FASTGNN: A Topological Information Protected Federated Learning Approach for Traffic Speed Forecasting

Chenhan Zhang, *Student Member, IEEE*, Shuyu Zhang, James J. Q. Yu, *Senior Member, IEEE*, and Shui Yu, *Senior Member, IEEE*

***Abstract*—Federated learning has been applied to various tasks in intelligent transportation systems to protect data privacy through decentralized training schemes. The majority of the state-of-the-art models in intelligent transportation systems (ITS) are graph neural networks (GNN)-based for spatial information learning. When applying federated learning to the ITS tasks with GNN-based models, the existing frameworks can only protect the data privacy; however, ignore the one of topological information of transportation networks. In this article, we propose a novel federated learning framework to tackle this problem. Specifically, we introduce a differential privacy-based adjacency matrix preserving approach for protecting the topological information. We also propose an adjacency matrix aggregation approach to allow local GNN-based models to access the global network for a better training effect. Furthermore, we propose a GNN-based model named attention-based spatial-temporal graph neural networks (ASTGNN) for traffic speed forecasting. We integrate the proposed federated learning framework and ASTGNN as FASTGNN for traffic speed forecasting. Extensive case studies on a real-world dataset demonstrate that FASTGNN can develop accurate forecasting under the privacy preservation constraint.**

***Index Terms*—Deep learning, federated learning, graph neural networks (GNN), traffic speed forecasting.**

Manuscript received October 30, 2020; revised December 17, 2020 and January 14, 2021; accepted January 21, 2021. Date of publication January 29, 2021; date of current version August 20, 2021. This work was supported in part by the General Program of Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515011032 and in part by the Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation under Grant 2020B121201001. The work of Chenhan Zhang and Shui Yu was supported in part by Australian under Grant ARC DP200101374. Paper no. TII-20-4975. *(Corresponding author: James J. Q. Yu.)*

Chenhan Zhang is with the Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and also with the University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: zhangch@mail.sustech.edu.cn).

Shuyu Zhang and James J. Q. Yu are with the Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: 11712122@mail.sustech.edu.cn; yujq3@sustech.edu.cn).

Shui Yu is with the School of Computer Science, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: shui.yu@uts.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3055283.

Digital Object Identifier 10.1109/TII.2021.3055283

I. Introduction

THE development and broad adoption of the Internet of Things (IoT) have greatly revolutionized people's lifestyles and industrial production. An intelligent transportation system (ITS) is among the most typical application scenarios of IoT [1]. For example, we can see a massive deployment of IoT-based sensor networks for urban traffic data collection, which provides overwhelming data for handling traffic problems such as traffic state forecasting. In this context, recent years have witnessed an unprecedented advancement of data-driven problem-solvers for traffic state forecasting, such as deep learning-based approaches [2].

Graph neural networks (GNN) lies at the most cutting-edge of deep learning techniques capable of learning spatial information from the complicated topology of data. GNN-based approaches have been widely adopted in traffic state forecasting tasks, and remarkable results have been shown in the literature [3]. Nonetheless, most methods highly rely on large-scale data for centralized training. In ITS, such massive traffic data are generally collected and shared by different providers, including government organizations (e.g., California Department of Transportation) and companies (e.g., AutoNavi). The collaborations among these providers usually involve data exchange, but the data may contain personal privacy (e.g., plate number, travel record). Thus, there exist potential privacy leakage issues. With this in mind, different organizations are requested to avoid data exchange by storing their user data locally to preserve personal privacy, making it challenging for them to train a powerful model collaboratively.

With the emerging federated learning (FL) technology, the aforementioned collaborative problems have been vastly resolved [4]. FL serves as a learning framework for multiple data providers, allowing providers to build an effective model collaboratively while keeping their data locally. Comprehensive and successful cases have demonstrated that FL can tradeoff between model performance and privacy [5], [6]. While existing FL frameworks have been successfully applied to many deep learning-based approaches, we found few cases involving GNN-based models. There are two main challenges to combining FL and GNN-based models. First, unlike regular deep learning-based models, GNN-based models need to handle not only the input data feature but also topological information.

Existing FL aggregation algorithms are not capable of handling topological information, which may limit their use in GNN scenarios. Second, the conventional FL framework can only protect the privacy of the data feature. In ITS, the topological information privacy is also important since the topological information may contain sensitive information (e.g., the relationships among mobile data contributors, the number of deployed sensor stations).

To address the above two problems, we propose a FL framework named federated attention-based spatial-temporal graph neural networks (FASTGNN). The proposed framework integrates a novel FL strategy towards topological information protection and a GNN-based model named attention-based spatial-temporal graph neural networks (ASTGNN) for traffic speed forecasting. Specifically, in the proposed FL strategy, we introduce a differential privacy (DP)-based local-network adjacency matrix preserving approach, and it enables each organization's topological information in the FL framework can be well-preserved. A local-network topological information aggregation mechanism is also devised, which allows the local models can take advantage of a DP-processed global topological information to guarantee its performance. In the proposed ASTGNN model, a graph attention mechanism and gated recurrent unit networks are adopted, and they make ASTGNN possess excellent spatial-temporal feature learning capacity for developing accurate network-wide traffic speed predictions. In such a configuration, FASTGNN can develop promising traffic speed forecasting without compromising privacy.

The main contributions of this article are summarized below.

1) We propose a topological information protected FL framework FASTGNN for traffic speed forecasting problem. This framework integrates a GNN-based predictor utilizing the advanced spatial-temporal techniques. Such a framework can provide robust privacy-preserving traffic speed forecasting through training models locally by different organizations without raw data and topological information exchange.
2) In the proposed FL framework, we introduce a DP-based approach for adjacency matrix preserving to protect the topological information. We also develop an adjacency matrix aggregation mechanism to generate a preserved global-network adjacency matrix. These two approaches guarantee our framework achieves the tradeoff between privacy and performance.
3) A series of comprehensive case studies on a real-world traffic dataset are conducted to demonstrate the efficacy of the proposed FASTGNN framework.

The rest of this article is organized as follows. In Section II, we review the related literature on traffic speed forecasting and privacy-preserving in ITS. Section III gives a basic formulation of the investigated traffic speed forecasting problem and the federated learning framework. We elaborate on the proposed ASTGNN model and FASTGNN framework in Section IV. Section V presents the results and discussion of the case studies. Finally, Section VI concludes this article.

II. Literature Review

A. GNN-Based Approaches for Traffic Forecasting

Since spatial correlation in traffic data has been demonstrated useful in predicting the time-series, researchers start to involve spatial feature exploiting mechanisms in traffic forecasting approaches [7]. In [8] and [9], CNN is adopted to extract the spatial independencies of the traffic data. Despite its effectiveness in feature extraction, CNN-based approaches are constrained to only process the grid-like spatial structure. In the meantime, traffic data is sampled from the traffic network, which is non-Euclidean. To overcome this problem, GNN is employed by treating the irregular traffic networks as graphs where spatial information can be fully learned [10]–[12]. Graph attention network is an extension of GNN, which applies attention mechanism to graphs [13]. In [14], the authors proposed a traffic flow estimator where both spatial and temporal attentional factors are computed to extract the spatial-temporal dependencies in traffic data. Shi *et al.* [15] devise an attention mechanism that can exploit short- and long-term dependencies in time-series.

B. Privacy Problems in ITS

The process of data exchange among users and organizations in ITS may leak private information [16]. These increase the privacy-preserving awareness of ITS participants. Zhou *et al.* [17] proposed a privacy-preserving transportation traffic measurement approach for cyber-physical road systems, which adopts maximum-likelihood estimation (MLE) to develop the prediction. In [18], the authors proposed an autonomous privacy-preserving authentication approach, where vehicles can effectively conceal their information using pseudonyms. In [19], the author proposed a secure ITS movement analysis scheme that allows participants to generate the private/public key pair of their own accord. Nonetheless, these approaches suffer from massive data within a limited processing time and preserve privacy in exchange for the model performance. Recent years have witnessed the development of federated learning (FL) models. FL can train machine learning models in a privacy-preserving manner, where the training datasets are distributed across multiple devices while preventing data breach [4]. Liu *et al.* [20] proposed a FL-based GRU neural network algorithm for traffic forecasting to achieve privacy-preserving traffic flow prediction. Albaseer *et al.* [21] proposed a federated semisupervised learning scheme to utilize the unlabeled data in ITS. Smarakoon *et al.* [22] optimized the communications delay incurred by FL in their proposed distributed resource allocation approach for vehicular networks. Feng *et al.* [23] proposed a privacy-preserving mobility prediction framework via federated learning. In [24], the authors proposed a FL-based mobility-aware proactive edge caching approach for vehicular networks. Qolomany *et al.* [25] proposed a particle swarm optimization approach to optimize the hyperparameter settings for the local machine-learning models in the FL framework, and demonstrated its capacity on a traffic prediction task. Qi *et al.* [26] introduced blockchain technique into FL for privacy-preserving traffic flow prediction.

TABLE I
DEFINED SYMBOLS IN THE PROPOSED FRAMEWORK

Symbol	Definition
	FASTGNN framework
$\mathcal{G}$	Transportation network.
$\mathcal{V}$	Set of road segments (nodes) in $\mathcal{G}$.
$\mathcal{N}$	Number of road segments (nodes).
$\mathcal{E}$	Set of road intersections (edges).
$\mathcal{A}$	Adjacency matrix of $\mathcal{G}$.
$\tilde{\mathcal{A}}$	Privacy-preserved $\mathcal{A}$.
$\tilde{\mathcal{A}}^{aggre}$	Aggregated $\tilde{\mathcal{A}}$.
$\mathcal{T}$	Number of time stamps in the past.
s	Number of time stamps in the future.
X^t	Observed speeds of all roads at time t.
$\hat{X}^t$	Forecasted speeds of all roads at time t.
$f(\cdot)$	Learnable function to forecast the speed in the future.
p	Number of participated organizations.
$\mathcal{O}$	Set of participated organizations.
$\mathcal{G}^*$	Set of local-networks.
$\mathcal{D}_i$	Database of organization O_i.
M_i	Local model of organization O_i.
ϕ_i	Parameters of local model M_i.
$Z_{\mathcal{G}_i^*}$	Topological information of local network $\mathcal{G}_i^*$.
$Z^{(pp)}_{\mathcal{G}_i^*}$	Privacy-preserved $Z_{\mathcal{G}_i^*}$.
$\mathcal{M}$	Number of random projection.
	ASTGNN model
h^t	Network-wide feature vector at time t.
F	Dimension of the vector h^t.
$Att_{v_i \leftarrow v_j}$	The attention score that v_i perceive from v_j.
W, a, U	Weight matrix (vector).
$\text{concat}(\cdot)$	Concatenation operation.
$\alpha^t_{v_i \leftarrow v_j}$	Attention coefficient of $Att_{v_i \leftarrow v_j}$.
$\hat{\alpha}^t_{v_i \leftarrow v_j}$	Filtered $Att_{v_i \leftarrow v_j}$ for connected node pair.
$\text{N}(i)$	Set of immediately adjacent nodes of node v_i.
r^t	Reset gate vectors of GRU at time t.
z^t	Update gate vectors of GRU at time t.
$\sigma(\cdot)$	Non-linear activation function.

While many insightful privacy-preserving and FL-based approaches for traffic forecasting have been proposed, most of them are based on relatively simple temporal models such as [20]. In this article, the proposed FL framework is devised for GNN-based models, especially for privacy-preserving of topological information, which attempts to fill the research gap.

III. PRELIMINARY

The symbols defined in this article are summarized in Table I. In this section, we first define the problem of traffic forecasting on transportation networks. Then, the problem of traffic forecasting with graph-based deep learning models in the FL framework is introduced.

A. Traffic Speed Forecasting on Transportation Networks

A transportation network can be represented by an undirected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V}$ is the set of nodes which we define each node as a road segment and $\mathcal{E}$ is the set of edges, and $\mathcal{A} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is the adjacency matrix of $\mathcal{G}$ where $\mathcal{N}$ is the number of nodes in $\mathcal{G}$. $\forall\, v_i, v_j \in \mathcal{V}$, if v_i and v_j are connected, $(v_i, v_j) \in \mathcal{E}$ and entry $\mathcal{A}_{ij} = 1$ (otherwise $\mathcal{A}_{ij} = 0$). Denote the traffic speed observed on $\mathcal{G}$ as a graph-wide feature matrix $\mathcal{X} \in \mathbb{R}^{\mathcal{N}\times\mathcal{Q}}$ where $\mathcal{Q}$ is the number of incorporated features of each node. Let vector $\mathcal{X}^t \in \mathbb{R}^{\mathcal{N}}$ denote the traffic speed observation at time t, the problem can be thus defined as learning a function $f(\cdot)$ to develop traffic speed predictions $\hat{\mathcal{X}}^{t+1}, \hat{\mathcal{X}}^{t+2}, \ldots, \hat{\mathcal{X}}^{t+s}$ in the following s time stamps, given historical traffic speed observations of $\mathcal{T}$ stamps $\mathcal{X}^{t-\mathcal{T}+1}, \mathcal{X}^{t-\mathcal{T}+2}, \ldots, \mathcal{X}^t$.

B. Federated Learning on Transportation Networks

In this article, we construct the FL framework for traffic speed forecasting on the transportation network. We define a "global-network" $\mathcal{G}$ as the entire transportation network of an area. This area is divided and conquered by several organizations (e.g., companies, governments). Let $\mathcal{O} = \{O_1, O_2, \ldots, O_p\}$ denote the organization set where p is the number of organizations. Thus, each organization operates a local-network $\mathcal{G}_i^*$ of $\mathcal{G}$. Let $\mathcal{G}^* = \{\mathcal{G}_1^*, \mathcal{G}_2^*, \ldots, \mathcal{G}_p^*\}$ denote the local-network set. The respective databases of these organizations are $\mathcal{D}_i$, which collect traffic speed data from their operated local-networks. Particularly, we have $\mathcal{D}_i = (\mathcal{X}_i^*, Z_{\mathcal{G}_i^*})$ where $\mathcal{X}_i^*$ and $Z_{\mathcal{G}_i^*}$ are the historical traffic speed data and topological information (e.g., road connectivity) collected from local-network $\mathcal{G}_i^*$, respectively. Additionally, this article is based on the assumption that the organizations do not have overlapping regions and data with each other, i.e., for any two organizations i and j, $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$. This is a common assumption among the literature, see [20], [23], and [27] for some examples. Our goal is to train a powerful model in the cloud that can predict the global-network-wide traffic speed with local traffic speed data from $\mathcal{D}_i$. Nonetheless, due to privacy concerns, these organizations are prohibited from sharing the raw traffic data and the topological information of their operated local-networks (i.e., they can only access their respective local-networks).

To achieve our goal under the aforementioned privacy constraints, it is required to adopt a secure parameter aggregation mechanism (SPAM) [4] in the FL framework. Particularly, the graph-based deep learning model M_i constructed by each organization O_i computes a group of updated model parameters ϕ_i utilizing the local training data from $\mathcal{D}_i$ and the topological information of the corresponding local-network $\mathcal{G}_i^*$. After all the organizations complete the parameters' updating, their respective parameters are uploaded to the cloud. The global model is finally developed by aggregating these uploaded parameters. SPAM guarantees that no traffic speed data leakage happens among the organizations.

IV. METHODOLOGY

This section first introduces the proposed attention-based spatial-temporal graph neural networks (ASTGNN) as the local graph deep learning-based model for traffic speed forecasting. Then, we elaborate on our proposed FL framework Federated-ASTGNN (FASTGNN).

A. Attention-Based Spatial-Temporal Graph Neural Networks

For the network-wide traffic speed forecasting problem, we propose ASTGNN as the local forecasting model. As illustrated in Fig. 1, ASTGNN consists of four modules: feature embedding module, spatial dependency capture module, temporal dependency capture module, and prediction output module.

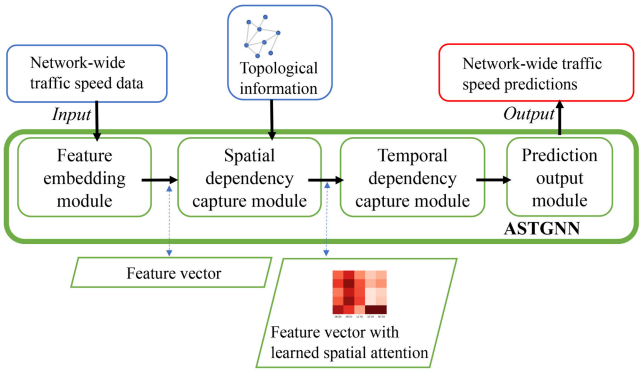


Fig. 1. Framework of ASTGNN.

1) Feature Embedding Module: The feature embedding module transforms the input time-series data into feature vectors, which can be processed by the spatial dependency capture module afterward. Specifically, given a sequence (length $= T$) of network-wide time-series speed values $\mathcal{X}^1, \mathcal{X}^2, \ldots, \mathcal{X}^T$, each feature vector can be formulated as

$$h^t = [\mathcal{X}^{t-F+1}, \mathcal{X}^{t-F+2}, \ldots, \mathcal{X}^t] \tag{1}$$

where $h^t \in \mathbb{R}^{F \times \mathcal{N}}$ is the network-wide feature vector at time t; F is the dimension of the vector whose physical meaning is equivalent to the past window size (i.e., $\mathcal{T}$). That means that we actually embed a sequence of speed data whose length is the same as the past window size into a feature vector. In this way, we can obtain a sequence of feature vectors $h^1, h^2, \ldots, h^T$.

2) Spatial Dependency Capture Module: Spatial dependency capture module is used to exploit the spatial dependency among different road segments (nodes) in the transportation network (graph). We construct this module by following graph attention networks (GAT) [13], which utilizes the attention mechanism to obtain the spatial correlations. The operational steps of this module can be described as the following steps.

1) We commence by computing the attention score. For any ordered pair of nodes $(v_i, v_j) \in \mathcal{V}$, the attention score v_i perceive from v_j can be formulated as

$$Att_{v_i \leftarrow v_j} = a^T \cdot \text{concat}(W h_i^t, W h_j^t) \tag{2}$$

where $Att_{v_i \leftarrow v_j}$ denotes the attention score, h_i^t and h_j^t are feature vector of node v_i and v_j at time t, respectively, $W \in \mathbb{R}^{F^h \times F}$ is a weight matrix that can transform feature vector into a higher-level dimension F^h, $\text{concat}(\cdot)$ denotes the concatenation operation, $a \in \mathbb{R}^{2F^h}$ is a weight vector, and $\cdot^T$ denotes the transposition operation.

2) Subsequently, we use activation functions to normalize the attention score and obtain the attention efficient, which can be expressed as

$$\alpha_{v_i \leftarrow v_j}^t = \text{softmax}\left(\text{LeakyReLU}\left(e_{ij}\right)\right) \tag{3}$$

where $\alpha_{v_i \leftarrow v_j}^t \in [0, 1]$ denotes the attention coefficient, $\text{LeakyReLU}(\cdot)$ denotes the leaky rectified linear units activation function, and $\text{softmax}(\cdot)$ denotes the softmax activation function.

3) Next, we filter the obtained attention coefficient to survive the attention coefficients only for connected node pair, which can be formulated as

$$\hat{\alpha}_{v_i \leftarrow v_j}^t = \alpha_{v_i \leftarrow v_j}^t \odot \mathcal{A}_{ij} \tag{4}$$

where $\mathcal{A}_{ij}$ is the entry for node v_i and v_j in the adjacency matrix $\mathcal{A}$, and $\odot$ denotes the Hadamard product. We can deduce that when $\mathcal{A}_{ij} = 1$, the attention coefficient survives, otherwise be discarded (i.e., equal to 0).

4) Finally, the attention coefficients are employed to update the feature vector of node v_i, which can be formulated as

$$\hat{h}_i^t = \sigma\left(\sum_{j \in \text{N}(i)} \hat{\alpha}_{v_i \leftarrow v_j}^t W^o h_j^t\right) \tag{5}$$

where $\hat{h}_i^t$ is the updated feature vector of node v_i at time t which is regarded as the output of this module, $\text{N}(i)$ is the set of immediately adjacent nodes of node v_i, W^o is a weight matrix.

3) Temporal Dependency Capture Module: The temporal dependency capture module is designed to learn the potential temporal dependency of data. We employ two layers of GRU neural networks in this module. GRU introduces a collection of gating units and cell states to process the input information, which can solve the gradient vanishing problem in the learning process. The gating units have two types, i.e., reset gate r and update gate z. Given the input data x^t,[1] the hidden layer output h_g^t can be computed by

$$z^t = \sigma\left(W^{(z)} x^t + U^{(z)} h_g^{t-1}\right) \tag{6}$$

$$r^t = \sigma\left(W^{(r)} x^t + U^{(r)} h_g^{t-1}\right) \tag{7}$$

$$\tilde{h}_g^t = \tanh\left(W x^t + r^t \odot U h_g^{t-1}\right) \tag{8}$$

$$h_g^t = z^t \odot h_g^{t-1} + (1 - z^t) \odot \tilde{h}_g^t \tag{9}$$

where $W^{(z)}$, $W^{(r)}$, $U^{(z)}$, and $U^{(r)}$ are the weight matrices connecting x^t and h_g^{t-1} to two gates, $\tilde{h}_g^t$ is the intermediate candidate activation.

4) Prediction Output Module: A fully connected layer is employed in this module to produce the traffic speed of future s time stamps. Such a linear transformation conducted by a full-connected layer is formulated as

$$\mathcal{X}^{\hat{t}+1}, \mathcal{X}^{\hat{t}+2}, \ldots, \mathcal{X}^{\hat{t}+s} = W^{(fc)} h_g^t + b \tag{10}$$

where $W^{(fc)} \in \mathbb{R}^{C \times s}$ is a weight matrix that maps the hidden output of GRU in the temporal module to s prediction output, and b is the bias.

B. Federated Learning Framework for ASTGNN

In the previous section (Section IV-A), we introduce the proposed ASTGNN model for traffic speed forecasting. In this section, we introduce the proposed FL framework for ASTGNN,

[1] x^t is the output of the spatial module, i.e., $\hat{h}^t$, we use x^t here to avoid confounding notations.

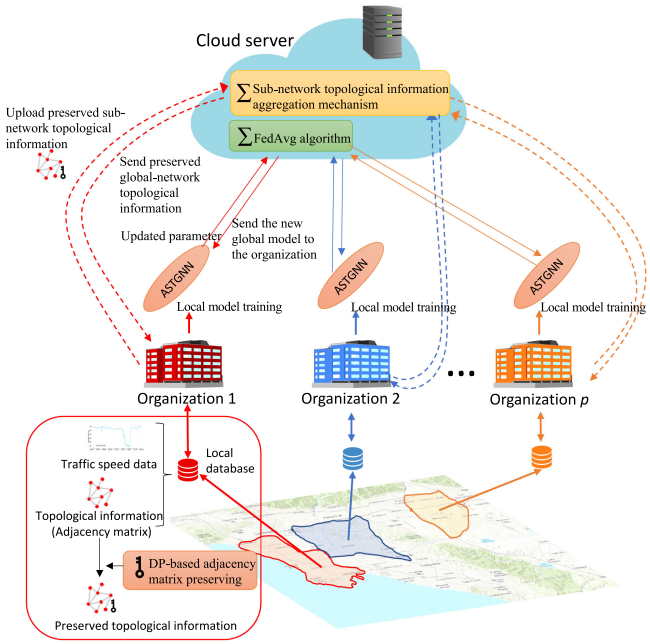


Fig. 2. Framework of FASTGNN.

Algorithm 1: Communication Protocol of FASTGNN.

1: The organizations apply a privacy-preserving algorithm to its local-network topological information and obtain preserved topological information $Z_{\mathcal{G}_i^*}^{(pp)}$
2: The organizations upload $Z_{\mathcal{G}_i^*}^{(pp)}$ to the cloud server, the latter aggregate the uploaded $Z_{\mathcal{G}_i^*}^{(pp)}$ and develop one of the global-network $Z_{\mathcal{G}}^{(pp)}$
3: The cloud distributes the copies of the global model and $Z_{\mathcal{G}}^{(pp)}$ to all organizations, and each organization trains its copies using local data
4: Each organization uploads the learned model parameters ϕ_i to the cloud. Since the private data and topological information is not shared in the entire process, the privacy-preserving is guaranteed
5: The cloud server aggregates ϕ_i by SPAM as introduced in Section III-B to build a new global model. Subsequently, the new model is distributed to the organizations

namely, FASTGNN. As illustrated in Fig. 2, each organization operates an ASTGNN as the local model, whose input is traffic speed data and topological information from its local traffic database. The DP-based adjacency matrix preserving algorithm is implemented at the organization end to protect the local topological information. The cloud server is in charge of aggregating the preserved local topological information and ASTGNN model parameters and broadcast the aggregated ones. The detailed elaboration of related algorithms can be seen in the following.

1) FASTGNN Communication Protocol: As defined in Section III-B, each organization can only access its own traffic data and local-network topological information for local models' training. One concern with training local models using only local-network topological information is that local-networks do not contain all the essential topological information for computing attention coefficients by ASTGNN. This issue may lead to the final low learning effect (the related experimental comparison will be demonstrated in Section V-C). Thus it is requisite to feed the topological information of the global-network to the local models for obtaining better results. To achieve this without compromising the privacy of local-network topological information, we propose a FL communication protocol as presented in Algorithm 1.

We then detail the adopted privacy-preserving algorithm for topological information, the local-network topological information aggregation mechanism, SPAM, and the entire FL process.

2) DP-Based Adjacency Matrix Preserving: In this article, we regard the adjacency matrix of local-network as the carrier of topological information. We introduce a DP-based approach to provide privacy-preserving to the adjacency matrix while keeping its utility in the learning process of ASTGNN. This approach is based on [28], which leverages the theories of DP and random matrix to the adjacency matrix privacy-preserving. Specifically, given the to be preserved adjacency matrix $\mathcal{A} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, the algorithm is presented below.

1) Generate two Gaussian random matrices $R^{(p)} \in \mathbb{R}^{\mathcal{N}\times\mathcal{M}}$ and $R^{(q)} \in \mathbb{R}^{\mathcal{M}\times\mathcal{M}}$ where $\mathcal{M}$ is the number of random projection [29] that have $\mathcal{M} \ll \mathcal{N}$. In this way, each entry of $R^{(p)}$ and $R^{(q)}$ are independently sampled from Gaussian distribution $N_1(0, 1/\mathcal{M})$ and $N_2(0, \sigma^2)$[2]
2) Compute the projection matrix $\mathcal{A}^{(p)} \in \mathbb{R}^{\mathcal{N}\times\mathcal{M}}$ by $\mathcal{A}^{(p)} = \mathcal{A}R^{(p)}$. By doing this, each row of $\mathcal{A}$ is projected from a high dimension $\mathbb{R}^{\mathcal{N}}$ into a low dimension $\mathbb{R}^{\mathcal{M}}$.
3) Perturb $\mathcal{A}^{(p)}$ with the Gaussian random matrix $R^{(q)}$ by $\tilde{\mathcal{A}}^{(p)} = \mathcal{A}^{(p)} + R^{(q)}$. We then project $\tilde{\mathcal{A}}^{(p)}$ back to the dimension $\mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ by $\tilde{\mathcal{A}} = \tilde{\mathcal{A}}^{(p)}(R^{(q)})^T$. Matrix $\tilde{\mathcal{A}}$ is the output of the algorithm.

The perturbed matrix $\tilde{\mathcal{A}}$ is regarded as the preserved one of the original adjacency matrix $\mathcal{A}$. The top eigenvectors of the adjacency matrices are mainly utilized in GNN-based models to compute the spatial correlations [13], [30]. The adoption of random projection as described in Step i preserves the top eigenvectors of $\mathcal{A}$, which provides a guarantee for the effectiveness of the preserved adjacency matrix in the subsequent ASTGNN predictor. Furthermore, this algorithm enables us to involve a small amount of random perturbation, which further improves the utility of the perturbed matrix. In the case studies of this work, we empirically set $\mathcal{M} = 10$ and $\sigma = 0.5$. Regarding the mathematical analysis of this algorithm, interested readers can refer to [28].

3) Local-Network Topological Information Aggregation Mechanism: Step ii of the FASTGNN communication protocol requires the cloud server to aggregate the uploaded $Z_{\mathcal{G}}^{(pp)}$ (i.e., the preserved adjacency matrix). Thus, we propose an adjacency

[2]With abuse of notation, σ in this section exclusively denotes the variance of distribution N_2.

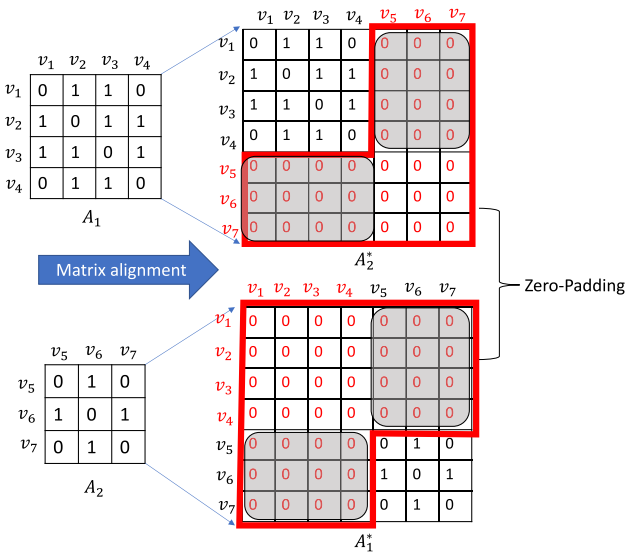


Fig. 3. Adjacency matrix alignment. The red frame highlights the padding entries. The shadowed region highlights the entries entailing the connectivity among objective local-network and other local-networks.

matrix aggregation mechanism. Given a group of uploaded preserved local-network adjacency matrices $\{\tilde{\mathcal{A}}_1, \tilde{\mathcal{A}}_2, \ldots, \tilde{\mathcal{A}}_p\}$ where p is the number of involved local-networks, their corresponding sizes are $\{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_p\}$. Since the sizes of these matrices are different, we first use a matrix alignment approach to make them possess the same size while keeping their own topological information. Specifically, as shown in Fig. 3, we align the dimensions of them to the size of the global-network (i.e., $\mathcal{N}$) by using zero-padding and thus obtain a group of aligned matrices $\{\tilde{\mathcal{A}}_1^{(a)}, \tilde{\mathcal{A}}_2^{(a)}, \ldots, \tilde{\mathcal{A}}_p^{(a)}\}$ where $\forall \tilde{\mathcal{A}}_i^{(a)}, \tilde{\mathcal{A}}_i^{(a)} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$. Further, considering the significance of the connectivity (i.e., edges) among different local-networks (as the shadowed region shown in Fig. 3) for learning the attention, we construct a random connection for them. Specifically, we generate a Gaussian random matrix with the same size as the shadowed regions using the approach as introduced in Section IV-B2 and symmetrically replace the original parts. Finally, we obtain the aggregated preserved adjacency matrix by adding the aligned matrices together, which can be formulated as

$$\tilde{\mathcal{A}}^{(aggre)} = \sum_{i}^{p} \tilde{\mathcal{A}}_i^{(a)}. \tag{11}$$

Particularly, denote $[\tilde{\mathcal{A}}^{(aggre)}]$ as the entry of $\tilde{\mathcal{A}}^{(aggre)}$, we threshold its value by

$$\forall \left|[\tilde{\mathcal{A}}^{(aggre)}]\right| < \frac{p}{\mathcal{M}}, [\tilde{\mathcal{A}}^{(aggre)}] \leftarrow 0. \tag{12}$$

4) Learning Process of FASTGNN: In FASTGNN, we use FedAvg [31] algorithm as SPAM to aggregate the uploaded parameters and develop a new global model. The FedAvg algorithm can be formulated as

$$\phi_{(g)}^{t+1} = \frac{1}{p}\sum_{i=1}^{p} \phi_i \tag{13}$$

where ϕ_i is the parameter of the local model, p is the number of organizations (i.e., the number of local models), and $\phi_{(g)}^{t+1}$ is the aggregated parameter for the new global model. FedAvg algorithm can help train high-quality global with a small cost of communication.

Finally, as shown in Algorithm 2, the entire learning process of each round in FASTGNN consists of three steps

1) The cloud server broadcasts the global model with initial parameters $\phi_g^0 = (W, W^o, W^{(z)}, W^{(r)}, U^{(z)}, U^{(r)}, W^{(fc)})$ and the preserved adjacency matrix of global-network $\tilde{\mathcal{A}}^{(aggre)}$ to the organizations.
2) Each organization O_i trains its local data $\mathcal{X}_i^*$ using $\tilde{\mathcal{A}}^{(aggre)}$ and updates the initial local model parameter ϕ_i^t for E_l epochs of an optimizer with mini-batch size B to obtain ϕ_i^{t+1}.
3) The cloud server aggregates each organization's ϕ_i^{t+1} through FedAvg algorithm and obtains a new global model with the aggregated parameter $\phi_{(g)}^{t+1}$.

5) Theoretical Discussion of DP-Based Adjacency Matrix Preserving on Model Performance: Many existing studies have demonstrated that the noise added by DP algorithms to the data may lead to degenerated learning and further affect the

Algorithm 2: FASTGNN.

Input: Organizations $\mathcal{O} = \{O_1, O_2, \ldots, O_p\}$; The number of rounds (i.e., global epochs), E; The preserved adjacency matrix of global-network, $\tilde{\mathcal{A}}^{(aggre)}$; The size of local mini-batch, S; The number of local epochs, E_l; The learning rate, η; The gradient optimizer for ASTGNN, $\mathcal{L}(\cdot,\cdot)$.
Output: Parameter ϕ_i.
Server (k, ω):
initialize global model parameters ϕ_g^0
broadcast $\tilde{\mathcal{A}}^{(aggre)}$ to organizations
foreach round $t = 1, 2, \ldots, t \in E$ **do**
  foreach organization $O \in \mathcal{O}$ **in parallel do**
    $\phi_{(g)}^{t+1} \leftarrow \text{LocalModelUpadate}\ (O, \tilde{\mathcal{A}}^{(aggre)}, \phi_{(g)}^t)$
  end for
  $\phi_{(g)}^{t+1} \leftarrow \frac{1}{p}\sum_{i=1}^{p} \phi_i$
end for
LocalModelUpadte $(O, \tilde{\mathcal{A}}^{(aggre)}, \phi_{(g)}^t)$:
$\mathcal{B} \leftarrow$ (divide $\mathcal{X}_)^*$ in to batches of size B)
foreach epoch $e = 1, 2, \ldots, e \in E_l$ **do**
  foreach batch $b = 1, 2, \ldots, b \in \mathcal{B}$ **do**
    $\phi_i \leftarrow \phi_i - \eta \cdot \mathcal{L}(\tilde{\mathcal{A}}^{(aggre)}, \phi_i)$
  end for
end for
return ϕ_i to cloud server

model performance [32], [33]. In our proposed approach, the noises are added to the adjacency matrices rather than the data. In the learning process of each local model, the adopted aggregated DP-processed global adjacency matrix $\tilde{\mathcal{A}}^{(aggre)}$ is used to only filter the attention coefficients as described in (4). Since $\tilde{\mathcal{A}}^{(aggre)}$ approximates a binary matrix (i.e., (0,1)-matrix) after DP processing and aggregation, the values of attention coefficients will not be affected significantly. Thus, promising final model performance can be guaranteed. Furthermore, the existing performance loss is due to the disparity between the original global topology and the new global topology after DP processing and aggregation on the adjacency matrices. The related case study will be demonstrated in Section V-C.

V. Experiments

In this article, we propose FASTGNN as a FL framework to address the traffic speed forecasting problem with privacy-preserving concern. To fully assess the performance of the proposed framework, we carry out three comprehensive case studies on a real-world traffic dataset. First, we investigate the accuracy of forecasting speed using the proposed framework and the comparison with baselines. Subsequently, an ablation study is conducted to evaluate the critical components of FASTGNN. Lastly, we exhibit the performance of FASTGNN under different organization numbers.

A. System Configuration

1) Dataset Description and Preprocessing: PeMSD7 is the experimental dataset in this work, which is a public dataset collected from the caltrans performance measurement system (PeMS) in District 7 of California. We select 228 out of 39 000 sensor stations in PeMSD7 to construct the final dataset as a tailored one for our case studies. The time interval of speed data is set to 5 min, and the period of the dataset is from May 1st to June 30th of 2012.[3] Linear interpolation is employed to recover the missing data when there exist missing data points. We apply Z-score to normalize the data before input to the models. The training, validation, and testing sets are correspondingly constructed for supervised learning, each of which contains 60%, 20%, and 20% of all data, respectively.

To simulate the distributed training scenario of FASTGNN, we first construct the adjacency matrix of the entire traffic network (i.e., global-network) $\mathcal{A}$ by

$$[\mathcal{A}_{ij}] = \begin{cases} 1, \text{ if } i \neq j \text{ and } \exp\left(-\dfrac{\text{dist}(v_i, v_j)}{\varsigma^2}\right) \geqslant \varepsilon \\ 0, \text{ otherwise.} \end{cases} \tag{14}$$

where $[\mathcal{A}_{ij}]$ is the entry of $\mathcal{A}$ that denotes the connectivity between node v_i and node v_j, which is decided by their Euclidean space distance $\text{dist}(v_i, v_j)$; ε and ς^2 are the user-controlled parameters that control the density of graph, and we set their values to 0.5 and 10, respectively. Note that since we define the network as an undirected graph, the adjacency matrix is symmetrical, i.e., $[\mathcal{A}_{ij}] = [\mathcal{A}_{ji}]$. Then, we partition the global-network into p subnetworks for corresponding p organizations randomly. Let $\mathcal{V}_u, \mathcal{V}_v$ denote any two subnetworks' node sets, we have $\mathcal{V}_u \cap \mathcal{V}_v = \emptyset$. We can thus obtain subnetworks' adjacency matrices $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_p\}$.

2) Experiment Setting: The proposed FASTGNN is implemented with PyTorch, and all tests are conducted on a computing server with an Intel(R) Xeon(R) E5-2620 v4 CPU and eight nVidia GeForce RTX 2080 Ti GPUs. When training FASTGNN, the objective dimension of the weight matrix W in (2) (i.e., F^h) is set to 144, and the numbers of neurons in the two GRU layers are set to 64 and 256, respectively. All the neural networks-based models are trained with Adam optimizer for 50 epochs, and the batch size and learning rate are set to 50 and $1e^{-3}$, respectively.[4] Unless otherwise stated, we simulate FASTGNN with the number of organizations $p = 4$. In terms of the traffic speed forecasting, the past time window is 60 min (i.e., 12 timestamps), and we use these to predict speed in the next 45 min (i.e., nine timestamps). With regards to accuracy comparison, we adopt root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) as the metrics to evaluate the forecasting accuracy of all approaches. Particularly, MAPE is considered as the most preferable one among the three metrics (see [35] and [36] for examples), which can be defined as

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - \hat{X}_i}{X_i}\right| \times 100\% \tag{15}$$

where X_i and $\hat{X}_i$ are the observed and the forecasted traffic speeds at time i, respectively.

B. Accuracy of Forecasting Traffic Speed

We first investigate the accuracy of forecasting traffic speed with PeMSD7 dataset. Specifically, FASTGNN is compared with the following baselines and state-of-the-art approaches:

1) historical average (HA);
2) autoregressive integrated moving average (ARIMA);
3) linear support victor regression (LSVR);
4) diffusion convolutional recurrent neural network (DCRNN) [37];
5) graph WaveNet [38]; and
6) spatio-temporal graph convolutional networks (STGCN) [30].

To make a fair comparison, we configure the baseline approaches with the default hyperparameters in their respective literature.

The forecasting results are presented in Table II for 45 min ahead traffic speed forecasting. From the simulation results, traditional approaches, i.e., HA, ARIMA, and LSVR, have the worst performance with relatively large forecasting errors, which implies their shortage in handling nonlinearity. Comparatively, the graph deep learning-based approaches, i.e.,

[3]Only weekdays' data is contained to avoid atypical traffic, which is in accordance with the literature. See [30], [34] for examples.

[4]For FASTGNN, it denotes that the global epoch size $E = 50$ and the size of local mini-batch $S = 50$.

TABLE II
COMPARISON OF TRAFFIC SPEED FORECASTING ACCURACY

Approach	Accuracy			Graph-based	Privacy-preserving
	RMSE	MAE	MAPE (%)		
HA	7.20	4.01	10.61	–	–
ARIMA	9.45	6.33	16.10	–	–
LSVR	8.28	4.53	11.49	–	–
DCRNN	7.14	4.11	9.92	✓	–
Graph WaveNet	6.23	3.51	9.03	✓	–
STGCN	**5.80**	**3.47**	8.56	✓	–
FASTGNN	5.83	3.50	**8.36**	✓	✓

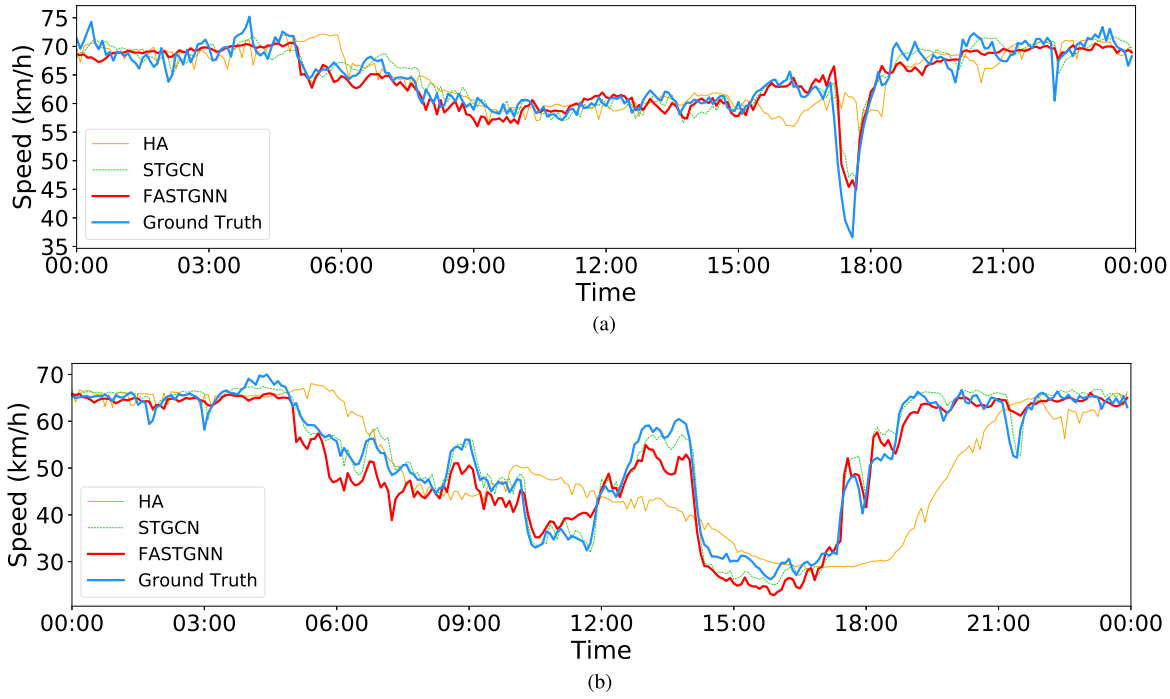


Fig. 4. Traffic speed forecasting curves in a day. (a) and (b) present results from two different sensor stations, respectively.

DCRNN, Graph WaveNet, STGCN, and FASTGNN, perform much better than the conventional approaches with an average improvement of 2.08 (RMSE), 1.34 (MAE), and 3.76% (MAPE). Particularly, the proposed FASTGNN can achieve the same performance level as STGCN, whose accuracy of MAPE even surpasses STGCN by 0.20%. This demonstrates the efficacy of the adopted technical scheme for spatial-temporal learning. Furthermore, FASTGNN is the only one among these approaches that can both deal with spatial information and achieve privacy-preserving through a decentralized training scheme in the proposed FL framework. It indicates that FASTGNN can achieve outstanding performance and privacy-preserving at the same time.

Besides, to better illustrate the forecasting performance of FASTGNN, we present and compare the forecasting curves developed by FASTGNN, HA, and STGCN. As shown in Fig. 4, FASTGNN can produce traffic speed prediction with a small deviation and accurately reflect the oscillation on ground truth.

C. Ablation Study on FASTGNN

To evaluate the several scheme designs in the proposed FASTGNN, we conduct ablation studies in this section. Specifically, we first transform FASTGNN into the following variants by adding particular constraints and compare their MAPE performance with FASTGNN.

1) **FASTGNN-V1**: Without the DP-based adjacency matrix preserving approach.
2) **FASTGNN-V2**: Without local-networks aggregation, i.e., each local model of FASTGNN can only access the local-network other than the global-network for training.
3) **FASTGNN-V3**: Without considering the connectivity among different local-networks when constructing the aggregated global-network.
4) **ASTGNN**: Naive ASTGNN model without FL as introduced in Section IV-A.

The results are presented in Table III. Comparing FASTGNN and ASTGNN, it can be seen that performance degeneration

TABLE III
COMPARISON OF ABLATION TESTS

	FASTGNN	V1	V2	V3	ASTGNN
RMSE	5.83	5.51	9.29	7.98	5.33
MAE	3.50	3.20	7.27	6.10	3.21
MAPE (%)	8.36	8.03	16.10	12.33	7.84

TABLE IV
ACCURACY OF FASTGNN WITH DIFFERENT ORGANIZATION NUMBERS

$p=$	2	4	6	8	12	16
RMSE	5.73	5.83	5.96	6.03	6.18	6.22
MAE	3.31	3.50	3.58	3.76	4.05	4.36
MAPE (%)	8.02	8.36	8.76	9.25	9.79	10.38

due to the adoption of FL's decentralized training is not significant, where the accuracy only suffers from a 0.52% MAPE penalty. This indicates that the combo of adopted techniques can ensure the learning effect of ASTGNN in FL framework under privacy-preserving. Especially when we compare FASTGNN with FASTGNN-V1, the minuscule difference of accuracy performance implies that the adoption of DP-based adjacency matrix preserving approach does not veritably weaken the topological information of the network and further affect the spatial learning effect of the model, which proves the effectiveness of this approach. By contrast, a large performance gap is observed between FASTGNN and FASTGNN-V2, where there is a 7.74% accuracy difference. Since in FASTGNN-V2 the local model can only access the local-network for training where the latter can only provide limited topological information for training a generalized model applicable to the global-network, this results in the striking performance degeneration. It can also shed light on the necessity of adopting a local-network aggregation mechanism to construct a shareable global network for each local training. A similar conclusion can be drawn when comparing FASTGNN and FASTGNN-V3. FASTGNN-V3 performs worse than FASTGNN by 3.97% MAPE. While in the setting of FASTGNN-V3, the local-networks are aggregated, the connectivity among them is absent. This results in the declined performance of V3.

D. Performance Comparison of FASTGNN Under Different Organization Numbers

In the above tests, the default organization number is set as $p = 4$. Nonetheless, the number of organizations in real scenarios may vary a lot. It is interesting to investigate the impact of different organization numbers on the performance of FASTGNN. In this experiment, we set $p \in \{2, 4, 6, 8, 12\}$ for FASTGNN and compare the accuracy performance under this group of settings.

As the results are shown in Table IV, we can observe the number of organizations that have a negative correlation with the performance of FASTGNN. More organizations involve increasing groups of local topological information and model parameters, which makes it challenging for cloud sever to perform

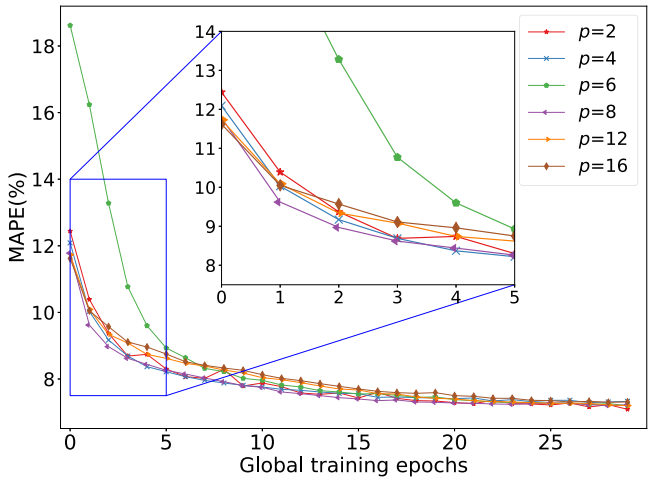


Fig. 5. Visualization of training process for 30 global epochs with different organization numbers.

TABLE V
COMPARISON OF TRAFFIC SPEED FORECASTING ACCURACY ON METR-LA

Approach	Accuracy		
	RMSE	MAE	MAPE (%)
HA	7.80	4.16	13.02
ARIMA	12.11	6.01	15.04
LSVR	12.01	5.92	14.81
DCRNN	7.24	3.41	9.67
Graph WaveNet	6.49	3.01	9.22
STGCN	**6.11**	**2.98**	**8.84**
FASTGNN	6.42	3.03	9.15

the aggregation algorithms. We can draw the same conclusion from the convergence curves as shown in Fig. 5, where the larger number of involved organizations, the more difficult the learning curves converge. It is worth mentioning that no matter how many organizations are involved in our simulation, their respective data and topological information are obtained by dividing the same global-network (c.f., Section V-A1). This may make the results contrast not distinct. We will conduct refined tests in future work.

E. Generalization Ability

In the above case studies, we test the performance of FASTGNN on the PeMSD7 dataset. To assess the generalization ability of FASTGNN, we adopt another dataset METR-LA to examine the forecasting accuracy of FASTGNN. METR-LA is a public dataset, which contains traffic data collected from 207 loop detectors in the highway of Los Angeles County. The experiment setting are configured as the same as it on PeMSD7 for the sake of fairness.

Table V presents the simulation results. For the results, we can observe that conventional machine learning approaches (i.e., ARIMA and LSVR) perform worse than on PeMSD7. This implies that the data of METR-LA is more unstable and changeable than that of PeMSD7. In this context, FASTGNN can still obtain

matched performance compared with the three state-of-the-art baselines, where the MAPE of FASTGNN is only 0.31% higher than that of STGCN. This indicates that FASTGNN is capable of handling data with different time-series fluctuation and topology.

VI. Conclusion

In this article, we proposed the FASTGNN framework for traffic speed forecasting with federated learning for privacy preservation. FASTGNN integrates a GNN-based model ASTGNN for local training and a novel FL strategy to protect the shared topological information. Specifically, we introduced a number of techniques, including a DP-based adjacency matrix preserving approach and a local-network topological information aggregation mechanism, which can make the local topological information be aggregated into a shareable global-network without sharing their raw information. We assessed the performance of FASTGNN on a PeMS dataset and compare it with state-of-the-art approaches. The simulation results show a satisfactory forecasting accuracy for FASTGNN. We also conducted an ablation study on FASTGNN to validate the efficacy of its components. Furthermore, we investigated the performance of FASTGNN under different organization numbers.

In the future, we plan to focus on more fine-grained research on FASTGNN, including the communication overhead, generalization ability on different datasets, etc.

References

[1] W. He, G. Yan, and L. D. Xu, "Developing vehicular data cloud services in the iot environment," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1587–1595, May 2014.

[2] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[3] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. Part C: Emerg. Technol.*, vol. 99, pp. 144–163, 2018.

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[5] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.

[6] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "Deepfed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2020.3023430.

[7] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive Mobile Comput.*, vol. 50, pp. 148–163, 2018.

[8] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, Apr. 2017, Art. no. 818.

[9] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, Jun. 2017, Art. no. 1501.

[10] C. Chen *et al.*, "Gated residual recurrent graph neural networks for traffic prediction," in *Proc. AAAI Conf. Artif. Intel.*, 2019, pp. 485–492.

[11] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, "Gstnet: Global spatial-temporal network for traffic flow prediction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2286–2293.

[12] C. Zhang, J. J. Yu, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166246–166256, 2019.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[14] L. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, "An effective spatial-temporal attention based neural network for traffic flow prediction," *Transp. Res. Part C: Emerg. Technol.*, vol. 108, pp. 12–28, 2019.

[15] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, "A spatial-temporal attention approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2020.2983651.

[16] L. Li *et al.*, "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, Jul. 2018.

[17] Y. Zhou, Z. Mo, Q. Xiao, S. Chen, and Y. Yin, "Privacy-preserving transportation traffic measurement in intelligent cyber-physical road systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3749–3759, May 2016.

[18] V. Sucasas, G. Mantas, F. B. Saghezchi, A. Radwan, and J. Rodriguez, "An autonomous privacy-preserving authentication scheme for intelligent transportation systems," *Comput. Secur.*, vol. 60, pp. 193–205, 2016.

[19] S. O. Ogundoyin, "An anonymous and privacy-preserving scheme for efficient traffic movement analysis in intelligent transportation system," *Secur. Privacy*, vol. 1, no. 6, 2018, Art. no. e50.

[20] Y. Liu, J. J. Yu, 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, Aug. 2020.

[21] A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha, "Exploiting unlabeled data in smart cities using federated edge learning," in *Proc. Int. Wireless Commun. Mobile Comput.*, 2020, pp. 1666–1671.

[22] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.

[23] J. Feng, C. Rong, F. Sun, D. Guo, and Y. Li, "PMF: A. privacy-preserving human mobility prediction framework via federated learning," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–21, 2020.

[24] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2020.3017474.

[25] B. Qolomany, K. Ahmad, A. Al-Fuqaha, and J. Qadir, "Particle swarm optimized federated learning for industrial IoT and smart city services," in *Proc. GLOBECOM 2020–2020 IEEE Glob. Commun. Conf.*, Taipei, Taiwan, 2020, pp. 1–6.

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

[27] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[28] F. Ahmed, R. Jin, and A. X. Liu, "A random matrix approach to differential privacy and structure preserved social network graph publishing," 2013, *arXiv:1307.0475*.

[29] T. Sarlos, "Improved approximation algorithms for large matrices via random projections," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci.*, 2006, pp. 143–152.

[30] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.

[31] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[32] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, Jan. 2020.

[33] C. Dwork *et al.*, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3/4, pp. 211–407, 2014.

[34] Z. Cui, R. Ke, and Y. Wang, "Deep stacked bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction," in *6th Int. Workshop Urban Comput. (UrbComp 2017)*, 2016.

[35] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3940–3951, Oct. 2019.

[36] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.

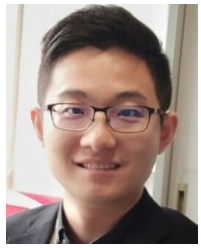
[37] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. ICLR*, 2018.
[38] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.

Chenhan Zhang (Student Member, IEEE) received the B.Eng. degrees in telecommunication engineering from the University of Wollongong, Wollongong, Australia, and Zhengzhou University, Zhengzhou, China, in 2017 and 2018, respectively, and the M.S degree in engineering management from the City University of Hong Kong, Hongkong, in 2019. He is currently working toward the Ph.D. degree in computer science with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia.

His research interests include deep learning, intelligent transportation systems, and privacy-preserving in AI.

Shuyu Zhang is currently working toward the Graduation in computer science with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China.

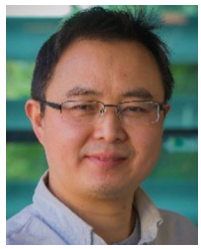
His research interests include urban computing, deep learning, and federated learning.



James J. Q. Yu (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electrical and electronic engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively.

He was a Postdoctoral Fellow with the University of Hong Kong, from 2015 to 2018. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China, and an honorary Assistant Professor with the Department of Electrical and Electronic Engineering, University of Hong Kong. He is also the Chief Research Consultant of GWGrid Inc., Zhuhai, and Fano Labs, Hong Kong. His research interests include smart city technologies, deep learning and big data, intelligent transportation systems, and energy systems.

Dr. Yu is an Editor for the *IET Smart Cities Journal*.



Shui Yu (Senior Member, IEEE) is a Professor with the School of Computer Science, University of Technology Sydney, Australia. He has authored or coauthored three monographs and edited two books, more than 350 technical papers, including top journals and top conferences, such as IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, and INFOCOM. His h-index is 48. His research interest includes big data, security and privacy, networking, and mathematical modelling.

Dr. Yu initiated the research field of networking for big data in 2013, and his research outputs have been adopted by industrial systems. He is currently serving a number of prestigious Editorial Boards, including IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (Area Editor), *IEEE Communications Magazine*, and IEEE INTERNET OF THINGS JOURNAL. He is a Member of AAAS and ACM, and a Distinguished Lecturer of IEEE Communication Society.