL in Section III. To better introduce our framework for traffic prediction, we elaborate the overall framework of FC-FedGCN in Section IV. Section V describes the datasets used for experimentation, with the subsequent analysis of the experimental outcomes. Finally, a summary and outlook for future work were given in Section VI.

II. Related Works

This section focuses on the research of traffic prediction and FL in graph learning.

A. Traffic Prediction

Statistical and traditional machine learning models are prominent data-driven techniques for predicting traffic.

In the realm of time-series analysis, the autoregressive integrated moving average (ARIMA) model and its extensions are well-established approaches rooted in classical statistical methods. These have been extensively utilized for traffic forecasting tasks, as evidenced by [19], [20], and [21]. However, these models are typically tailored for smaller datasets and often fall short when faced with intricate and fluctuating time series data.

Deep learning models leverage a far more extensive array of features and intricate architectures compared to their classical counterparts, which is why GCNs are employed for modeling non-Euclidean spatial structures. This approach aligns more closely with the inherent structure of traffic road networks. Liu et al. [22] proposed a spatio-temporal dual adaptive graph convolutional network model for spatio-temporal traffic prediction. The model uses a dual adaptive adjacency matrix composed of static and dynamic graph structure learning matrices to solve the problem of inaccurate predictions caused by fluctuations in the relationship between road sections due to factors such as traffic accidents, weather conditions and other events. Wang et al. [23] proposes a method to perform multi-dimensional cross-attention and spatio-temporal graph convolution. This method fully considers the cross information and constructs an attention cross view between each pair of dimensions in the channel, temporal and spatial domains to model the cross-dimensional dependencies of traffic data. Bao et al. [24] proposes a novel approach based on

TABLE I
SUMMARY OF RELATED WORKS

Related Work	Papers	Large Dataset	Temporal Dependency	Spatial Dependency	Privacy Protection
Traffic Prediction	Model-based Methods [19] [20] [21]	✗	✓	✗	✗
	Machine Learning-based Methods [22] [23] [24] [25] [26] [27] [28] [29]	✓	✓	✓	✗
Federated Learning	Traffic Flow Prediction [30] [31]	✓	✗	✗	✓
	Federated Graph Learning [32] [33] [34] [35]	✓	✗	✗	✓
Ours	FC-FedGCN	✓	✓	✓	✓

prior knowledge enhanced time-varying graph convolutional network to describe dynamic and long-term spatio-temporal correlations. In addition, the papers [25], [26], [27], [28], [29] has improved the GCN model in terms of time and space.

B. Federated Learning

Federated learning is a distributed learning framework that plays an important role in traffic flow prediction, graph learning, and other aspects.

In terms of traffic flow prediction, [30] proposed a federated learning algorithm for network traffic prediction (Fed-NTP) based on the long short-term memory (LSTM) algorithm, trained the model locally, and implemented the LSTM algorithm in a decentralized manner by using the federated learning algorithm on the vehicular ad hoc network (VANET) dataset, and predicted the network traffic based on the most influential features of the network traffic in the road and network. Reference [31] proposed a federated learning framework based on a consortium blockchain, where model updates from distributed vehicles are verified by miners to prevent unreliable model updates and then stored on the blockchain.

In federated graph learning, [32], [33], [34], and [35] aim to incorporate graph neural network (GNN) models into a federated learning framework. The primary focus of these studies is on leveraging the federated system for enhanced privacy protection. Additionally, they seek to improve the accuracy of GNN model training within this distributed learning environment. In the federated learning paradigm, the integration of GCNs is a significant step forward, as it allows for the modeling of complex graph-structured data while maintaining the privacy of individual participants' data. This is achieved by training the models on decentralized data without the need to centralize the data itself, thus ensuring that sensitive information remains confidential.

As shown in Table I, and through the summary of the above related works, we found that there are two problems in the current research in the field of traffic prediction: 1) Insufficient consideration of the structural characteristics of traffic data; 2) The privacy of data is not well protected during the training process. Therefore, this paper uses a FC-FedGCN framework for traffic prediction, which solves the above problems well.

III. Preliminaries and Problem Formulation

A. Formal Concept Analysis

FCA is a mathematical theory used to describe and analyze the classification structure of data and used to process conceptual data systems. The core idea of FCA is to regard concepts as associations between sets of objects and sets of attributes, and to reveal the relationships between concepts through formal methods. It can provide a rigorous mathematical framework to analyze conceptual data systems and process large-scale data sets.

Definition 1 (Formal Context): Formal context is consists of a set of objects and attributes in all research fields, usually represented by a binary relationship matrix, where the rows represent the set of objects, the columns represent the set of attributes, and the elements in the table mark whether the objects have the corresponding attributes. Formal context is represented as triplets $K = (O, P, R)$, where $O = \{o_1, o_2, \ldots\}$ is the object set, $P = \{p_1, p_2, \ldots\}$ is the attribute set, and R is the binary relationship between O and P. $R \in O \times P$, and $(o, p) \in R$ means that object o has attribute p.

In a traffic network $G = (V, E)$, $V = \{v_1, v_2, \ldots, v_m\}$ is the set of intersections, and $E = \{e_1, e_2, \ldots, e_n\}$ is the road connecting the two intersections. With this background, the objects and attributes in the formal context are all intersections v. If there exists a road e_{ij} between intersections v_i and v_j, the corresponding i_{th} row and j_{th} column of the formal context matrix is set to 1, otherwise set to 0. The modified adjacency matrix generated based on the following rules is the formal context of the social network:

$$I = \left\{ \begin{matrix} 1, (v_i, v_j) \in E \\ 1, i == j \\ 0.other \end{matrix} \right\} = a_{ij} \tag{1}$$

we denote this formal context as $K = (V, V, R)$.

Definition 2 (Concept Lattice): In FCA, a concept is composed of a set of objects and a set of attributes. And the following conditions are satisfied all objects in the collection have all the properties in the collection, and objects outside the collection do not have these properties. For a formal context triple $K = (O, P, R)$ or $K = (V, V, R)$, a concept can be represented as a binary pair (A, B), where $A \subseteq O$ and $B \subseteq P$, A and B are called the extent and intent of this concept respectively. A concept lattice $L(O, P, R)$ can be generated by topological relations between all concepts in $C(K)$.

Example 1: Fig. 1 shows a partial subway line diagram of a city. We take five transfer stations as an example to extract the traffic network diagram shown in Fig. 2. Fig. 2 depicts a traffic network consisting of 5 nodes and 7 edges. In this network, the nodes symbolize intersections, and the edges connecting these nodes represent the roads that link them. It is important to note that each node is characterized

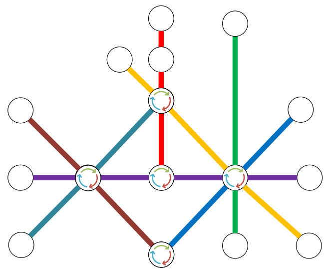


Fig. 1. A railway network.

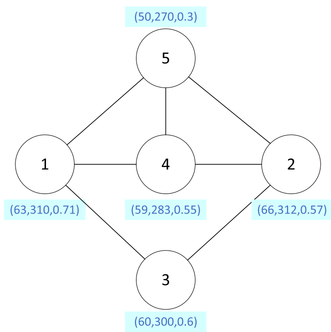


Fig. 2. A traffic network.

TABLE II
EXAMPLE FORMAL CONTEXT K

K	1	2	3	4	5
1	X		X	X	X
2		X	X	X	X
3	X	X	X		
4	X	X		X	X
5	X	X		X	X

by distinct features such as speed, traffic flow, and occupancy. Table II shows its formal context K with $O = \{1, 2, 3, 4, 5\}$ and $P = \{1, 2, 3, 4, 5\}$. Here, both objects and attributes are represented by the nodes themselves, and the road of an edge between two nodes is denoted by an "X". It can be observed that object $\{1\}$ has the same attributes $\{1, 3, 4, 5\}$, and attributes $\{1, 3, 4, 5\}$ also correspond to the same objects $\{1\}$. Thus we get the concept $(\{1\}, \{1, 3, 4, 5\})$. $\{1\}$ is the extent of this concept, and $\{1, 3, 4, 5\}$ is the intent of this concept. Due to the symmetric nature of this formal context, there exists an inverse symmetrical concept $(\{1, 3, 4, 5\}, \{1\})$ under this example.

Definition 3 ([36], [37] Generator): Given a formal concept (A, B) of $C(K)$, $Q \subseteq A$ is a generator of A if $Q' = B$ (and hence $Q'' = A$, $'$ represents Generator).

Definition 4 ([37], [38] Concept Stability): Let $K = (O, P, R)$ be a formal context. Given a formal concept (A, B) of $C(K)$, the concept stability σ of (A, B) is defined as follows:

$$\sigma(A, B) = \frac{|\{Q \subseteq A \mid Q' = B\}|}{2^{|A|}} \tag{2}$$

The stability of a concept is a metric that assesses the resilience of the concept's definition against variations or disturbances within the data. Specifically, it evaluates the condition where the removal of certain entities from the set of objects (extent) associated with a formal concept (A, B) does not lead to a change in the concept's intent. A concept with higher stability indicates a more consistent and reliable structure, as represented by the vertices of the concept within a graph. This also implies a stronger distinction and separation of the concept when viewed in the context of the entire data structure.

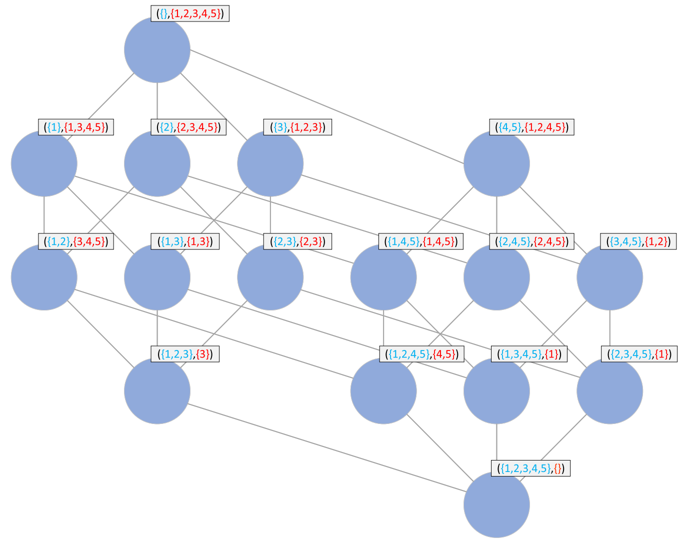


Fig. 3. Concept lattice of Example 1 (each node represents a concept which is formed by extent (in blue) and intent (in red)).

Definition 5 ([37], [38], [39] Concept Separation): Let $K = (O, P, R)$ be a formal context. Given a formal concept (A, B) of $C(K)$, we define the concept separation ξ of (A, B) as follows:

$$\xi(A, B) = \frac{|A|\,|B|}{\sum_{e \in A} |f(A)| + \sum_{i \in B} |g(B)| - |A|\,|B|} \tag{3}$$

Separation is a metric that quantifies the uniqueness of a concept within a dataset, relative to the distribution of objects across the various extents of formal concepts. It is determined by the ratio of the distinct area that a particular concept occupies to the overall area encompassing all the objects and attributes associated with that concept. Typically, a concept with a more focused distribution, or a smaller spread, is deemed more representative.

Definition 6 ([15] Equiconcept): Given a concept (A, B) of $C(K)$, if $A = B$, then concept (A, B) is an equiconcept.

It is clear that two concepts are considered identical if they share the same extent and intent, meaning there is a one-to-one correspondence between the elements of the node sets that constitute each concept. Such concepts are referred to as equiconcepts.

Equiconcepts hold a distinctive position within the structure of a concept lattice. In our prior research, we have established that there exists an equivalence between equiconcepts and maximal cliques [15]. This implies that the process of identifying equiconcepts within a concept lattice is analogous to the task of locating maximal cliques in the underlying graph, and this equivalence can be formulated as:

$$\zeta(K) \equiv maximal\_clique(G) \tag{4}$$

TABLE III
CONCEPTS THAT GENERATED BY FIG. 2

extent	intent	$\sigma(C)$	$\xi(C)$	equiconcept
{1}	{1,3,4,5}	0.5	0.267	NO
{2}	{2,3,4,5}	0.5	0.267	NO
{3}	{1,2,3}	0.5	0.272	NO
{1,2}	{3,4,5}	0.25	0.462	NO
{1,3}	{1,3}	0.25	0.4	YES
{2,3}	{2,3}	0.25	0.4	YES
{4,5}	{1,2,4,5}	0.75	0.5	NO
{1,2,3}	{3}	0.125	0.273	NO
{1,4,5}	{1,4,5}	0.375	0.6	YES
{2,4,5}	{2,4,5}	0.375	0.6	YES
{3,4,5}	{1,2}	0.375	0.462	NO
{1,2,4,5}	{4,5}	0.1875	0.5	NO
{1,3,4,5}	{1}	0.1875	0.267	NO
{2,3,4,5}	{2}	0.1875	0.267	NO
{1,2,3,4,5}	{}	0.09375	0	NO
{}	{1,2,3,4,5}	0	0	NO

where K is the formal context of the graph G, $\zeta(K)$ denotes the set of equiconcepts for formal context K, and $maximal\_clique(G)$ denotes the set of maximal cliques in graph G.

Example 2: Continuing the Example 1, Table III presents the concept stability, separation and the identification of equiconcepts. It is observed that the stability of concept $(\{1\}, \{1, 3, 4, 5\})$ is 1/2, where 2 represents the base of the set $\{1, 3, 4, 5\}$ and 1 represents the number of generators of this set. The separation of concept $(\{1\}, \{1, 3, 4, 5\})$ is 4/15, where 4 is the common area of this concept and it is shown as the red part in Table II, while 15 is the total area occupied by the extent and intent and is shown as the red and gray part in Table II. Within the table, three concepts are recognized as equiconcepts, characterized by identical extents and intents. These equiconcepts are typically observed to possess higher stability and separation values compared to other concepts. This heightened stability and separation suggest that the equiconcepts are associated with maximal cliques within the graph that are more robust and distinct.

B. Formal Concept-Enhanced Graph Convolutional Neural Networks

GCN is a neural network architecture specifically designed to process graph structured data. It learns the features of nodes by utilizing the topological structure information in the graph and performing convolution operations on the nodes in the graph. In this section, we denote a graph as $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_m\}$ is called the vertices set and $E = \{e_1, e_2, \ldots, e_j\}$ is called the edges set. $A \in \mathbb{R}^{m\times m}$ and $X \in \mathbb{R}^{m\times n}$ are the adjacency matrix and feature matrix of the graph respectively, where m is the number of nodes of graph G, n is the length of the nodes' feature.

Example 3: Continuing with Example 2, we can see through Table III that four of the sixteen concepts generated are equiconcepts, therefore we generate a matrix with dimension 5×4, and perform "$\oplus$" operation as defined in Eq.(9) with the original feature matrix to generate a new feature matrix X^*.

The iterative process of FC-GCN can be summarized as follows:

$$H^{l+1} = \theta(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}(X^*)^l W^l) \tag{5}$$

Among them, $\tilde{A} = A + I$, $\tilde{A}$ is the modified adjacency matrix of the data set input to this layer, D is the degree of $\tilde{A}$ and $\tilde{D} = D + I$, so $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is the Laplacian matrix, X^* is the updated feature using FCA, W^l is the trainable parameter matrix of the l layer of GCN, H^l and H^{l+1} are the input and output of l_{th} layer respectively, H_o is the original feature matrix and H_eq is the added feature matrix.

C. Federated Learning

In the diagram illustrated as Fig. 4, we examine a horizontal federated learning framework consisting of a single server and several clients. Within this system, each edge device utilizes its own local dataset to train models. Subsequently, the server is responsible for consolidating the parameters of the models that have been trained by the edge devices. After aggregation, the server disseminates the synthesized model parameters back to the edge devices. This process enables the devices to proceed with the subsequent phase of model training. We assume that device k has the dataset $D_k = \{x_{k,j}, y_{k,j}\}_{j=1}^{n_k}$, $n_k = |D_k|$ indicates the amount of data in the dataset D_k, $x_{k,j}$ denotes the j_{th} input data of the k_{th} device, $y_{k,j}$ is the label of $x_{k,j}$. The overall dataset $D = \bigcup_{k\in N} D_k$, the overall sample $n = \sum_{k=1}^{N} n_k$. The goal of training is to find model parameters ω to minimize the loss function on the overall dataset, local parametric model is shown as the following formula [10]:

$$\omega_{t+1}^k \longleftarrow \omega_{t+1}^k - \eta \bigtriangledown F_k(\omega) \tag{6}$$

where $F_k(\omega) = \frac{1}{n_k}\sum_{\{x_{k,j}, y_{k,j}\}\in D_k} f(\omega, x_{k,j}, y_{k,j})$ is local loss function, loss function $f(\omega, x_{k,j}, y_{k,j})$ measures the inaccuracy of the model parameter ω on the data pair $(x_{k,j}, y_{k,j})$. After that the central server aggregates all the parameters and returns them to the clients through the following formula [10]:

$$\omega_{t+1} = \sum_{k=1}^{M} \frac{m_k}{m}\omega_{t+1}^k \tag{7}$$

which $\frac{m_k}{m}$ is the proportion of samples in client k in round $t + 1$ to all samples of the M clients participating in training.

D. Problem Statement

In this paper, we first adopt the formal concept-enhanced graph convolutional network to predict traffic rate, flow and occupancy. Then, we illustrate the use of the characteristics of federated learning framework to solve the problem of data privacy and achieve the purpose of improving the accuracy of model training.

1) Problem Statement: Suppose there are k clients, and each client has a set of traffic data sets G_k, $G_k = (V_k, E_k, X_k)$, where $V_k = \{v_{k_i}\}_{i=1}^N$ is the node set of a network with N nodes, $E_k = \{(v_{k_i}, v_{k_j}) \mid 1 \le i, j \le N, i \ne j\}$ is the edge set of this traffic network, and $X_k = \{x_{k_i}\}_{i=1}^N$ is the feature set of G_k. Our goal is minimize the overall loss function of all

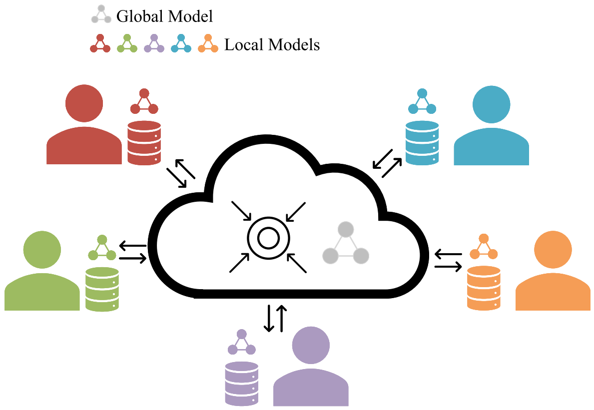


Fig. 4. Federated learning.

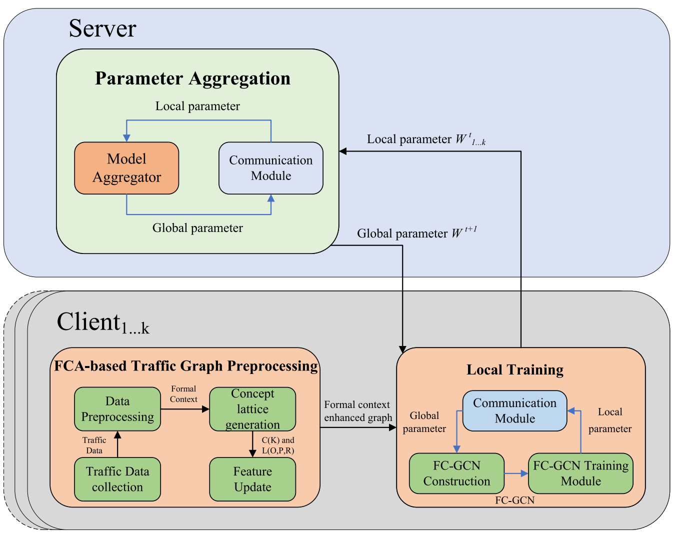


Fig. 5. Framework of formal concept-enhanced federated graph learning for traffic prediction.

clients by optimizing the parameters and predict future traffic characteristics such as flow at each node. Consequently, the primary goal of the loss function is to reduce the discrepancy between predicted and real traffic data to the greatest extent possible.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{8}$$

As shown in Fig 5, each client trains the local FC-GCN model, generates its own model parameters, and uploads them to the central server for aggregation via Eq. 7. The server sends the aggregated parameters back to all clients to update local parameters via Eq. 6 for next round of training. In the first round of training, each client only uses local raw data as training data to obtain the traffic prediction model and its parameters. In subsequent training, each client updates local data and trains through global parameters.

IV. Formal Concept-Enhanced Federated Graph Learning

In this section, we introduce the FC-FedGCN framework for traffic prediction. As shown in Fig. 6, the framework is divided into three modules, i.e., FCA-based graph processing, FC-GCN model training and Federated Learning. In the FCA data processing module, we conduct an analysis of the original data to extract concept features, which enrich the feature representations of the original nodes. It processes results in the generation of a formal concept-enhanced graph and features for each client. In the FC-GCN Model training module, each client abstracts the traffic network graph into graph data, and uses the traffic flow, occupancy, speed and other data collected by sensors as features for FC-GCN model training. The Federated Learning module is bifurcated into client and server components, where the formal concept-enhanced graph is trained, leading to the acquisition of an updated model, and achieve the prediction results of traffic data.

A. FCA-Based Graph Preprocessing

In order to improve the accuracy of GCN in traffic data feature extraction, this study integrated the FCA concept lattice feature extraction method based on traditional GCN, so that the model can benefit from higher-level structured information. Specifically, FCA mines equiconcepts between nodes through the concept lattice structure, and uses the feature update mechanism of the maximal cliques to fuse the equiconcept matrix with the original feature matrix to generate an enhanced feature matrix. This feature enhancement operation not only improves the model's ability to represent different traffic nodes, but also significantly reduces the model's training error on complex graph structures. In this way, FC-GCN can simultaneously capture local and global features when node features are updated, thereby improving the model's prediction accuracy in multi-regional transportation networks. Compared with traditional GCN, which only relies on node neighborhood information extraction, this method is more suitable for handling the heterogeneity of cross-regional data and achieving better feature representation under the constraints of data privacy.

In the FCA-based graph preprocessing module, each client processes its local data independently. Given a traffic network $G = (V, E)$ as the local graph input to the FCA data processing module, we first process the local graph data and add self-loops to each node. In the corresponding graph adjacency matrix, the elements on the diagonal are all 1. Then we perform FCA on the processed data to obtain its concepts $C(K)$ and corresponding concept lattices $L(V, V, R)$. By calculating and evaluating the stability of the concepts in the concept set, we obtain the new graph node characteristic value, which we call conceptual features. After feature splicing operation, we finally obtain formal concept-enhanced graph and features.

As previously stated, an equiconcept is a unique concept characterized by an identical extent and intent, which corresponds to the structure of maximal clique within a clustering framework. Maximal cliques represent significant structural components of a graph. The features of an equiconcept are derived from the extraction of these maximal cliques, thereby enriching the graph's feature matrix. This

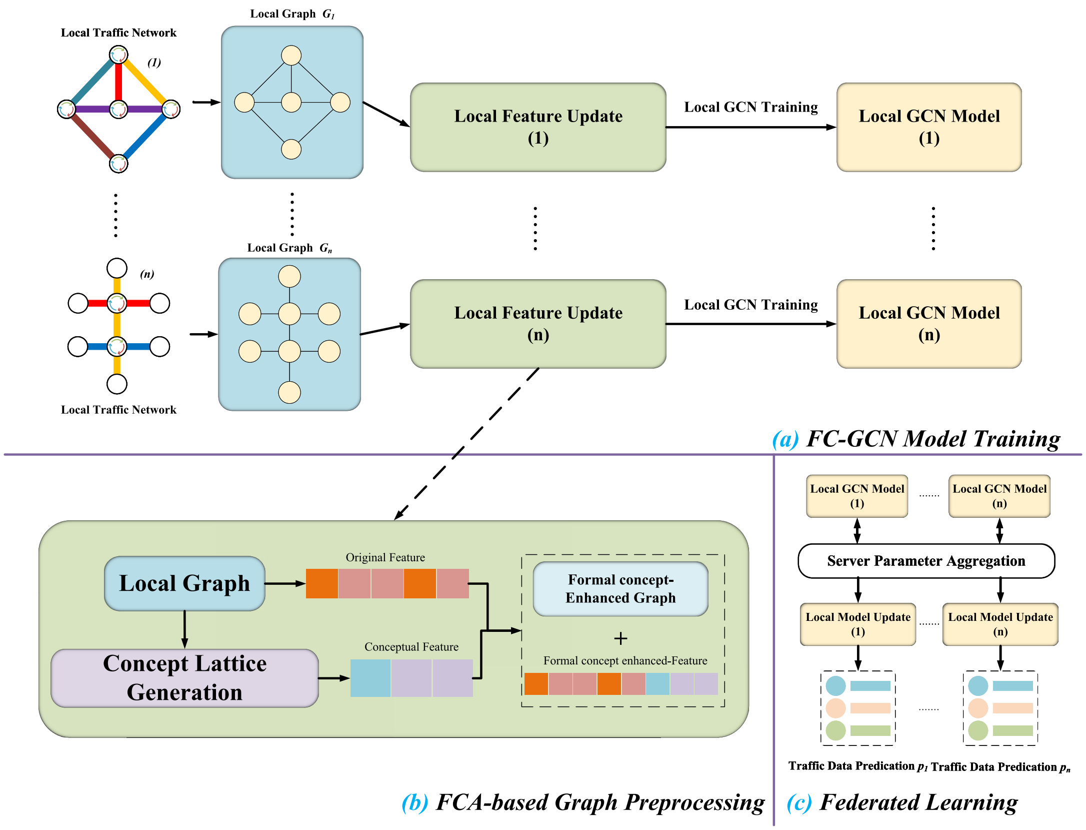


Fig. 6. Overall flowchart of FC-FedGCN for traffic predication.

Algorithm 1 Feature Update

Input:
Graph feature matrix $X \in \mathbb{R}^{m \times n}$, concepts set $C(K)$;
Output:
updated feature matrix $X^* \in \mathbb{R}^{m \times (n+eq)}$;
Initialize maximal clique matrix f, maximal clique set ζ, matrix X^*
begin
  for $\forall (E, I) \in C(K)$
    if $E == I$
      $\zeta = \zeta \cup (E, I)$
      $eq = eq + 1$
  end for
  for $(E, I)_j \in \zeta$
    if $v_i \in E$ $f_{ij} = 1$
    else $f_{ij} = 0$
  end for
  $X^* = X \oplus f$
end

enhancement effectively boosts the model's training efficiency and accuracy.

FC-GCN uses the FCA method to preprocess the input graph data and update the feature representation of its nodes. We extract the equiconcepts with number eq in $C(K)$ and generate a 0-1 matrix $f \in \mathbb{R}^{m \times eq}$. Then use the matrix f to update the feature matrix X of the graph to generate the updated feature matrix X^*, which defines the operation "$\oplus$" as following:

$$X^* = X \oplus f, X^* \in \mathbb{R}^{m \times (n+eq)} \tag{9}$$

The updating process and results are shown in Algorithm 1.

B. Federated Learning

Federated learning is bifurcated into two primary components: the client and the server. The client's role is to train the local model, while the server's role is to aggregate the parameters from the models trained by the clients and then disseminate these aggregated parameters back to the clients. This cycle of training and parameter sharing continues to iteratively refine the model. In this section, we delve into the specifics of how federated learning is applied in the context of formal concept-enhanced federated graph learning.

Under FL, each client performs model training locally and shares parameters based on the concept feature matrix extracted by FCA and the update rules of GCN. Through parameter exchange between multiple clients, FL realizes the fusion of cross-regional traffic data features, but does not directly transmit data, thereby ensuring data privacy. Compared with the traditional FL model, the FC-FedGCN proposed in this paper can improve the adaptability and generalization of the global model to traffic characteristics of different regions while maintaining data privacy through the aggregation of concept features.

1) Local FC-GCN Model Training by Clients: In the federated learning framework, each client is tasked with training the FC-GCN model utilizing the local data derived from the formal concept-enhanced graph. Prior to commencing the inaugural training round, clients receive the divided formal concept-enhanced graph data from the Formal Concept-Enhanced Graph module, which serves as the foundation for the initial model training. Subsequently, they initiate the first training round.

At the onset of the initial training round, client k initializes the local weight parameters W^k and feeds in the Laplace matrix along with the formal concept-enhanced feature matrix to facilitate model training. Upon completion of the first round, client k uploads the local model parameters to the server, where they are integrated with other client parameters to form the new global model parameters $\bar{W}_1$. Client k then proceeds with the second round of training using these updated parameters.

Following this, the clients engage in several training iterations to refine the feature weights in accordance with their respective local graph data. The methodology for training is encapsulated in the subsequent equation:

$$H^{l+1} = \theta(\tilde{Q}^l (X^*)^l W^l), \quad \tilde{Q}^l = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \tag{10}$$

where X^* is the feature matrix of the original graph data H_o with the added features H_{eq} with operation "$\oplus$" to obtain the formal concept-enhanced feature matrix, and it is used as the initial feature input for the first round of training.

Specifically, each training iteration is composed of two principal phases: the operation of the graph convolution layer and the activation of the softmax function. The local FC-GCN training process for each client $client_k$ is described in Algorithm 2.

Algorithm 2 Local FC-GCN Training of $Client_k$

```
for each training round t
    Construct an FC-GCN G_i = {V_i^1, V_i^2, ..., V_i^L}
    for each v ∈ V_i^l
        if t == 1
            Z^{l+1} = Σ_{v∈V_i} Q̃_i^l (X*)^l W_i^l
        if t > 1
            Z^{l+1} = Σ_{v∈V_i} Q̃_i^l (X*)^l W_i^l + Σ_{u∈V_j} Q̃_j^l (X*)^l W_j^l
            (j ≠ i)
        end if
            H^{l+1} = θ(Z^{l+1})
    end for
end for
```

The subsequent training rounds following the initial one, once the clients have received the global parameters from the server, client k will then proceed to update the local parameters for the upcoming training round. Degree matrix D and adjacency matrix A are shown as:

$$D_{uv}^{t+1} = \begin{cases} \dfrac{|N'(v)|}{|N(v)|} D_{uv}^t, & u \in N'(v) \\ D_{uv}^t, & other \end{cases} \tag{11}$$

$$A_{uv}^{t+1} = \begin{cases} 1, & u \in N'(v) \\ 0, & other \end{cases} \tag{12}$$

where $N(v)$ denotes the neighbor of node v at the t_{th} round of training and $N'(v)$ denotes the neighbor node of v at the $(t+1)_{th}$ round of training.

The purpose of traffic forecasting is to ensure that the predictions closely align with the actual traffic conditions.

2) Parameter Aggregation by Server: In every cycle of federated learning, clients send their local model parameters to the server. Upon receiving all the parameters $W = \{W_1, W_2, \ldots, W_k\}$, the server consolidates these local model parameters to derive the global parameter $\bar{W}$. This global parameter is then disseminated back to all clients, initiating the subsequent round of training. The parameter aggregation formula is shown in Eq.(13):

$$\bar{W}^{t+1} = \sum_{i \in W} \frac{|V_i^t|}{\sum_{i \in W} |V_i^t|} W_i^t \tag{13}$$

where v_i^t is the nodes set of of client i at the t_{th} training round, $\bar{W}^{t+1}$ is the global parameter at the $(t+1)_{th}$ round of training. When the $client_i$ receives the global parameters, it will update the local parameters according to the global parameters with the following equation:

$$W_i^{t+1} = \sum_{j \in W} \frac{|V_j| W_j^t}{\sum_{j \in W} |V_j|} \tag{14}$$

where V_j is the set of nodes with labels in the $client_j$.

V. Experiments

In this section, we present a comprehensive overview of our experimental setup and findings. Furthermore, we also present comparative results between our approach and other existing baselines. The source code has been uploaded to github.[1]

A. Datasets and Settings

In the experiment, we use the well-known traffic flow dataset PEMS. The PEMS dataset contains traffic flow data from multiple sensor sites, covering a large number of traffic parameters such as vehicle flow, vehicle speed, and lane occupancy. We use PEMS04[2] and PEMS08[3] as training data on different clients. PEMS04 contains data from 3848 detectors on 29 highways in the San Francisco Bay Area from January 1, 2018 to February 28, 2018, collected every 5 minutes, so the shape of the original traffic data after reading is (307, 16992, 3), where the 3D features are flow, occupy, speed. PEMS08 contains data from 1979 detectors on 8 highways in San Bernardino, Southern California, from July 1, 2016 to August 31, 2016, and its data shape is (170, 17856, 3). When the number of clients is greater than two, we use the above datasets as training data on different clients.

During the conducted experiments, the input data underwent normalization, resulting in a range confined between 0 and 1. The dataset is then divided, with 80% allocated for training

[1] https://github.com/Kaiiiii1/traffic_predicition
[2] https://gitcode.com/open-source-toolkit/5072d
[3] https://gitcode.com/open-source-toolkit/73aea

the model and the remaining 20% reserved for testing the model. The objective is to forecast traffic speeds for the subsequent 15, 30, 45, and 60 minutes. We have three adjustable parameters: $client\_num$ representing the number of clients, $epochs$ indicating the number of training iterations for the local models, and $round$ denoting the number of times parameters are aggregate. We then evaluate the traffic prediction model using the following evaluation metrics in different models:

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{15}$$

- Root Mean Square Error ($RMSE$):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{16}$$

- Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{17}$$

- Variance (Var):

$$Var = Var = 1 - \frac{Var(y - \hat{y})}{Var(y)}. \tag{18}$$

- Accuracy (Acc):

$$Acc = 1 - \frac{|y_i - \bar{y}|}{|y|} \tag{19}$$

MAE is used to measure the average absolute difference between the predicted value and the true value. The smaller the value, the smaller the model prediction error. $RMSE$ measures the standard deviation between the predicted value and the true value, which can reflect the degree of dispersion of the prediction error. R^2 is used to measure the ability of the model to explain the variables. The closer the value is to 1, the better the model can explain the variability of the data. Var is used to evaluate the volatility of the data set and the fit of the model. Acc measures the degree of match between the model prediction results and the true value, which is usually determined by comparing the consistency of the prediction results with the true results.

B. Results and Analysis

For comparison we extend the following schemes for FC-FedGCN as follows.

- ***History Average model (HA)*** [40]: HA utilizes the mean traffic data from previous time periods as a forecast.
- ***Autoregressive Integrated Moving Average model (ARIMA)*** [41]: ARIMA adjusts the observed time series to a parametric model for forecasting future traffic data.
- ***Support Vector Regression model (SVR)*** [42]:SVR employs historical data for model training to establish the correlation between inputs and outputs, subsequently forecasting by inputting anticipated future traffic data.
- ***GCN*** [9]: GCN serves as a feature extraction technique for graph-structured data. The primary objective of GCN is to distill meaningful features from the graph data, enabling it to effectively carry out various tasks, including vertex classification and edge prediction.
- ***T-GCN*** [26]: T-GCN innovative model fuses the strengths of GCN and Gated Recurrent Units (GRU). The GCN component is tasked with deciphering intricate topological patterns to grasp spatial relationships, while the GRU component is designed to understand the fluctuating nature of traffic data, thereby capturing temporal dynamics.
- ***AST-GCN*** [25]: AST-GCN incorporates external factors by categorizing them into dynamic and static attributes and employs an attribute enhancement module to encode and assimilate these elements into the spatio-temporal graph convolutional framework.
- ***DCRNN*** [27]: DCRNN approach to modeling traffic flow treats it as a diffusion process across a directed graph. This method introduces a diffusion convolutional recurrent neural network that adeptly integrates the spatial and temporal aspects of traffic flow. By employing bidirectional random walks on the graph, it effectively captures spatial dependencies.
- ***Graph WaveNet*** [29]: Graph WaveNet accurately captures the hidden spatial dependencies in the data by developing a novel adaptive dependency matrix and learning it through node embedding.
- ***STGCN*** [28]: STGCN addresses the problem on graph-structured data and constructs the model using full convolutional layers, resulting in significantly faster training and reduced parameter count.
- ***FC-GCN***: FC-GCN is a graph convolutional neural network to extract and refine the features from the input convolutional layer. By incorporating information of equiconcept, it enhances the feature representation, thereby improving the model's predictive accuracy.
- ***FC-FedGCN***: FC-FedGCN is a Federated Graph framework, which train graph data on each client and exchange parameters between sever and clients, to improve the model's accuracy and protect data privacy.

Experiment1–Comparison of Model Indicators with Baseline Methods

Table IV presents a comparison of the FC-FedGCN model with other baseline methods on the PEMS datasets. The results indicate that the FC-FedGCN model consistently achieves superior predictive accuracy across all considered evaluation metrics and for nearly all forecast horizons. This underscores the efficacy of the FC-FedGCN model in the domain of traffic flow prediction. $*$ means that the values are too small to be negligible, indicating that the model's prediction effect is poor. Specifically, the FC-GCN and FC-FedGCN models demonstrated a reduction in Root Mean Square Error (RMSE) of approximately 11.3% and 50.6%, respectively, compared to the Historical Average (HA) model, with corresponding increasing in accuracy of about 2.75% and 8.53%. A lower RMSE value also indicates that there are few outliers in the predicted values. When compared to the Autoregressive Integrated Moving Average (ARIMA) model, the RMSE is reduced by around 14.5% and 52.4%, and the accuracy

TABLE IV
THE PREDICTION RESULTS OF FC-FEDGCN AND OTHER BASELINE METHODS

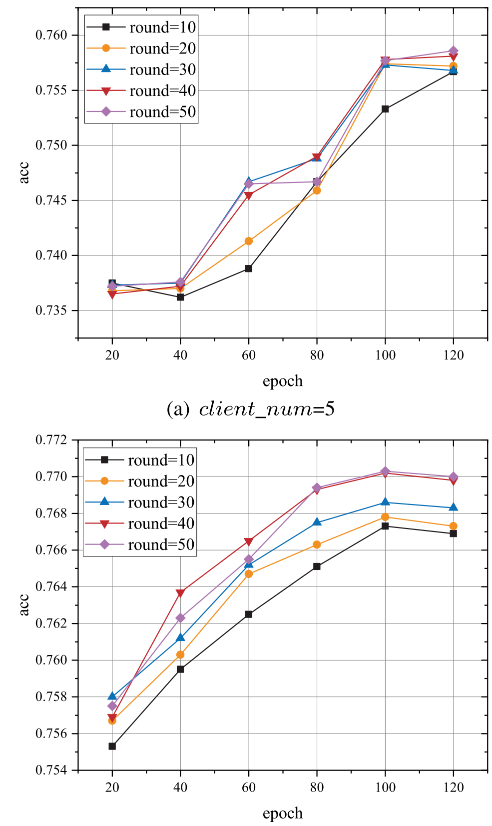
Model \ Metric	MAE	$RMSE$	R^2	Var	Acc
HA	3.1968	4.9198	0.7914	0.7914	0.6807
$ARIMA$	4.6557	6.2151	*	0.0121	0.4278
SVR	4.7236	7.5368	0.8367	0.8121	0.6961
GCN	4.2178	5.6418	0.6689	0.6147	0.6433
$T-GCN$	2.7209	4.0507	0.8421	0.8297	0.7033
$AST-GCN$	2.7054	4.0231	0.8521	**0.8524**	0.7089
$DCRNN$	3.1768	4.5157	0.8282	0.8325	0.6998
$GraphWaveNet$	3.2197	4.5769	0.8431	0.8380	0.7045
$STGCN$	2.6758	4.0268	**0.8486**	0.8479	0.7203
$FC-GCN$	2.8003	6.0232	0.7872	0.7693	0.7082
$FC-FedGCN$	**2.4886**	**3.9094**	0.8173	0.8416	**0.7260**

is enhanced by 28.04% and 33.82%. Similarly, against the Support Vector Regression (SVR) model, the FC-GCN and FC-FedGCN models show a decrease in RMSE by 6.81% and 48.1%, with accuracy improvements of about 1.21% and 6.99%.

These superior results can be attributed to the limitations of traditional methods like HA, ARIMA, and SVR in dealing with intricate and non-stationary time series data. The GCN model's subpar performance is attributed to its focus solely on spatial features without delving deeply into the nuanced spatial characteristics of traffic data. Moreover, despite ARIMA's reputation as a well-established traffic forecasting technique, its accuracy is surpassed by the HA model, primarily due to its challenges in managing long-term and non-stationary datasets. The ARIMA model's forecasting process, which involves calculating the error at each node and averaging these values, is susceptible to increased overall error if certain data points exhibit significant fluctuations.

The FC-FedGCN model adds a federated learning framework and performs feature aggregation on traffic data in various regions while protecting data privacy. Compared with the FC-GCN model, the RMSE is reduced by 44.3% and the model accuracy is increased by 5.78%. Additionally, compared with other machine learning-based methods such as T-GCN, AST-GCN, DCRNN, Graph WaveNet and STGCN, FC-FedGCN has better performance in terms of MAE, RMSE, and accuracy. Specifically, the improvement of MAE is 6% to 21.6%, RMSE is 2.8% to 14.6%, and accuracy is 0.7% to 3.6%. These experimental results demonstrate that the FC-FedGCN has higher performance for traffic prediction.

In addition, we can conclude that the FC-FedGCN model has a higher consistency in prediction results than the HA model, ARIMA model, and SVR model. Although the FC-FedGCN model has a smaller R^2 and limited explanability, the smaller variance means that the prediction results given by the model are relatively stable and do not fluctuate greatly. Small fluctuations in the data set will also lead to smaller variance in the prediction results even if the model has a weak explanability.



(a) $client\_num$=5

(b) $client\_num$=10

Fig. 7. The influence of different parameters on model accuracy.

Experiment2–Interpretation of FC-FedGCN

(1) The effect of hyperparameters. We train the traffic prediction accuracy under different parameters $client\_num = 5$, $epoch \in [20, 120]$ and $round \in \{20, 40\}$, $client\_num$ is the number of clients, $epoch$ is the client training epoch, and $round$ is the number of rounds of parameter exchange between the client and the server.

As shown in Fig. 7, the horizontal axis is $epoch$ and the vertical axis is model accuracy acc. Note that, the blue curve round = 20 and the red curve round = 40. We can observe that as epoch increases, the model accuracy increases accordingly, but after the local training round increases to 100 rounds, the increase in model accuracy slows down, which may be caused by overfitting of the model training. Increasing the number of server parameter aggregations can also improve the accuracy of model training. For example, in Fig. 7(a), the red curve is mostly above the blue curve; in addition, the increase in the number of clients can also play the same role. This is because during the federated learning training process, each client shares local data by sharing parameters, enriching local features and improving model accuracy.

(2) The effect of different length of train data. Figs. 8 to 11 show the results of using 3, 6, 9, and 12 time steps as training data and predicting the traffic flow in the next time step. The upper sub-figure of each figure is the prediction result of the last 100 time steps from February 22, 2018 to the beginning, and the lower figure is the prediction result of the last 30 time

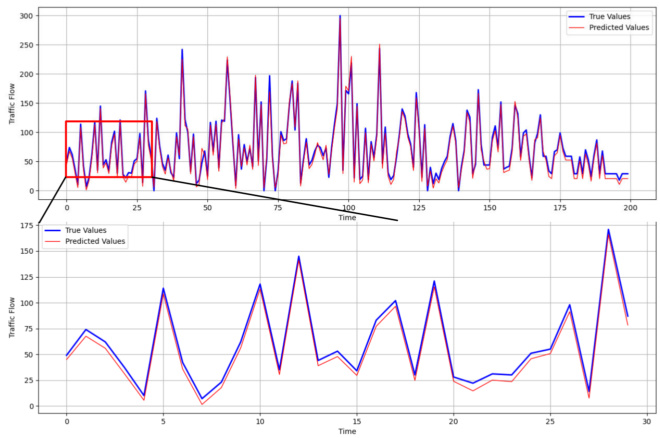


Fig. 8. The visualization results for prediction of 15 minutes.

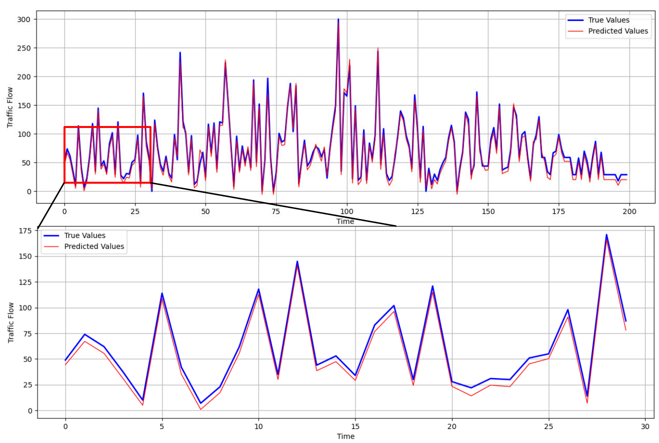


Fig. 9. The visualization results for prediction of 30 minutes.

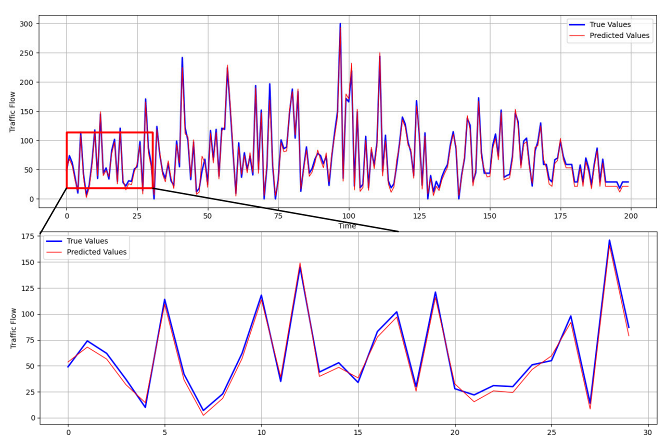


Fig. 10. The visualization results for prediction of 45 minutes.

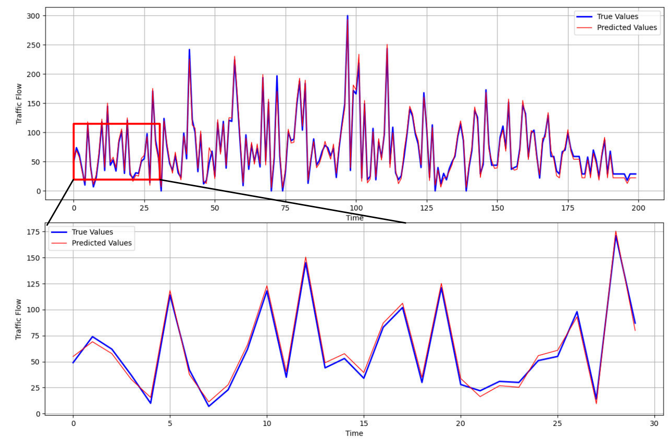
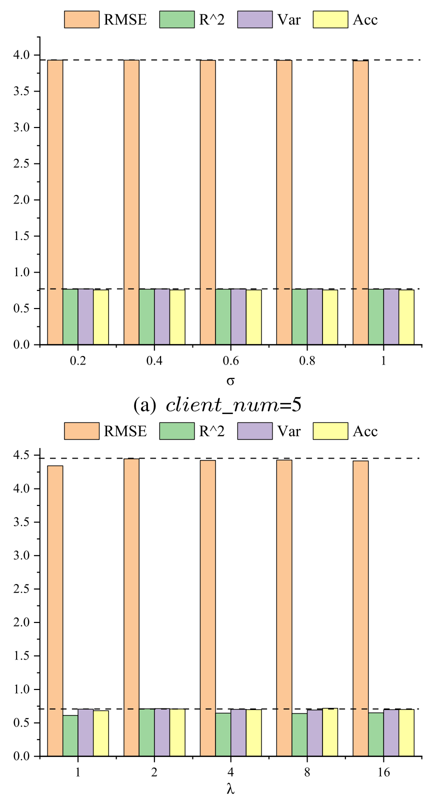


Fig. 11. The visualization results for prediction of 60 minutes.



(a) $client\_num$=5

(b) $client\_num$=10

Fig. 12. Perturbation analysis. (a) The results of adding gaussian perturbation on PEMS. (b) The results of adding poisson perturbation on PEMS.

steps from February 22, 2018. From the visualized results, we can draw the following conclusions:

- For different time step settings, the model can predict the traffic flow well. The prediction results of the model are similar to the changing trend of the real traffic flow;
- The model trained with a longer time step performs better in predicting traffic flow, which is most obvious when the time step is increased to 12, but the improvement in model performance from 6 to 3 time steps is not obvious;
- To pinpoint the inflection point in traffic flow trends, the FC-FedGCN model exhibits significant discrepancies during both peak and off-peak periods. This could be attributed to the abrupt shifts in traffic conditions influenced by elements like weather and temporal factors.

Experiment3–Perturbation Analysis and Robustness

In the real-world data collection process, noise is an unavoidable factor. To evaluate the noise resistance of the FC-FedGCN model, we conducted perturbation analysis experiments to test the model's robustness. During these experiments, we introduce two types of common random noise into the dataset: Gaussian noise following the distribution $N(0, \sigma^2)$ where σ takes values from the set {0.2, 0.4, 0.6, 0.8, 1}, and Poisson noise following the distribution $P(\lambda)$ where λ takes values from the set {1, 2, 4, 8, 16}. After adding the noise, we normalized the noise matrix values to the range [0, 1]. The results of the experiments, using various evaluation metrics, are as follows.

Fig. 12(a) illustrates the impact of adding Gaussian noise to the PEMS dataset. The horizontal axis represents the Gaussian distribution parameter σ, the vertical axis shows the variation in the evaluation metrics, and different colors represent different metrics. Fig. 12(b) similarly presents the outcomes of introducing Poisson noise to the PEMS dataset. The results indicate that the evaluation metrics exhibit minimal changes regardless of the type of noise distribution applied. Consequently, the FC-FedGCN model demonstrates robustness and is capable of effectively managing datasets with high levels of noise.

VI. Conclusion

This paper proposes the FC-FedGCN framework for traffic prediction. This proposed approach not only uses the FCA to optimize the graph convolutional network model training process and improve the model training accuracy, but also adopts the federated learning framework to prevent the leakage of original data, further improving the model training accuracy while eliminating privacy issues between users. The extensive experiments on the FC-FedGCN framework with the PEMS datasets demonstrate that the model trained by it has certain advantages in traffic prediction. In future research, we will investigate the security of the FC-FedGCN framework and optimize its security performance in combination with advanced methods.

References

[1] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1544–1561, Apr. 2022.

[2] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2021.

[3] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, "A spatial–temporal attention approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4909–4918, Aug. 2021.

[4] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.

[5] A. Boukerche and J. Wang, "Machine learning-based traffic prediction models for intelligent transportation systems," *Comput. Netw.*, vol. 181, Nov. 2020, Art. no. 107530.

[6] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: A comprehensive review," *Comput. Social Netw.*, vol. 6, no. 1, pp. 1–23, Dec. 2019.

[7] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1725–1735.

[8] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[9] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 7370–7377.

[10] F. Chen, P. Li, T. Miyazaki, and C. Wu, "FedGraph: Federated graph learning with intelligent sampling," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1775–1786, Aug. 2021.

[11] L. Li, Y. Fan, Y. K. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, Sep. 2020, Art. no. 106854.

[12] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*. Cham, Switzerland: Springer, 2012.

[13] Y. Yang, F. Hao, B. Pang, G. Min, and Y. Wu, "Dynamic maximal cliques detection and evolution management in social Internet of Things: A formal concept analysis approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1020–1032, May 2022.

[14] F. Hao, D.-S. Park, G. Min, Y.-S. Jeong, and J.-H. Park, "K-cliques mining in dynamic social networks based on triadic formal concept analysis," *Neurocomputing*, vol. 209, pp. 57–66, Oct. 2016.

[15] F. Hao, Z. Pei, and L. T. Yang, "Diversified top-k maximal clique detection in social Internet of Things," *Future Gener. Comput. Syst.*, vol. 107, pp. 408–417, Jun. 2020.

[16] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[17] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Syst.*, vol. 216, Jan. 2021, Art. no. 106775.

[18] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastruct. Deep Learn.*, Dec. 2018, pp. 1–8.

[19] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, 2003.

[20] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2024, no. 1, pp. 116–125, Jan. 2007.

[21] X. Li et al., "Prediction of urban human mobility using large-scale taxi traces and its applications," *Front. Comput. Sci.*, vol. 6, no. 1, pp. 111–121, 2012.

[22] Y. Liu, T. Feng, S. Rasouli, and M. Wong, "ST-DAGCN: A spatiotemporal dual adaptive graph convolutional network model for traffic prediction," *Neurocomputing*, vol. 601, Oct. 2024, Art. no. 128175.

[23] L. Wang, D. Guo, H. Wu, K. Li, and W. Yu, "TC-GCN: Triple cross-attention and graph convolutional network for traffic forecasting," *Inf. Fusion*, vol. 105, May 2024, Art. no. 102229.

[24] Y. Bao, J. Liu, Q. Shen, Y. Cao, W. Ding, and Q. Shi, "PKET-GCN: Prior knowledge enhanced time-varying graph convolution network for traffic flow prediction," *Inf. Sci.*, vol. 634, pp. 359–381, Jul. 2023.

[25] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, and H. Li, "AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting," *IEEE Access*, vol. 9, pp. 35973–35983, 2021.

[26] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Aug. 2019.

[27] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[28] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.

[29] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," 2019, *arXiv:1906.00121*.

[30] S. S. Sepasgozar and S. Pierre, "Fed-NTP: A federated learning algorithm for network traffic prediction in VANET," *IEEE Access*, vol. 10, pp. 119607–119616, 2022.

[31] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Gener. Comput. Syst.*, vol. 117, pp. 328–337, Apr. 2021.

[32] K. Zhang, C. Yang, X. Li, L. Sun, and S. M. Yiu, "Subgraph federated learning with missing neighbor generation," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 6671–6682.

[33] K. Hu, J. Wu, Y. Li, M. Lu, L. Weng, and M. Xia, "FedGCN: Federated learning-based graph convolutional networks for non-Euclidean spatial data," *Mathematics*, vol. 10, no. 6, p. 1000, Mar. 2022.

[34] C. Chen, Z. Xu, W. Hu, Z. Zheng, and J. Zhang, "FedGL: Federated graph learning framework with global self-supervision," *Inf. Sci.*, vol. 657, Feb. 2024, Art. no. 119976.

[35] X. Ni, X. Xu, L. Lyu, C. Meng, and W. Wang, “A vertical federated learning framework for graph convolutional network,” 2021, *arXiv:2106.11593*.

[36] S. O. Kuznetsov, “On stability of a formal concept,” *Ann. Math. Artif. Intell.*, vol. 49, nos. 1–4, pp. 101–115, Aug. 2007.

[37] J. Gao, F. Hao, Z. Pei, and G. Min, “Learning concept interestingness for identifying key structures from social networks,” *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 3220–3232, Oct. 2021.

[38] S. O. Kuznetsov and T. Makhalova, “On interestingness measures of formal concepts,” *Inf. Sci.*, vols. 442–443, pp. 202–219, May 2018.

[39] M. Klimushkin, S. Obiedkov, and C. Roth, “Approaches to the selection of relevant concepts in the case of noisy data,” in *Proc. Int. Conf. Formal Concept Anal.*, Agadir, Morocco. Cham, Switzerland: Springer, Jan. 2010, pp. 255–266.

[40] J. Liu and W. Guan, “A summary of traffic flow forecasting methods,” *J. Highway Transport. Res. Develop.*, vol. 21, no. 3, pp. 82–85, Mar. 2004.

[41] C. Xu, Z. Li, and W. Wang, “Short-term traffic flow prediction using a methodology based on autoregressive integrated moving average and genetic programming,” *Transport*, vol. 31, no. 3, pp. 343–358, 2016.

[42] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

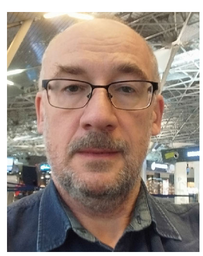
Kai Wu received the B.Sc. degree in Internet of Things engineering from Hohai University, China, in 2022. He is currently pursuing the M.Sc. degree with the School of Computer Science, Shaanxi Normal University, China. His research interests include graph learning and formal concept analysis.

Fei Hao received the Ph.D. degree in computer science and engineering from Soonchunhyang University, South Korea, in 2016. From 2020 to 2022, he was a Marie Curie Fellow with the University of Exeter, Exeter, U.K. He is currently a Professor with the School of Computer Science, Shaanxi Normal University, Xi’an, China. He has published more than 150 papers in the leading international journals and conference proceedings, such as IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Services Computing, IEEE Transactions on Network Science and Engineering, *IEEE Communications Magazine*, IEEE Internet Computing, *ACM Transactions on Multimedia Computing, Communications, and Applications*, ACM SIGIR, and IEEE GLOBECOM. His research interests include social computing, soft computing, big data analytics, pervasive computing, and data mining. He is also a member of ACM, CCF, and KIPS. In addition, he was a recipient of the Best Paper Award from IEEE GreenCom 2013. He was a recipient of the Outstanding Service Award at IEEE DSS 2018, the IEEE SmartData 2017, the IEEE Outstanding Leadership Award at IEEE CPSCom 2013, and the 2015 Chinese Government Award for Outstanding Self-Financed Students Abroad.

Ruoxia Yao received the Ph.D. degree in computer science and technology from East China Normal University, China, in 2005. From April 2010 to May 2011, she was a Visiting Scholar with the University of Minnesota, USA. She is currently a Professor with the School of Computer Science, Shaanxi Normal University, China. She has published more than 50 papers in international conferences and journals, such as *Physica D: Nonlinear Phenomena*, *Chaos, Solitons and Fractals*, *Communications in Nonlinear Science and Numerical Simulation*, *Physics Letters A*, and *Chinese Physics Letters*. Her research interests include integrable systems, symbolic computing, and deep learning.

Jinhai Li received the M.Sc. degree in science from Guangxi University, Nanning, China, in 2009, and the Ph.D. degree in science from Xi’an Jiaotong University, Xi’an, China, in 2012. He is currently a Professor with Kunming University of Science and Technology, Kunming, China. In recent years, he has published more than 40 articles in IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Systems, Man, and Cybernetics-Systems, IEEE Transactions on Big Data, IEEE Transactions on Emerging Topics in Computational Intelligence, and *Pattern Recognition*. His research interests include big data, cognitive computing, granular computing, and formal concept analysis. He is also an Area Editor of *International Journal of Approximate Reasoning* and an Associate Editor of *International Journal of Machine Learning and Cybernetics*.

Geyong Min (Member, IEEE) received the B.Sc. degree in computer science from the Huazhong University of Science and Technology, China, in 1995, and the Ph.D. degree in computing science from the University of Glasgow, U.K., in 2003. He is currently a Professor of high performance computing and networking with the Department of Mathematics and Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, U.K. His research interests include next generation internet, wireless communications, multimedia systems, information security, ubiquitous computing, modeling, and performance engineering.



Sergei O. Kuznetsov received the degree from Moscow Institute of Physics and Technology. He defended Doctor of Science thesis on “Machine Learning Models Based on Concept Lattices” in 2002 at the Computing Center, Russian Academy of Science. He is currently a Full Professor with HSE University, Moscow, being the Head of the School for Data Analysis and Artificial Intelligence, International Laboratory for Intelligent Systems, and an Academic Supervisor of the data science master program.