Knowledge and Information Systems
https://doi.org/10.1007/s10115-025-02393-7

RESEARCH

FedTPS: traffic pattern sharing for personalized federated traffic flow prediction

Hang Zhou[1] · Wentao Yu[1] · Sheng Wan[1] · Yongxin Tong[2] · Tianlong Gu[3] · Chen Gong[1]

Received: 27 December 2024 / Revised: 16 February 2025 / Accepted: 7 March 2025

Abstract
Traffic flow prediction plays a critical role in ensuring the efficiency of transportation systems, which has motivated extensive research into capturing spatial-temporal dependencies within road networks. However, most existing approaches depend on centralized data, potentially raising privacy concerns as traffic data is often managed by different traffic administration departments and restricted from distribution. To address this issue, federated learning (FL) allows collaborative model training without exchanging raw data. Nevertheless, traditional FL methods are designed to optimize a model that performs well globally, making them inadequate for handling the naturally non-independent and identically distributed traffic data across different regions. To overcome this limitation, we propose a new framework termed "*personalized Federated learning with Traffic Pattern Sharing*" (FedTPS), which exploits the sharing of underlying common traffic patterns across regions while preserving region-specific characteristics in a personalized manner. Specifically, discrete wavelet transform is employed to decompose the traffic data and extract low-frequency components in each client that reflect stable traffic dynamics. The clients then learn representative traffic patterns from these stable traffic dynamics and store them in traffic pattern repositories. Afterward, these repositories

✉ Sheng Wan
wansheng315@hotmail.com

✉ Chen Gong
chen.gong@njust.edu.cn

Hang Zhou
hangzhou@njust.edu.cn

Wentao Yu
wentao.yu@njust.edu.cn

Yongxin Tong
yxtong@buaa.edu.cn

Tianlong Gu
gutianlong@jnu.edu.cn

1 School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

2 SKLSDE, Beihang University, Beijing, China

3 Engineering Research Center of Trustworthy AI (Ministry of Education), Jinan University, Guangzhou, China

Published online: 09 April 2025

are shared with a central server, which enables the identification and integration of common traffic patterns to improve global learning. Meanwhile, the model components capturing spatial-temporal dependencies are retained for local training, ensuring adaptation to region-specific data. Intensive experiments on four real-world traffic datasets firmly demonstrate the superiority of our proposed FedTPS over traditional FL methods across various estimation errors.

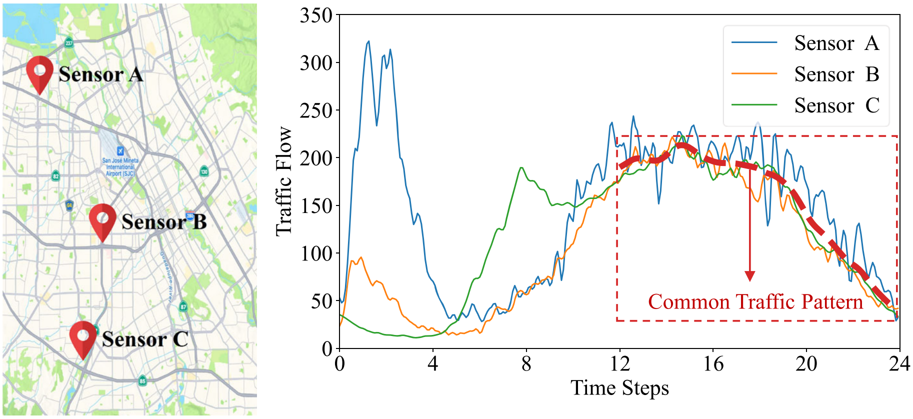
Keywords Spatial-temporal data · Traffic flow prediction · Personalized federated learning · Graph neural network

1 Introduction

Accurate and real-time traffic flow prediction (TFP) plays a pivotal role in enhancing urban management by enabling efficient traffic control, reducing congestion, and optimizing travel routes [34]. The primary objective of TFP is to estimate future traffic conditions by uncovering the spatial-temporal dependencies based on historical traffic data and relevant features [28].

Traditional TFP methods [14, 29, 31] simply apply statistical time series models to traffic forecasting tasks. However, these approaches often rely on the assumption of stationarity, which limits their ability to capture the complex and dynamic relationships in traffic data. With the advancements in deep learning, researchers have explored combining convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to model the spatial-temporal dependencies among different traffic routes [20, 38, 46]. Nevertheless, the non-Euclidean nature of traffic networks inherently conflicts with the grid-like structures assumed by CNNs, which limits their ability to capture the spatial dependencies of road networks effectively. To address this limitation, recent studies have introduced graph neural networks (GNNs) to TFP tasks [26, 47], leveraging graph structures to accurately represent the spatial dependencies between roads. However, these methods typically rely on pre-defined graphs, which may be incomplete or biased. In response, emerging research has shifted toward adaptively learning graph structures [2, 16, 43], enabling more precise characterization of the complex interactions inherent in real-world traffic dynamics. Although these methods achieve promising performance, their training relies on centralized traffic data collected from different regions. In practice, traffic data is often collected and managed by different traffic administration departments based on the zoning of the city, province, or state. Since traffic data may contain sensitive information, such as individual locations and travel trajectories [22], centralizing these data will probably raise privacy concerns, thereby limiting the practical applicability of these approaches.

Federated learning (FL), which adopts a distributed training strategy without exchanging raw data, provides a natural solution to the aforementioned privacy issues. In FL, model training is conducted locally on clients (i.e., traffic administration departments in all regions in our problem) and only model parameters, rather than raw data, are uploaded to a central server for collaborative learning, which helps ensure data privacy [35, 48]. Up to now, various efforts have been made to achieve accurate TFP using the FL framework [32, 36, 42, 50]. However, traffic data recorded by sensors across different locations and time stamps can exhibit significant variability. For instance, traffic sensors in high-density urban areas may capture entirely different traffic flow when compared with that in suburban or rural regions. This non-independent and identically distributed (non-IID) nature of traffic data



(a) Sensor location

(b) Traffic flow recorded by different sensors

Fig. 1 The observation on PEMS04 dataset. Traffic flow recorded by sensors from different regions (with locations shown in (**a**)) may share common traffic patterns (indicated by the red dashed line in (**b**)) (Colour figure online)

from different clients can result in unstable training, slow convergence, and degradation in the performance of the global model in FL [25].

To address this issue, personalized federated learning (PFL) has emerged as an effective approach. Unlike traditional FL, which aims to develop a single global model for all clients, PFL methods focus on customizing models to the specific data distributions of each individual client. This personalization enables PFL to capture the unique characteristics of local traffic data while still benefiting from collaborative learning [24, 30, 49]. Although these methods enhance model personalization to some extent, they ignore the underlying common knowledge shared across different regions, which is actually critical in collaborative model training. To be specific, due to similar functional characteristics of different regions (e.g., commercial and residential areas) or consistent travel behaviors during certain periods (e.g., morning and evening rush hours), traffic data from different regions may share certain common traffic patterns with similar temporal characteristics [21] despite the data heterogeneity caused by unique local factors. Although these common traffic patterns may fluctuate due to varying traffic conditions across regions, they generally exhibit stable traffic dynamics. For instance, as shown in Fig. 1, sensors A, B, and C are located in distinct regions, but the traffic flows they record exhibit similar stable traffic dynamics during certain periods. This observation inspires the sharing of common traffic patterns within the FL framework for performance enhancement.

To effectively explore and utilize common traffic patterns across different regions, in this work, we propose *personalized* ***Fed****erated learning with* ***T****raffic* ***P****attern* ***S****haring* (FedTPS), a novel federated framework for TFP. Our objective is to improve local performance by leveraging the sharing of common traffic patterns across different regions while preserving the region-specific data characteristics in a personalized manner. To be specific, we employ discrete wavelet transform (DWT) to decompose the traffic data in each client, isolating low-frequency components that capture stable traffic dynamics. Furthermore, we design a parameterized traffic pattern repository for each client to learn and store representative traffic patterns from the stable traffic dynamics. In the aggregation phase of FL, the traffic pattern repositories from different clients are aggregated on the server side to obtain common traffic patterns, which facilitates the sharing of global knowledge and enhances collaborative model training. Meanwhile, the model components capturing spatial-temporal dependencies of traffic data are retained in each client for local training to preserve region-specific characteristics. We have conducted intensive experiments on four popular TFP datasets in the FL scenario, which demonstrates the superiority of FedTPS against multiple baseline methods.

This paper is an extended version of our previous conference paper [54], where FedTPS was first introduced. Compared with the initial version [54], we provide more algorithmic details by explaining the extraction of traffic patterns, elaborating on why DWT is effective in capturing stable traffic dynamics, and detailing how the traffic pattern repository is integrated with the overall framework. Besides, we analyze the resource overhead of different methods to demonstrate the efficiency of our proposed FedTPS. Furthermore, we present additional experimental results on the visualizations of matched traffic patterns. These results are critical for a deeper understanding of the common traffic patterns learned by FedTPS.

2 Related work

Our approach primarily focuses on the TFP task in FL scenarios. In this section, we will review the related works in these areas that are relevant to our research.

2.1 Traffic flow prediction

As transportation systems grow in complexity and scale, efficient traffic management becomes crucial for alleviating traffic congestion and planning travel routes. TFP is one of the critical tasks in traffic management that focuses on forecasting traffic volumes at specific times and locations within a traffic network. In the early stages, some studies simply applied statistical methods for time series models to TFP tasks, such as historical average (HA) [14], Kalman filter (KF) [31], and auto-regressive integrated moving average (ARIMA) [29]. However, these methods typically assume linearity in traffic data, which is inadequate for handling the complex dependencies inherent in traffic data. To capture nonlinear correlations in traffic data, many deep learning-based time series models, such as RNN [38], temporal convolutional network (TCN) [39], and Transformer [53], have been applied to TFP with the advancement of deep learning technologies. These models have shown great power in handling complex temporal dependencies, thereby enhancing prediction accuracy. Unlike general time series prediction, TFP not only focuses on temporal dependencies but also needs to account for spatial dependencies of different road segments. To capture the spatial dependencies, some studies have combined CNN with time series models to achieve improved performance [20, 38, 46]. However, CNN-based approaches typically treat road segments as grid-based data, thereby overlooking the complex non-Euclidean structures inherent in traffic road networks.

In recent years, researchers have increasingly focused on leveraging the strengths of GNNs to capture the complex spatial dependencies of traffic data. These methods integrate GNNs with time series models to enhance TFP. For example, DCRNN [26] models the dynamics of traffic flow as diffusion processes and introduces diffusion convolutional operations to capture spatial dependencies. Besides, STGCN [47] combines graph convolutional network (GCN) with TCN to capture comprehensive spatial-temporal correlations through modeling multi-scale traffic networks. However, these methods are typically based on pre-defined graphs, which limits their ability to fully capture the latent dynamic relationships between road segments. To further explore the dynamism of traffic networks, Graph WaveNet [43] adaptively learns a normalized adjacency matrix to capture the spatial dependencies. Building upon this, AGCRN [2] incorporates node-specific adaptive graph convolutional layers, enabling the model to capture unique spatial interactions for each node. Besides, StemGNN [3] employs spectral graph convolution to adaptively capture the spatial dependencies among nodes in the traffic network, while MegaCRN [16] learns node-level prototypes in the meta-node bank for updating the auxiliary graph adaptively. Additionally, some methods employ attention mechanisms to capture time-varying spatial dependencies among traffic roads. For instance, GMAN [52] utilizes graph attention network (GAT) and temporal attention to model the relationships between historical and future time stamps. Meanwhile, ASTGNN [13] develops a dynamic graph convolution module, which employs self-attention to capture the spatial correlations in a dynamic manner. STWave [9] disentangles traffic data into trends and events and applies a sampling strategy-based GAT to achieve accurate forecasts with reduced computational costs.

However, in practice, traffic data from different regions are often managed by different traffic administration departments. Since these data may contain sensitive information, such as travel trajectories and vehicle identification numbers, sharing them across regions is often prohibited due to privacy concerns. This reality renders most existing research efforts for TFP, which rely on centralized training data, impractical for real-world scenarios.

2.2 Federated learning

Most deep learning models are deployed on the central server, requiring training data to be uploaded to the server for model training. This process poses a risk of exposing sensitive information contained in the data. FL is a distributed machine learning paradigm that enables collaboratively training models across decentralized devices or clients, where data remains local. In FL, clients train models locally and share only model parameters, rather than raw data, ensuring data privacy and addressing the concerns associated with centralized data storage and processing. The traditional FL method FedAvg [35] aggregates model weights sent from local clients on the server and downloads the aggregated model back to the clients for further training. This iterative process continues until a satisfactory global model is obtained. Although FedAvg is efficient and scalable, it assumes that data across different clients are independently and identically distributed (IID), which is rarely met in real-world scenarios. The presence of non-IID data, where data distributions vary significantly between clients, can lead to challenges such as performance degradation in the global model. To deal with this problem, FedProx [23] proposes a regularization term aimed at minimizing the discrepancy between local models and the global model, thereby preventing local models from deviating too far from their local training data. From the perspective of global aggregation, FedAtt [15] enables flexible aggregation by adaptively assigning weights according to the contribution of each local model to the global model. Besides, FedFed [45] shares the performance-sensitive features to mitigate data heterogeneity while keeping the performance-robust features locally. Unlike traditional FL methods, which train a global model for all clients, clustered FL approaches group similar clients together for aggregation, thereby reducing the impact of heterogeneity [7, 10]. For example, FedGroup [6] utilizes the Euclidean distance of decomposed cosine similarity to cluster clients into multiple groups, which effectively reduces the divergence between clients. Different from the above methods, PFL proposes to train a personalized model suitable for the local data of each client. This approach effectively addresses the issue of data heterogeneity in federated scenarios and has demonstrated promising performance in recent years [5]. Some PFL methods divide the model into shared modules that participate in federated aggregation and private modules that are retained locally [1, 27, 37]. For example, FedPer [1] shares the feature extractor as a common base layer while retaining a private classifier to preserve local knowledge. Besides, PerFedAvg [8] employs model-agnostic meta-learning to train a meta-model that generates initial local models for each client, upon which the clients train personalized models with local data. Additionally, pFedMe [40] utilizes Moreau envelopes to decouple the optimization of personalized models from global model learning. FedALA [51] captures the desired information in the global model for client models through adaptive local aggregation.

Due to the capability to preserve data privacy, FL has recently been applied to spatial-temporal forecasting tasks in various studies. For example, FedGRU integrates FL with gated recurrent units and adopts an ensemble clustering-based approach to capture the correlations in traffic data. Furthermore, CNFGNN [36] models the temporal dependencies on the client side, while capturing the spatial dependencies among clients through the GNN on the server, where alternating optimization is employed to reduce communication costs. Spatial-temporal data inherently exhibits temporal and spatial diversity, leading to data heterogeneity across different clients. To address this issue, some studies have employed PFL to enhance model performance. For example, FedDA [49] employs a dual attention mechanism to construct the global model by aggregating both intra-cluster and inter-cluster models, rather than simply averaging the weights of local models. Besides, FML-ST [24] constructs a global spatial-temporal graph based on meta-learning, where clients customize their models by evaluating the differences between the global graph and local graphs. Additionally, FUELS [30] adaptively aligns positive and negative pairs based on semantic similarity and incorporates auxiliary contrastive tasks to inject detailed spatial-temporal heterogeneity into the latent representation space. To exploit the spatial relationships across clients, FedGTP [44] performs adaptive learning of inter-client spatial dependencies.

However, the aforementioned methods fail to effectively leverage the underlying common knowledge (e.g., common traffic patterns) within spatial-temporal data across different regions, and thus their performance could be limited. To address this issue, our FedTPS extracts common traffic patterns from traffic data to capture global knowledge while retaining personalized models.

3 Problem description

This section provides a formal definition of the setting for the federated TFP problem investigated in this study. The traffic road network of a city can be represented as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of nodes, each corresponding to a traffic sensor that records traffic data, and $\mathcal{E}$ represents the set of edges, which indicate the roads connecting these sensors. Additionally, $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ represents the weighted adjacency matrix depicting the proximity (e.g., geographical distance, causal dependencies, or traffic series similarity) between nodes. The notation $| \cdot |$ denotes the cardinality of a set.

In practice, traffic sensors in different regions of a city are often managed by distinct traffic administration departments. Suppose there are M traffic administration departments, each governing one of the M regions. Then the m-th ($m = 1, 2, \ldots, M$) region is associated with a subset of sensors $\mathcal{V}_m$, forming a subgraph of the global traffic network $\mathcal{G}_m \subseteq \mathcal{G}$ along with its corresponding private dataset $\mathcal{D}_m = \{\mathbf{X}_1, \ldots, \mathbf{X}_t, \ldots, \mathbf{X}_T\}$, where $\mathbf{X}_t \in \mathbb{R}^{|\mathcal{V}_m| \times d}$ represents the observed d-dimensional features recorded by the sensors within the local traffic network

$\mathcal{G}_m$ at time stamp t, and T represents the total number of time stamps. The objective is to accurately predict the traffic flow at the locations of these sensors.

Most existing methods rely on centralized data collection, which is impractical due to privacy restrictions on sharing traffic data across departments. In response, we propose using FL to collaboratively train TFP models without exchanging private data. In federated TFP, each traffic administration department can be considered as a client that trains a TFP model to capture the spatial-temporal dependencies of traffic roads from historical traffic data recorded by local sensors and make accurate predictions of future traffic flow. To be specific, the task for the m-th client is to train a model $f_{\mathbf{W}_m}$ parameterized by $\mathbf{W}_m$ such that it can predict the traffic flow for the future T_2 time stamps based on the historical T_1 time stamps, namely

$$\mathbf{X}_{t-T_1+1}, \ldots, \mathbf{X}_t \xrightarrow{f_{\mathbf{W}_m}} \mathbf{X}_{t+1}, \ldots, \mathbf{X}_{t+T_2}. \tag{1}$$

The objective of federated TPF is to collaboratively train TFP models across multiple clients without compromising data privacy. The classic FL method, i.e., FedAvg [35], aggregates model parameters at the server after local training according to the following formula:

$$\overline{\mathbf{W}} \leftarrow \sum_{m=1}^{M} \frac{|\mathcal{V}_m|}{|\mathcal{V}|} \mathbf{W}_m. \tag{2}$$

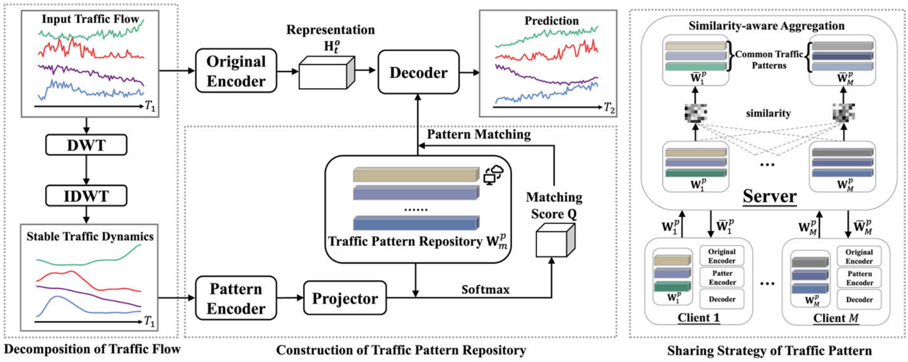
After aggregation, the global model is redistributed to clients for the subsequent training round. However, due to the non-IID traffic data across different regions, this approach to training a global TFP model for all clients often results in suboptimal performance. To address this issue, PFL is implemented by training a customized model for each client to enhance the performance on local traffic data. The objective of PFL can then be formulated as

$$\min_{\mathbf{W}_1,\ldots,\mathbf{W}_M} \frac{1}{M} \sum_{m=1}^{M} \frac{|\mathcal{V}_m|}{|\mathcal{V}|} \mathcal{L}_m \left(\mathbf{W}_m, \mathcal{D}_m\right), \tag{3}$$

where $\mathcal{L}_m$ is the loss function of the m-th client.

4 Methodology

This section provides a detailed explanation of the proposed FedTPS framework (see Fig. 2). During the local training phase (see Fig. 2a), the model first decomposes the traffic flow to extract stable traffic dynamics. Subsequently, the original traffic data and the obtained stable traffic dynamics are fed into different encoders to obtain their corresponding representations. Afterward, the representations of stable traffic dynamics are projected into a query space to calculate the matching scores with the traffic patterns in the traffic pattern repository. Based on the matching scores, the matched pattern is computed as a weighted sum of the patterns in the repository. Finally, the representation of original traffic data is concatenated with the matched traffic pattern and passed into the decoder to forecast future traffic flow. During this process, the model of each client learns representative traffic patterns from the stable traffic dynamics and stores them in the traffic pattern repository. During the global aggregation phase (see Fig. 2b), the traffic pattern repository is uploaded to the server, where similarity-aware aggregation is performed. By aligning and aggregating the traffic pattern repositories from different clients, common traffic patterns are derived and redistributed to each client, facilitating the sharing of global knowledge among clients. At the same time, the components



(a) The local training phase (b) The global aggregation phase

Fig. 2 The framework of FedTPS. During the local training phase, stable traffic dynamics are extracted through the decomposition of traffic flow and are further utilized to construct the traffic pattern repository consisting of representative traffic patterns on each client. During the global aggregation phase, each client uploads the traffic pattern repository to the server and shares the repository with other clients via similarity-aware aggregation to derive common traffic patterns

that learn region-specific characteristics are retained locally as personalized modules. Next, we detail the critical steps of FedTPS by presenting the graph convolutional recurrent unit (GCRU) (see Sect. 4.1), explaining the extraction of traffic patterns (see Sect. 4.2), and describing the sharing strategy of traffic patterns (see Sect. 4.3).

4.1 Adaptive graph convolutional recurrent unit

The inherent graph structure of traffic networks makes them highly compatible with models that integrate GCN and GRU, enabling the simultaneous exploration of spatial and temporal dependencies in traffic data [3, 26]. Based on this foundation, some methods [2, 16, 43] have introduced adaptive adjacency matrices to model the dynamic spatial correlations within traffic networks. Following these prior works, our local TFP model employs an encoder-decoder architecture composed of GCRUs with an adaptive adjacency matrix, which can be represented as

$$\mathbf{u}_t = \sigma(\mathrm{Gconv}_u(\mathbf{X}_t, \mathbf{H}_{t-1}, \tilde{\mathbf{A}})), \tag{4}$$

$$\mathbf{r}_t = \sigma(\mathrm{Gconv}_r(\mathbf{X}_t, \mathbf{H}_{t-1}, \tilde{\mathbf{A}})), \tag{5}$$

$$\mathbf{C}_t = \tanh(\mathrm{Gconv}_C(\mathbf{X}_t, (\mathbf{r}_t \odot \mathbf{H}_{t-1}), \tilde{\mathbf{A}}), \tag{6}$$

$$\mathbf{H}_t = \mathbf{u}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{u}_t) \odot \mathbf{C}_t, \tag{7}$$

where $\tilde{\mathbf{A}} = \mathrm{softmax}\left(\mathrm{ReLU}(\mathbf{E}\mathbf{E}^\top)\right) \in \mathbb{R}^{|\mathcal{V}_m| \times |\mathcal{V}_m|}$ denotes the adaptive adjacency matrix, obtained based on learnable parameter $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}_m| \times e}$. The notation $\mathrm{Gconv}(\mathbf{X}_t, \mathbf{H}_{t-1}, \tilde{\mathbf{A}})$ denotes the graph convolution operation over the current input $\mathbf{X}_t$ and the previous hidden states $\mathbf{H}_{t-1} \in \mathbb{R}^{|\mathcal{V}_m| \times h}$ where h denotes the dimensionality of the hidden state of each node. Here, the update gate, reset gate, and candidate state of GRU at time t are indicated by $\mathbf{u}_t$, $\mathbf{r}_t$, and $\mathbf{C}_t$, respectively, while the corresponding graph convolution operations are, respectively denoted as Gconv_u, Gconv_r, and Gconv_C. The notation $\sigma(\cdot)$ represents an activation function, such as the sigmoid function used in this paper, and $\odot$ represents element-wise product. Note that all the mathematical notations related to the m-th ($m = 1, 2, \ldots, M$) client above and

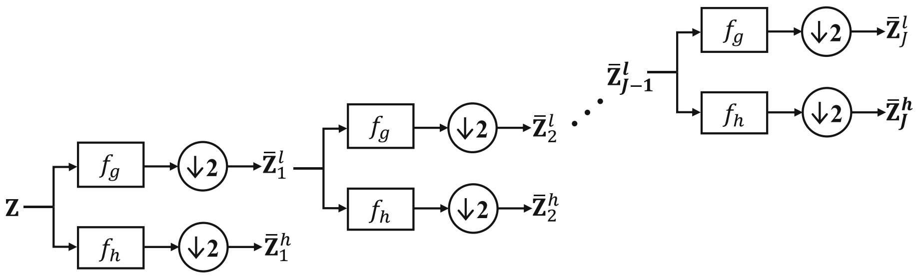


Fig. 3 Illustration of J-level DWT

hereinafter should be accompanied by the subscript m. However, to simplify the notation, we omit the subscript m if no notational confusion is incurred.

4.2 Extraction of traffic patterns

Since common traffic patterns generally exhibit stable dynamics, we apply DWT to decompose the traffic flow on each client and extract the low-frequency components that represent these stable traffic dynamics. These stable dynamics are then utilized to construct a traffic pattern repository during the local training phase. This repository serves as a collection of representative traffic patterns that can be shared within the FL framework to derive common traffic patterns for clients, thereby enhancing the collaborative learning process. The detailed process will be outlined in the following part.

4.2.1 Decomposition of traffic flow

Since traffic flow is primarily influenced by human activities, traffic data from different regions may share common traffic patterns [21]. Arising from similar functions of areas or consistent travel behaviors, these patterns manifest as stable traffic dynamics. However, most existing federated TFP methods [24, 30, 49] overlook the common traffic patterns across different regions. In fact, these patterns represent underlying global knowledge that can enhance TFP by capturing predictable and recurring traffic behaviors over time and across locations. To bridge this gap, we propose extracting stable traffic dynamics from traffic data across different regions. This approach facilitates clients in effectively learning representative traffic patterns, which can be further utilized during aggregation to derive common traffic patterns for clients.

To achieve this goal, we utilize DWT [9] to decompose the traffic data into waveforms of different frequencies. The low-frequency component, which corresponds to stable traffic dynamics, is expected to contain the common traffic patterns across regions. By isolating these stable dynamics, we aim to capture the traffic patterns that can be shared among clients for effective TFP. To be specific, given traffic data $\mathbf{Z} = [\mathbf{X}_{t-T_1+1}; \mathbf{X}_{t-T_1+2}; \ldots; \mathbf{X}_t] \in \mathbb{R}^{T_1 \times |\mathcal{V}_m| \times d}$, the J-level DWT is performed to obtain the low-frequency component $\bar{\mathbf{Z}}_j^l$ and high-frequency component $\bar{\mathbf{Z}}_j^h$ at the j-th level, namely

$$\bar{\mathbf{Z}}_j^l = (\downarrow 2)(f_g \star \bar{\mathbf{Z}}_{j-1}^l), \tag{8}$$

$$\bar{\mathbf{Z}}_j^h = (\downarrow 2)(f_h \star \bar{\mathbf{Z}}_{j-1}^l), \tag{9}$$

where f_g and f_h represent the low-pass and high-pass filters of a 1D orthogonal wavelet, respectively. The symbol $\star$ denotes the convolution operation and $(\downarrow 2)$ represents naive down-sampling halving the length of each component. The process of J-level DWT is illustrated in Fig. 3. To reduce the computation overhead, we only employ one-level DWT, which can be performed as a preprocessing step prior to training. As a result, DWT will not compromise the efficiency and scalability of our method. The low-frequency component, which represents stable traffic dynamics, is then transformed back to the time domain through inverse DWT (IDWT), which reaches

$$\mathbf{Z}^l = f_g^{-1} \star (\uparrow 2)\bar{\mathbf{Z}}_1^l, \tag{10}$$

where f_g^{-1} is the inverse low-pass filter and $(\uparrow 2)$ denotes the naive up-sampling operation doubling the length of each component. These low-frequency components capture the periodicity and trends that remain relatively stable despite short-term fluctuations, which makes them ideal for identifying common traffic patterns. In contrast, high-frequency components are more sensitive to region-specific, short-term fluctuations such as traffic incidents, weather conditions, or sudden surges in traffic volume. By isolating the low-frequency components, it becomes possible to focus on the more stable and generalizable aspects of traffic dynamics, which are crucial for the extraction of representative traffic patterns.

After decomposition, the original traffic time series $\mathbf{Z}$, which contains high-frequency components, is fed into the original encoder to obtain the learned representations $\mathbf{H}_t^o$. On the other hand, the low-frequency component $\mathbf{Z}^l$ is passed through a separate encoder, yielding the learned representation of the stable traffic dynamics $\mathbf{H}_t^l$.

4.2.2 Construction of traffic pattern repository

After learning the representations of stable traffic dynamics, we aim to leverage and share the global knowledge contained within them across different clients in the FL framework. However, since these stable traffic dynamics are derived by decomposing the raw data of clients, directly sharing them may raise privacy concerns. More importantly, rather than sharing all the stable dynamics, we aim to share only the common part of the stable traffic dynamic. To achieve this, we further encode these stable traffic dynamics through learnable parameters to obtain traffic patterns. Considering the observed variations in traffic patterns across different traffic road networks [21], our goal is to learn a set of representative traffic patterns for each client. These representative patterns capture the essential and generalized traffic dynamics of regions, minimizing privacy concerns while still enabling the sharing of common patterns for model training.

Due to the efficient ability to store and retrieve information, memory networks have achieved notable success in fields such as computer vision [41] and anomaly detection [11, 17]. Their utility has recently extended to spatial-temporal data analysis [16, 21, 33] to learn representative information. Drawing inspiration from memory networks, we construct a learnable traffic pattern repository $\mathbf{W}^p \in \mathbb{R}^{N \times c}$, where N and c denote the number of the representative traffic patterns and the dimension of each pattern, respectively. To learn representative traffic patterns from the data, we first project the stable traffic dynamics representation $\mathbf{H}_t^l$ to a query space, which can be formulated as

$$\mathbf{H}_t^q = \mathbf{H}_t^l \mathbf{W}^q. \tag{11}$$

where $\mathbf{W}^q \in \mathbb{R}^{h \times c}$ is a learnable parameter matrix. The query matrix $\mathbf{H}_t^q \in \mathbb{R}^{|\mathcal{V}_m| \times c}$ can be further used to identify relevant patterns from the traffic pattern repository. Subsequently, we compute the matching scores between the query matrix and the patterns stored in the repository. These scores are determined by the similarity between the query and the stored patterns, which can be represented as

$$\mathbf{Q} = \mathrm{softmax}\left(\mathbf{H}_t^q \mathbf{W}^{p\top}\right), \tag{12}$$

where $\mathbf{Q} \in \mathbb{R}^{|\mathcal{V}_m| \times N}$ is the matching score matrix. This operation allows us to quantify the relevance of each stored pattern to the current query. Subsequently, we calculate the matched traffic patterns $\mathbf{P}_t \in \mathbb{R}^{|\mathcal{V}_m| \times c}$ as a weighted sum of the patterns in $\mathbf{W}^p$ based on the computed matching scores, and obtain

$$\mathbf{P}_t = \mathbf{Q}\mathbf{W}^p. \tag{13}$$

Finally, we concatenate the matched patterns $\mathbf{P}_t$ with the representations of the original traffic data $\mathbf{H}_t^o$ and feed them into the decoder to obtain predictions $\mathbf{Z}' = [\mathbf{X}_{t+1}; \mathbf{X}_{t+2}; \ldots; \mathbf{X}_{t+T_2}] \in \mathbb{R}^{T_2 \times |\mathcal{V}_m| \times d}$, where ℓ_1 loss function is adopted to optimize the training process. The learnable parameters at the m-th client are denoted by $\mathbf{W}_m^{e_1}$, $\mathbf{W}_m^{e_2}$, $\mathbf{W}_m^d$, $\mathbf{W}_m^q$, and $\mathbf{W}_m^p$, where $\mathbf{W}_m^{e_1}$ and $\mathbf{W}_m^{e_2}$ refer to the parameters of the original encoder and the pattern encoder, respectively. Besides, $\mathbf{W}_m^d$ refers to the parameters of the decoder, $\mathbf{W}_m^q$ refers to the parameters of the projector, and $\mathbf{W}_m^p$ refers to the learnable traffic pattern repository, which stores the representative traffic patterns, and some of them can be shared across clients.

4.3 Sharing strategy of traffic pattern

In the FL framework, the server aggregates the models uploaded by clients and distributes the aggregated model back to the clients for the next round of training. In the proposed FedTPS, the model on the m-th client can be divided into two parts: the traffic pattern repository $\mathbf{W}_m^p$ that stores the representative traffic patterns and other modules (i.e., $\mathbf{W}_m^{e_1}$, $\mathbf{W}_m^{e_2}$, $\mathbf{W}_m^d$, and $\mathbf{W}_m^q$) that learn the spatial-temporal dependencies of local traffic data. Building upon the constructed traffic pattern repository, the core idea of FedTPS is to aggregate representative traffic patterns from clients to derive common traffic patterns and share global knowledge, while preserving personalized local learning. This strategy ensures that clients benefit from the global knowledge of common traffic patterns while simultaneously refining personalized models adapted to their unique local traffic conditions.

Furthermore, different from the widely adopted averaging aggregation [35], we propose a similarity-aware aggregation strategy to enhance the alignment of traffic patterns across different clients during the aggregation process. In particular, let $\mathbf{W}_m^p[i]$ denote the i-th traffic pattern in the repository of the m-th client. The server computes the cosine similarity between $\mathbf{W}_m^p[i]$ and the patterns from repositories of other clients. Afterward, the server selects and aggregates the top-k most similar patterns from each client based on the cosine similarity, where k is a hyperparameter fixed throughout the entire training process. The aggregation process can be expressed as

$$\overline{\mathbf{W}}_m^p[i] \leftarrow \frac{1}{M} \sum_{n=1}^{M} \frac{1}{k} \sum_{j \in \mathcal{S}_k} \mathbf{W}_n^p[j], \tag{14}$$

where $\mathcal{S}_k$ indicates the set of k indices of the representative patterns in $\mathbf{W}_n^p$ that are most similar to $\mathbf{W}_m^p[i]$. Afterward, the server redistributes the aggregated traffic pattern repository to each client for the subsequent round of local training.

Algorithm 1 FedTPS on the client side

Input: Historical traffic flow $\mathbf{Z}$ from private dataset $\mathcal{D}_m$; number of local rounds R_1; traffic pattern repository $\overline{\mathbf{W}}_m^p$.
Output: Prediction of future traffic flow $\mathbf{Z}'$.

Download traffic pattern repository $\overline{\mathbf{W}}_m^p$ from the server;
Update the traffic pattern repository $\mathbf{W}_m^p \leftarrow \overline{\mathbf{W}}_m^p$;
for each local rounds $r = 1, 2, \ldots, R_1$ **do**
    Compute low-frequency component $\mathbf{Z}^l$ via Eqs. (8) and (10);
    Compute the representations $\mathbf{H}_t^o$ and $\mathbf{H}_t^l$ via Eqs. (4)–(7);
    Compute the matched pattern $\mathbf{P}_t$ via Eqs. (11)–(13);
    Concate $\mathbf{H}_t^o$ and $\mathbf{P}_t$, and predict future traffic flow $\mathbf{Z}'$ through the decoder;
    Update learnable parameters $\mathbf{W}_m^{e1}$, $\mathbf{W}_m^{e2}$, $\mathbf{W}_m^d$, $\mathbf{W}_m^q$, and $\mathbf{W}_m^p$ via gradient optimization;
end for
Upload $\mathbf{W}_m^p$ to the server.

Algorithm 2 FedTPS on the server side

Input: Number of clients M; number of communication rounds R_2; number of selected patterns k; the traffic pattern repository from each client.
Output: Traffic pattern repository $\overline{\mathbf{W}}_m^p$ for client m.

Initialize the global traffic pattern repository $\overline{\mathbf{W}}^{p(1)}$;
for each communication round $r = 1, 2, \ldots, R_2$ **do**
    for client $m \in \{1, 2, \cdots, M\}$ **in parallel do**
        if $r = 1$ **then**
            Send $\overline{\mathbf{W}}^{p(1)}$ to client m;
        else
            $\overline{\mathbf{W}}_m^{p(r)} \leftarrow$ aggregate $\mathbf{W}_{1:M}^{p(r)}$ via Eq. (14);
            Send $\overline{\mathbf{W}}_m^{p(r)}$ to client m;
        end if
        Perform Algorithm 1 on client m;
    end for
end for

Through iterative training and aggregation of traffic pattern repositories, common traffic patterns provide additional global knowledge that enhances the TFP process. This enables the clients to gain a generalized understanding of traffic dynamics across different regions. At the same time, the other components of the model, which focus on learning region-specific spatial-temporal dependencies, are excluded from the aggregation process. This strategy enables clients to learn personalized models, allowing the model to benefit from common traffic patterns while learning regional characteristics. Moreover, it helps mitigate the negative impact of regional discrepancies in the FL framework.

In summary, our proposed method of traffic pattern extraction and sharing strategy enables the FL framework to leverage these common patterns to enhance TFP while maintaining model personalization for each client. By integrating global knowledge from common traffic patterns with region-specific characteristics learned from personalized models, our approach enhances the ability of the model to predict traffic dynamics for clients of different regions. The detailed implementation of our FedTPS framework for the client side and the server side is provided in Algorithms 1 and 2, respectively.

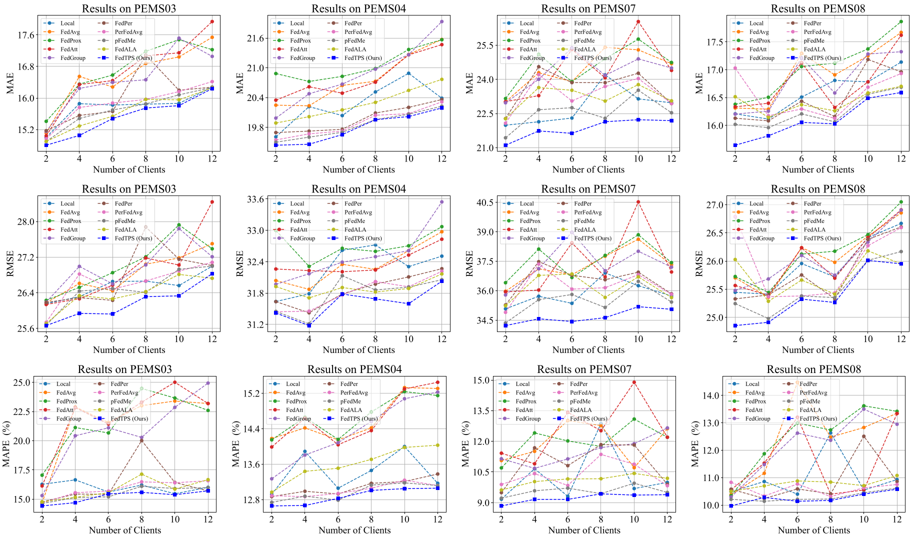


Fig. 4 The performance of various methods on four datasets, with varied client numbers

Table 1 Dataset statistics

Datasets	# Samples	# Nodes	Sample rate (mins)	Time span
PEMS03	26,208	358	5	09/2018–11/2018
PEMS04	16,992	307	5	01/2018–02/2018
PEMS07	28,224	883	5	05/2017–08/2017
PEMS08	17,856	170	5	07/2016–08/2016

5 Experiments

To evaluate the effectiveness of our proposed model, we perform a series of comparative experiments on four real-world highway traffic datasets in FL scenarios. In this section, we first introduce the experimental settings, including details on the datasets, evaluation metrics, baseline methods, and implementation. Subsequently, we present the comprehensive experimental results, including performance comparison, resource overhead analysis, ablation study, parametric sensitivity analysis, and case study.

5.1 Experimental setup

5.1.1 Datasets description and preprocessing

We evaluate the effectiveness of our proposed FedTPS on four widely used datasets for TFP, including *PEMS03*, *PEMS04*, *PEMS07*, and *PEMS08*. These datasets contain traffic flow information gathered by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS) [4], where the numeric identifiers correspond to the district code. The overview of the statistical details of these datasets is listed in Table 1.

In line with the practice of previous methods [12], we divide the datasets into training set, validation set, and test set in chronological order with the ratio of 6 : 2 : 2. For each of the four datasets, we use the traffic data from the past 12 time stamps to predict the traffic flow for the upcoming 12 time stamps. Before training, we apply a standard normalization procedure to the datasets to ensure a stable training process. To simulate the FL scenario, we employ the graph partitioning algorithm, i.e., METIS [18] to evenly partition the global traffic road network, with each client assigned a subgraph of the global traffic road network. This partitioning restricts each client to localized data, simulating a realistic decentralized learning environment.

5.1.2 Evaluation metrics

In this paper, we use three widely adopted evaluation metrics to assess the performance of different methods in the TFP task: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE), which are defined as follows:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}\left|\mathbf{X}_t - \hat{\mathbf{X}}_t\right|, \tag{15}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{X}_t - \hat{\mathbf{X}}_t\right)^2}, \tag{16}$$

$$MAPE = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{\mathbf{X}_t - \hat{\mathbf{X}}_t}{\mathbf{X}_t}\right|, \tag{17}$$

where $\mathbf{X}_t$ denotes the ground truth of all nodes at time stamp t and $\hat{\mathbf{X}}_t$ denotes the prediction value. We evaluate the performance of the TFP task on the client side, and then average the performances across all clients.

5.1.3 Baseline methods

Unlike previous node-level federated TFP methods [32, 36], where each sensor is treated as an independent client, our approach targets subgraph-level federated TFP tasks. In this setting, each client possesses a subset of sensors, representing a region of the global traffic network. To ensure a fair comparison, we evaluate our method against the following nine baseline methods:

- *Local* This baseline method represents a scenario where all clients train their models independently without any sharing of model parameters. Each client updates its model based solely on its own local data, and there is no communication between clients during the training process.
- *FedAvg* [35] FedAvg is the classical FL algorithm that aggregates the locally updated models by applying a simple averaging strategy. After each round of local training, the averaged model is redistributed to all clients for further updates.
- *FedProx* [23] FedProx is a FL algorithm that incorporates a proximal term in the local training objective. This term prevents the local models from deviating too far from the global model, which ensures convergence and stability in heterogeneous data settings.

- *FedAtt* [15] FedAtt is a FL algorithm that employs an attention mechanism to weigh the aggregation of local model parameters. By assigning different importance to local models, it enables a flexible model aggregation.
- *FedGroup* [6] FedGroup is a FL framework that groups clients based on the similarity between their parameter updates. It employs a data-driven measure to efficiently cluster clients, mitigating the concentration phenomenon in high-dimensional data clustering.
- *FedPer* [1] FedPer is a PFL algorithm that shares common base layers across clients while keeping personalized layers locally. This strategy allows clients to benefit from global knowledge while maintaining model customizations.
- *PerFedAvg* [8] PerFedAvg is a PFL algorithm where an initial model is trained in a federated manner and subsequently fine-tuned to adapt to the local data of each client.
- *pFedMe* [40] pFedMe is a PFL algorithm that utilizes the global model to optimize personalized models for each client by updating personalized models through a meta-learning approach.
- *FedALA* [51] FedALA is a PFL algorithm that adaptively aggregates the global and local models to align with the local objective of each client. It seeks to strike a balance between global knowledge and local adaptation by adjusting the aggregation process.

5.1.4 Implementation details

In our encoder-decoder architecture, both the encoder and decoder modules are designed with 64 GCRUs. To ensure a fair comparison, all baseline methods adopt the same encoder-decoder architecture as the local model. For the traffic pattern repository, we set the dimension of each traffic pattern c to 64 and the size of the traffic pattern repository N to 40 for *PEMS07* and 20 for *PEMS03*, *PEMS04*, and *PEMS08*, respectively. The number of selected patterns k in the aggregation process is set to 2. We employ the Adam optimizer [19] with a learning rate of 0.001 and the batch size is set to 128. The local training epochs and global communication rounds are fixed to 1 and 200, respectively, for all FL methods. The default number of clients is set to 4. All methods are implemented using Python 3.8.8 and PyTorch 1.9.1, with all experiments conducted on one GeForce RTX 3090 GPU.

5.2 Performance comparison

We present the comparison of the prediction performance between FedTPS and nine baseline methods across four datasets. As shown in Table 2, our proposed FedTPS outperforms the baseline methods in most cases. This improvement can be attributed to the ability of FedTPS to effectively share common traffic patterns across different clients, which facilitates collaborative model training while minimizing the adverse effects of regional discrepancies in traffic data. Furthermore, when compared with Local, conventional FL methods (i.e., FedAvg, FedProx, and FedAtt) show a noticeable performance decline, which may be attributed to the heterogeneity of traffic data across different clients. Since these conventional methods train a global model for all clients, they are insufficient in capturing region-specific traffic characteristics, leading to performance degradation. Differently, PFL methods train customized models for each client, allowing them to achieve better performance by learning the unique traffic characteristics for each region. This highlights the importance of personalization in FL, particularly for tasks like TFP, where data are collected at varying times and locations, resulting in significant data heterogeneity.

Table 2 Overall performance on four datasets

Method	*PEMS03*			*PEMS04*			*PEMS07*			*PEMS08*		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
Local	15.86	26.31	16.65	20.22	31.79	13.89	22.14	35.72	10.66	16.11	25.41	10.86
FedAvg [35]	16.55	26.61	22.90	20.23	31.87	14.42	24.29	37.12	11.51	16.29	25.36	11.16
FedProx [23]	16.35	26.52	21.13	20.73	32.31	14.66	25.10	38.12	12.41	16.51	25.44	11.87
FedAtt [15]	16.34	26.27	22.84	20.62	32.23	14.64	23.29	36.04	10.90	16.40	25.39	11.53
FedGroup [6]	16.24	26.99	20.43	20.48	32.17	13.80	24.01	37.12	10.68	16.26	25.68	11.48
FedPer [1]	15.56	26.29	15.43	19.72	31.42	12.99	24.56	37.48	11.68	16.08	25.40	10.24
PerFedAvg [8]	15.76	26.82	15.55	19.67	31.46	12.87	24.21	37.36	10.42	16.17	25.37	10.33
pFedMe [40]	15.48	26.44	15.13	19.60	31.21	12.88	22.67	35.55	9.58	15.96	24.98	**10.14**
FedALA [51]	15.29	26.34	15.16	20.02	31.71	13.44	23.64	36.78	10.03	16.14	25.29	10.70
FedTPS	**15.05**	**25.94**	**14.70**	**19.46**	**31.18**	**12.67**	**21.74**	**34.57**	**9.16**	**15.81**	**24.91**	10.28

The best results are highlighted in boldface

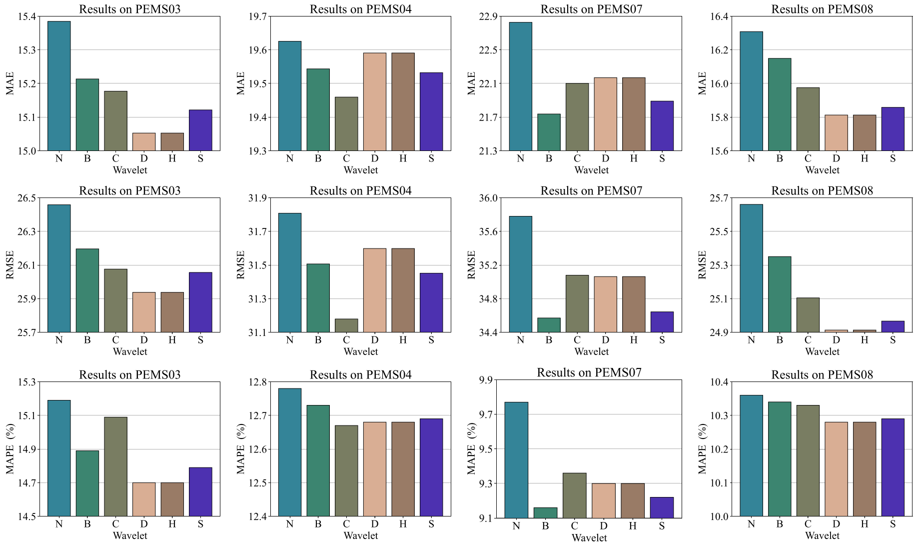


Fig. 5 Effect of DWT on four datasets. N, Not using DWT; B, Biorthogonal; C, Coiflets; D, Daubechies; H, Haar; S, Symlets

We further investigate the impact of varying numbers of clients on the performance of FedTPS. The prediction performance of different methods in FL framework with different numbers of clients across four datasets is presented in Fig. 4. It can be observed that, as the number of clients increases, the performance of most methods tends to degrade. Since the data amount of the dataset is constant, increasing the number of clients leads to the lack of local data and the loss of correlation information between traffic roads, thereby hindering the training of local models. Despite this, FedTPS usually demonstrates strong performance by leveraging the common traffic patterns across different clients to enhance model training. This highlights the effectiveness of our proposed method under the FL frameworks, even as the number of clients increases.

5.3 Resource overhead

In addition to model prediction performance, resource overhead is a critical factor for practical deployment, particularly in FL scenarios where client resources may be constrained. We measure the computation overhead (i.e., the training time per communication round) and communication overhead (i.e., the parameters transmitted per communication round) of different methods on the *PEMS08* dataset. As shown in Table 3, FedTPS requires 0.81 min per communication round, with the primary additional computation overhead arising from local updates for the pattern encoder and the traffic pattern repository. Compared with traditional FL methods, FedTPS only incurs an additional 0.05–0.25 min, resulting in a significant improvement in prediction performance. While perFedAvg and pFedMe also demonstrate good performance, they require additional training steps to personalize the local model, making training time longer than FedTPS.

Regarding communication overhead, although there are differences in training and aggregation strategies, most methods upload and download all model parameters during each communication round, resulting in communication overhead consistent with FedAvg. In

Table 3 Comparison of the computation overhead and the communication overhead of different methods on *PEMS08* dataset

Method	Computation (time/round) (min)	Communication (Param./round) (KB)
Local	0.55	–
FedAvg [35]	0.56	2364
FedProx [23]	0.69	2364
FedAtt [15]	0.76	2364
FedGroup [6]	0.60	2364
FedPer [1]	0.59	1174
PerFedAvg [8]	1.16	2364
pFedMe [40]	3.29	2364
FedALA [51]	0.68	2364
FedTPS	0.81	40

contrast, FedPer transmits the common base layers of the model and retains the personalized layers locally, therefore reducing communication overhead. FedTPS only uploads and downloads the traffic pattern repository, which contains the common traffic patterns, rather than all model parameters during each communication round. This reduces communication overhead substantially while maintaining encouraging performance.

5.4 Ablation study

Our proposed FedTPS integrates global knowledge sharing and local model personalization to enhance model performance. As described in Sect. 4, FedTPS leverages DWT decomposition to extract stable traffic dynamics, which are then used to learn representative traffic patterns. Meanwhile, clients retain the modules that learn spatial-temporal dependencies locally, which ensures personalized model adaptation while sharing only traffic pattern repositories for collaborative learning. To illustrate the contributions of these two modules, we conduct a series of ablation studies.

5.4.1 Effectiveness of DWT decomposition

To evaluate the effectiveness of our proposed DWT decomposition, we conduct a series of ablation studies across four datasets comparing the performance of the models with or without DWT. In the variant without DWT, the model directly feeds the original traffic data into the pattern encoder without performing any wavelet decomposition. Additionally, we investigate the impact of different wavelet bases in the variants with DWT, including Biorthogonal, Coiflets, Daubechies, Haar, and Symlets. As illustrated in Fig. 5, compared with the variant without DWT, applying DWT to decompose the traffic data enhances the performance across all datasets. This improvement can be attributed to the ability of DWT to isolate stable traffic dynamics, which facilitates the identification of common traffic patterns effectively. The server aggregates the traffic pattern repositories from the clients to derive common traffic patterns, thereby providing global knowledge to enhance model training. Furthermore, we observe that the choice of wavelet bases influences the model performance across the datasets. Specifically, Daubechies and Haar wavelets demonstrate the best performance for

PEMS03 and *PEMS08* datasets, Coiflets wavelets yield the optimal results for *PEMS04*, and Biorthogonal wavelets are most effective for *PEMS07*.

5.4.2 Effectiveness of traffic pattern sharing strategy

To evaluate the contribution of the proposed traffic pattern sharing strategy with the similarity-aware aggregation, we conduct a comparative analysis between FedTPS and its variants. These variants differ in terms of which components of the local model are shared across clients, where the same aggregation method as FedAvg [35] is adopted. As shown in Table 4, on most datasets, the aggregation strategy that shares encoder-decoder parameters (i.e., "ED" in Table 4) results in a performance degradation compared with the strategy that does not share parameters (i.e., "None" in Table 4). This suggests that directly sharing model parameters of clients across different regions can introduce the interference of region-specific characteristics from other clients, which disrupts the model ability to effectively learn from local data. Although the variant that shares all parameters (i.e., "All" in Table 4) can somewhat mitigate the performance decline with the help of common traffic patterns, it is still inevitably influenced by discrepancies of different regions. This indicates that a direct sharing of all model parameters across clients, without distinguishing between common and region-specific knowledge, is insufficient to overcome the heterogeneity of traffic data from different regions.

Differently, the strategy that shares traffic pattern repositories (i.e., "PR" in Table 4) demonstrates good performance since it only aggregates the traffic pattern repositories to benefit from common traffic patterns, while keeping other model components locally. This allows clients to retain region-specific knowledge in a personalized manner, which enhances the overall model ability to adapt to local variations in traffic dynamics while still benefiting from the global knowledge contained in common traffic patterns. Furthermore, unlike the model variants using averaging-based aggregation (i.e., "None", "All", "ED", and "PR" in Table 4), our proposed FedTPS that employs similarity-aware aggregation outperforms the others. By aligning the traffic patterns from different regions based on their similarity, FedTPS effectively reduces the negative effects of regional discrepancies. This alignment improves the model ability to integrate and share knowledge, thereby showing improved performance.

5.5 Parametric sensitivity

In the proposed FedTPS framework, there are two key hyperparameters that require manual tuning, i.e., the size of the traffic pattern repository N, and the number of selected patterns k during the aggregation process. In this section, we analyze in detail the impact of these hyperparameters on model performance.

The effect of varying the size of the traffic pattern repository N is shown in Fig. 6. It can be observed that the optimal value of N is closely related to the number of traffic sensors in the datasets. Specifically, datasets with a large number of sensors tend to benefit from a large size repository of traffic patterns. For instance, in the *PEMS07* dataset, which contains a relatively high number of sensors, the performance improves as the size of the traffic pattern repository N increases. This is due to the presence of a great variety of traffic patterns in large-scale datasets, which necessitates a large repository to effectively learn and store the representative traffic patterns. Conversely, for small datasets with fewer sensors, such as *PEMS08* dataset, a large repository size is unnecessary, as a small set of traffic patterns is sufficient to capture the traffic pattern.

Table 4 Comparative analysis of different sharing strategies

Shared Component	*PEMS03*			*PEMS04*			*PEMS07*			*PEMS08*		
	MAE	RMSE	MAPE/%	MAE	RMSE	MAPE/%	MAE	RMSE	MAPE/%	MAE	RMSE	MAPE/%
None	15.29	26.30	15.04	19.65	31.69	12.78	23.54	37.07	9.94	15.90	25.06	10.56
All	15.24	26.18	15.22	20.19	31.98	13.29	23.33	36.28	10.47	15.98	24.92	10.72
ED	15.38	26.46	15.19	20.55	32.48	14.11	24.37	37.11	11.67	16.28	25.28	11.44
PR	15.13	26.33	14.87	19.59	31.60	12.68	22.61	35.67	9.63	15.87	24.97	10.33
FedTPS	**15.05**	**25.94**	**14.70**	**19.46**	**31.18**	**12.67**	**21.74**	**34.57**	**9.16**	**15.81**	**24.91**	**10.28**

The best results are highlighted in boldface

None, Nonparameter sharing; All, Sharing all parameters; ED, Sharing encoder–decoder parameters; PR, Sharing the traffic pattern repository

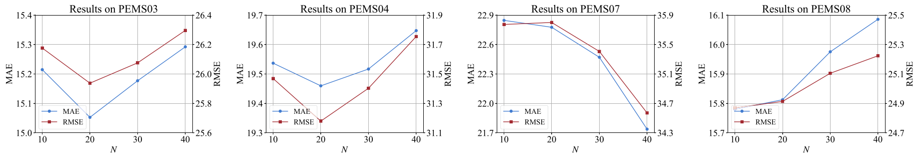


Fig. 6 Sensitivity analysis of the pattern number N in different datasets

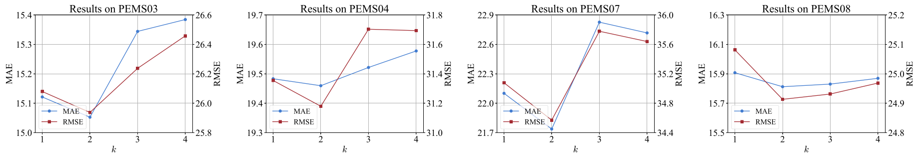


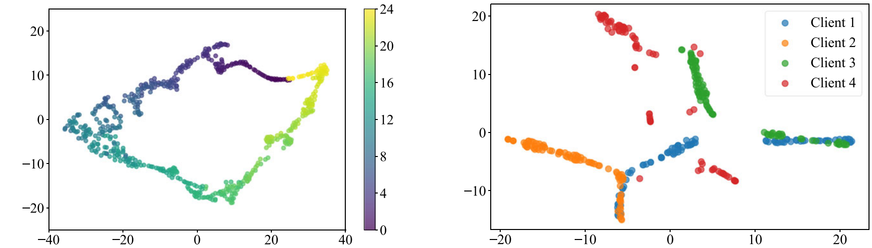
Fig. 7 Sensitivity analysis of the number of selected patterns k during aggregation in different datasets

The impact of varying k on model performance is shown in Fig. 7. We observe that our model achieves the best performance across all datasets when the number of selected patterns k is set to 2. When k is too small, FedTPS may fail to capture sufficient common patterns, limiting effective knowledge sharing between clients. On the other hand, when k is set too large, the aggregation process may lead to misalignment in shared knowledge and suboptimal model performance.

5.6 Case study

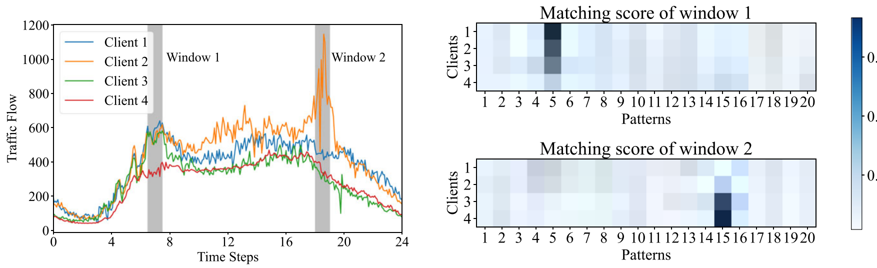
To further explore the interpretability and effectiveness of the common traffic patterns learned by FedTPS, we analyze the matched traffic patterns across different time stamps and locations on *PEMS08* dataset. First, we use t-distributed Stochastic Neighbor Embedding (t-SNE) to visualize the matched traffic patterns for traffic data at different times of the day in Fig. 8a. We observe that the matched traffic patterns exhibit clear daily periodicity. This aligns well with consistent travel behaviors, indicating that FedTPS can effectively learn regular temporal characteristics of common traffic patterns from historical traffic data. Secondly, we visualize the matched traffic patterns of traffic data recorded by sensors from different clients in Fig. 8b. Although these sensors are located in different regions, some of them tend to cluster together. This is due to similar temporal characteristics present in the traffic flow of different regions. FedTPS can effectively leverage the traffic pattern repository to share such global knowledge in the form of common traffic patterns across different clients.

To provide a detailed illustration, we visualize the traffic flow recorded by sensors from different clients, along with their corresponding matching scores of traffic pattern repository in Fig. 9. For time window 1 (i.e., the period from 6:30 am to 7:30 am), the traffic flows recorded by the sensors from clients 1, 2, and 3 exhibit consistent trends, resulting in similar matching scores with the traffic pattern repository. The traffic flow recorded by the sensor from client 4 shows noticeable differences when compared with other traffic flow records, which is also reflected in its matching scores. Likewise, for time window 2 (i.e., the period from 6:00 pm to 7:00 pm), the traffic flows recorded by the sensors from clients 3 and 4 display similar characteristics, and their matching scores also align closely, whereas the matching scores of clients 1 and 2 differ from the others. These visualizations provide further insight into the interpretability of FedTPS and demonstrate its ability to share and utilize common traffic patterns effectively across clients.



(a) The matched traffic patterns at different times of the day
(b) The matched traffic patterns from different clients

Fig. 8 The visualization of matched traffic patterns for traffic data across different time stamps and locations on *PEMS08* dataset



(a) The traffic flow recorded by sensors from different clients
(b) The matching scores of sensors from different clients

Fig. 9 The visualization of traffic flow and matching scores from different clients on *PEMS08* dataset

6 Conclusion

In this paper, we introduce FedTPS, a novel PFL framework designed to address the challenge of data heterogeneity in federated TFP. Different from previous works that overlook the underlying global knowledge represented by common traffic patterns across regions, the proposed FedTPS decomposes traffic data to extract stable traffic dynamics for learning representative traffic patterns. Additionally, by incorporating the similarity-aware aggregation strategy, our framework enables clients to leverage common traffic patterns from different regions, which enhances global knowledge sharing while preserving local spatial-temporal dependencies to maintain region-specific characteristics. This balance between collaborative learning and personalization allows FedTPS to effectively mitigate the adverse effects of heterogeneous data. Intensive experiments conducted on four widely-used TFP datasets confirm the effectiveness and superiority of our FedTPS over multiple baseline methods.

Author Contributions HZ wrote the main manuscript text. All authors reviewed the manuscript.

Funding Chen Gong was supported by NSF of China (Nos: 62336003, 12371510), and NSF for Distinguished Young Scholar of Jiangsu Province (No: BK20220080). Sheng Wan was supported by Postdoctoral Fellowship Program of CPSF (No: GZC20233503), China Postdoctoral Science Foundation (Nos: 2023M741708, 2023TQ0159), and NSF of Jiangsu Province (No: BK20241469). Tianlong Gu was supported by NSF of China (No: U22A2099).

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Arivazhagan MG, Aggarwal V, Singh AK, et al (2019) Federated learning with personalization layers. arXiv preprint arXiv:1912.00818
2. Bai L, Yao L, Li C, et al (2020) Adaptive graph convolutional recurrent network for traffic forecasting. In: Advances in neural information processing systems, pp 17804–17815
3. Cao D, Wang Y, Duan J, et al (2020) Spectral temporal graph neural network for multivariate time-series forecasting. In: Advances in neural information processing systems, pp 17766–17778
4. Chen C (2002) Freeway performance measurement system (PEMS). PhD thesis, University of California, Berkeley
5. Collins L, Hassani H, Mokhtari A, et al (2021) Exploiting shared representations for personalized federated learning. In: International conference on machine learning, PMLR, pp 2089–2099
6. Duan M, Liu D, Ji X, et al (2021) Fedgroup: efficient federated learning via decomposed similarity-based clustering. In: 2021 IEEE international conference on parallel & distributed processing with applications, big data & cloud computing, sustainable computing & communications, social computing & networking, pp 228–237
7. Duan M, Liu D, Ji X et al (2022) Flexible clustered federated learning for client-level data distribution shift. IEEE Trans Parallel Distrib Syst 33(11):2661–2674
8. Fallah A, Mokhtari A, Ozdaglar A (2020) Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach. In: Advances in neural information processing systems, pp 3557–3568
9. Fang Y, Qin Y, Luo H, et al (2023) When spatio-temporal meet wavelets: disentangled traffic forecasting via efficient spectral graph attention networks. In: 2023 IEEE 39th international conference on data engineering, IEEE, pp 517–529
10. Ghosh A, Chung J, Yin D et al (2022) An efficient framework for clustered federated learning. IEEE Trans Inf Theory 68(12):8076–8091
11. Gong D, Liu L, Le V, et al (2019) Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1705–1714
12. Guo S, Lin Y, Feng N, et al (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI conference on artificial intelligence, pp 922–929
13. Guo S, Lin Y, Wan H et al (2021) Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Trans Knowl Data Eng 34(11):5415–5428
14. Hamilton JD (2020) Time series analysis. Princeton University Press, Princeton
15. Ji S, Pan S, Long G, et al (2019) Learning private neural language modeling with attentive aggregation. In: 2019 International joint conference on neural networks, IEEE, pp 1–8
16. Jiang R, Wang Z, Yong J, et al (2023) Spatio-temporal meta-graph learning for traffic forecasting. In: Proceedings of the AAAI conference on artificial intelligence, pp 8078–8086
17. Jiang T, Chen W, Zhou H et al (2024) Towards semi-supervised classification of abnormal spectrum signals based on deep learning. Chin J Electron 33(3):721–731
18. Karypis G (1997) Metis: unstructured graph partitioning and sparse matrix ordering system. Technical report
19. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: International conference on learning representations
20. Lai G, Chang WC, Yang Y, et al (2018) Modeling long- and short-term temporal patterns with deep neural networks. In: The 41st international ACM SIGIR conference on research & development in information retrieval, pp 95–104
21. Lee H, Jin S, Chu H, et al (2021) Learning to remember patterns: pattern matching memory networks for traffic forecasting. In: International conference on learning representations
22. Li L, Liu J, Cheng L et al (2018) Creditcoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. IEEE Trans Intell Transp Syst 19(7):2204–2220
23. Li T, Sahu AK, Zaheer M, et al (2020) Federated optimization in heterogeneous networks. In: Proceedings of machine learning and systems, pp 429–450

24. Li W, Wang S (2022) Federated meta-learning for spatial-temporal prediction. Neural Comput Appl 34(13):10355–10374
25. Li X, Jiang M, Zhang X, et al (2020) FedBN: federated learning on non-IID features via local batch normalization. In: International conference on learning representations
26. Li Y, Yu R, Shahabi C, et al (2018) Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: International conference on learning representations
27. Liang PP, Liu T, Ziyin L, et al (2020) Think locally, act globally: federated learning with local and global representations. arXiv preprint arXiv:2001.01523
28. Lin C, Han G, Du J et al (2020) Spatiotemporal congestion-aware path planning toward intelligent transportation systems in software-defined smart city IoT. IEEE Internet Things J 7(9):8012–8024
29. Lippi M, Bertini M, Frasconi P (2013) Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning. IEEE Trans Intell Transp Syst 14(2):871–882
30. Liu Q, Sun S, Liang Y, et al (2024) Personalized federated learning for spatio-temporal forecasting: A dual semantic alignment-based contrastive approach. arXiv preprint arXiv:2404.03702
31. Liu W, Zheng Y, Chawla S, et al (2011) Discovering spatio-temporal causal interactions in traffic data streams. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1010–1018
32. Liu Y, James J, Kang J et al (2020) Privacy-preserving traffic flow prediction: a federated learning approach. IEEE Internet Things J 7(8):7751–7763
33. Liu Z, Zheng G, Yu Y (2023) Cross-city few-shot traffic forecasting via traffic pattern bank. In: Proceedings of the 32nd ACM international conference on information and knowledge management, pp 1451–1460
34. Lv Y, Duan Y, Kang W et al (2014) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16(2):865–873
35. McMahan B, Moore E, Ramage D, et al (2017) Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics, PMLR, pp 1273–1282
36. Meng C, Rambhatla S, Liu Y (2021) Cross-node federated graph neural network for spatio-temporal data modeling. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery and data mining, pp 1202–1211
37. Peterson D, Kanani P, Marathe VJ (2019) Private federated learning with domain adaptation. arXiv preprint arXiv:1912.06733
38. Shi X, Chen Z, Wang H, et al (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems
39. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems
40. T Dinh C, Tran N, Nguyen J (2020) Personalized federated learning with Moreau envelopes. In: Advances in neural information processing systems, pp 21394–21405
41. Vinyals O, Blundell C, Lillicrap T, et al (2016) Matching networks for one shot learning. In: Advances in neural information processing systems
42. Wang H, Zhang R, Cheng X, et al (2022) Federated spatio-temporal traffic flow prediction based on graph convolutional network. In: 2022 14th International conference on wireless communications and signal processing, IEEE, pp 221–225
43. Wu Z, Pan S, Long G, et al (2019) Graph wavenet for deep spatial-temporal graph modeling. In: Proceedings of the 28th international joint conference on artificial intelligence, pp 1907–1913
44. Yang L, Chen W, He X, et al (2024) FedGTP: exploiting inter-client spatial dependency in federated graph-based traffic prediction. In: Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining, pp 6105–6116
45. Yang Z, Zhang Y, Zheng Y, et al (2024) Fedfed: feature distillation against data heterogeneity in federated learning. In: Advances in neural information processing systems
46. Yao H, Wu F, Ke J, et al (2018) Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of the AAAI conference on artificial intelligence
47. Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the 27th international joint conference on artificial intelligence, pp 3634–3640
48. Yue S, Deng Y, Wang G et al (2024) Federated offline reinforcement learning with proximal policy evaluation. Chin J Electron 33(6):1–13
49. Zhang C, Dang S, Shihada B, et al (2021) Dual attention-based federated learning for wireless traffic prediction. In: IEEE INFOCOM 2021-IEEE conference on computer communications, IEEE, pp 1–10
50. Zhang C, Zhang S, James J et al (2021) FASTGNN: a topological information protected federated learning approach for traffic speed forecasting. IEEE Trans Ind Inf 17(12):8464–8474

51. Zhang J, Hua Y, Wang H, et al (2023) FedALA: adaptive local aggregation for personalized federated learning. In: Proceedings of the AAAI conference on artificial intelligence, pp 11237–11244
52. Zheng C, Fan X, Wang C, et al (2020) GMAN: a graph multi-attention network for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence, pp 1234–1241
53. Zhou H, Zhang S, Peng J, et al (2021) Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI conference on artificial intelligence, pp 11106–11115
54. Zhou H, Yu W, Wan S, et al (2024) Traffic pattern sharing for federated traffic flow prediction with personalization. In: 2024 IEEE international conference on data mining

Hang Zhou is currently pursuing the MS degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He is affiliated with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Security. His research interests include federated learning and spatial -temporal prediction.

Wentao Yu received his BS degree from Chengdu University of Information Technology (CUIT), Chengdu, China, in 2018 and his MS degree from Xidian University, Xi'an, China, in 2021. He is currently pursuing a PhD degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology (NJUST), Nanjing, China. He is affiliated with the PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Security. His supervisor is Professor Chen Gong. His research focuses on federated learning and graph representation learning. Currently, he is also a visiting PhD student at Hong Kong Baptist University under the supervision of Professor Bo Han. He has published research papers in top-tier conferences such as AAAI and has received notable recognition, including the "ICDM-2024 Best Student Paper Runner-Up Award".

Sheng Wan received the PhD degree in Computer Science and Technology from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2023. He is currently an associate professor with the School of Artificial Intelligence, Nanjing Agricultural University, Nanjing, China. His research interests include graph machine learning, weakly supervised learning, and hyperspectral image processing. He has published more than ten papers in top-tier conferences such as NeurIPS and AAAI, and prominent journals such as IEEE T-NNLS, and IEEE T-GRS.

Yongxin Tong received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology in 2014. He is currently a professor in the School of Computer Science and Engineering, Beihang University. His research interests include federated learning, spatiotemporal data analytics, crowdsourcing, and reinforcement learning. He received the championship of KDD Cup 2020 RL track and VLDB 2014 Excellent Demonstration Award. He is a member of the IEEE.

Tianlong Gu (Senior Member, IEEE) received the PhD degree in automatic control from Zhejiang University, Hangzhou, China, in 1996. From 1998 to 2002, he was a postdoctoral fellow with the School of Electrical and Computer Engineering, Curtin University, Perth, Australia, and a research fellow with the School of Engineering, Murdoch University, Perth, Australia. He is currently a professor and the director of the Engineering Research Center of Trustworthy AI (Ministry of Education), Jinan University, Guangzhou, China. His research interests include trustworthy artificial intelligence, artificial intelligence ethics, data governance, and formal methods.

Chen Gong received his dual doctoral degree from Shanghai Jiao Tong University (SJTU) and University of Technology Sydney (UTS), respectively. Currently, he is a full professor of Shanghai Jiao Tong University. His research interests mainly include machine learning, data mining, and learning-based vision problems. He has published more than 130 technical papers at prominent journals and conferences such as JMLR, IEEE T-PAMI, IEEE T-NNLS, IEEE T-IP, IEEE T-CYB, ICML, NeurIPS, ICLR, CVPR, ICCV, AAAI, IJCAI, and ICDM. He serves as the associate editor for IEEE T-CSVT, Neural Networks, and NePL, and also the Area Chair or Senior PC member of several top-tier conferences such as AAAI, IJCAI, ICML, ICLR, ECML-PKDD, AISTATS, ICDM, and ACM MM. He won the “Excellent Doctorial Dissertation Award” of Chinese Association for Artificial Intelligence, “Young Elite Scientists Sponsorship Program” of China Association for Science and Technology, “Wu Wen-Jun AI Excellent Youth Scholar Award”, and the Scientific Fund for Distinguished Young Scholars of Jiangsu Province. He was also selected as the “Global Top Chinese Young Scholars in AI” released by Baidu, and "World’s Top 2% Scientists" released by Stanford University.