Traffic Pattern Sharing for Federated Traffic Flow Prediction with Personalization

Hang Zhou[†‡§], Wentao Yu[†‡§], Sheng Wan[†‡§*], Yongxin Tong[¶], Tianlong Gu[‖] and Chen Gong[†‡§*]
[†]School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[‡]Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, China
[§]Jiangsu Key Laboratory of Image and Video Understanding for Social Security, China
[¶]SKLSDE, Beihang University, China
[‖]Engineering Research Center of Trustworthy AI (Ministry of Education), Jinan University, China
wansheng315@hotmail.com chen.gong@njust.edu.cn

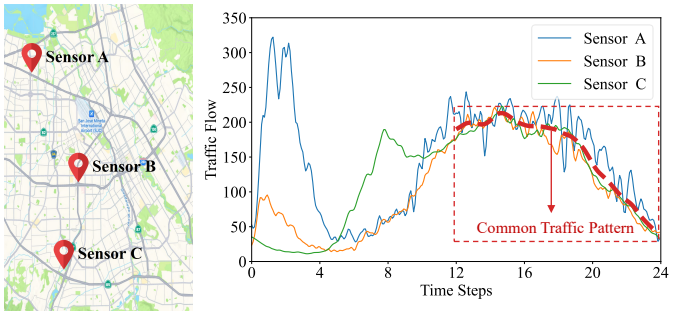
***Abstract*—Accurate Traffic Flow Prediction (TFP) is crucial for enhancing the efficiency and safety of transportation systems, so it has attracted intensive researches by exploiting spatial-temporal dependencies within road networks. However, existing works only consider the case of centralized data collection with all traffic data observed, which may raise privacy concerns as each region of a city may have its own traffic administration department and the traffic data is not allowed to distribute. Therefore, this paper proposes to use Federated Learning (FL) to address this issue by allowing all clients (*i.e.*, traffic administration departments in all regions in our problem) to collaboratively train TFP models without exchanging raw data, thereby offering a solution in maintaining data privacy. Nevertheless, most existing FL methods aim to learn a global model that performs well universally, so they cannot well handle the non-Independent and Identically Distributed (non-IID) traffic data naturally over different regions. To cope with this problem, this paper develops a new FL framework termed "*personalized Federated learning with Traffic Pattern Sharing*" (FedTPS) to solve federated TFP problem. Our FedTPS critically exploits the underlying common traffic patterns (*e.g.*, morning and evening rush hours) shared across different city regions and meanwhile maintaining the region-specific data characteristics in a personalized FL manner. Specifically, to extract the common traffic patterns, we decompose the traffic data in each client via using discrete wavelet transform, where the low-frequency components uncover the stable traffic dynamics of different regions and thus can be considered as the common traffic patterns. These common patterns are then shared among different clients through traffic pattern repositories on the server side to aid the global collaborative traffic flow modeling. Moreover, the model components capturing spatial-temporal dependencies in traffic data are retained for local training, thereby enabling personalized learning based on regional characteristics. Intensive experiments on four real-world traffic datasets firmly demonstrate the superiority of our proposed FedTPS over other compared typical FL methods in terms of various estimation errors.**

***Index Terms*—spatial-temporal data, traffic flow prediction, personalized federated learning**

I. Introduction

Traffic Flow Prediction (TFP) targets to forecast the future traffic conditions at specific locations or roadway segments via using historical traffic data and relevant features [1]. Accurate and real-time traffic prediction offers substantial benefits to urban management and travel planning [2].

*Corresponding authors.



(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 may share common traffic patterns (indicated by the red dashed line in (b)).

Early-staged deep learning-based TFP methods [3]–[5] often employ a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to model the diverse dependencies among different traffic routes. Although these deep learning-based methods generally outperform traditional statistical approaches [6]–[8], their performances could be still limited due to the inherent non-Euclidean nature of the traffic network. To better capture the spatial dependencies within traffic networks, in recent studies, Graph Neural Networks (GNNs) have been applied to TFP tasks [9], [10]. However, these methods are usually based on pre-defined graphs, and thus may fail to characterize the dynamic nature of traffic networks. In response, recent research has shifted towards adaptively learning graph structures [11]–[13], in order to accurately reveal real-world traffic dynamics.

Despite of the good performance achieved by the aforementioned methods, they are typically performed in a centralized manner, where all training data are collected onto a central server to train a global model. However, traffic data is often collected by different traffic administration departments according to the zoning of the city, province, or state. Since traffic data may include sensitive information, such as travel trajectories of individuals and vehicle identification numbers [14], centralizing these data will probably raise privacy

issues. To address these challenges, Federated Learning (FL) has emerged as a solution. In FL, model training is conducted locally on clients and only model parameters rather than raw data are uploaded to the server, which helps ensure data privacy [15], [16]. Up to now, various attempts have been made for accurate TFP using the FL framework [17]–[20].

However, one major challenge in FL is the non-Independent and Identically Distributed (non-IID) issue, where heterogeneity of data among different clients may lead to unstable training and performance deterioration [21]. This issue is particularly pronounced in traffic data recorded by sensors across various locations and time stamps. Recently, Personalized Federated Learning (PFL) methods that develop customized models for each client have been proved to be effective in addressing the heterogeneity of traffic data [22]–[24]. Although these methods enhance model personalization to some extent, they ignore the underlying common knowledge across different regions which is actually critical in collaborative model training. To be specific, despite the traffic data from different regions may be heterogeneous, they share certain common traffic patterns with similar temporal characteristics [25]. These common traffic patterns may arise from 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). Although 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 inspires the sharing of common traffic patterns in the FL framework for performance enhancement.

Therefore, in this work, we propose *personalized Federated learning with Traffic Pattern Sharing* (FedTPS), a federated framework for TFP that effectively explores and utilizes common traffic patterns. Our objective is to improve local performance by sharing the common traffic patterns across different regions while maintaining 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, where the low-frequency components reflecting stable traffic dynamics can be considered as the common traffic patterns. Afterwards, we design a traffic pattern repository for each client to further extract and store representations of common traffic patterns. In the aggregation phase of FL, we aggregate the traffic pattern repositories from different clients on the server side to effectively share the common patterns, which facilitates collaborative model training. Meanwhile, to preserve region-specific characteristics, the model components capturing spatial-temporal dependencies of traffic data are retained in each client for local training. We have conducted intensive experiments on four popular TFP datasets in the FL scenario, which demonstrates the superiority of FedTPS against multiple baseline methods.

II. Related Work

In this section, the related works on TFP and FL for spatial-temporal forecasting will be reviewed.

A. Traffic Flow Prediction

TFP aims to forecast traffic volumes at specific times and locations. In the early stages, statistical methods, such as Historical Average (HA) [6], Kalman Filter (KF) [7], and Auto-Regressive Integrated Moving Average (ARIMA) [8], were commonly employed for TFP. However, these methods often assume linearity in traffic data, which is inadequate for handling the complex dynamics in traffic flow. With the advance of deep learning technologies, many deep learning-based time series models have been applied to TFP, such as RNN [3], Temporal Convolutional Network (TCN) [26], and Transformer [27]. These models have shown great power in capturing nonlinear correlations in traffic data and handling dynamic temporal dependencies, thereby improving prediction accuracy.

With the emergence of GNNs, various methods have been developed to integrate GNNs with time series models, in order to capture the spatial-temporal dependencies of traffic data. For example, DCRNN [9] models the dynamics of traffic flow as diffusion processes and introduces diffusion convolutional operations to capture spatial dependencies. Besides, STGCN [10] combines Graph Convolutional Network (GCN) with TCN for TFP. To further explore the dynamism of traffic networks, Graph WaveNet [11] and AGCRN [12] adaptively learn adjacency matrices to capture the spatial dependencies. Building upon this, StemGNN [28] extracts temporal correlations of sequences through Gated Recurrent Unit (GRU) to learn adjacency matrices, while MegaCRN [13] computes the weights of adjacency matrices through a learned meta-node bank. Additionally, some methods employ attention mechanisms to capture time-varying spatial dependencies among traffic roads. For instance, GMAN [29] utilizes Graph Attention Network (GAT) and temporal attention to model the relationships between historical and future time stamps. Meanwhile, ASTGNN [30] develops a dynamic graph convolution module, which employs self-attention to capture the spatial correlations in a dynamic manner. STWave [31] applies a sampling strategy based GAT to decouple complex traffic data and achieves accurate forecasts with reduced computational costs.

However, most existing research efforts still rely on centralized training data. Traffic data from different regions often belong to different traffic administration departments. Due to privacy issues, sharing of traffic data among different regions may be prohibited, which makes the application of traditional techniques impractical for real-world TFP.

B. Federated Learning

FL is a machine learning paradigm that enables collaboratively training models across decentralized devices or clients with local data. This technique avoids the need to exchange data, thereby preserving privacy and security. The traditional

FL method FedAvg [15] aggregates model weights sent from local clients on the server and downloads the aggregated model back to the clients. However, the heterogeneity of data among different clients poses a critical challenge. To deal with this problem, FedProx [32] 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. FedAtt [33] achieves flexible aggregation through adaptive weight. Besides, FedFed [34] shares the performance-sensitive features to mitigate data heterogeneity. Recently, PFL [35]–[39] becomes popular in dealing with highly heterogeneous data, which proposes to train a personalized local model suitable for each client in a collaborative manner. Therefore, PFL is usually more effective than traditional FL methods that only learn a single global model [32]–[34]. For example, FedPer [36] shares a common base layer while providing individualized local layers for each client to preserve local knowledge. Additionally, through model-agnostic meta-learning, PerFedAvg [37] learns a meta-model to generate initial local models for each client, in order to improve the local performance.

Recently, various FL methods have been developed for spatial-temporal forecasting. For example, FedGRU [17] introduces an ensemble clustering-based FL scheme to capture the spatial-temporal correlation of traffic data. Furthermore, CNFGNN [20] aggregates parameters based on spatial dependencies captured by GNNs on the server. Considering the heterogeneity of spatial-temporal data, some studies aim to enhance the performance of models by learning personalized models for each client. For instance, FedDA [22] proposes a dual attention scheme, which aggregates both intra- and inter-cluster models, rather than simply averaging the weights of local models. In FML-ST [23], meta-learning is integrated into the FL scenario to solve the heterogeneity problem in spatial-temporal prediction. Analogously, FUELS [24] incorporates auxiliary contrastive tasks to inject detailed spatial-temporal heterogeneity into the latent representation space. However, the aforementioned methods overlook the common knowledge (*e.g.*, common traffic patterns) within spatial-temporal data, and thus their performance could be limited.

III. Problem Description

In this section, we formally define the setting of our investigated federated TFP problem. The traffic road network of a city can be represented as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V}$ denotes the set of nodes, where each node corresponds to a sensor to record traffic data, and $\mathcal{E}$ denotes the set of edges corresponding to the roads connecting the sensors. Besides, $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ 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 reality, traffic sensors in different regions of a city may belong to different traffic administration departments. Consequently, suppose there are M traffic administration departments governing M different regions. Then the m-th ($m = 1, 2, \ldots, M$) region possesses a subset of sensors $\mathcal{V}_m$, therefore forming a subgraph of the global traffic network $\mathcal{G}_m \subseteq \mathcal{G}$ along with its corresponding private dataset $\mathcal{D}_m = \{\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_t^{(i)}, \ldots, \mathbf{x}_T^{(i)}\}_{i=1}^{|\mathcal{V}_m|}$, where $\mathbf{x}_t^{(i)} \in \mathbb{R}^d$ represents the observed d-dimensional features recorded by sensor i at time stamp t, and T represents the total number of time stamps. Therefore, our target is to precisely predict the traffic flow at the location of sensors.

However, most existing works rely on centralized data collection, which is impractical since the traffic data possessed by different traffic administration departments is not allowed to distribute due to privacy issues. To overcome this challenge, we propose to use FL to collaboratively train TFP models without the need to exchange private traffic 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 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:

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

where $\mathbf{X}_t = [\mathbf{x}_t^{(1)}; \mathbf{x}_t^{(2)}; \ldots; \mathbf{x}_t^{(|\mathcal{V}_m|)}] \in \mathbb{R}^{|\mathcal{V}_m|\times d}$ represents the observation of the local traffic network $\mathcal{G}_m$ at time stamp t.

The objective of federated TPF is to collaboratively train TFP models across clients without compromising data privacy. The classic FL method, *i.e.*, FedAvg [15], aggregates model parameters at the server after local training according to the following formula:

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

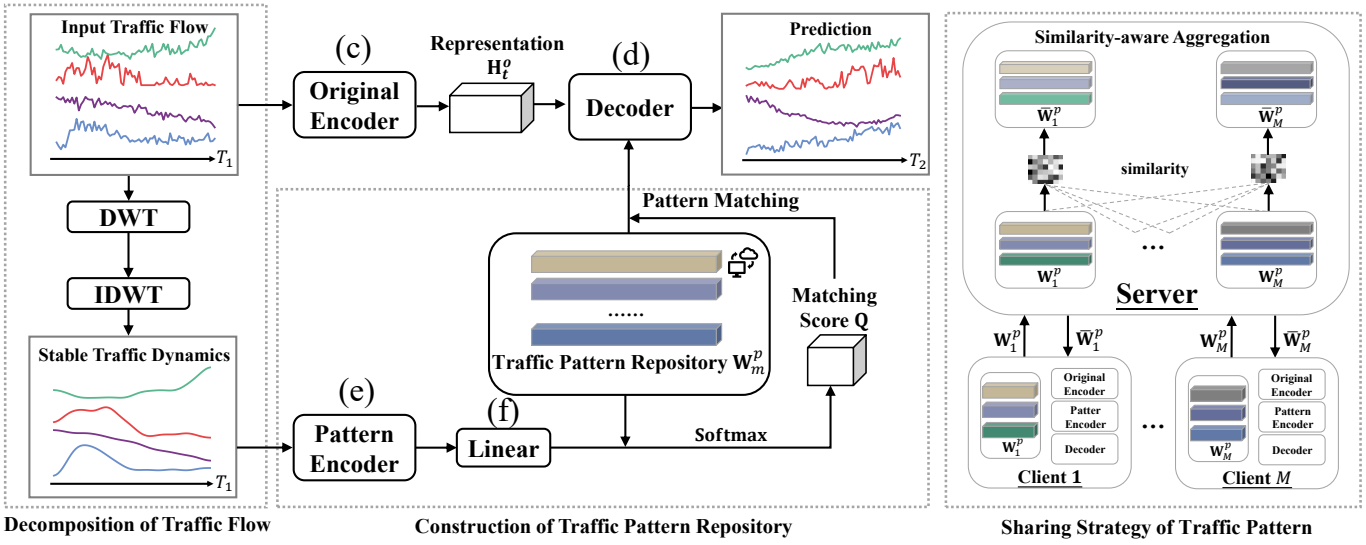
The aggregated model is then redistributed to clients for the next round of training. Due to the non-IID traffic data across different regions, this way of learning a global TFP model may exhibit suboptimal performance. To address this issue, PFL is implemented by training a customized model for each client to enhance its 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.

IV. Methodology

This section details the proposed FedTPS (see Fig. 2). During the local training phase (see Fig. 2(a)), stable traffic dynamics are firstly obtained through the decomposition of traffic flow. To construct the traffic pattern repository, the stable traffic dynamics representation obtained through the pattern encoder (see Fig. 2(e)) is firstly projected to a query space through a linear layer (see Fig. 2(f)) to compute the matching scores with patterns in the repository. Then the



(a) The local training phase

(b) The global aggregation phase

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

matched common pattern is computed as a weighted sum of the patterns in the repository via matching scores. Afterwards, the representations of original traffic data obtained through the original encoder (see Fig. 2(c)) and the matched common traffic pattern are fed into the decoder (see Fig. 2(d)) for TFP. During the global aggregation phase (see Fig. 2(b)), the traffic pattern repository is uploaded to the server for similarity-aware aggregation, while the remaining components are used to learn region-specific characteristics locally. Next, we detail the critical steps of FedTPS by presenting the graph convolutional recurrent unit (GCRU) (see Section IV-A), explaining the process of common traffic pattern extraction (see Section IV-B), and describing the sharing strategy of traffic patterns (see Section IV-C).

A. Adaptive Graph Convolutional Recurrent Unit

The inherent graph structure of traffic networks is well-suited for methods integrating GCN and GRU to concurrently explore spatial and temporal dependencies in traffic data [9], [28]. Based on this foundation, some methods [11]–[13] have introduced adaptive adjacency matrices to model the dynamic spatial correlations within traffic networks. By following these prior works, our local TFP model employs an encoder-decoder architecture (*i.e.*, (c), (d), and (e) in Fig. 2) 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 current input $\mathbf{X}_t$ and previous hidden states $\mathbf{H}_{t-1} \in \mathbb{R}^{|\mathcal{V}_m|\times h}$ where h denotes the dimensionality of hidden state of each node. 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. 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 hereinafter should be accompanied by the subscript m. However, to simplify the notation, we omit the subscript m if no notational confusion is incurred.

B. Extraction of Common Traffic Patterns

To extract common traffic patterns, DWT is firstly employed to decompose the traffic flow on each client to obtain stable traffic dynamics. Then, we utilize them to construct a traffic pattern repository during the local training phase. They are detailed as follows.

1) Decomposition of Traffic Flow: In reality, traffic data from different regions may share common traffic patterns that manifest as stable dynamics due to similar functionalities or consistent travel behaviors [25]. However, in most existing FL frameworks for TFP [22]–[24], the underlying global knowledge represented by common traffic patterns among different regions is largely ignored. Therefore, we aim to extract common traffic patterns manifesting stable traffic dynamics from the diverse traffic data. This can not only facilitate the sharing of common patterns but also help mitigate

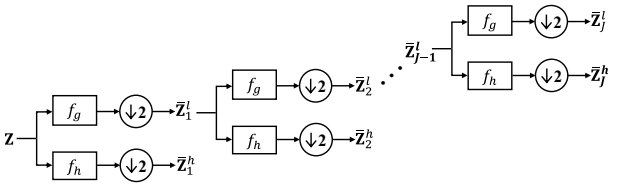


Fig. 3. Illustration of J-level DWT.

the side effects of discrepancies arising from region-specific characteristics. To achieve this goal, we employ DWT [31] to decompose the traffic data into waveforms of different frequencies, with the expectation that the low-frequency component corresponding to stable traffic dynamics can be considered as the common traffic pattern. 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. We only employ one-level DWT in our model to reduce computational overhead. 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.

After decomposition, the original traffic time series $\mathbf{Z}$ and its low-frequency component $\mathbf{Z}^l$ are separately fed into different encoders to obtain the learned representations $\mathbf{H}_t^o$ and $\mathbf{H}_t^l$, respectively.

2) Construction of Traffic Pattern Repository: After obtaining the common traffic patterns represented by the low-frequency component, we intend to exploit the knowledge contained in these patterns for subsequent sharing among the different clients. However, since these patterns are generated from traffic data on each client individually, directly sharing them may pose a risk of privacy leakage. In addition, given the observed variation in traffic patterns across different traffic road networks [25], we aim to learn a set of representative traffic patterns for each client to facilitate pattern sharing.

Memory networks, which have achieved notable success in computer vision [40] and anomaly detection [41], [42] due to their powerful representation abilities, have been increasingly adopted for spatial-temporal data analysis [13], [25], [43]. Inspired by the 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. We first adopt a linear layer to project the stable traffic dynamics representation $\mathbf{H}_t^l$ to a query space, which can be formulated as

$$\mathbf{H}_t^q = \mathbf{H}_t^l * \mathbf{W}^q + \mathbf{b}^q, \tag{11}$$

where $\mathbf{H}_t^q \in \mathbb{R}^{|\mathcal{V}_m| \times c}$ denotes query matrix and "$*$" denotes the dot product operation. $\mathbf{W}^q \in \mathbb{R}^{h \times c}$ and $\mathbf{b}^q \in \mathbb{R}^c$ are learnable parameters. Then we compute the matching scores $\mathbf{Q}$ with patterns in the repository as follows:

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

Subsequently, we calculate the matched traffic patterns $\mathbf{P}_t \in \mathbb{R}^{|V| \times c}$ as a weighted sum of the patterns in $\mathbf{W}^p$, and obtain

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

Equation (12) and (13) are used to retrieve the most relevant common traffic patterns for a given query matrix. Finally, we concatenate $\mathbf{P}_t$ 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 $\ell 1$ 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 linear layer, and $\mathbf{W}_m^p$ refers to the learnable traffic pattern repository.

C. Sharing Strategy of Traffic Pattern

Based on the constructed traffic pattern repository, FedTPS aims to share the global knowledge contained in the common traffic patterns across clients in a personalized manner. The model on the m-th client can be divided into two parts: the traffic pattern repository $\mathbf{W}_m^p$ that represents the common traffic patterns and other model parameters (*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. Our core idea is to share the learnable traffic pattern repository within the FL framework while maintaining the rest model parameters for local training. Moreover, to improve the alignment of traffic patterns from different clients during the aggregation process, we devise the similarity-aware aggregation rather than the conventional averaging aggregation. Specifically, by denoting $\mathbf{W}_m^p[i]$ as the i-th representative pattern in the repository of the m-th client, we calculate the cosine similarity of $\mathbf{W}_m^p[i]$ with patterns from repositories of other clients. Then we select and aggregate the top-k similar patterns from each client, which can be expressed as

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

where $\mathcal{S}_k$ indicates the set of k indices of the representative patterns in $\mathbf{W}_n^p$ that are most similar to $\mathbf{W}_m^p[i]$. Afterwards, 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; federated traffic pattern repository $\overline{\mathbf{W}}_m^p$.
Output: Prediction of future traffic flow $\mathbf{Z}'$.

Download federated traffic pattern repository $\overline{\mathbf{W}}_m^p$ from the server;
Update the traffic pattern repository $\mathbf{W}^p{}_m \leftarrow \overline{\mathbf{W}}_m^p$;
for each local rounds $r = 1, 2, \ldots, R_1$ **do**
    Compute low-frequency component $\mathbf{Z}^l$ via (8) and (10);
    Compute the representations $\mathbf{H}_t^o$ and $\mathbf{H}_t^l$ via (4), (5), (6), and (7);
    Compute the matched pattern $\mathbf{P}_t$ via (11), (12) and (13);
    Concate $\mathbf{H}_t^o$ and $\mathbf{P}_t$, and predict future traffic flow $\mathbf{Z}'$ through decoder;
    Calculate gradients and update learnable parameters $\mathbf{W}_m^{e_1}$, $\mathbf{W}_m^{e_2}$, $\mathbf{W}_m^d$, $\mathbf{W}_m^q$, and $\mathbf{W}_m^p$;
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 $\mathbf{W}_m^p$ from client m.
Output: Federated traffic pattern $\overline{\mathbf{W}}_m^p$ for client m.

Initialize $\overline{\mathbf{W}}^{p(1)}$;
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 (14);
            Send $\overline{\mathbf{W}}_m^{p(r)}$ to client m;
        Perform Algorithm 1 on client m;

Through iterative training and aggregation of pattern repositories, common traffic patterns serve as additional global knowledge to further guide the TFP process. Meanwhile, the remaining model components, which learn the spatial-temporal dependencies specific to the region of each client, do not participate in the process of aggregation, thereby forming the personalized FL style and mitigating the adverse effects of region-specific variations.

In summary, through our proposed method of common pattern extraction and sharing strategy, the federated framework can effectively explore common traffic patterns to enhance TFP capabilities with the personalized model. We provide the pseudocode of our FedTPS on the client side and server side in Algorithm 1 and Algorithm 2, respectively.

TABLE I
DATASET STATISTICS

Datasets	# Samples	# Nodes	Sample Rate	Time Span
PEMS03	26208	358	5 mins	09/2018 - 11/2018
PEMS04	16992	307	5 mins	01/2018 - 02/2018
PEMS07	28224	883	5 mins	05/2017 - 08/2017
PEMS08	17856	170	5 mins	07/2016 - 08/2016

V. EXPERIMENTS

To evaluate the performance of our model, we carried out comparative experiments on four real-world highway traffic datasets in FL scenarios. First, we introduce the experimental settings, followed by a detailed presentation of the results, which includes performance comparison, ablation study, and parametric sensitivity.

A. Experimental Setup

1) Datasets Description and Preprocessing: We evaluate our proposed framework on four widely used datasets for TFP, including *PEMS03*, *PEMS04*, *PEMS07*, and *PEMS08*. These datasets consist of traffic flow data collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS) [44], with the number representing the district code. The statistical details of these datasets are listed in Table I.

Following the practice of previous methods [45], we split the datasets into training set, validation set, and test set in chronological order with the ratio of $6:2:2$. Across all four datasets, we use 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 [46] to evenly partition the global traffic road network, with each client possessing a subgraph of the global traffic road network.

2) Evaluation Metrics: The evaluation metrics in this paper include 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.

TABLE II
OVERALL PERFORMANCE ON FOUR DATASETS. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLDFACE**.

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 [15]	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 [32]	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 [33]	16.34	26.27	22.84	20.62	32.23	14.64	23.29	36.04	10.90	16.40	25.39	11.53
FedPer [36]	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 [37]	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 [38]	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 [39]	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

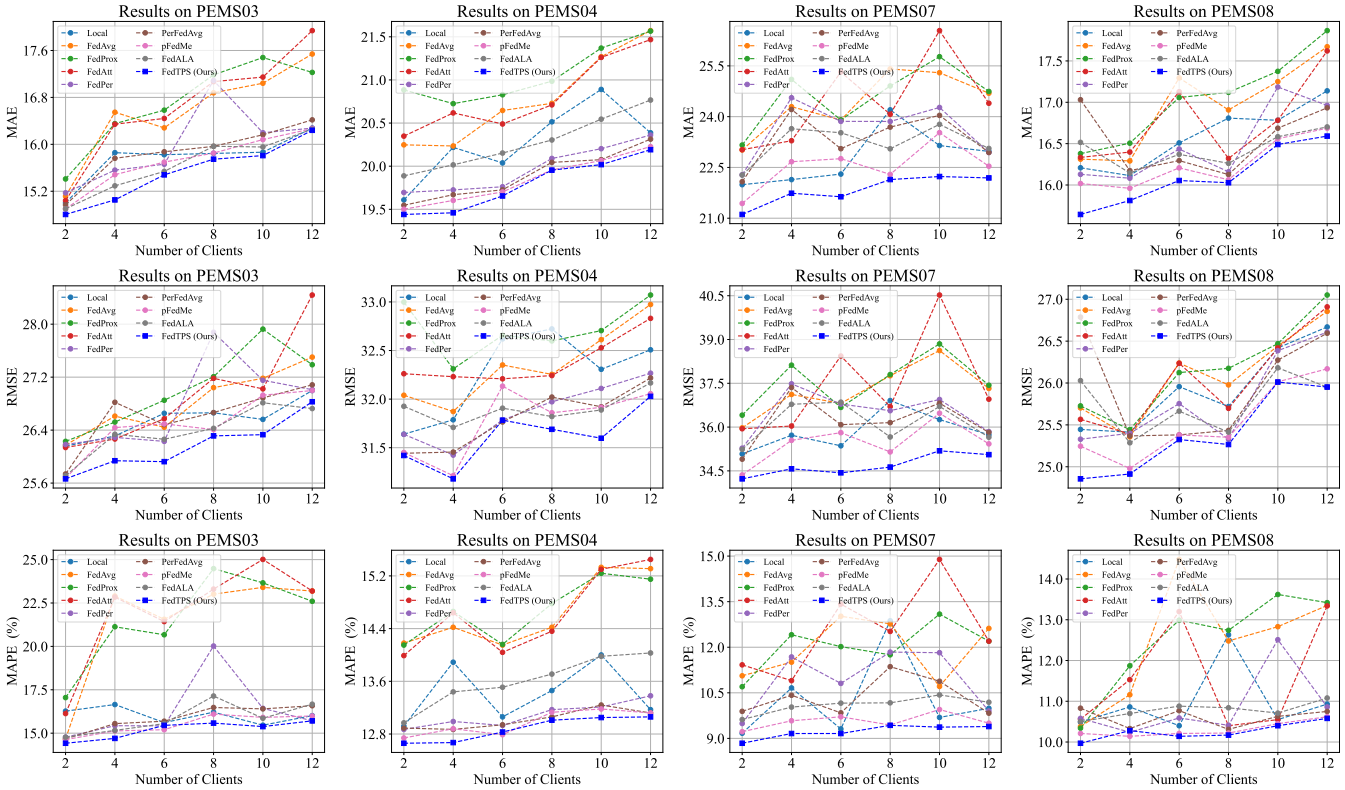


Fig. 3. The performance on four datasets, with varying client numbers.

3) Baseline Methods: Different from the previous node-level federated TFP methods [17], [20], where each sensor is considered as a client, our method is aimed at subgraph-level federated TFP task, where each client possesses a subset of sensors. To ensure the fairness of experiment, we compare our method with the following eight baseline methods:

- **Local**: A baseline method where all clients train their models locally without sharing model parameters.
- **FedAvg** [15]: The classic FL algorithm aggregating the locally updated models via averaging strategy.
- **FedProx** [32]: A FL algorithm preventing the deviation of local models towards their corresponding data via using a proximal term.
- **FedAtt** [33]: A FL algorithm using attention mechanism to weight the aggregation of local and global model parameters.
- **FedPer** [36]: A PFL algorithm sharing common base layers across clients while keeping the personalized layers locally.
- **PerFedAvg** [37]: A PFL algorithm training an initial model that can be fine-tuned to adapt to the local data of clients.
- **pFedMe** [38]: A PFL algorithm utilizing the global model to optimize personalized models.
- **FedALA** [39]: A PFL algorithm adaptively aggregating the global and local models towards the local objective.

4) Implementation Details: In our encoder-decoder architecture, both the encoder and decoder contain 64 GCRUs. To ensure fairness in our experiments, all baseline methods employ the same encoder-decoder architecture as the local

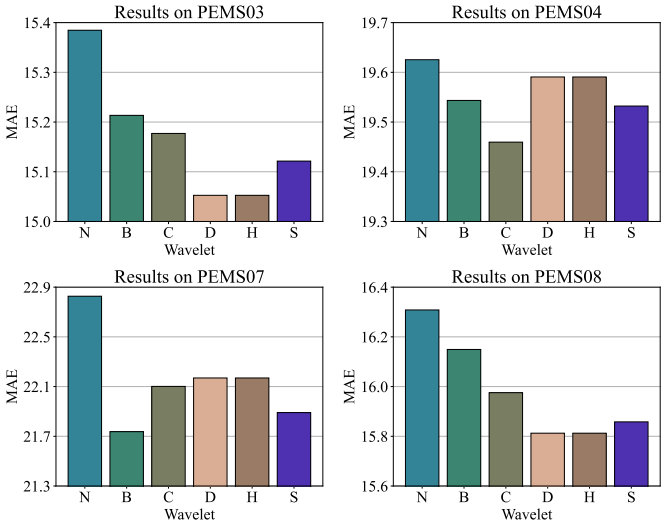


Fig. 4. Effect of DWT on four datasets. N: Not using DWT; B: Biorthogonal; C: Coiflets; D: Daubechies; H: Haar; S: Symlets.

model. We set the size of the traffic pattern repository N to 40 for *PEMS07* and 20 for *PEMS03*, *PEMS04*, and *PEMS08*. The dimension of representative patterns is set to 64 and the number of selected patterns k in the aggregation process is set to 2. We use the Adam optimizer 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 at 1 and 200, respectively, for all FL methods. The default number of clients is set to 4. We implement all the methods on Python 3.8.8 using PyTorch 1.9.1 and conduct all experiments on one GeForce RTX 3090 GPU.

B. Performance Comparison

We evaluate the performances of all methods under three metrics on four TFP datasets, with the results listed in Table II. We clearly observe that our proposed FedTPS outperforms the baseline methods in most scenarios. We infer that this good performance benefits from the effective use of common traffic patterns sharing which facilitates collaborative model training and minimizes the side effects of region-specific discrepancies. Additionally, conventional FL methods (*i.e.*, FedAvg, FedProx, and FedAtt) suffer obvious performance degradation compared with Local, which could be due to the heterogeneity of traffic data from different clients. In contrast, personalized FL methods train customized models for each client and thus can achieve better performance.

To further evaluate the performance of our FedTPS across FL frameworks of different numbers of clients, we investigate the impact of varying numbers of clients on the performance of different methods. As shown in Fig. 3, there is a general trend of increasing prediction error as the number of clients increases over all methods which can be attributed to the division of the traffic network into multiple subgraphs. To be specific, the correlations between traffic sensors are disrupted and the amount of data available to each client is reduced, thereby hindering the training of local models. Nevertheless, FedTPS generally demonstrates promising performance, indicating the effectiveness of our proposed method under the FL frameworks of different numbers of clients.

C. Ablation Study

As described in the methodology, our proposed FedTPS extracts common traffic patterns through DWT decomposition and enables clients to learn personalized models while sharing global knowledge represented by common traffic patterns. To illustrate the contributions of these two modules, we carry out a series of ablation studies.

1) Effectiveness of DWT Decomposition: To understand the contribution of DWT decomposition, we conduct ablation studies comparing the performance of models with or without DWT. In the variant without DWT, the original traffic data are fed into the pattern encoder. Moreover, for the variants with DWT, we further explore the effects of different wavelet bases, including Biorthogonal, Coiflets, Daubechies, Haar, and Symlets. As illustrated in Fig. 4, decomposing the traffic data using DWT enhances the performance across all datasets. This is because the stable traffic dynamics obtained through DWT can effectively capture the common traffic patterns, which is beneficial to FL. Furthermore, we observe that different wavelet bases yield varied performance enhancements across the datasets. Specifically, Daubechies and Haar wavelets demonstrate the best performance for *PEMS03* and *PEMS08* datasets, Coiflets wavelets show the optimal results for *PEMS04*, and Biorthogonal wavelets are most effective for *PEMS07*.

2) Effectiveness of Traffic Pattern Sharing Strategy: To validate the effectiveness of the proposed traffic pattern sharing strategy with the similarity-aware aggregation, we compare FedTPS with its variants. These variants share different components of the local model, where the same aggregation method as FedAvg [15] is adopted. As shown in Table III, on most datasets, the aggregation strategy that shares encoder-decoder parameters (*i.e.*, ED in Table III) results in a substantial performance degradation compared with the strategy that does not share parameters (*i.e.*, None in Table III). These outcomes indicate that directly sharing model parameters of clients across different regions can introduce the interference of region-specific characteristics from other clients and thereby disrupt the learning process. Although the variant that shares all parameters (*i.e.*, All in Table III) can somewhat mitigate the decline in performance with the help of common traffic patterns, it is still inevitably influenced by discrepancies of different regions. In contrast, the strategy that shares traffic pattern repositories (*i.e.*, PR in Table III) demonstrates good performance since it only shares common traffic patterns that represent global knowledge, while retaining region-specific knowledge in a personalized FL manner, allowing each client to learn a personalized model. Furthermore, unlike the model variants using averaging aggregation (*i.e.*, None, All, ED, and PR in Table III), our proposed FedTPS utilizing the similarity-aware aggregation can help align the common traffic

TABLE III
COMPARATIVE ANALYSIS OF DIFFERENT SHARING STRATEGIES. NONE: NON-PARAMETER SHARING; ALL: SHARING ALL PARAMETERS; ED: SHARING ENCODER-DECODER PARAMETERS; PR: SHARING THE TRAFFIC PATTERN REPOSITORY.

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**

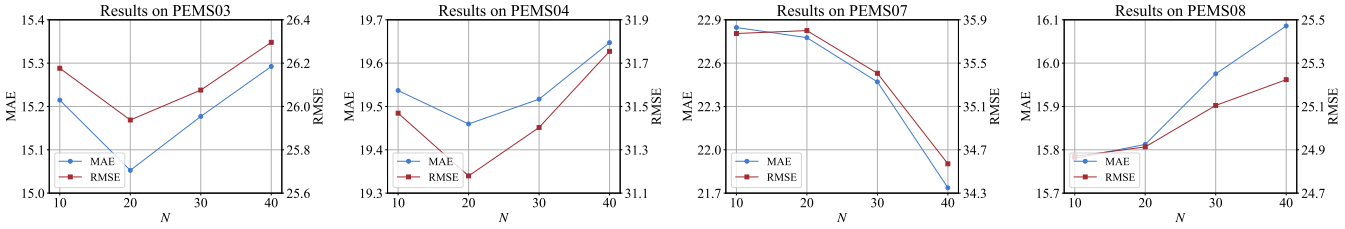


Fig. 5. Sensitivity analysis of the pattern number N in different datasets.

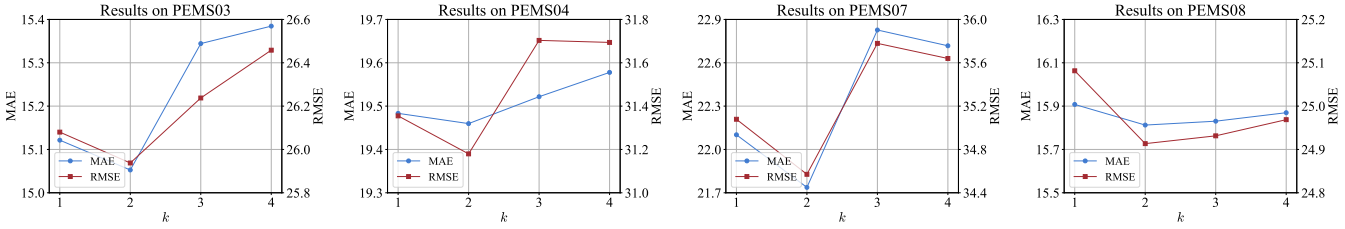


Fig. 6. Sensitivity analysis of the number of selected patterns k during aggregation in different datasets.

patterns from different regions, thereby showing improved performance.

D. Parametric Sensitivity

In the proposed FedTPS framework, two critical hyper-parameters need to be manually pretuned, *i.e.*, the size of the traffic pattern repository N and the number of selected patterns k in the aggregation process. In this subsection, we will evaluate the sensitivity of the performance to different hyperparameter settings of the proposed FedTPS.

The impact of varying N is shown in Fig. 5. We observe that the optimal value of N is related to the number of traffic sensors in the dataset. Generally, datasets with a large number of sensors benefit from a large-size traffic pattern repository (*e.g.*, *PEMS07*). This allows for comprehensive learning of common traffic patterns, which leads to enhanced performance.

The results for different values of k are presented in Fig. 6. We observe that our model achieves the best performance across all datasets when $k = 2$. If k is too small, FedTPS may overlook the traffic patterns that are beneficial to knowledge sharing. Conversely, if k is too large, it may incorporate patterns that do not align well, resulting in suboptimal performance.

VI. CONCLUSION

In this paper, we propose FedTPS, a new PFL framework to address the challenge of data heterogeneity in federated TFP via sharing common traffic patterns. Different from previous works that overlook the underlying global knowledge represented by common traffic patterns from different regions, the proposed FedTPS decomposes the traffic data to acquire common traffic patterns, which can be shared across different clients. Meanwhile, by devising the similarity-aware aggregation strategy, clients can learn from the common traffic patterns of different regions globally while maintaining personalized components learning from local spatial-temporal dependencies to preserve region-specific characteristics. Experimental evaluations conducted on four widely-used TFP datasets confirm the effectiveness and superiority of our FedTPS over multiple baseline methods.

ACKNOWLEDGMENT

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). Chen Gong was supported by NSF of China (Nos: 62336003, 12371510), NSF of Jiangsu Province (No: BZ2021013), and NSF

for Distinguished Young Scholar of Jiangsu Province (No: BK20220080).

REFERENCES

[1] C. Lin, G. Han, J. Du, T. Xu, L. Shu, and Z. Lv, "Spatiotemporal congestion-aware path planning toward intelligent transportation systems in software-defined smart city iot," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8012–8024, 2020.

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2014.

[3] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NeurIPS*, vol. 28, 2015.

[4] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *SIGIR*, 2018, pp. 95–104.

[5] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *AAAI*, vol. 32, no. 1, 2018.

[6] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.

[7] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in *KDD*, 2011, pp. 1010–1018.

[8] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, 2013.

[9] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, 2018.

[10] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI*, 2018, pp. 3634–3640.

[11] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI*, 2019, pp. 1907–1913.

[12] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *NeurIPS*, vol. 33, 2020, pp. 17 804–17 815.

[13] R. Jiang, Z. Wang, J. Yong, P. Jeph, Q. Chen, Y. Kobayashi, X. Song, S. Fukushima, and T. Suzumura, "Spatio-temporal meta-graph learning for traffic forecasting," in *AAAI*, vol. 37, no. 7, 2023, pp. 8078–8086.

[14] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, 2018.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017, pp. 1273–1282.

[16] S. Yue, Y. Deng, G. Wang, J. Ren, and Y. Zhang, "Federated offline reinforcement learning with proximal policy evaluation," *Chinese Journal of Electronics*, vol. 33, no. 6, pp. 1–13, 2024.

[17] Y. Liu, J. James, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7751–7763, 2020.

[18] C. Zhang, S. Zhang, J. James, and S. Yu, "Fastgnn: A topological information protected federated learning approach for traffic speed forecasting," *IEEE Trans. Ind. Inform.*, vol. 17, no. 12, pp. 8464–8474, 2021.

[19] H. Wang, R. Zhang, X. Cheng, and L. Yang, "Federated spatio-temporal traffic flow prediction based on graph convolutional network," in *WCSP*, 2022, pp. 221–225.

[20] C. Meng, S. Rambhatla, and Y. Liu, "Cross-node federated graph neural network for spatio-temporal data modeling," in *KDD*, 2021, pp. 1202–1211.

[21] X. Li, M. JIANG, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," in *ICLR*, 2020.

[22] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *INFOCOM*, 2021.

[23] W. Li and S. Wang, "Federated meta-learning for spatial-temporal prediction," *Neural Computing and Applications*, vol. 34, no. 13, pp. 10 355–10 374, 2022.

[24] Q. Liu, S. Sun, Y. Liang, J. Xue, and M. Liu, "Personalized federated learning for spatio-temporal forecasting: A dual semantic alignment-based contrastive approach," *arXiv preprint arXiv:2404.03702*, 2024.

[25] H. Lee, S. Jin, H. Chu, H. Lim, and S. Ko, "Learning to remember patterns: Pattern matching memory networks for traffic forecasting," in *ICLR*, 2021.

[26] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, vol. 27, 2014.

[27] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[28] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong *et al.*, "Spectral temporal graph neural network for multivariate time-series forecasting," in *NeurIPS*, vol. 33, 2020, pp. 17 766–17 778.

[29] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *AAAI*, vol. 34, no. 01, 2020, pp. 1234–1241.

[30] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Trans. Knowledge Data Eng.*, vol. 34, no. 11, pp. 5415–5428, 2021.

[31] Y. Fang, Y. Qin, H. Luo, F. Zhao, B. Xu, L. Zeng, and C. Wang, "When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks," in *ICDE*, 2023, pp. 517–529.

[32] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine learning and systems*, vol. 2, 2020, pp. 429–450.

[33] S. Ji, S. Pan, G. Long, X. Li, J. Jiang, and Z. Huang, "Learning private neural language modeling with attentive aggregation," in *IJCNN*, 2019, pp. 1–8.

[34] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, and B. Han, "Fedfed: Feature distillation against data heterogeneity in federated learning," in *NeurIPS*, vol. 36, 2023, pp. 60 397–60 428.

[35] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *ICML*, 2021, pp. 2089–2099.

[36] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[37] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *NeurIPS*, vol. 33, 2020, pp. 3557–3568.

[38] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," in *NeurIPS*, vol. 33, 2020, pp. 21 394–21 405.

[39] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Fedala: Adaptive local aggregation for personalized federated learning," in *AAAI*, vol. 37, no. 9, 2023, pp. 11 237–11 244.

[40] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *NeurIPS*, vol. 29, 2016.

[41] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *ICCV*, 2019, pp. 1705–1714.

[42] T. Jiang, W. Chen, H. Zhou, J. He, and P. Qi, "Towards semi-supervised classification of abnormal spectrum signals based on deep learning," *Chinese Journal of Electronics*, vol. 33, no. 3, pp. 721–731, 2024.

[43] Z. Liu, G. Zheng, and Y. Yu, "Cross-city few-shot traffic forecasting via traffic pattern bank," in *CIKM*, 2023, pp. 1451–1460.

[44] C. Chen, "Freeway performance measurement system (pems)," Ph.D. dissertation, University of California, Berkeley, 2002.

[45] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *AAAI*, vol. 33, no. 01, 2019, pp. 922–929.

[46] G. Karypis, "Metis: Unstructured graph partitioning and sparse matrix ordering system," *Technical report*, 1997.