GOF-TTE: Generative Online Federated Learning Framework for Travel Time Estimation

Zhiwen Zhang*, Hongjun Wang*, Jiyuan Chen, Zipei Fan†, Xuan Song†, and Ryosuke Shibasaki

Abstract—Estimating the travel time of a path is an essential topic for intelligent transportation systems. It serves as the foundation for real-world applications, such as traffic monitoring, route planning, and taxi dispatching. However, building a model for such a data-driven task requires a large amount of users' travel information, which directly relates to their privacy and thus is less likely to be shared. The non-Independent and Identically Distributed (non-IID) trajectory data across data owners also make a predictive model extremely challenging to be personalized if we directly apply federated learning. Finally, previous work on travel time estimation does not consider the real-time traffic state of roads, which we argue can significantly influence the prediction. To address the above challenges, we introduce GOF-TTE for the mobile user group, **G**enerative **O**nline **F**ederated Learning Framework for **T**ravel **T**ime **E**stimation, which I) utilizes the federated learning approach, allowing private data to be kept on client devices while training, and designs the global model as an online generative model shared by all clients to infer the real-time road traffic state. II) apart from sharing a base model at the server, adapts a fine-tuned personalized model for every client to study their personal driving habits, making up for the residual error made by localized global model prediction. We also employ a simple privacy attack to our framework and implement the differential privacy mechanism to further guarantee privacy safety. Finally, we conduct experiments on two real-world public taxi datasets of DiDi Chengdu and Xi'an. The experimental results demonstrate the effectiveness of our proposed framework.

Index Terms—Ubiquitous; Urban Computing, Federated Learning, Travel Time Estimation

I. INTRODUCTION

With the popularization of mobile devices and the maturity of real-time positioning technology, the trajectories of people and vehicles are frequently recorded, resulting in a large amount of position-based data which mainly comes from mobile phones [1], public bicycles [2], public transportation cards [3] and taxis [4]. These data is taken from snippets of everyday life, reflecting both individual behavior and city dynamics. Recently the emerging idea of utilizing these data to promote the construction of smart city draws considerable

attention, among which the modeling of travel time estimation (TTE) serves as a key building block. TTE is the fundamental problem in traffic monitoring, route planning, navigation and taxi dispatching. Having an accurate prediction of the travel time for a given path not only provides convenience to the individual in travel scheduling, but also helps the traffic control center in alleviating road congestion.

However, although the problem has been widely studied in the past, ranging from Bayesian inference [5], Markov chains [6], to deep learning models [7, 8], they require the travel data from all users which, may contain sensitive identifiable information such as driving habits and locations, to be collected and then trained a model in a centralized way. These conventional methods are faced with severe privacy challenges with the increasing awareness of data security, since one must take the risk of privacy leakage when sharing his own travel information to public organizations.

Fortunately, recent progress in federated learning [9] has shed light on such privacy issues. The fundamental idea of federated learning is to keep user's data locally instead of sharing while collaboratively training a global model by exchanging only training parameters (i.e., model weights). More specifically, the centralized learning mechanism aggregates all clients' data to the cloud sever and trains a centralized model before different clients download it. On the contrary, Fig. 2 (a) reveals the federated learning policy, which is proposed to train a local model with local data independently at each client/device, and then sends their model weights to the server where the updates from different clients are integrated into the global model. The global model later in turn sent its weights to update all local model for each device/client. Ideally, the global model obtained from federated learning has similar or better results than the model obtained from centralized data training on a central server [10].

Learning from non-IID data is one of the major challenges of federated learning, particularly in the case of TTE, since the travel styles and active areas vary from person to person, resulting in a highly non-IID dataset for each individual user. While the centralized learning procedure shuffles and mixes the entire dataset to draw batches that hold almost the same data distribution, federated learning suffers when local data is non-IID [11]. Therefore, the global model trained through federated learning might not perform well for each client in the presence of statistical data heterogeneity (i.e., the non-IID data from different clients). We adapt the idea from [12, 13] to train an extra personalized model for each client to study his/her personal driving habit (i.e., encodes it as latent representation). The personalized model is trained towards the

Hongjun Wang, Zhiwen Zhang, Jinyu Chen, Zipei Fan, Xuan Song and Ryosuke Shibasaki are with the SUSTech-UTokyo Joint Research Center on Super Smart City, Department of Computer Science and Engineering and Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology (SUSTech), Shenzhen, China. E-mail: {wanghj2020,11811810}@mail.sustech.edu.cn; zhangzhiwen@csis.u-tokyo.ac.jp, fanzipei@iis.u-tokyo.ac.jp and songx@sustech.edu.cn.

Zipei Fan, Zhiwen Zhang, Xuan Song, and Ryosuke Shibasaki are also with The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, 277-8561, Japan; emails: fanzipei@iis.u-tokyo.ac.jp; zhangzhiwen@csis.u-tokyo.ac.jp; songx@sustech.edu.cn and shiba@csis.u-tokyo.ac.jp

† Corresponding to Zipei Fan;
* Zhiwen Zhang, Hongjun Wang equal contribution;

Fig. 1. The application of global traffic state in real life. The traffic state is shown in different colors where 1) red - very congested 2) yellow - congested 3) orange - slow, and 4) green - unblocked. A comprehensive perception of the global traffic state helps a lot when planning a route or estimating travel time.

residual error between ground truth and the prediction from the global model. With the cooperation of both a global and personalized model, the final result can be more "personal" and thus more accurate.

Moreover, as mentioned by [14], travel time relies heavily on two parts: 1) the statically spatial features of the city road graph (e.g. number of traffic lights, number of lanes, and speed limit), and 2) the dynamically changing real-time traffic condition (i.e., the traffic conditions at different time slots are temporally-variant due to many random variables such as the daily activities of people, road works, and accidents). *Li et al.,* [14] proposed to generate a dynamic grid-based global traffic state by combining these heterogeneous factors (both spatial and temporal) together to improve the prediction accuracy. Nevertheless, we argue that it has two important drawbacks: 1) the grid-based traffic state suffers from its coarse-grained characteristics, lacking the modeling for the state of roads, where the travel really happens. 2) its real-time traffic state is calculated manually using time slots and sub-trajectories rather than using machine learning models to generate, so it is less likely to contain the hidden and high-level traffic information such as traffic accidents. Instead of introducing another new module, we directly design our global federated learning model to be an online generative model that generates the real-time road-based traffic state tensor, inferring the estimated current travel time for each road segment and road intersection. We believe that the real-time traffic state can provide us with more timely traffic information that can have a significant improvement in our prediction. Specifically, we design a spatio-temporal cross product to produce the traffic state by fusing the spatial representations (both road segments and intersections) and the temporal embedding representations.

Additionally, in the application, the global traffic state addresses even more importance in the field of the intelligent transportation system. It contributes to traffic control, route planning, vehicle scheduling, and other tasks, and plays an important role in alleviating traffic congestion and ensuring public transportation safety. Recently, many companies, such as DiDi [15, 16], Baidu [17, 18] and Amap [19, 20], have applied various deep learning methods to traffic state prediction, including studying global traffic states from myriad trajectories

[14]. Thereby, our generative design in federated learning satisfies the interests of both the service provider and individuals, allowing the former to monitor the global traffic state without invading the personal privacy of city residents and providing the latter with convenience. Fig. 1 shows one famous case from the view of both the government and individuals. For the government, the traffic state improves service efficiency, such as ride-sharing services, taxi scheduling, and TTE. For people, the traffic state announces the urban communication efficiency in real-time, which provides tips for the personal daily trip plan.

Summing all mentioned work together, we name our framework as GOF-TTE, **G**enerative **O**nline **F**ederated Learning Framework for **T**ravel **T**ime **E**stimation. The overall structure is shown in Fig. 2 (b) and the detailed global and personalized model design is shown in Fig. 5. The contribution of this paper can be summarized as follows:

- To the best of our knowledge, we are the first to design an online generative federated learning system to estimate the travel time. The online system can provide a more accurate estimation because of its real-time perception of the global traffic state. The traffic state generated via federated learning captures high-level traffic information that companies can collect without privacy leakage.
- Considering non-IID caused by personal driving habits and the inconsistency with data among clients, we propose a personalized federated learning strategy, which reserves a fine-tuned personalized model in the client's device and shares a global model among clients.
- A practical privacy attack has been given in the travel time estimation task, and the differential privacy technology has been applied when the clients upload his weights to servers.

The remainder of our paper is summarized as follows: Sec. II introduces several basic conceptions (e.g., road Network, route, differential privacy) for our proposed GOF-TTE. Sec. III provides our proposed federated framework and data preprocessing. Sec. IV describes our model design and the privacy-preserving technique in detail. Sec. V conducts extensive experiments to evaluate our proposed framework. Sec. VI systematically discusses the related work in TTE algorithms, federated learning architecture, and privacy-preserving methods. Sec. VIII finally summaries the conclusion and discusses the future works.

II. PRELIMINARIES

In this section, we will briefly introduce the definition of urban road network, route, travel time estimation problem and differential privacy technology.

Definition 2.1 Road Network. A road network is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \pi_v, \pi_e)$, where $\mathcal{V}$ denotes the set of nodes (i.e. road intersections) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges/links (i.e. road segments). π serves as a feature function that can be applied to both $v \in \mathcal{V}$ and $e \in \mathcal{E}$. $\pi_v(v)$ denotes the features of v, such as its speed limits and whether it has a traffic light. $\pi_e(e)$ denotes the features of e, for example, its road types, road length and the number of road lanes.

Definition 2.2 Route. A route $r_i : e_1 \to v_1 \to e_2 \to \cdots \to v_{|r_i|-1} \to e_{|r_i|}$ is an alternating sequence with links and nodes.

Definition 2.3 Travel Time Estimation (TTE). Given a route r_i, the task of travel time estimation is to infer the overall traveling time $\tilde{y}$ necessary for the route in the city.

Definition 2.4 (Differential Privacy [21]) A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two adjacent inputs $d, d' \in \mathcal{D}$ and any subset of outputs $S \subseteq \mathcal{R}$, there holds

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta,$$

where $\Pr$ denotes the probability space that is over the coin flips of the mechanism $\mathcal{M}$. Intuitively, differential privacy-based noise is added to the query results so that an attacker cannot tell whether a particular text is in $\mathcal{D}$. In addition, we say that $\mathcal{M}$ is ε-differentially private if $\delta = 0$.

III. FRAMEWORK DESIGN

A. Framework Overview

Under the background of the contradiction between the data island phenomenon and data fusion demand, federated learning has been proposed to train the model without sharing data but only exchanging training parameters in the middle stage. Ideally, the shared model obtained from federated learning has similar or better results than the model got from centralized data training on a central server. We here compare the differences between the classical federated system and our proposed framework. The direct deployment of a federated TTE framework (Fig. 2 (a)) for map providers (such as Amap and Google) will lead to several challenges listed as follows:

1. In travel time estimation, due to different driving habits and preferences of each driver, the privacy data among drivers has the non-IID characteristics, so the local trajectory data distribution for each user can vary with the user's locations and preferences.

2. As we mentioned, the global traffic state plays an important role for both the problem of estimating travel time [22, 14] and the companies interests. How to learn the traffic road state from the individual remains a challenge.

3. The direct upload and aggregation of the localized global model parameters may also leak the individual private information in the TTE task.

Our architecture (Fig. 2 (b)) can ensure that all participants will not disclose personal information to the server, and the server is only responsible for the secure aggregation of local models' parameters, and sends the results to all clients, ensuring that the large family can share a joint trained model. Repeating this process until the loss function converges completes the training of the entire joint model. As shown in Fig. 2 (b), the detailed descriptions of this system are listed as follows:

1. Each client first trains the localized global model with local data, and then trains the personalized model towards residual error.

2. The clients upload their localized global model's parameters with differential privacy to server.

3. The server performs security aggregation.

4. The companies/organizations obtained the current global traffic state from the global model.

5. The server sends the results of the aggregation back to the clients.

6. Participants update their localized global models and repeat Step 1.

B. Data Preprocessing

1) Taxi Trajectories.: We use two public taxi trajectory datasets from the DiDi Express platform. Xi'an dataset was generated by about 18,000 online car hails in October 2016 in Xi'an, China. Moreover, the Chengdu dataset comprises 9 737 557 trajectories of 14864 taxis in August 2014 in Chengdu, China. The GPS points of both datasets have been mapped to roads via a map-matching tool FMM [23] and the time interval of sampling GPS points is 2-4s, ensuring that the vehicle trajectories can correspond to the actual road information.

2) Map Gridding: We employ standard quantization to divide the urban map into multiple uniform sub-regions to understand the driver's preferences. Formally, we divide the geographical space of the city into multiple areas that do not intersect.

3) Road Network Feature Extraction: This paper uses two road networks: Chengdu Road Network and Xi'an Road Network. Both of them are extracted from OpenStreetMap. Xi'an road network contains 4780 edges (i.e., road segments) and 3832 nodes (i.e., road intersections), and the Chengdu road network contains 8221 edges and 5182 nodes. To take care of the privacy-preserving concerns and to generate an unbiased traffic state, we here extract the urban road network features from OpenStreetMap (OSM)[1], which is a free, open-source, and editable map service jointly created by the public network. Furthermore, we extract the road segments features from *highway tag*, which are listed as follows:

- *Road types*: freeway, trunk, primary, secondary, etc.;
- *special types* : whether it is bridge, tunnel, footway, etc.;
- *Otherwise features*: speed limits, length of road, width of road , number of lanes, etc. ;

and intersections from *node tag*, such as

- *Tags of node*: e.g. speed camera, traffic signals, crossing signs, turn circle and bus stop;
- *Junction types*: e.g. X-juction, T-junction and 5-way junction:
- *Otherwise features*: e.g. unique ID and GPS coordinates;

4) Drivers' Profile Feature Extraction: Most existing literature tries to understand and analyze driving behavior [24, 25]. However, intuitively, driving behavior is also an essential factor in the problem of TTE. For example, novice drivers are unfamiliar with the urban road network and driving skills. So their expected travel time tends to be longer than experienced drivers. Based on this assumption, we extracted the following drivers' profile features.

$c_{d,1} \& c_{d,2}$: Break start & end time. Each taxi driver could take a break during work. The driver's preference for break and end time can speculate about his/her usual working time.

---

[1]https://www.openstreetmap.org/

(a) Conventional federated learning framework.

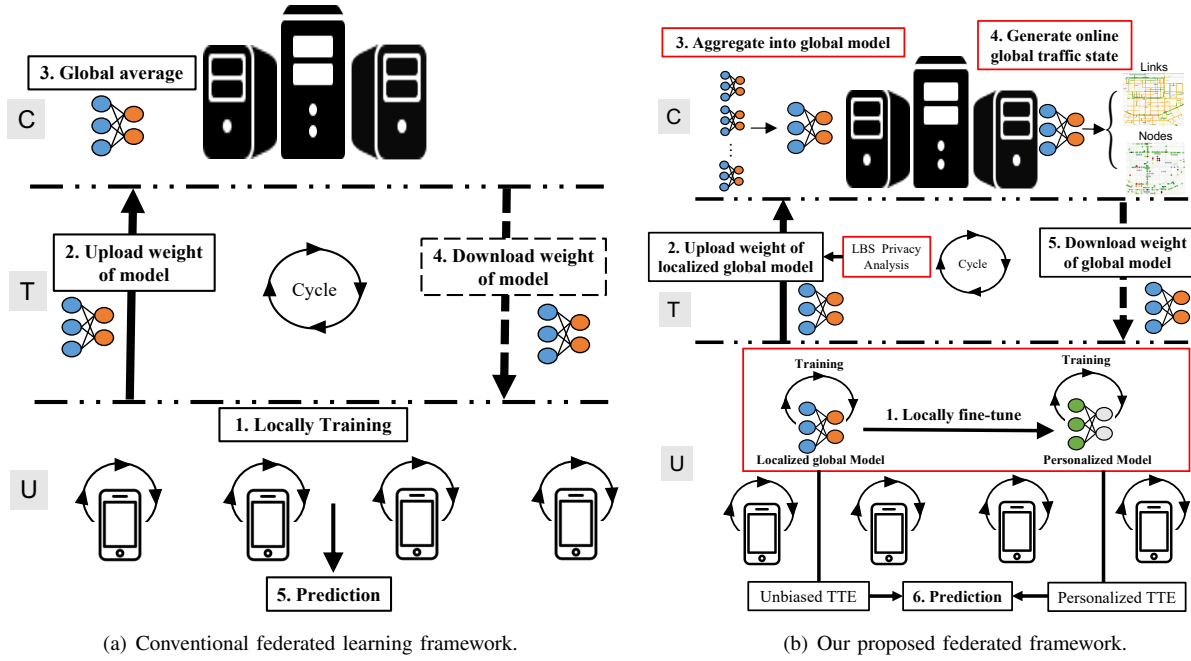(b) Our proposed federated framework.

Fig. 2. Workflow comparison between the directed federated learning system and our proposed framework. "C", "T" and "U" in figure denote the cloud server, transfer environment, and mobile devices (clients) respectively.
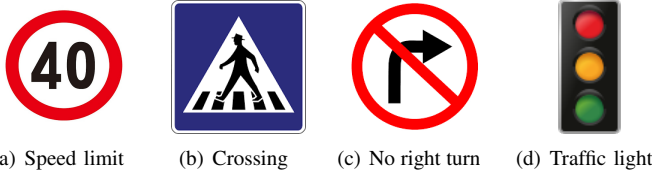

(a) Speed limit   (b) Crossing   (c) No right turn   (d) Traffic light

Fig. 3. The sample cases of *tags of intersection (node)* in OSM.

As a matter of experience, the longer you work, the slower you drive. The distributions of both two features are shown in Fig. 4(a) and Fig. 4(b), respectively. Here, the x-axis is the average break start & end time for each day.

$c_{d,3} \& c_{d,4}$: Frequently visited regions and roads. Different regions and road segments have different road conditions. Therefore, we can speculate on the driver's preference and threshold for congestion. For example, if a driver always visits the downtown region at the morning peak, we guess he/she does not avoid the congestion.

$c_{d,5}$: Average driving distance. Taxi drivers can have his/her preference for the distances of served trips. Some drivers could prefer to choose the orders for long trips because each trip can earn much more. Other drivers could intend to choose short trips due to the relaxation and convenience. Fig. 4(c) presents the drivers' average driving distance for each order. From this point, we know whether one driver prefers the long-distance order or not. Generally, the longer you drive, the slower you go.

$c_{d,6}$: Number of trips served. Each driver has his/her strat-egy for looking for passengers. Fig. 4(d) provides the drivers' number of orders for each day. This is an indicator to distinguish the novice and experienced drivers. In general, the travel time for experienced drivers is shorter.

From the Fig. 4, we can find that taxi drivers obviously have different driving habits, which could influence their estimation performance.
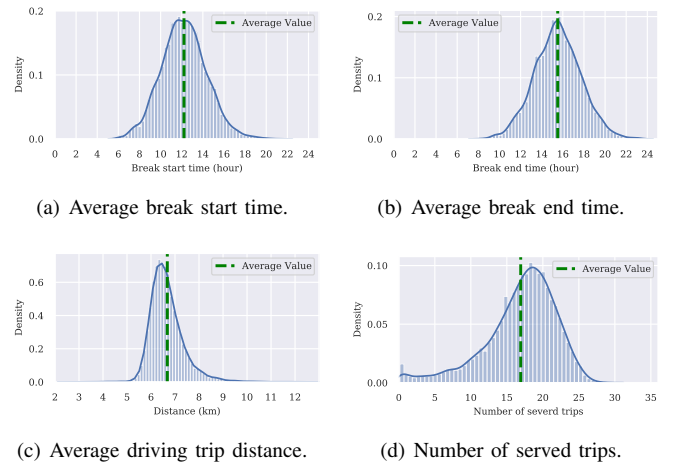

(a) Average break start time.

(b) Average break end time.

(c) Average driving trip distance.

(d) Number of served trips.

Fig. 4. Drivers' profile feature analysis.

IV. METHODOLOGY

The detailed base and personalized model design is shown in Fig. 5. The base model is shown at the top of the figure,
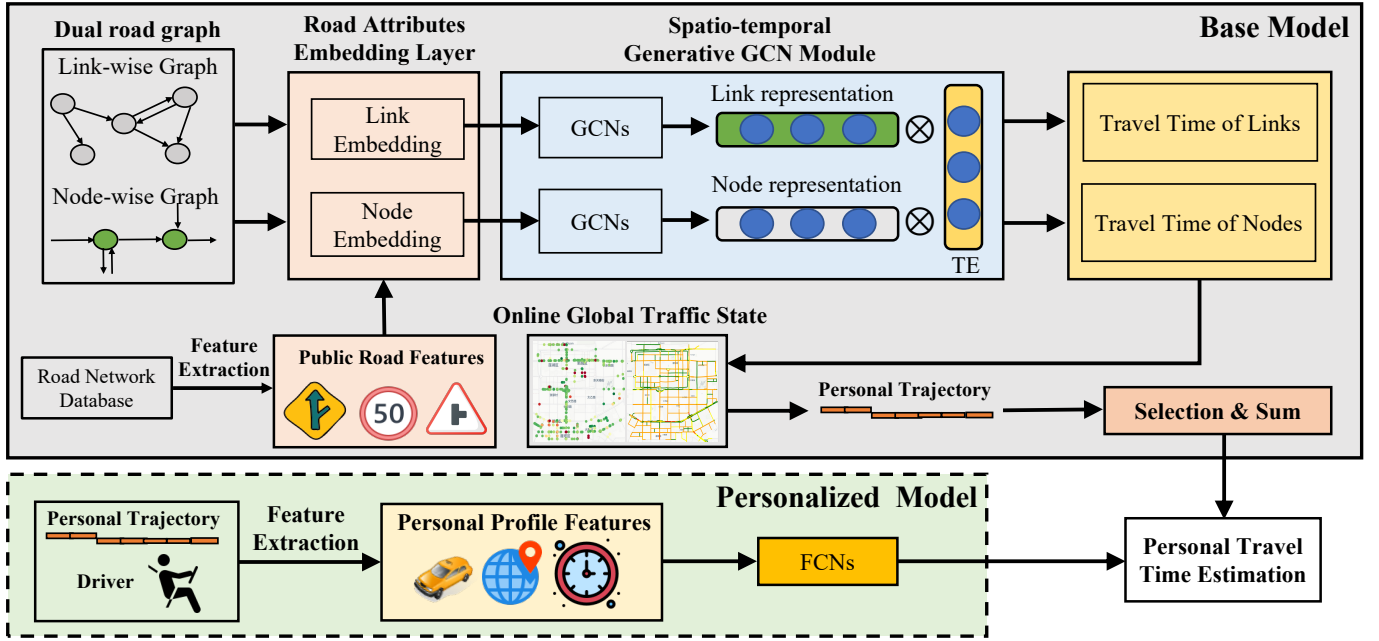
Fig. 5. The model design of our federated learning framework that can be divided into two parts: a base model and a personalized model. Here, TE: temporal embedding; FCNs: fully connected networks.

which generates the online global traffic state for the cloud servers and infers the non-personalized travel time for each client.

In the subsequent sections, we will introduce our proposed base model, personalized model, privacy-preserving mechanism, and online training procedure.

A. Base Model

In this study, the base model mainly provide the functions of two parts: 1) **Localized global model**: this is the well-trained version of base model after the local training for each client, which could also provide non-personalized TTE for each client; 2) **Global model**: the global model is the aggregated model after each round of communication, which aims to generate online global traffic state for cloud server, including both road segment-wise and intersection-wise road network.

1) Road Attributes Embedding Layer: To accurately estimate the travel time, the spatial feature of the city road network is very important. To get there, we embed both the features of roads segments and road intersections into the model through an embedding layer. Fig. 5 shows an overview of this procedure. The input of the layer is a dual road graph, where link-wise graph and node-wise graph characterize the dependencies of road segments and intersections respectively.

2) Spatio-temporal Generative GCN Module: In this sub-section, we will introduce our spatio-temporal generative module, which contains the spatial GCNs layer and a spatio-temporal cross-product layer. The former aims to capture the spatial dependencies of the road network and the latter aims to generate the global state comprised of road segments and intersections, respectively.

Spatial GCNs Layer. Recently, a bunch of works on Graph Convolutional Network (GCN) [26] have been proposed to capture the complex relations in the road network [27]. In this study, we employ the GCN into our framework to process the public features from road network. Formally, given a graph signal $h^{(l)} \in \mathbb{R}^{N \times d^{(l)}}$, where N denotes the number of samples (links/nodes) and $d^{(l)}$ is the features size in l^{th} layer, a typical formulation of graph convolutional layer with C-hop is

$$GCN_{\mathcal{G}}(h^{(l)}; W^{(l)}, \theta^{(l)}) = \sigma\left(\sum_{c=0}^{C} \theta_c^{(l)} L^c h^{(l)}\right) W^{(l)}. \quad (1)$$

In our case, the $h^{(l)}$ stands for the hidden state of (links/nodes) from the output of the *Road Attributes Embedding Layer*.

Spatio-Temporal Cross Product Layer. In fact, the global traffic state varies from time to time (e.g., the changes in non-rushing and rush hour). Meanwhile, the online system requires a time-variant feature representations to update the current traffic state. To solve this problem, we model the temporal dynamically correlations to combine with the static spatial representations. More specifically, we encode the day-of-week and time-of-day of each time step t_k into $\mathbb{R}^7$ and $\mathbb{R}^{\mathcal{K}}$ using one hot encoding, and concatenate them with the embedding of HolidayID (holiday or not). With the aforementioned background of TTE in federated learning, we here consider the task of base model is to recover the real traffic state, which can be formulated as the tensor completion problem, in which the two modal components (both spatial and temporal aspects) have been widely modeled and analyzed [28, 29]. Since the urban road traffic state has a typical spatio-temporal distribution characteristics [27], which can be divided into two levels of intersections and road segments in spatial aspect, and into time-of-day, day-of-week and HolidayID in temporal

aspect. The structured time-space data of traffic state can be organized into the form of multidimensional tensor, which lays a foundation for the application of tensor theory in data completion [30]. In addition, a temporal attention mechanism is also employed to capture the relations of adjacent time steps. Let one day be into K time steps by the time interval. Then the model reconstructs the road traffic state by employing the 1st order CP decomposition between the hidden state $h^{(l+1)}$ generated by the GCNs and the temporal embedding $\mathcal{T} \in R^{K \times \mathcal{I}}$, where $\mathcal{I}$ is the size of temporal embedding vector. Thus, we can obtain the output of spatio-temporal cross product layer:

$$
\begin{aligned}
Y_e &= \left(h_e^{(l+1)} W_{\mu_e} + b_{\mu_e} \right) \otimes \text{att}(\mathcal{T}) \\
&= \left(h_e^{(l+1)} W_{\mu_e} + b_{\mu_e} \right) \otimes \frac{\exp\left(\mathcal{T}_{t_k}\right)}{\sum_{t_k}^K \exp\left(\mathcal{T}_{t_k}\right)}
\end{aligned}
\tag{2}
$$

$$
Y_v = \left(h_v^{(l+1)} W_{\mu_v} + b_{\mu_v} \right) \otimes \frac{\exp\left(\mathcal{T}_{t_k}\right)}{\sum_{t_k}^K \exp\left(\mathcal{T}_{t_k}\right)}
\tag{3}
$$

where $W_{\mu_e}, W_{\mu_v} \in \mathbb{R}^{d^{(l+1)} \times \mathcal{J}}$, $b_{\mu_e} \in \mathbb{R}^{|\mathcal{E}|}$ and $b_{\mu_v} \in \mathbb{R}^{|\mathcal{V}|}$ are the parameters in the fully connected layer to reconstruct the output of spatial GCNs layer. Based on the spatio-temporal cross product of spatial feature vectors $h^{(l+1)} W_\mu + b_\mu$ (for both nodes and links) and the temporal attention-based vector $\text{att}(\mathcal{T})$, $Y_e^{t_k} \in \mathbb{R}^{|\mathcal{E}|}$ and $Y_v^{t_k} \in \mathbb{R}^{|\mathcal{V}|}$ are the outputs of the base model and denoted as the current traffic state of links and nodes at the t_k time step, respectively.

3) Localized Global Model Prediction: Since we have obtained the estimation time of road segments and intersections from Eq. (3), next we will introduce how to generate the non-personalized travel time-based prediction for each client. With the aforementioned definition of the route in Sec. II, a route r in this study is an alternating sequence of links and nodes, which occurs at the time step t_k. Therefore, we have the localized global output of this route:

$$
\widehat{y} = \sum_{e_i \in r}^{|r|} Y_{e_i}^{t_k} + \sum_{v_j \in r}^{|r|} Y_{v_j}^{t_k}
\tag{4}
$$

After local training of the base model based on the trajectory data of each client, we can acquire the localized global model and then upload this trained model to cloud server, which is used to conduct federated model aggregation.

4) Online Global Traffic State Generation: When the cloud server collects weights of the trained localized global model from each user/client, we here aim to generate an online global traffic state based on the aggregated global model. Fig. 6 shows a schematic of our aggregation process for the online global traffic state, which provides an extensible and flexible method to infer the online average travel time for each road segment and road intersection from the road network. Technically, we divide the time of day into five parts: 1) Midnight hours (0:00–7:00); 2) Morning rush hours (7:00–9:00); 3) Non-rush hours (9:00–17:00); 4) Evening rush hours (17:00–19:00), 5) Evening hours (19:00–24:00). This design could avoid the online aggregation process's high computation and communication cost based on the relatively

stable traffic state in the non-rush hours. In our case, let Δ be a parameter indicating the online aggregation frequency, we set the time interval $\Delta=0.5$ hour during 7:00-9:00 (frequently changing traffic state in the morning peak), $\Delta=2$ hours during 9:00-17:00 (stable traffic state), $\Delta=0.5$ hour during 17:00-19:00 (evening peaks), $\Delta=2$ hours during 19:00-23:00 (stable traffic state) and $\Delta=4$ hours during 23:00-7:00 (sleeping time). Our online learning process for global traffic state can be viewed as a repetitive prediction process, as shown in Fig. 6. For example, selected users take part in the training based on their trajectories during 07:00-07:30 and upload their trained localized models to the server. When clients send a TTE query for their trajectories during 07:30-08:00, the cloud server will directly provide each client's global traffic state of the last time step for non-personalized estimation through Eq. (4). In addition, our proposed online aggregation also avoids the missing value problem due to the observation uncertainty or limited reports from crowdsourcing systems. Formally, the task of online global traffic state generation can be described as inferring the citywide travel time for each road segment and intersection by acquiring the fusion result $Y_e \in \mathbb{R}^{|\mathcal{E}|}$ and $Y_v \in \mathbb{R}^{|\mathcal{V}|}$ from the aggregate global model. Relying on the spatio-temporal feature representation of the global model, the server can acquire the complete road condition, which accords with the structure of the road network.

B. Personalized Model

Due to the various driving habits of people, the trajectory data collected by different mobile users are very likely to be unevenly distributed. Therefore, many common optimization algorithms with the assumption of IID data are not suitable for our distributed traffic prediction for the mobile user group. If we directly use the localized global model to predict travel time for clients, there will be bias which reduces prediction accuracy [11]. In this work, we introduce a personalized model for each client to study their personal driving habits. We here use the drivers' profile features described in Sec. III-B. For these dense features, for example, $f_{d,5}$, $f_{d,6}$, and sparse features, for example, $f_{d,1}, f_{d,2}$, we apply a linear transformation and embedding technique, respectively. After embedding the profile features, we concatenate the above hidden states and employ a fully connected layer to produce the personal travel time bias as

$$
\widehat{bias} = W_u^{(l)} X_u^{(l)} + b_u^{(l)}
\tag{5}
$$

where $X_u^{(l)}$ is the hidden state of personal features in l^{th} layer. $W_u^{(l)}$ and $b_u^{(l)}$ are the parameters of the fully connected layer in l^{th} layer. We consider the final output to be the cooperation between the localized global model and the personalized model. The former produces the non-personalized prediction and the latter makes up for the residual error between the ground truth and the former's output.

C. Privacy-preserving Mechanism

1) Privacy Attack: In distributed machine learning, both the centralized worker-server architecture and the decentralized
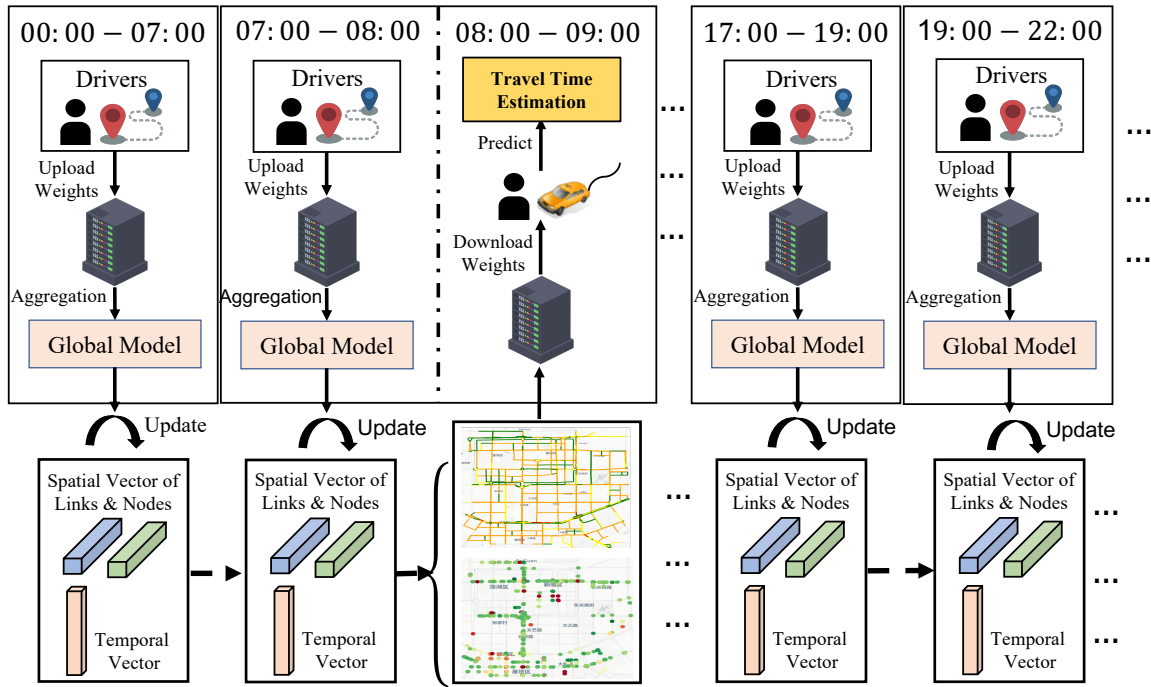
Fig. 6. Online traffic state aggregation process of our proposed system.

all-Reduce architecture involve the sharing of local gradients. In the former, worker gradients are visible on the server, while in the latter, neighbor gradients are shared. Recently, a bunch of work [31, 32, 33] has raised people's awareness about the security of gradients. In our case, since the proposed model architecture is composed of a base model and a personalized model, an intuitive privacy attack approach is to compare the differences between the base model's parameters delivered from server at $t-1$ round and current parameters uploaded by client device at t round (Fig. 7), which is based on the assumption that the difference between localized global model parameters and the global model parameters is approximately equal to the differences between gradients [31]. Fig. 8 represents two cases of privacy attack by using difference attack. We show the top 500 road segments with the largest change in the link embedding layer between the global model and the localized global model. Through the figure, we can see that the privacy attack achieves 52.60% and 83.50% overlap respectively with the ground truth.

2) Model Privacy-preserving with Differential Privacy: Differential privacy [34] defined in Sec. II can solve two defects of traditional privacy protection methods. First, the differential privacy-based protection method assumes that the attacker can obtain information about all records except the target record, and the sum of this information can be understood as the maximum background knowledge that the attacker can master. Under this maximum background knowledge assumption, differential privacy-based protection does not need to take into account any possible background knowledge possessed by the attacker, because this background knowledge is unlikely to provide more information than maximum background knowledge. Second, based on a solid mathematical foundation, differential privacy technology has a strict definition of privacy
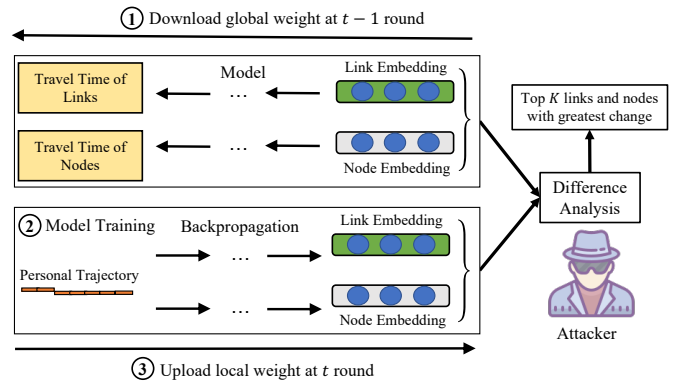

Fig. 7. Schematic of privacy attack in base model.

protection and provides a quantitative evaluation method, which makes the privacy protection level with different parameter settings comparable. In the final, there is no doubt that the privacy protection method could influence the model performance, and we need to achieve a good balance between the privacy protection and model prediction performance.

More specifically, we aim to evaluate the privacy risk based on differential privacy and to analyze the trade-off between model performance and security guarantee. In this work, we construct the noisy data with a differential privacy-based Laplace mechanism, and add them into the weight of the local globalized model when each client uploads their trained models onto a cloud server. The Laplace mechanism preserves $(\varepsilon, 0)$-differential privacy, which perturbs each coordinate with noise drawn from the Laplace distribution.
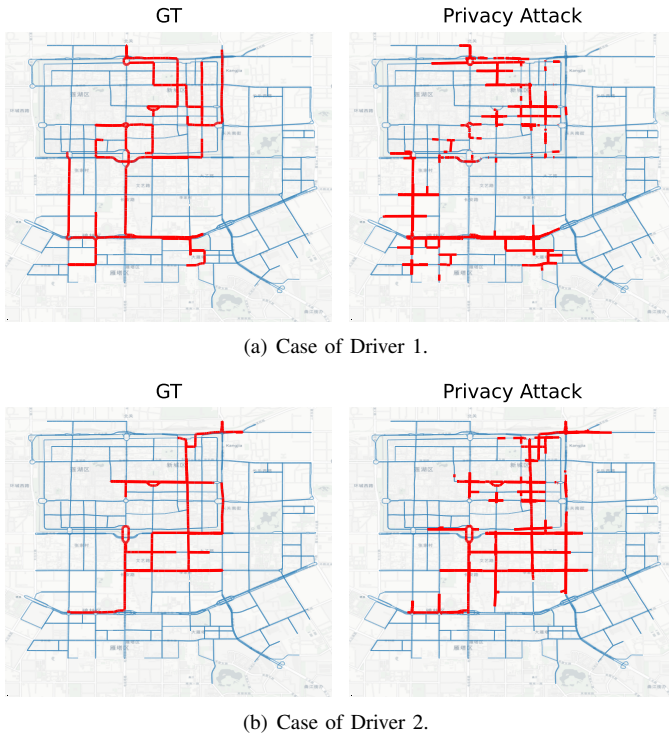
(a) Case of Driver 1.

(b) Case of Driver 2.

Fig. 8. The comparisons between the ground-truth road segments and the privacy attack result for two driver cases from Xi'an datasets.

D. Model Training Procedure

Step by step training in personalized federated learning is a common strategy [35, 36], which means that it defines the training priorities between the localized global model and the personalized model. In our case, inspired by the federated learning with personalized layers [37], we first train the base model to learn the non-personalized traffic state for each client m, and MSE is chosen as the loss function for TTE. Then the objective function can be defined as

$$\mathcal{L}_{\theta_b} = \sum_{i}^{n_m} \|y_i - \widehat{y_i}\|^2 \tag{6}$$

where n_m, y and $\tilde{y}$ denotes the number of trajectory for client m, the ground truth and estimated value respectively. θ_b are all trainable parameters in the base model. In this study, the stochastic gradient descent (SGD) algorithm as the default optimizer for both base and personalized models, and the FedAvg [9] is set as federated learning strategy. However, another gordian knot is that there is an unequal amount of data between users, since each driver receives different taxi orders and travels different distances every day (Fig. 4). We here utilize a simple method mentioned in [9] that the global model F^{t+1} is equal to the weighted-average sum of all clients: $F^{t+1} \leftarrow \sum_{m=1}^{M} \frac{n_m}{n} f_m^t$, where f_k^t is the local model, α is the learning rate, M is the amount of selected clients and each client owns n_m samples at t round, and n is the total number of samples for all selected clients at t round. After that, we freeze the localized global model and train the fine-tuned personalized model to learn each client/user's personal

driving habits, which is to fit the residual error in Eq. (6) as

$$\mathcal{L}_{\theta_p} = \sum_{i}^{n_m} \|y_i - \tilde{y_i}\|^2, \tag{7}$$

$$= \sum_{i}^{n_m} \left\| y_i - \widehat{y_i} - \widehat{bias} \right\|^2, \tag{8}$$

where θ_p are all trainable parameters in the personalized model, and $\tilde{y} = \widehat{y} - \widehat{bias}$ is the final output of our proposed model (as is shown in Fig. 5). In the final, to protect the client's privacy risk of model weights, we construct the noise based on differential privacy and add them into trained localized global model for each client before uploading them to the cloud server. Then the cloud server can acquire an aggregate global model and conduct the next round of federated learning. Here, our federated training procedure at time step t_k can be shown in the Algorithm 1. It is noted that we do not design complicated structure for the personalized model to avoid the potential overfitting on limited personal trajectory data.

Algorithm 1: Model Training procedure

---

1 **Global parameters:** F^t is the global model at t round; α is the base model learning rate; M is the amount of selected clients.

2 **Local parameters:** f_m is the localized global model for client m; p_m is the personalized model for client m, *epoch* is the number of local epoch; $\mathcal{D}_m$ is the personal privacy trajectory data on local device; ε is the parameter of differential privacy.

3 **Server:**

4 initialize F^0

5 **for** *each round* $t \in \{1, 2, ...\}$ **do**

6     $I_t \leftarrow$ random set of M client's;

7   **for** *each client* $m \in I_t$ ***in parallel*** **do**

8         $f_m \leftarrow$ **ClientUpdate**(F^{t-1}, *epoch*, α);

9   **end**

10     $F^{t+1} \leftarrow \sum_{m=1}^{M} \frac{n_m}{n} f_m^t$

11 **end**

12 **Client Update:**

13 **for** $i \in \{1, 2, ..., epoch\}$ **do**

14     $f_m \leftarrow f_m - \alpha \nabla \mathcal{L}_{\theta_b}(f; \mathcal{D}_m)$ // $\nabla \mathcal{L}_{\theta_b}(f; \mathcal{P}_m)$ obtained from Eq. (6)

15 **end**

16 $f_m \overset{+noise}{\longleftarrow} f_m$ // construct noise with differential privacy ε

17 **return** f_m

18 **Fine-tune** personalized model p_m based on $\mathcal{L}_{\theta_p}$ for each client m before inference.

V. EXPERIMENTS

A. Baselines

In terms of travel time estimation models, we compare our proposed methods with baseline models including DeepTTE, WDR, MURAT, DeepGTT, T-GCN, and ConSTGAT in different training strategies (e.g. Fed-Avg and centralized training).

TABLE I
OVERALL PERFORMANCE COMPARISON UNDER TWO PUBLIC DATASETS. HERE, "+PERSONALIZED" DENOTES THE TRAINING PROCEDURE OF LOCALIZED GLOBAL MODEL AND PERSONALIZED MODEL DESCRIBED IN SEC.IV-D

Training Strategy	Models	Chengdu			Xi'an		
		MAE (sec)	RMSE (sec)	MAPE (%)	MAE (sec)	RMSE (sec)	MAPE (%)
Centralization	T-GCN	201.28	297.59	12.09	225.89	361.91	14.50
	ConSTGAT	199.62	295.53	11.78	187.57	295.33	12.03
	DeepTTE	194.47	293.15	12.36	167.85	266.26	11.28
	WDR	231.22	355.76	15.88	207.52	304.63	14.46
	MURAT	267.83	394.29	17.78	225.39	353.16	14.37
	DeepGTT	251.71	369.26	18.12	247.83	367.25	15.09
Fed-Avg	T-GCN	281.45	393.61	17.25	305.68	457.92	19.34
	ConSTGAT	279.14	390.87	16.82	253.82	373.67	16.04
	DeepTTE	271.93	387.73	17.64	227.14	336.89	15.04
	WDR	353.29	469.52	24.56	291.96	404.72	19.29
	MURAT	369.17	458.16	24.39	311.85	427.68	21.24
Fed-Avg	Localized Global Model	285.53	403.36	18.21	238.37	368.16	15.57
Fed-PA		291.98	411.46	18.27	234.26	356.84	15.69
Fed-Ensemble		272.43	393.94	17.28	229.37	331.41	14.93
+Personalized (ours)		**237.46**	**360.38**	**17.11**	**209.32**	**311.59**	**14.28**

(a) The impact of selected clients on Chengdu

(b) The impact of selected clients on Xi'an

(c) The impact of local epoch on Chengdu
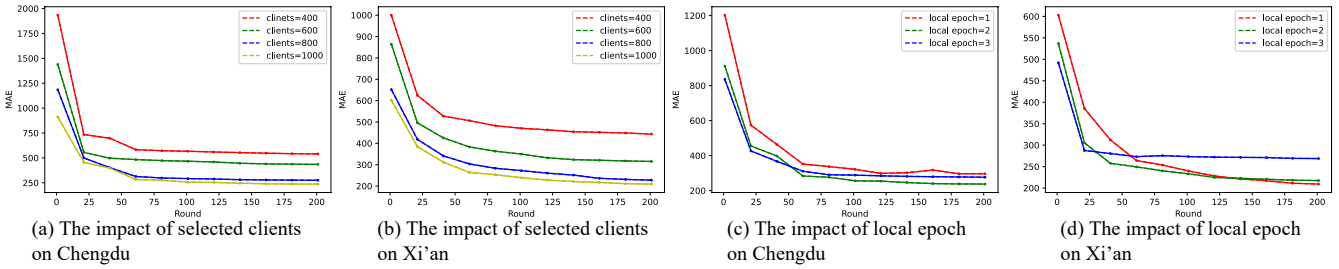
(d) The impact of local epoch on Xi'an

Fig. 9. The effects of the number of selected clients and the local training epoch.

- **DeepTTE** [8] is an end-to-end deep learning framework, which infers the trajectory travel time of both the entire path and each local path simultaneously.
- **WDR** [38] predicts the travel time along a given route at a specific departure time, and then jointly trains wide linear models, and recurrent neural networks together to take full advantage of all three kinds of models.
- **MURAT** [39] is a multi-task representation learning method by utilizing the underlying road network and the spatio-temporal prior knowledge.
- **DeepGTT** [14] learns the travel time distribution through the deep generative model, which takes the real-time traffic condition into consideration.
- **T-GCN** [40] is a temporal graphical model that combines the GCN and GRU to simultaneously characterize the spatio-temporal dependencies.
- **ConSTGAT** [17] adopts a graph attention mechanism to explore the joint relations of spatio-temporal information.

All baseline models but DeepGTT (due to structural reason) are implemented in both centralized training and federated learning strategies. For centralized training, data is shared by different users, and the model trained with shared data succeeds in capturing the universal mobility patterns and achieves better modelling performance. However, privacy issues are completely ignored. And in federated learning, the model is directly trained and executed in personal devices with only local private data. Without sharing users' private

data, this training strategy protects the personal privacy but fails to provide competitive performance. In this paper, we also apply three federated learning strategies to our proposed system. Among them, Fed-PA and Fed-Ensemble are recent personalized federated learning strategies for urban computing.

- **Fed-Avg** [9] In Federated Averaging, a subset of the total devices are selected and trained locally for E number of epochs at each round, and then the updates of resulting local model are averaged.
- **Fed-PA** [41] Feng et al. proposed a personal adaptor to be fine-tuned in the local device for better personal pattern modelling. In our experiment, the time embedding layer is filtered by a trainable same-size vector with sigmoid function using the multiply operation.
- **Fed-Ensemble** [42] Yu et al. proposed an ensemble clustering-based scheme for traffic flow prediction by grouping organizations/users into clusters before applying the Fed-Avg algorithm. In our experiment, we use the average locations of every user's trajectories to cluster users and then implement federated averaging in the corresponding clusters to get the global model set. Finally, the global model is generated by the ensemble learning scheme.

B. Metrics

We evaluate the performance of travel time estimation with RMSE (Root Mean Square Error), MAE (Mean Absolute

Error), and MAPE (Mean Absolute Percentage Error).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}\|\tilde{y}_i - y_i\|^2},$$

$$MAE = \frac{1}{n}\sum_{i}^{n}|\tilde{y}_i - y_i|,$$

$$MAPE = \frac{1}{n}\sum_{i}^{n}|\frac{\tilde{y}_i - y_i}{y_i}|,$$

where n, y and $\tilde{y}$ denotes the number of trajectory samples, the ground truth and estimated value respectively.

C. Experimental Setup

The experiments are implemented with PyTorch 1.6.0 and Python 3.6 and the models are trained with a RTX2080 GPU on Ubuntu 16.04 OS platform. We trained the models using SGD optimizer with an initial learning rate of 0.001 on both Chengdu and Xi'an datasets, and early stopping is applied to avoid overfitting. For the hyper-parameters of federated training, we set clients=1000, local epoch=2 in several federated learning strategies, except for our proposed training strategy *Base + Personalized*. This is because the local training of our prediction model is a two-step training process. Here, we implement local epoch=2 to train the localized global model and local epoch=1 to fine-tuned personalized model.

D. Experimental Results

This section presents the performance comparison with baseline models under different training strategies, including centralized training and several federated learning strategies. Then we analyze the effect of hyper-parameters in the training procedure of our proposed travel time estimation system. Especially, we also conducted a real-world case study in Xi'an, which visualizes the learned global state of our proposed system on the road network.

1) Overall Performance: We compare our proposed methods with different estimation models and training strategy combinations under two real-world datasets. These can be divided into privacy-preserving models under federated training and privacy-leakage models under centralized training. Table I shows the performance of travel time estimation. From these results, we first find that centralized training usually achieves better performance than federated training regarding the same prediction model. On the other hand, the best performances are both achieved by privacy-leakage DeepTTE. Furthermore, we can conclude that our proposed method is superior to other privacy-preserving models. First, our global model combined with the personalized model achieves the best performance among privacy-preserving models with an improvement of 12.67% and 7.85% under two datasets, even though it still has competitive performance compared with the best privacy-leakage models. Second, our federated training procedure also performs better than other federated learning strategies, such as Fed-PA and Fed-Ensemble. Besides, only the global model with Fed-Avg has achieved near the performance of

DeepTTE with Fed-Avg. This means that the trained traffic states of local drivers have extreme reliability. In summary, our proposed methods achieve promising performance while protecting personal privacy in federated training.

2) Simulation Performance Under Different Scenarios: The simulation experiments have been conducted to verify the correctness and usefulness of our online TTE framework. The corresponding result comparisons are shown in Table II. To be clear, we here present the model performance in non-rush hours (0:00-7:00, 9:00-17:00, and 19:00-24:00) and rush hours (7:00-9:00 and 17:00-19:00), respectively. As we have heard, since traffic state tends to be congested during the rush hours, it is a challenge to accurately estimate travel time. We take an example for Chengdu datasets, our proposed framework that integrates with the personalized model, still improves at least 15%, 16%, and 8% in RMSE, MAE, and MAPE. In terms of non-rush hours, our model obtains at least 18%, 6%, and 15% improvement in these three metrics. We can conclude that our model significantly outperforms overall federated strategy and TTE baselines, demonstrating that the generalization of our proposed framework can better adapt to different scenarios.

3) The Effects of Hyper-parameters on Federated Learning: In the experiment, we first analyze the effects of the number of selected clients. In each training round of the proposed system, we only choose a fraction of clients to participate in the current optimization procedure. Fig. 9 (a) and Fig. 9 (b) present the effects of the number of selected clients on the final performance on two datasets. Generally, we can observe that the final performance is improved continuously, with the number of clients increasing from 400 to 1000 for both datasets. This indicates that the generated global traffic states need large amounts of users' trajectories to train. We also study the effects of the local training epoch of the localized global model on the system performance in Fig. 9(c) and Fig. 9(d). We find that local epoch=2 and local epoch=1 are the optimal settings for Xi'an and Chengdu datasets, respectively. This is because Xi'an's road network is simpler than Chengdu, and the training process for the Xi'an dataset needs fewer iterations than Chengdu.

4) Case Study: Our proposed system can not only conduct path travel time estimation for personal trips, but also learn the global traffic state of the links and nodes.

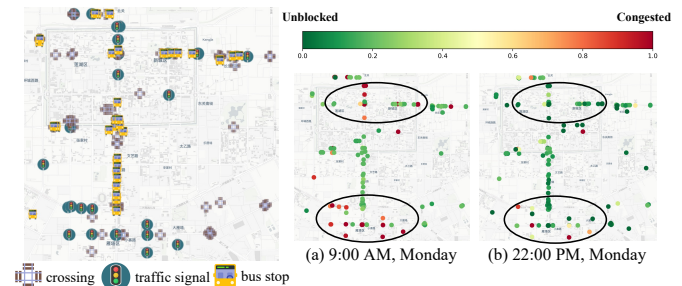

(a) 9:00 AM, Monday   (b) 22:00 PM, Monday

Fig. 10. The time-consuming ratio for some nodes in the Xi'an road network. Here, we select three types of nodes including "crossing", "traffic signal" and "bus stop".

TABLE II
OVERALL SIMULATION PERFORMANCE COMPARISON OF OUR PROPOSED METHOD AND BASELINE STRATEGIES FOR TRAVEL TIME ESTIMATION (ON CHENGDU AND XI'AN DATASET) UNDER TWO SCENARIOS, INCLUDING BOTH NON-RUSH AND RUSH HOURS.

Data	Training Strategy	Models	Non-Rush Hours			Rush Hours		
			MAE (sec)	RMSE (sec)	MAPE (%)	MAE (sec)	RMSE (sec)	MAPE (%)
Chengdu	Fed-Avg	Localized Global Model	314.20	405.26	18.93	347.15	488.16	20.78
	Fed-PA		323.39	413.33	19.24	364.77	507.82	21.04
	Fed-Ensemble		307.27	403.65	18.91	321.32	463.59	19.23
	+Personalized (ours)		**256.92**	**384.18**	**16.17**	**291.46**	**410.33**	**18.75**
Xi'an	Fed-Avg	Localized Global Model	262.275	369.851	16.20	289.782	445.517	17.78
	Fed-PA		269.947	377.217	16.47	304.492	463.461	18.00
	Fed-Ensemble		256.489	368.381	16.18	268.219	423.091	16.46
	+Personalized (ours)		**214.455**	**350.611**	**13.85**	**243.290**	**374.479**	**16.01**

(a) Computed speed by entire trajectory data on Xi'an dataset (Ground Truth)

(b) Global traffic state trained by clients=600

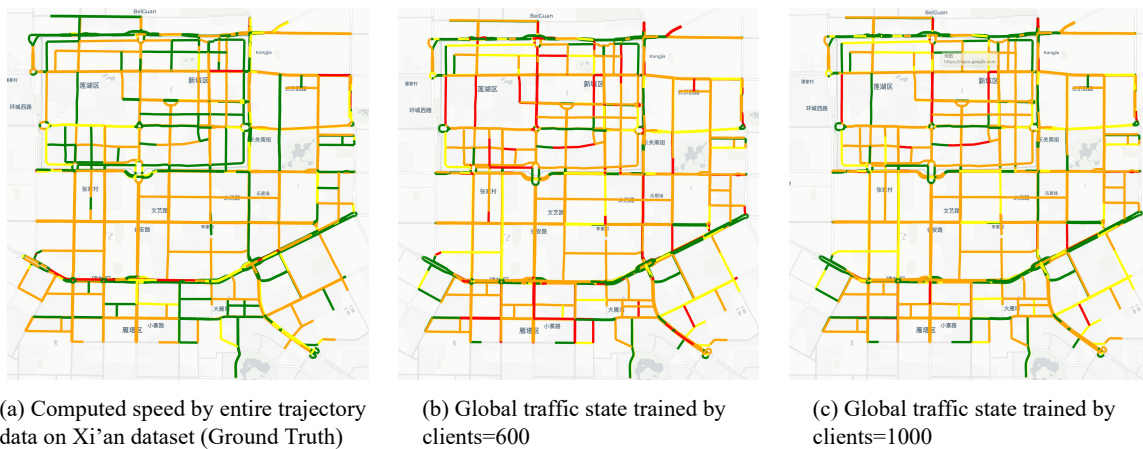(c) Global traffic state trained by clients=1000

Fig. 11. The aggregated global traffic state in Xi'an. Here we use four kinds of colors to represent the different road states, which can be defined as: 1) red - very congested, 2) orange - congested, 3) yellow - slow, 4) green - unblocked.

On the one hand, we first provide the aggregated global traffic state for different nodes, which are depicted in Fig. 10. On the other hand, we select three types of nodes in a road network and use the time-consuming ratio to represent congestion status. The time-consuming ratio is calculated by dividing the corresponding mean value of learned travel time distributions by the maximum mean value among all nodes (note that we have filtered out the top 1% most enormous node travel time). From Fig. 10, we can find that nodes with the "traffic signal" type are more time-consuming than the other two types of nodes, and most nodes in the morning peak are easy to become congested. These show that our node travel time estimation is reasonable in both spatial and temporal aspects.

On the other hand, we depict the aggregated global states of links and compare them with the ground truth computed by the original taxi trajectories. Especially, we compare the aggregated road conditions with two different numbers of selected clients (clients=600 and clients=1000). We mark it with the unblocked state for the road segments without taxi trajectories. Since the speed limit varies from road to road, primarily defined by road type or road length, we use four colors to represent the different road states (very congested, congested, slow, and unblocked). We divide the limiting-velocity for each road type equally. For example, the rate-limiting of the primary road is $60kph$, so the interval between very congested is $[0, 15)$, congested is $[15, 30)$, slow is $[30, 45)$, unblocked is $[45, 60)$. The compared result is shown in Fig. 11. Comparing the aggregated road condition and ground truth, the generated traffic states under the selected client's two settings are similar to the ground truth. Most road segments have the same road states. Furthermore, we find that the more clients participate in the federated aggregation process, the more accurate the global traffic state will be.

E. Privacy Risk Analysis

As the introduction about analyzing the privacy risk of uploading the base model in Sec. IV-C, we conduct the privacy risk analysis for our proposed framework, and show the effectiveness of the protection on local devices. Here, we have two objectives of privacy analysis. One is to validate our local data protection method based on our differential privacy mechanism; the other is to evaluate the effects of differential privacy on the prediction accuracy of travel time.
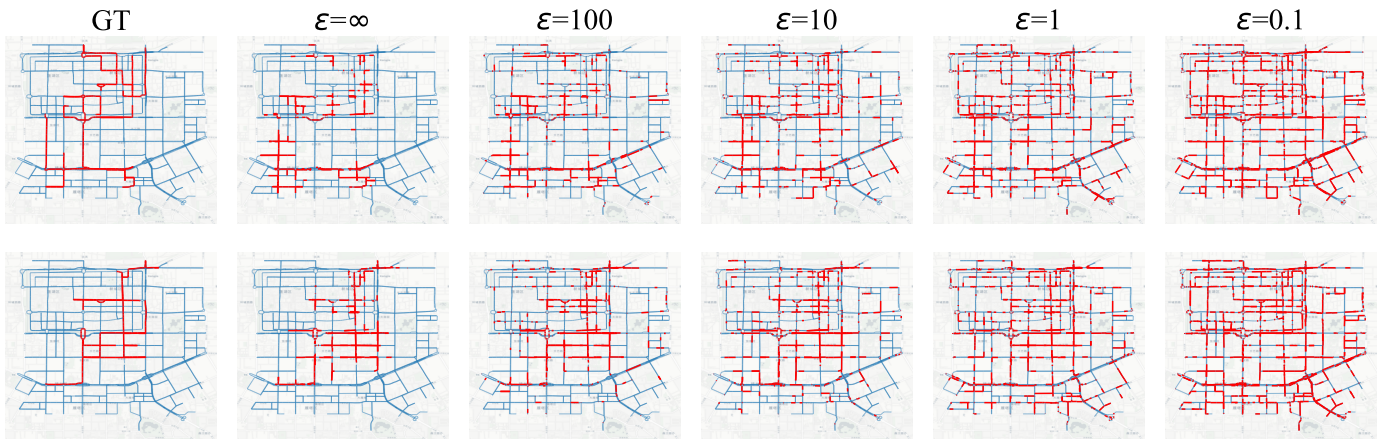
| GT | ε=∞ | ε=100 | ε=10 | ε=1 | ε=0.1 |

Fig. 12. The effectiveness of attack risk with changing differential privacy parameter ε.

(a) Attack results on Xi'an   (b) Attack results on Chengdu
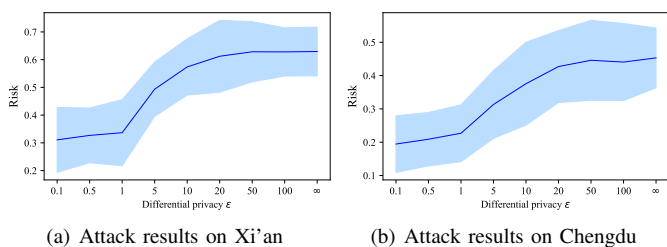
Fig. 13. The privacy attack result with different differential privacy parameter ε.

1) Privacy-preserving Performance: To evaluate the effectiveness of our privacy protection method, we define the privacy $risk = \frac{|\Omega_i \cap \tilde{\Omega}_i|}{|\Omega_i|}$ metrics to measure the privacy-preserving performance with different parameter settings of ε for differential privacy, where Ω_i is the set of personal real travelled road segments and $\tilde{\Omega}_i$ is the set of road segments revealed by privacy attack.

In this study, we measure the privacy-preserving performance from two aspects: a qualitative and quantitative analysis. As shown in Fig. 12, we compare the effectiveness of attack risk with different setting of differential privacy parameter ε, which is the same case with Fig. 8. We can observe that the attack risk can be reduced significantly by utilizing differential privacy mechanisms. Furthermore, we can also see that from left to right, the noise intensifies, and the risk (overlap) declines dramatically. As Fig. 12 shows, it is difficult to attack mobile users' location privacy when the ε reduces to 10. Moreover, when ε is lower than 10, the privacy attack would have unobservable effects, which ensures the user's privacy.. In addition, we conduct the quantitative analysis of privacy risk, and Fig. 13 represents the average risk of overall clients/users in federated training procedure for both datasets, respectively. As Fig. 13 shows, we can also know that the clients are exposed to high risk when the strength of noise is small. Especially, the trend of privacy risk for Xi'an, which is shown in Fig. 13 (a), validates our conclusions.

2) The Effects of Differential Privacy on Prediction Performance: To efficiently protect location-based privacy on the local devices, we expect to utilize differential privacy mechanism in the ubiquitous computing system. However, greater noise always leads to the limited performance of the estimation task. Thus, how to balance the privacy protection and better performance becomes an important topic in ubiquitous computing. Table III presents the effects of differential privacy parameter ε on the travel time estimation performance under both two datasets. We further present the relative drops of estimation accuracy based on the performance without differential privacy ($\varepsilon = \infty$). We can observe that in terms of MAE, the performance in Chengdu and Xi'an gradually decreases when ε reduces from 100 to 10. In Chengdu dataset, the model performance is relatively sensitive when ε reduces from 100 to (1,0.1,0.01), compared with the Xi'an dataset. We suppose this may rely on the different road network structure between two cities. To investigate that, we here also depict the attack risk with changing differential privacy parameter ε in Fig. 12. We can find that the performance of Chengdu datasets that has relatively complex road network structure has low attack risk as Fig. 13 (b). For Xi'an dataset, the varying ε has relatively small influence on prediction performance when ε reduces from 100 to 1, although it has higher attack risk. In summary, our method can achieve better trade-off between the model performance and privacy risk of difference attack, and these ensure the availability of our proposed privacy-preserving mechanism.

TABLE III
MAE COMPARISON OF OUR METHOD WITH CHANGING DIFFERENTIAL
PRIVACY PARAMETER ε.

Dataset\ε	0.01	0.1	1	10	100	∞
Chengdu	367.51	353.24	338.22	297.90	253.15	237.46
Drops (%)	54.77	48.76	42.43	25.45	6.61	0
Xi'an	291.47	284.38	255.54	233.15	213.67	209.32
Drops (%)	39.25	35.86	22.08	11.38	2.09	0

VI. RELATED WORK

A. Travel Time Estimation

Existing approaches of travel time estimation can be classified into four groups: road segment based, path based, deep

learning based and graph neural network based methods.

Road Segment Based and Path Based Methods. The road segment based methods use the speed data sampled from loop detectors to infer each individual road segment's travel time independently [43] but ignore the correlations between the road segments, such as the factors of traffic lights, and left/right turns. Path based approaches have been proposed to resolve those issues, which can roughly be divided into two types: 1) nearest neighbor search [44, 45], which estimates the travel time by averaging the historical trajectories' travel time. 2) trajectory regression methods [7], which predicts the travel time of road segments via public features, for instance, road types, road lanes, road length, etc.

Deep Learning Based Methods. With the development of deep learning in these years, various studies have been designed to enhance the performance of travel time estimation. For instance, DeepTTE [8] presents an end-to-end framework that predicts the travel time of the whole path directly by using geo-convolution on the GPS sequences to capture spatial features. *Li et al.* [14] designs a deep generative model to learn the travel time distribution for each road segment using grid-based traffic conditions and basic road network features. *Wang et al.* [38] models the problem of travel time estimation as a spatio-temporal regression problem by a Wide-Deep-Recurrent learning model [46], and this work considers various features including spatial information (the characteristics of route, such as the road segment and traffic light information), temporal information (such as a month and a day, the holiday indicator), traffic information (such as estimated road condition) and personalized information (such as driver profile).

Graph Neural Network Based Methods. Very recently, the advances of graph neural network have promoted a myriad of work in travel time estimation problem. For example, ConSTGAT [17] employs a graph attention mechanism onto the spatial-temporal features by integrating traffic and contextual information, in which the input feature includes the road segment-based information, historical road condition and departure time of each trajectory. Deepist [47] enhances the power of convolutional neural network to capture moving behaviors embedded in paths by introducing the image of urban in grid level. GraphTTE [48] designs a Multi-layer Spatio-temporal Graph frame, which consists of static and dynamic networks. In particular, the dynamic networks implement the GCN and gate recurrent unit to model the historical traffic characteristics, and the static network employs GCN to model the basic road attributes. Especially, *Hong et al.* [49] transforms the road network into a multi-relational network and introduce a vehicle-trajectories based network to jointly consider the traffic behavior pattern (traffic speed, volume).

However, those graph-based approaches ignore the complex relations between road segments and intersections from road network. Meanwhile, their methods need to model the real traffic dynamics, which is difficult to acquire due to the power and communication limitations of mobile devices. In this paper, we model the adjacencies in both node-wise graph and link-wise graph simultaneously. And the online traffic state can be obtained via our aggregated global model.

B. Privacy Protection Methods

We here discuss the related work from two directions: 1) user data protection mechanism, and 2) federated Learning.

User Data Protection Mechanisms. a series of techniques aiming to protect the user trajectories have been proposed, which can be classified with two categories [50]: 1) real-time continuous location-based services, and 2) publication of historical trajectories. The former intends to protect the scenario, for example, asking the Apps to give the traffic conditions that are 1 km around me. Multiple techniques have been proposed including spatial cloaking [51], mix zones [52] and path confusion [53]. The latter schemes to protect the trajectories, and its related works include clustering-based [54], generalization-based [55], suppression-based [56], and grid-based [57] approaches.

Federated Learning. It is a mechanism to coordinate the joint training model of multiple participants, which is emerging in the era of big data and the development of artificial intelligence technology [10]. Meanwhile, communication cost and communication efficiency have become one of the key bottlenecks of federated learning. A bunch of work has been proposed to analyze how to reduce the communication cost of federated learning [9, 58]. Further, federated learning does not exchange data directly and has a higher privacy guarantee than traditional centralized machine learning training, but federated learning itself does not provide comprehensive and sufficient privacy protection. Recently, researchers have applied differential privacy operation [21] on the whole model in the aggregation stage of federated learning [59].

Personalized Federated Learning. Personalized federated learning [60] differs from traditional federated learning in that it does not require all participants to end up using the same model, but allows each participant to fine-tune the model based on their own data to generate their own unique personalized model. Models often perform better on local test sets after personalized tweaks. A bunch of work has been proposed from meta-learning [61], multi-task learning [62], hypernetwork [63], knowledge distillation [64] to personalized model [65]. For example, the goal of meta-learning [61] is to train highly adaptive models that can be trained to solve new tasks in a small number of samples.And in multitasking learning [62], models among the clients learn together by taking advantage of commonalities and differences between tasks.

In this paper, our proposed federated learning system can be viewed as a case of *base model + personalized model*, which estimates the travel time by fusing the outputs of both localized global and personalized model. Moreover, our aggregated global model can generate real-time road condition, which plays an important role of the intelligent transportation system, compared with single fusion of the previous personalized model strategies.

VII. CONCLUSION

In allusion to the TTE problem, this paper proposes an online personalized federated learning framework to fill the gap between personal privacy and model performance by dynamically updating the global traffic state. Specifically, we

estimate the travel time through the training strategy of the *base model + personalized model*, where the base model serves to produce the unbiased global traffic state for all clients, and the latter is only trained locally to act on fitting the personalized driving habits through the profile features. We evaluate the effectiveness of our proposed framework from three aspects: 1) overall performance compared with state-of-the-art in travel time estimation models and federated learning strategies; 2) a case study depicting the global state from links and nodes, respectively, to show the aggregation performance of our system. 3) privacy risk analysis from both quantitative and qualitative views, respectively, to prove the effectiveness of our privacy-preserving mechanism.

In the future, we will further test the performance with more federated learning strategies, such as clustering the personal driving features together, since the drivers' habits have multiple similarities. Meanwhile, we will also reduce the communication cost since the online algorithm frequently needs to upload and download model weight between clients and the cloud server.

VIII. ACKNOWLEDGMENT

REFERENCES

[1] Francesco Calabrese, Zbigniew Smoreda, Vincent D Blondel, and Carlo Ratti. Interplay between telecommunications and face-to-face interactions: A study using mobile phone data. *PloS one*, 6(7):e20814, 2011.

[2] Neal Lathia and Licia Capra. How smart is your smartcard? measuring travel behaviours, perceptions, and incentives. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 291–300, 2011.

[3] Jon Edward Froehlich, Joachim Neumann, and Nuria Oliver. Sensing and predicting the pulse of the city through shared bicycling. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[4] Gang Pan, Guande Qi, Zhaohui Wu, Daqing Zhang, and Shijian Li. Land-use classification using taxi gps traces. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):113–123, 2012.

[5] Soknath Mil and Mongkut Piantanakulchai. Modified bayesian data fusion model for travel time estimation considering spurious data and traffic conditions. *Applied Soft Computing*, 72:65–78, 2018.

[6] Jiyoun Yeon, Lily Elefteriadou, and Siriphong Lawphongpanich. Travel time estimation on a freeway using discrete time markov chains. *Transportation Research Part B: Methodological*, 42(4):325–338, 2008.

[7] Tsuyoshi Idé and Masashi Sugiyama. Trajectory regression on road networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, 2011.

[8] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng. When will you arrive? estimating travel time based on deep neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[9] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[10] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[11] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[12] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *arXiv preprint arXiv:2103.00710*, 2021.

[13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154, 2017.

[14] Xiucheng Li, Gao Cong, Aixin Sun, and Yun Cheng. Learning travel time distributions with deep generative model. In *The World Wide Web Conference*, pages 1017–1027, 2019.

[15] Yiwen Sun, Yulu Wang, Kun Fu, Zheng Wang, Changshui Zhang, and Jieping Ye. Constructing geographic and long-term temporal graph for traffic forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3483–3490. IEEE, 2021.

[16] Yang Liu, Fanyou Wu, Baosheng Yu, Zhiyuan Liu, and Jieping Ye. Building effective large-scale traffic state prediction system: Traffic4cast challenge solution. *arXiv preprint arXiv:1911.05699*, 2019.

[17] Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2697–2705, 2020.

[18] Xiaomin Fang, Jizhou Huang, Fan Wang, Lihang Liu, Yibo Sun, and Haifeng Wang. Ssml: Self-supervised meta-learner for en route travel time estimation at baidu maps. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2840–2848, 2021.

[19] Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3074–3082, 2020.

[20] Amap tech. Exploration and practice of deep learning application in amap eta. [EB/OL]. https://www.alibabacloud.com/blog/597638 Accessed April 20, 2021.

[21] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer, 2006.

[22] JQ James. Citywide estimation of travel time distributions with bayesian deep graph learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[23] Can Yang and Gyozo Gidofalvi. Fast map matching, an algorithm integrating hidden markov model with precomputation. *International Journal of Geographical Information Science*, 32(3):547–570, 2018.

[24] Saad Ezzini, Ismail Berrada, and Mounir Ghogho. Who is behind the wheel? driver identification and fingerprinting. *Journal of Big Data*, 5(1):1–15, 2018.

[25] Weishan Dong, Jian Li, Renjie Yao, Changsheng Li, Ting Yuan, and Lanjun Wang. Characterizing driving styles with deep learning. *arXiv preprint arXiv:1607.03611*, 2016.

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[27] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[28] Guangyin Jin, Huan Yan, Fuxian Li, Jincai Huang, and Yong Li. Spatial-temporal dual graph neural networks for travel time estimation. *arXiv preprint arXiv:2105.13591*, 2021.

[29] Tobias Skovgaard Jepsen, Christian S Jensen, and Thomas Dyhre Nielsen. Relational fusion networks: Graph convolutional networks for road networks. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[30] Zipei Fan, Xuan Song, and Ryosuke Shibasaki. Cityspectrum: a non-negative tensor factorization approach. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 213–223, 2014.

[31] Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated learning*, pages 17–31. Springer, 2020.

[32] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? *arXiv preprint arXiv:2003.14053*, 2020.

[33] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[34] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[35] Binbin Guo, Yuan Mei, Danyang Xiao, and Weigang Wu. Pfl-moe: personalized federated learning based on mixture of experts. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 480–486. Springer, 2021.

[36] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

[37] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[38] Zheng Wang, Kun Fu, and Jieping Ye. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 858–866, 2018.

[39] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1695–1704, 2018.

[40] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2019.

[41] Jie Feng, Can Rong, Funing Sun, Diansheng Guo, and Yong Li. Pmf: A privacy-preserving human mobility prediction framework via federated learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–21, 2020.

[42] Yi Liu, JQ James, Jiawen Kang, Dusit Niyato, and Shuyu Zhang. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal*, 7(8):7751–7763, 2020.

[43] John Rice and Erik Van Zwet. A simple and effective method for predicting travel times on freeways. *IEEE Transactions on Intelligent Transportation Systems*, 5(3):200–207, 2004.

[44] Dalia Tiesyte and Christian S Jensen. Similarity-based prediction of travel times for vehicles traveling on known routes. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–10, 2008.

[45] Hongjian Wang, Xianfeng Tang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. A simple baseline for travel time estimation using large-scale trip data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–22, 2019.

[46] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[47] Tao-yang Fu and Wang-Chien Lee. Deepist: Deep image-based spatio-temporal network for travel time estimation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 69–78, 2019.

[48] Qiang Wang, Chen Xu, Wenqi Zhang, and Jingjing Li. Graphtte: Travel time estimation based on attention-spatiotemporal graphs. *IEEE Signal Processing Letters*, 28:239–243, 2021.

[49] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. Heteta: heterogeneous information network embedding for estimating time of arrival. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2444–2454, 2020.

[50] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):1–41, 2015.

[51] Mohamed F Mokbel, Chi-Yin Chow, and Walid G Aref. The new casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd international conference on Very large data bases*, pages 763–774, 2006.

[52] Alastair R Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55, 2003.

[53] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Achieving guaranteed anonymity in gps traces via uncertainty-aware path cloaking. *IEEE Transactions on Mobile Computing*, 9(8):1089–1107, 2010.

[54] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *2008 IEEE 24th international conference on data engineering*, pages 376–385. Ieee, 2008.

[55] Mehmet Ercan Nergiz, Maurizio Atzori, and Yucel Saygin. Towards trajectory anonymization: a generalization-based approach. In *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 52–61, 2008.

[56] Manolis Terrovitis and Nikos Mamoulis. Privacy preservation in the publication of trajectories. In *The Ninth international conference on mobile data management (mdm 2008)*, pages 65–72. IEEE, 2008.

[57] Gyozo Gidofalvi, Xuegang Huang, and Torben Bach Pedersen. Privacy-preserving data mining on moving object trajectories. In *2007 International Conference on Mobile Data Management*, pages 60–68. IEEE, 2007.

[58] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[59] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[60] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. Decentralized collaborative learning of personalized models over networks. In *Artificial Intelligence and Statistics*, pages 509–517. PMLR, 2017.

[61] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[62] Canh T Dinh, Tung T Vu, Nguyen H Tran, Minh N Dao, and Hongyu Zhang. Fedu: A unified framework for federated multi-task learning with laplacian regularization. *arXiv preprint arXiv:2102.07148*, 2021.

[63] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hyper-networks. *arXiv preprint arXiv:2103.04628*, 2021.

[64] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[65] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. *arXiv preprint arXiv:2102.07078*, 2021.