FedGRU: Privacy-preserving Traffic Flow Prediction via Federated Learning

Yi Liu†, Shuyu Zhang†, Chenhan Zhang, James J.Q. Yu

***Abstract*— Existing traffic flow forecasting technologies achieve great success based on deep learning models on a large number of datasets gathered by organizations. However, there are two critical challenges. One is that data exists in the form of "isolated islands". The other is the data privacy and security issue, which is becoming more significant than ever before. In this paper, we propose a Federated Learning-based Gated Recurrent Unit neural network framework (FedGRU) for traffic flow prediction (TFP) to address these challenges. Specifically, FedGRU model differs from current centralized learning methods and updates a universe learning model through a secure aggregation parameter mechanism rather than sharing data among organizations. In the secure parameter aggregation mechanism, we introduce a Federated Averaging algorithm to control the communication overhead during parameter transmission. Through extensive case studies on the Performance Measurement System (PeMS) dataset, it is shown that FedGRU model can achieve accurate and timely traffic prediction without compromising privacy.**

I. Introduction

Urban residents, taxi drivers, business sectors, and government agencies have an immediate requirement on accurate and timely traffic flow information [1]. Such information can help the traffic sector to alleviate traffic congestion, control traffic light, and improve the efficiency of traffic operations and residents for developing better traveling plans [2]. Traffic flow prediction (TFP) is to provide such traffic flow information by using historical traffic flow data to predict the future [3]. TFP is regarded as a critical technology of the deployment of Intelligent Transportation System (ITS) subsystems, particularly the advanced traveler information, online car-hailing, and traffic management systems.

In the previous TFP literature, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and their variants have achieved gratifying results in predicting traffic flow. Such centralized machine learning methods are typically utilized to predict traffic flow by training with sufficient sensor data from mobile phones, cameras, radars, etc. In this context, these methods generally require data aggregation among public agencies and private companies. Indeed, the general public witnessed partnerships among public agencies and mobile service providers such as DiDi Chuxing, Uber, and Hellobike in recent years. These partnerships extend the capability and services of companies that provide real-time traffic flow forecasting, traffic management, car sharing, and personal travel applications.

Nevertheless, it is often overlooked that the data may contain sensitive private information (e.g., user's traveling track, home address etc.), which leads to potential privacy leakage. Therefore, different organizations should store their user data locally and avoid exchanges to protect users' privacy, which makes it challenging to train an effective model with the valuable data. While the assumption that an organization owns all the data is widely made in the literature, the acquisition of massive user data is not possible in real applications respecting privacy. To predict traffic flow in ITS without compromising privacy, reference [4] introduced a privacy control mechanism based on "k-anonymous diffusion," which can complete taxi order scheduling without leaking user privacy. Le Ny *et al.* in [5] proposed a differentially private real-time traffic state estimator system to predict traffic flow. However, these privacy-preserving methods cannot achieve the trade-off between accuracy and privacy, rendering degraded system performance.

To address the data privacy leakage issue, we incorporate a privacy-preserving machine learning technique named federated learning (FL) [6] for TFP in this work. In FL, distributed organizations cooperatively train a globally shared model through their local data without exchanging the raw data. To accurately predict traffic flow, we propose an enhanced federated learning algorithm with a Gated Recurrent Unit neural network (FedGRU) in this paper. Through FL and its aggregation mechanism [7], FedGRU aggregates model parameters from different geographically located organizations to build a global deep learning model under privacy well-preserved conditions. Furthermore, contributed by the outstanding data regression capability of GRU neural networks, FedGRU can achieve accurate and timely traffic flow prediction for multiple organizations.

The main contributions are summarized as follows:

- We propose a novel privacy-preserving algorithm that integrates emerging federated learning with a practical GRU neural network for traffic flow prediction. Such an algorithm provides reliable data privacy preservation through a locally training model without raw data exchange.
- We introduce a Federated Averaging (FedAVG) algorithm in the secure parameter aggregation mechanism

†Equal contributions.

This work is supported in part by the General Program of Guangdong Basic and Applied Basic Research Foundation No. 2019A1515011032, in part by Guangdong Provincial Key Laboratory No. 2020B121201001, and in part by the Ministry of Education of China and the School of Entrepreneurship Education of Heilongjiang University No. 201910212133. *(Corresponding author: James J.Q. Yu.)*

The authors 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, China. Yi Liu is also with School of Data Science and Technology, Heilongjiang University, Harbin, China

which runs Stochastic Gradient Descent (SGD) on a selected subset of all organizations to aggregate model parameters to update the global model.
- FedGRU is a standard, stable, and extensible FL framework for ITS. As a privacy-preserving framework, it not only achieves accuracy close to traditional models but also can be combined with other and future state-of-the-art deep learning models.

The remainder of this paper is organized as follows. Section II reviews the literature on short-term TFP and privacy research in ITS. Section III defines the Centralized TFP Learning problem and Federated TFP Learning problem and proposes a security parameter aggregation mechanism. Section IV presents the FedGRU framework. Section V discusses the experimental results. Concluding remarks are described in Section VI.

II. Related Work

A. Traffic Flow Prediction

Traffic flow prediction (TFP) has always been a hot issue in ITS, which can improve the efficiency of real-time traffic control and urban planning. Although researchers have proposed many new models and methods, they can generally be divided into two categories: parametric and non-parametric models.

1) Parametric models: Parametric models predict future data by capturing existing data feature within its parameters. M. S. Ahmed *et al.* in [8] proposed the Autoregressive Integrated Moving Average (ARIMA) model in the 1970s to predict short-term freeway traffic. Since then, many researchers have proposed variants of ARIMA such as Kohonen-ARIMA (KARIMA), subset ARIMA, seasonal ARIMA, etc. These models further improve the accuracy of TP by focusing on the statistical correlation of the data.

2) Non-parametric models: With the improvement of data storage and computing, non-parametric models have achieved great success in TFP [9]. Davis and Nihan *et al.* in [10] proposed k-NN model for short-term traffic flow prediction. Lv *et al.* in [1] first applied the stacked autoencoder (SAE) model for TFP. Furthermore, SAE adopts a hierarchical greedy network structure to learn non-linear features and has better performance than support vector machines (SVM) [11] and feed-forward neural network (FFNN) [12]. Considering the temporal correlation of the data, Ma *et al.* in [13] and Tian *et al.* in [14] applied Long Short-Term Memory (LSTM) to achieve accurate and timely TFP. Fu *et al.* in [15] first proposed GRU neural network methods for TFP. In recent years, due to the success of convolutional networks and graph networks, Yu *et al.* in [16], [17] proposed graph convolutional generative autoencoder to address the real-time traffic speed estimation problem.

B. Privacy Research in Intelligent Transportation Systems

In ITS, many models and methods rely on training data from users or organizations. However, with the increasing privacy awareness of users and organizations, direct exchanges of data between users or organizations are advocated against the law. Brian *et al.* in [18] designed a data sharing algorithm based on information-theoretic k-anonymity principle. However, this algorithm may leak privacy during data sharing operations. Furthermore, the EU has promulgated GDPR, which means that as long as the organization has the possibility of revealing privacy in the data sharing process, such data transactions violate the law.

Although researchers have proposed some privacy-preserving methods to predict traffic flow in the literature, they still cannot meet the requirements of GDPR. In this paper, we explore a powerful privacy-preserving method with GRU for traffic flow prediction.

III. Problem Definition

We use the term "organization" throughout the paper to describe entities in TFP, such as urban agencies, private companies, and detector stations. We use the term "client" to describe computing nodes that correspond to one or multiple sensors in FL and use the term "device" to describe the sensor in the organizations. Let $\mathcal{C} = \{C_1, C_2, \cdots, C_n\}$ and $\mathcal{O} = \{O_1, O_2, \cdots, O_m\}$ denote the client set and organization set in ITS, respectively. Each client has q organizations. Each organization has k_i devices and their respective database D_i. We aim to predict the number of vehicles with historical traffic flow information from different organizations without sharing raw data and privacy leakage. We design a secure parameter aggregation mechanism as follows:

Secure Parameter Aggregation Mechanism: *Detector station O_i has N devices, and the traffic flow data collected by the N devices constitute a database D_i. The deep learning model constructed in O_i calculates updated model parameters p_i using the local training data from D_i. When all detector stations finish the same operation, they upload their respective p_i to the cloud and aggregate a new global model.*

According to Secure Parameters Aggregation, no traffic flow data is exchanged among different detector stations. The cloud aggregates the gradients uploaded by organizations to obtain a new global model without exchanging data.

In this paper, t and v_t represent the t-th timestamp in the time-series and traffic flow at the t-th timestamp, respectively. Let $f(\cdot)$ be the traffic flow prediction function, the centralized and federated TFP learning problems are defined as follows:

Centralized TFP Learning: *Given organizations $\mathcal{O}$, each organization's devices k_i, and an aggregated database $D = D_1 \cup D_2 \cup D_3 \cup \cdots \cup D_N$, the centralized TFP problem is to calculate $v_{t+s} = f(t+s, D)$, where s is the prediction window after t.*

Federated TFP Learning: *Given organizations $\mathcal{O}$ and each organization's devices k_i, and their respective database D_i, the federated TFP problem is to calculate $v_{t+s} = f_i(t+s, D_i)$ where $f_i(\cdot,\cdot)$ is the local version of $f(\cdot,\cdot)$ and s is the prediction window after t. Subsequently, the produced results are aggregated by a secure parameter aggregation mechanism.*

IV. METHODOLOGY

A. Federated Learning and Gated Recurrent Unit

Federated Learning (FL) [6] is a distributed machine learning (ML) paradigm that has been designed to train ML models without compromising privacy. With this scheme, different organizations can contribute to the overall model training while keeping the training data locally.

Particularly, FL problem involves learning a *single* and *globally* predicted model from the database separately stored in dozens of or even hundreds of organizations. We assume that each device k stores its local dataset $\mathcal{D}_k$ of size D_k. So we can define the local training dataset size $D = \sum_{k=1}^{K} D_k$. In a typical deep learning setting, given a set of input-output pairs $\{x_i, y_i\}_{i=1}^{|D_k|}$, where the input sample vector with d features is $x_i \in \mathbb{R}^d$, and the labeled output value for the input sample x_i is $y_i \in \mathbb{R}$. If we input the training sample vector x_i (e.g., the traffic flow data), we need to find the model parameter vector $\omega \in \mathbb{R}^d$ that characterrizes the output y_i (e.g., the value output of the traffic flow data) with loss function $f_i(\omega)$ (e.g., $f_i(\omega) = \frac{1}{2}(x_i^T\omega - y_i)$). Our goal is to learn this model under the constraints of local data storage and processing by devices in the organization with a secure parameter aggregation mechanism. For local data of client c, we aim to minimize the objective function as follows:

$$J_c(\omega) = \frac{1}{|D_k|}\sum_{i=1}^{|D_k|} f_i(\omega) + \lambda h(\omega), \tag{1}$$

where the local model parameter $\omega \in \mathbb{R}^d$, $\forall\lambda \in [0, 1]$, and $h(\cdot)$ is a regualarizer function. This characterizes the local model in the FL setting.

At the cloud, the global predicted model problem can be represented as follows:

$$\arg\min_{\omega\in\mathbb{R}^d} J(\omega)\ ,\ J(\omega) \equiv \frac{\sum_{k=1}^{|D_k|} D_k J_c(\omega)}{D}, \tag{2}$$

we recast the global predicted model problem in (2) as follows:

$$\arg\min_{\omega\in\mathbb{R}^d} J(\omega) := \frac{\sum_{k=1}^{|D_k|} f_i(\omega) + \lambda h(\omega)}{D}. \tag{3}$$

For TFP problem, we regard GRU neural network model as the local model in Equation (1). Cho *et al.* in [19] proposed the GRU neural network in 2014, which is a variant of RNN that handles time-series data. GRU is different from RNN is that it adds a "Processor" to the algorithm to judge whether the information is useful or not. The structure of the processor is called "Cell." A typical structure of GRU cell uses two data "gates" to control the data from processor: reset gate r and update gate z.

Let $X = \{x_1, x_2, \cdots, x_n\}$, $Y = \{y_1, y_2, \cdots, y_n\}$, and $H = \{h_1, h_2, \cdots, h_n\}$ be the input time series, output time series and the hidden state of the cells, respectively. At time step t, the value of update gate z_t is expressed as:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}), \tag{4}$$

where x_t is the input vector of the t-th time step, $W^{(z)}$ is the weight matrix, and h_{t-1} holds the cell state of the previous time step $t-1$. The update gate aggregates $W^{(z)}x_t$ and $U^{(z)}h_{t-1}$, then maps the results in $(0, 1)$ through a Sigmoid activation function. The reset gate r_t is computed similarly to the update gate:

$$r_t = \sigma(W^{(z)}x_t + U^{(r)}h_{t-1}). \tag{5}$$

The candidate activation $h_t{}'$ is denoted as:

$$h_t{}' = \tanh(Wx_t + r_t \odot Uh_{t-1}), \tag{6}$$

where $r_t \odot Uh_{t-1}$ represents the Hadamard product of r_t and Uh_{t-1}.

The final memory of the current time step t is calculated as follows:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t{}'. \tag{7}$$

Traditional learning methods typically include three steps: data processing, data fusion, and modeling. Among them, data fusion is used for traditional learning model that directly shares data among all parties to obtain a global database for training. However, such a centralized learning approach faces the challenge of new data privacy laws and regulations as organizations may disclose privacy when sharing data. FL is introduced into this context to address the above challenges.

B. Privacy-preserving Traffic Flow Prediction Algorithm

We develop a FL framework FedGRU to fully handle the data privacy infringement issues in traffic flow prediction task. We first introduce a FedAVG algorithm as an implementation of the secure parameter aggregation mechanism to collect gradient information. Then, we illustrate the federated traffic flow prediction learning architecture. Finally, we demonstrate the details of the FedGRU algorithm.

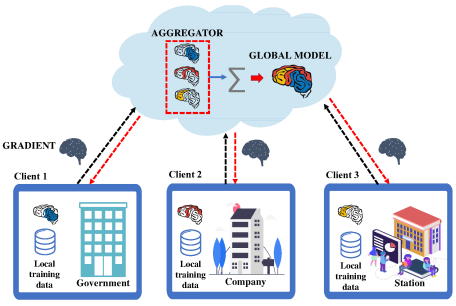


Fig. 1. Federated traffic flow prediction learning architecture.

1) FedAVG algorithm: A recognized problem in federated learning is the limited network bandwidth that bottlenecks cloud-aggregated local updates from the organizations. To reduce the communication overhead, each client uses its local data to perform gradient descent optimization on the current model. Then the central cloud performs a weighted average aggregation of the model updates uploaded by the clients. As shown in Algorithm 1, FedAVG consists of three steps:

(i) The cloud selects volunteers from organizations $\mathcal{O}$ to participate in this round of training and broadcasts global model ω^o to the selected organizations;

Algorithm 1: Federated Averaging (FedAVG) Algorithm.

Input: Organizations $\mathcal{O} = \{O_1, O_2, \cdots, O_N\}$. B is the local mini-batch size, E is the number of local epochs, α is the learning rate, $\nabla\mathcal{L}(\cdot;\cdot)$ is the gradient optimization function.

Output: Parameter ω.

Initialize ω^0 (Pre-trained by a public dataset);
foreach *round* $t = 1, 2, \cdots$ **do**
  $\{O_v\} \leftarrow$ select volunteer from organizations $\mathcal{O}$ participate in this round of training;
  Broadcast global model ω^o to organization in $\{O_v\}$;
  foreach *organization* $o \in \{O_v\}$ ***in parallel*** **do**
    Initialize $\omega_t^o = \omega^o$;
    ω_{t+1}^o ($\omega_{t+1}^o \leftarrow \mathrm{LocalUpdate}(o, \omega_t^o)$;
  $\omega_{t+1} \leftarrow \frac{1}{|\{O_v\}|} \sum_{o \in O_v} \omega_{t+1}^o$;
$\mathrm{LocalUpdate}(o, \omega_t^o)$: // Run on organization o ;
$\mathcal{B} \leftarrow$ (split $\mathcal{S}_o$ into batches of size B);
if *each local epoch* i *from 1 to* E **then**
  if *batch* $b \in \mathcal{B}$ **then**
    $\omega \leftarrow \omega - \alpha \cdot \nabla\mathcal{L}(\omega; b)$;
return ω to cloud

(ii) Each organization o trains data locally and updates ω_t^o for E epochs of SGD with mini-batch size B to obtain ω_{t+1}^o, i.e., $\omega_{t+1}^o \leftarrow \mathrm{LocalUpdate}(o, \omega_t^o)$;
(iii) The cloud aggregates each organization's ω_{t+1} through a secure parameter aggregation mechanism.

FedAVG algorithm is a critical mechanism in FedGRU to reduce the communication overhead in the process of transmitting parameters. This algorithm is an iterative process. For the i-th round of training, the models of the organizations participating in the training will be updated to the new global one.

2) Federated Learning-based Gated Recurrent Unit neural network algorithm: FedGRU aims to achieve accurate and timely TFP through combining FL and GRU without compromising privacy. The overview of FedGRU is shown in Fig. 1. It consists of four steps:

i) The cloud model is initialized through pre-training that utilizes domain-specific public datasets without privacy concerns;
ii) The cloud distributes the copy of the global model to all organizations, and each organization trains its copy on local data;
iii) Each organization uploads model updates to the cloud. The entire process does not share any private data, but instead sharing the encrypted parameters;
iv) The cloud aggregates the updated parameters uploaded by all organizations by the secure parameter aggregation mechanism to build a new global model, and then distributes the new global model to each organization.

Algorithm 2: Federated Learning-based Gated Recurrent Unit neural network (FedGRU) algorithm.

Input: $\{O_v\} \subseteq \mathcal{O}$, X, Y and H. The mini-batch size m, the number of iterations n and the learning rate α. The optimizer SGD.

Output: $J(\omega)$, ω and W_v^r, W_v^z, W_v^h.

According to X, Y, H and Equations (8)–(12), initialize the cloud model $J(\omega_0)$, ω_0, $W_v^{r_0}$, $W_v^{z_0}$, $W_v^{h_0}$, and H_v^0;
foreach *round* $i = 1, 2, 3, \cdots$ **do**
  $\{O_v\} \leftarrow$ select volunteer from organizations to participate in this round of training ;
  while g_ω *has not convergence* **do**
    foreach *organization* $o \in O_v$ ***in parallel*** **do**
      Conduct a mini-batch input time step $\{x_v{}^{(i)}\}_{i=1}^m$;
      Conduct a mini-batch true traffic flow $\{y_v{}^{(i)}\}_{i=1}^m$;
      Initalize $\omega_{t+1}^o = \omega_t^o$;
      $g_\omega \leftarrow \nabla_\omega \frac{1}{m} \sum_{i=1}^m \left(f_\omega(x_v^{(i)}) - y_v^{(i)}\right)^2$;
      $\omega_{t+1}^o \leftarrow \omega_t^o + \alpha \cdot SGD(\omega_t^o, g_\omega)$;
      Update the parameters $W_v^{r_0}$, $W_v^{z_0}$, $W_v^{h_0}$, and H_v^0;
      Update reset gate r and update gate z;
  Collect the all parameters from $\{O_v\}$ to update ω_{t+1}. (Referring to the Algorithm 1.);
return $J(\omega)$, ω and W_v^r, W_v^z, W_v^h

Given voluntary organization $\{O_v\} \subseteq \mathcal{O}$ and $o_v \in \{O_v\}$, referring to the GRU neural network in Section IV-A, we have:

$$z_v^t = \sigma(W^{(z_v)} + U^{(z_v)} h_v^{t-1}) \tag{8}$$

$$r_v^t = \sigma(W^{(r_v)} + U^{(r_v)} h_v^{t-1}) \tag{9}$$

$$h_v^{t\prime} = \tanh(W x_v^t + r_v^t \odot U h_v^{t-1}) \tag{10}$$

$$h_v^t = z_v^t \odot h_v^{t-1} + (1 - z_v^t) \odot h_v^{t'} \tag{11}$$

where $X = \{x_v^1, x_v^2, ..., x_v^n\}, Y = \{y_v^1, y_v^2, ..., y_v^n\}, H = \{h_v^1, h_v^2, ..., h_v^n\}$ denote o_v's input time series, o_v's output time series and the hidden state of the cells, respectively. According to Equation 3, the objective function of FedGRU is as follows:

$$\arg\min_{\omega} J(\omega) = \min \sum_{i=1}^{|D_v|} \sum_{t=1}^{T} \frac{1}{2} (y_d - y_v^t)^2 \tag{12}$$

The pseudocode of FedGRU framework is presented in Algorithm 2.

V. EXPERIMENTS

A. Dataset Pre-Processing and Evaluation Method

In this experiment, the proposed FedGRU is applied to the real-world data collected from the Caltrans Performance Measurement System (PeMS) [20] database for performance demonstration. The traffic flow data in PeMS database was

TABLE I

PERFORMANCE COMPARSION OF MAE, MSE, RMSE, AND MAPE FOR FEDGRU, LSTM, SAE, AND SVM

Metrics	MAE	MSE	RMSE	MAPE
FedGRU (default setting)	7.96	101.49	11.04	17.82%
GRU [15]	7.20	99.32	9.97	17.78%
SAE [1]	8.26	99.82	11.60	19.80%
LSTM [13]	8.28	107.16	11.45	20.32%
SVM [22]	8.68	115.52	13.24	22.73%

collected from over 39,000 individual detectors in real time. These sensors span the freeway system across all major metropolitan areas of the State of California [1]. In this paper, traffic flow data collected during the first three months of 2013 is used for experiments. We select the traffic flow data in the first two months as the training dataset and the third month as the testing dataset. Furthermore, since the traffic flow data is time-series data, we need to use them at the previous time interval, i.e., $x_{t-1}, x_{t-2}, \cdots, x_{t-r}$, to predict the traffic flow at time interval t, where r is the length of the history data window.

We adopt the the Mean Absolute Error (MAE), the Mean Square Error (MSE), the RMS Error (RMSE), and the Mean Absolute Percentage Error (MAPE) to show the prediction accuracy, i.e., prediction error. They are defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_p|, \tag{13}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_p)^2, \tag{14}$$

$$\text{RMSE} = [\frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_p|)^2]^{\frac{1}{2}}, \tag{15}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}|\frac{\hat{y}_p - y_i}{y_i}|. \tag{16}$$

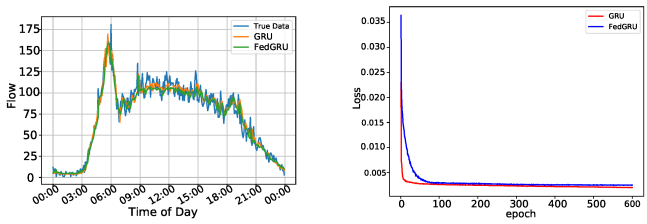
where y_i is the observed traffic flow, and y_i^p is the predicted traffic flow.

B. Experimental Setup

Without loss of generality, we assume that the detector stations are distributed and independent, and the data cannot be exchanged arbitrarily among them. In the secure parameter aggregation mechanism, PySyft [21] framework is adopted to encrypt the parameters[1].

For the cloud and each organization, we use mini-batch SGD for model optimization. PeMS dataset is split equally and assigned to 20 organizations. During the simulation, learning rate $\alpha = 0.001$, mini-batch size $m = 128$, and $|O_v| = 20$. Note that the client $C = 2$ of the FedGRU model is the default setting in FL [6]. All experiments are conducted using TensorFlow and PyTorch with Ubuntu 18.04.

[1]https://github.com/OpenMined/PySyft



(a) (b)

Fig. 2. (a) Traffic flow prediction of GRU model and FedGRU model. (b) Loss of GRU model and FedGRU model.

C. Experimental Results

1) Traffic Flow Prediction Accuracy: We compared the performance of the proposed FedGRU model with that of GRU, SAE, LSTM, and support vector machine (SVM) with an identical simulation configuration. Among these five competing methods, FedGRU is a federated machine learning model, and the rest are centralized ones. Among them, GRU is a widely-adopted baseline model that has better performance for traffic flow forecast tasks, as aforementioned in Section IV, and SVM is a popular machine learning model for general prediction applications [1]. In all investigations, we use the same PeMS dataset. The prediction results are given in Table I for 5-min ahead traffic flow prediction. From the simulation results, it can be observed that MAE of FedGRU is lower than those of SAEs, LSTM, and SVM but higher than that of GRU. Specifically, MAE of FedGRU is 9.04% lower than that of the worst case (i.e., SVM) in this experiment. This result is contributed by the fact that FedGRU inherits the advantages of GRU's outstanding performance in prediction tasks.

Fig. 2(a) shows a comparison between GRU and FedGRU for a 5-min traffic flow prediction task. We can find that the predict results of FedGRU model are very close to that of GRU. This is because the core technique of FedGRU to prediction is GRU structure, so the performance of FedGRU is comparable to GRU model. Furthermore, FedGRU can protect data privacy by keeping the training dataset locally. Fig. 2(b) illustrates the loss of GRU model and FedGRU model. From the results, the loss of FedGRU model is not significantly different from GRU model. This proves that FedGRU model has good convergence and stability. In a word, FedGRU can achieve accurate and timely traffic flow prediction without compromising privacy.

2) Performance Comparison of FedGRU Model Under Different Client Numbers: In Section V-C.1, the default client number is set $C = 2$. However, it is highly plausible that traffic data can be gathered by more than two entities, e.g., organizations and companies. In this experiment, we explore the impact of different client numbers (i.e., $C = 2, 4, 8, 10$) on the performance of FedGRU. The simulation results are presented in Fig. 3, where we observe that the number of clients has an adverse influence on the performance of FedGRU. The reason is that more clients introduce increasing communication overhead to the under-

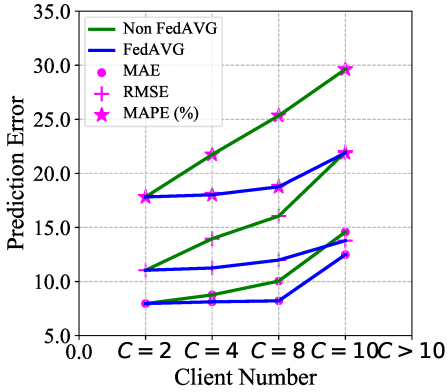


Fig. 3. The prediction error of FedGRU model with different client numbers.

lying communication infrastructure, which makes it more difficult for the cloud to simultaneously perform aggregation of gradient information. Furthermore, such overhead may cause communication failures in some clients, causing clients to fail to upload gradient information, thereby reducing the accuracy of the global model.

In this paper, we initially use FedAVG algorithm to alleviate the expensive communication overhead issue. FedAVG reduces communication overhead by i) computing the average gradient of a batch size samples on the client and ii) computing the average aggregation gradient from all clients. Fig. 3 shows that FedAVG performs well when the number of clients is less than 8, but when the number of clients exceeds 8, the performance of FedAVG starts to decline. The reason is that, when the number of clients exceeds a certain threshold (e.g., $C = 8$), the probability of client failure will increase, which causes FedAVG to calculate wrong gradient information. Nevertheless, FedAVG is significant for reducing communication overhead because the number of entities involved in predicting traffic flow tasks in real life is usually small.

VI. Conclusion

In this paper, we propose a FedGRU algorithm for traffic flow prediction with federated learning for privacy preservation. FedGRU does not directly access distributed organizational data but instead employs a secure parameter aggregation mechanism to train a global model in a distributed manner. It aggregates the gradient information uploaded by all locally trained models in the cloud to construct the global one for traffic flow forecasts. We evaluate the performance of FedGRU on a PeMS dataset and compared it with GRU, LSTM, SAE, and SVM, which all potentially compromise user privacy during the forecast. The results show that the proposed method performs comparably to the competing methods with minuscule accuracy degradation with privacy well-preserved. In the future, we plan to apply Graph Convolutional Network to the federated learning framework to predict traffic flow.

References

[1] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[2] N. Zhang, F.-Y. Wang, F. Zhu, D. Zhao, and S. Tang, "Dynacas: Computational experiments and decision support for ITS," *IEEE Intelligent Systems*, vol. 23, no. 6, pp. 19–23, 2008.

[3] C. Zhang, J. J. Q. Yu, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166 246–166 256, 2019.

[4] S. Madan and P. Goswami, "A novel technique for privacy preservation using k-anonymization and nature inspired optimization algorithms," *Available at SSRN 3357276*, 2019.

[5] J. L. Ny, A. Touati, and G. J. Pappas, "Real-time privacy-preserving model-based estimation of traffic flows," in *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, April 2014, pp. 92–102.

[6] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[7] X. Yuan, X. Wang, C. Wang, J. Weng, and K. Ren, "Enabling secure and fast indexing for privacy-assured healthcare monitoring via compressive sensing," *IEEE Transactions on Multimedia (TMM)*, vol. 18, no. 10, pp. 1–13, 2016.

[8] M. S. Ahmed, "Analysis of freeway traffic time series data and their application to incident detection," *Equine Veterinary Education*, vol. 6, no. 1, pp. 32–35, 1979.

[9] J. J. Q. Yu, A. Y. S. Lam, D. J. Hill, Y. Hou, and V. O. K. Li, "Delay aware power system synchrophasor recovery and prediction framework," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3732–3742, July 2019.

[10] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *Journal of Transportation Engineering*, vol. 117, no. 2, pp. 178–188, 1991.

[11] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[12] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, no. 1, pp. 43–62, 1997.

[13] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[14] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*. IEEE, 2015, pp. 153–158.

[15] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nov 2016, pp. 324–328.

[16] J. J. Q. Yu, W. Yu, and J. Gu, "Online vehicle routing with neural combinatorial optimization and deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3806–3817, Oct 2019.

[17] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3940–3951, Oct 2019.

[18] B. Y. He and J. Y. Chow, "Optimal privacy control for transport network data sharing," *Transportation Research Part C: Emerging Technologies*, 2019.

[19] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[20] C. Chao, *Freeway performance measurement system (pems)*, 2003.

[21] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, and J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *arXiv preprint arXiv:1811.04017*, 2018.

[22] M. A. Mohandes, T. O. Halawani, S. Rehman, and A. A. Hussain, "Support vector machines for wind speed prediction," *Renewable Energy*, vol. 29, no. 6, pp. 939–947, 2004.