IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-0926-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798018

FedTSE: Low-Cost Federated Learning for Privacy-Preserved Traffic State Estimation in IoV

Xiaoming Yuan[1], Jiahui Chen[1], Ning Zhang[2], Chunsheng Zhu[3], Qiang Ye[4], Xuemin (Sherman) Shen[5]
[1]Qinhuangdao Branch Campus, Northeastern University, Qinhuangdao 066004, China.
[2]Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada.
[3]SUSTech Institute of Future Networks, Southern University of Science and Technology, Shenzhen, 518055, China.
[4]Department of Computer Science, Memorial University of Newfoundland, St. John's, NL, A1B 3X5, Canada.
[5]Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, N2L 3G1, Canada.

***Abstract*—Traffic state estimation (TSE) is an important aspect of the Internet of Vehicles (IoV) for road path planning and better driving experience. In IoV, with the support of edge intelligence, real-time traffic data can be processed by edge computing (EC) servers for informed decision-making. However, collecting trajectory information from vehicles in a centralized manner may increase transmission delay and cause driver privacy leakage problems. In this paper, we firstly propose a federated learning (FL) framework for TSE, named FedTSE, with privacy preservation by jointly considering TSE accuracy, model computation, and transmission cost. Then, a TSE model is designed based on the long short-term memory (LSTM) as the local training model for joint prediction of vehicular speed and traffic flow. Considering the resource limitation of computation/communication, we further propose a deep reinforcement learning (DRL)-based algorithm for model parameter uploading/downloading decisions to improve the estimation accuracy of local models and balance the tradeoff between computation and communication cost. Simulation results show the proposed FedTSE achieves a lower cost and higher prediction training accuracy in TSE.**

***Index Terms*—Traffic state estimation, long short-term memory, edge computing, federated learning, internet of vehicles.**

I. Introduction

Traffic congestion is an inevitable problem in the process of urbanization. Excessive traffic congestion leads to an increase in automobile exhaust emissions and reduced traffic flowing efficiency [1]. Accurate traffic state estimation (TSE) is necessary to manage vehicular traffic and reduce congestion by monitoring some key vehicle indicators (e.g., space speed). The Internet of Vehicles (IoV) is foreseen to interconnect an increasing number of vehicles with more stringent delay requirements for data transmissions and processing. Connected autonomous vehicles are a typical example, where real-time TSE and decision-making are needed [2], [3]. The outcome of TSE can affect the driving experience and even drivers' safety of drivers in IoV.

TSE currently faces two major technical challenges. One is the privacy leakage problem in vehicular traffic data collection. Traffic data includes personal travel trajectory and driver performance data, related to travel preferences and driving habits. When transmitting those data to edge computing (EC) servers or cloud servers directly, data security issues are exposed. The other issue is the long service delay in TSE, which is composed of computation and communication delay [4]. Most existing works on TSE focus on improving the traffic prediction accuracy with centralized control [5], [6]. With the increase of network complexity and device numbers, enhancing the utilization of both communication and computing resources to reduce the cost with good service performance (e.g., delay) is crucial.

In order to preserve the privacy of data, the federated learning (FL) technology [7] emerges to allow clients to share the parameters of the machine learning models without transmitting the raw data. After introducing FL for distributed training, the frequent exchange of parameters can improve the model training accuracy but also consume substantial communication and computation resources [8]. Thus, the main cost in FL comes from the resource consumption in model computation and parameter exchange. Balancing the performance tradeoff between TSE accuracy and service performance is a challenging research issue.

In this paper, we propose an FL framework for TSE, called FedTSE in IoV with one EC and several road side units (RSUs). In FedTSE, we train a lightweight local model based on long short-term memory (LSTM) for TSE. All local TSE models are developed with the assistance of an EC server for parameter sharing under the proposed FedTSE framework. Moreover, we investigate how to optimize the parameter uploading/downloading decisions based on deep reinforcement learning (DRL) to reduce the FedTSE training cost under resource constraints. The main contributions of this paper are summarized as follows:

- We propose the FedTSE framework for traffic state estimation. In order to make the traffic estimation more efficient and accurate, we adopt LSTM as the local learning module for vehicular speed and traffic flow prediction by capturing temporal information. Moreover, we consider the influence of the actual training rounds of each local model while aggregating parameters.
- We formulate a computation and communication re-

This work was supported in part by the National Natural Science Foundation of China under Grant 61901099, 61972076 and Grant 61973069, in part by the Natural Science Foundation of Hebei Province under Grant F2020501037 and F2021501032, in part by the Fundamental Research Funds for the Central Universities under Grant N2123007 and in part by the Aeronautical Science Foundation of China under Grant 2020Z066050001.

source optimization problem for reducing transmission and model computation cost. An asynchronous DRL algorithm, asynchronous advantage actor-critic (A3C), is employed to solve the problem with a better convergence performance and of adapting to dynamic network environment intelligently.
- We conduct simulations to evaluate the performance of FedTSE in different communication modes. Results show that our FedTSE achieves lower cost and higher accuracy, compared with other baseline methods.

The remainder of this paper is organized as follows: In Section II, we review the related works on FL and TSE. We propose the FedTSE framework in Section III. In Section IV, we formulate an optimization problem for FedTSE. Section V presents the DRL algorithm to solve the problem. The simulations of FedTSE are given in Section VI, followed by the conclusions in Section VII.

II. Related Work

In this section, we review some recent works related to TSE and FL. Firstly, we introduce some popular TSE methods. Then, we focus on the works of FL.

A. Traffic State Estimation

TSE is mostly based on historical information such as traffic flow, speed and other road condition index [9], [10]. Currently, most of the works only consider the estimation of flow or speed. For speed estimation in traffic, Cui *et al*. [5] proposed a Traffic Graph Convolutional LSTM model for traffic speed prediction, considering spatiotemporal features of speed under the traffic graph convolutional networks. Similarly, Cui *et al*. [6] proposed a traffic state prediction algorithm based on LSTM and Bidirectional LSTM (BiLSTM) with missing raw data. Li *et al*. [11] proposed a spatial-Temporal fusion deep learning approach for traffic flow prediction. However, all of the above works lack the consideration of security in data sharing under centralized processing.

B. Parameter Sharing in Federated Learning

There are fewer research works on parameter uploading and downloading strategies. Parameter aggregation is usually based on FedAVG [12], [13], whose principle is using the weighted average method. In [14], the authors proposed a temporally weighted aggregation federated learning approach. Yuan *et al*. [15] integrated federated learning with a broad learning system in IoV in order to improve the efficiency and accuracy in data sharing and select clients according to user spatial similarity. Wang *et al*. [16] proposed a novel FL framework, which can achieve more efficient FL in multiple edge nodes without sharing their raw private data. In FL, the joining or exiting of nodes can also affect the performance of federated learning. In [17], the authors proposed a secure asynchronous FL algorithm for data sharing in IoV. This algorithm can remove some nodes in FL if they make the performance of FL worse. However, the parameter aggregation strategy is relatively fixed in these references, and the influence of dynamic changes of the environment on parameter aggregation is seldom considered.

III. System Model

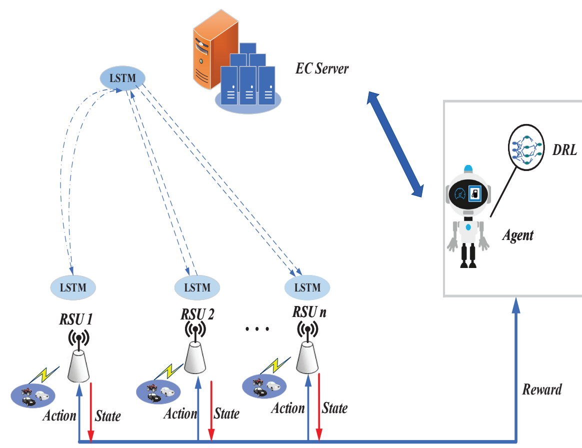


Fig. 1. The system model of FedTSE in IoV.

In this section, we present the process of FL-driven TSE. We first introduce how FL applied for TSE, and then introduce how the communication process during FL in edge computing driven IoV.

A. Federated Learning

Figure 1 shows the system model for TSE. In this scenario, we assume traffic trajectory data is collected by different RSUs [18]. The collected data may belong to different service providers, so we adopt FL to solve the data islands. Thus, the set of the model is defined as $\Omega = [\Omega_1, \Omega_2, ..., \Omega_N]$, which means there has N RSUs. The data sets of all RSUs are defined as $\mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N]$. In each $\mathcal{D}_i$, it includes two parts: input X_i, and label Y_i. In FL, the global model is set in the EC server which has powerful computation resources. In FL, the key is how to aggregate parameters effectively. Generally, the weighted average method is used to aggregate parameters. In view of the asynchronous nature of FL, the training epochs e_i^{actual} of each RSU is not consistent, so we need to assign the weight of the aggregation parameters to capture the influence of the rounds on the aggregation effect:

$$\mathcal{W}(e) = \sum_{i=1}^{N} \frac{size(\mathcal{D}_i)}{size(\mathcal{D})} \mathcal{W}_i(e) * Exp(-(e - e_i^{actual})), \quad (1)$$

where $\mathcal{W}(e)$ is the set of the parameters of global model at eth episode, and $\mathcal{W}_i(e)$ are the parameters of local model in ith RSU. $size(\mathcal{D}_i)$ means the size of the local raw data i, and $size(\mathcal{D})$ denotes the size of all raw data. Exp means natural constant and $Exp(-(e - e_i^{actual})) = \frac{1}{Exp((e-e_i^{actual}))}$. If $e - e_i^{actual} = 0$, the value of $\frac{1}{Exp((e-e_i^{actual}))}$ is one, which means that RSU i has the highest degree of participation in FL, so the temporal weight factor assigned is the largest. When the RSU is hardly involved in FL, $e - e_i^{actual} \to +\infty$. Then, we can get

$$\lim_{(e-e_i^{actual}) \to +\infty} \frac{1}{Exp((e - e_i^{actual}))} = 0$$

, which means the value of temporal weight factor is the lowest. The parameters are updated by:

$$\mathcal{W}_i(e+1) = \mathcal{W}(e) - \eta \frac{\partial \mathcal{L}(\mathcal{D}_i; \mathcal{W}(e))}{\partial \mathcal{W}(e)}, \tag{2}$$

where (2) means the gradient-descent approach is applied for training the model at each epoch and η is the learning rate.

B. Communications for Parameter Uploading/Downloading

In the FL environment, RSUs need to communicate with the EC server when uploading or downloading parameters. This process will consume communication resources. At the same time, data processing and model training also require computing resources on RSUs. Hence, it incurs some latency in transmission and execution in one FL epoch.

At each episode e, RSUs have the possibility to upload and download parameters. Thus, the uplink transmission rate between RSU i and the EC server is:

$$\mathcal{R}_i^{up}(e) = B_i^{up}(e)\log_2(1 + \frac{P_n^{tr}(e) \cdot g_n(e)}{N_0}), \tag{3}$$

where $B_i^{up}(e)$ is the uplink bandwidth. The channel gain $g_n(e) = d_{i,ec}^{\lambda}(e)$, where $\lambda = -4$ and $d_{i,ec}$ is the distance from RSU i to the EC server at time slot e. The transmission power is given as $P_n^{tr}(e)$. The transmission latency is given:

$$T_i^{up}(e) = \frac{size(\mathcal{W}_i(e))}{R_i^{up}(e)}. \tag{4}$$

Similarly, we can derive the formula of the downlink transmission rate between RSU i and the EC server is:

$$\mathcal{R}_i^{down}(e) = B_i^{down}(e)\log_2(1 + \frac{P_n^{tr}(e) \cdot g_n(e)}{N_0}), \tag{5}$$

where $B_i^{down}(e)$ is the downlink bandwidth. The transmission latency is given:

$$T_i^{down}(e) = \frac{size(\mathcal{W}(e))}{R_i^{down}(e)}. \tag{6}$$

IV. Problem Formulation for FedTSE

In this section, we adopt LSTM for local TSE. Furthermore, we construct the optimization problem of jointly considering computation and communication resources.

A. LSTM-based Model for Local Training

We use LSTM to capture the time correlation of vehicular speed and traffic flow. The $LSTM$ layer captures temporal features of vehicular speed and traffic flow as follows:

$$[\hat{Y}^f, \hat{Y}^r] = LSTM([X^f, X^r]), \tag{7}$$

where $[X^f, X^r]$ means that we merge X^f and X^r as the $LSTM$ input. $\hat{Y}^f$ and $\hat{Y}^r$ represent the predicted traffic flow and vehicular speed, respectively.

B. Cost Consumption in FedTSE

In FL, communication resources are consumed from bandwidth, power, and other resources during transmission. Computing resources are consumed mainly from local model training. Hence, the total delay of parameters transmission is given as:

$$T(e) = \sum_{i=1}^{N} \left\{ \alpha_i(e) T_i^{up}(e) + \beta_i(e) T_i^{down}(e) \right\}, \tag{8}$$

where $\alpha_i(e)$, $\beta_i(e)$ are binary decision variables, which indicate whether or not the local RSU chooses to parameter upload/download.

In the life-cycle of FL-driven TSE, the total latency is given as follows:

$$\mathcal{T} = \frac{1}{E} \sum_{e=1}^{E} \sum_{i=1}^{N} \left\{ \alpha_i(e) T_i^{up}(e) + \beta_i(e) T_i^{down}(e) \right\}. \tag{9}$$

The computation delay caused by model training mainly comes from the update of parameters. We assume f_i is the CPU frequency of local RSU and $\theta(e)$ is the calculation density [19]. Therefore, the computation delay of dealing the $\mathcal{W}_i$ is given as:

$$T_i^{exe}(e) = \frac{size(\mathcal{W}_i(e))\theta(e)}{f_i(e)}, \tag{10}$$

where $T_i^{exe}(e)$ means the delay of process in training the parameters of local model i.

According to the above analysis, the cost of TSE is mainly the calculation cost of model training and the communication cost of parameter upload/download. In order to maintain the accuracy of the model in TSE, the calculation cost is controlled by controlling the training rounds of the model.

In TSE, we use Mean Square Error (MSE) to evaluate accuracy, where $MSE_i(e) = \frac{1}{size(\mathcal{B})} \sum_{j=1}^{size(\mathcal{B})} (y_j - \Omega_i(x_j; \mathcal{W}_i))^2$ and y_j is the true value from the min-batch during training. Problems $\mathbf{P1}$: $\frac{1}{E} \sum_{e=1}^{E} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{size(\mathcal{B})} \sum_{j=1}^{size(\mathcal{B})} (y_j - \Omega_i(x_j; \mathcal{W}_i))^2 + e_i^{actual} T_i^{exe}(e)$ and $\mathbf{P2} : \mathcal{T}$ are sub-problems of TSE, and we want TSE to keep the model accurate and reduce the system cost, so we need to optimize computation cost $\mathbf{P1}$ and communication cost $\mathbf{P2}$ together. Hence, our multi-objective optimization function is given as:

$$\mathbf{P3} : \min a_1 \left\{ \frac{1}{E} \sum_{e=1}^{E} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{size(\mathcal{B})} \sum_{j=1}^{size(\mathcal{B})} (y_j - \Omega_i(x_j; \mathcal{W}_i))^2 + e_i^{actual} T_i^{exe}(e) \right\} + a_2 \mathcal{T} \tag{11}$$

$$s.t. \quad C1: \ e, e_i^{actual} \leq E \tag{11a}$$

$$C2: \ \alpha_i(e), \beta_i(e) \in \{0, 1\} \tag{11b}$$

$$C3: \ P_n^{tr}(e) \leq P_{max}(e), \tag{11c}$$

$$C4: \ B_i^{down}(e), B_i^{up}(e) \leq B_{max}(e), \tag{11d}$$

$$C5: \ i \in N, \tag{11e}$$

$$C6: \ f_i(e) \leq f_{max}(e), \theta(e) \leq \theta_{max}(e) \tag{11f}$$

$$C7: \epsilon \leq \varepsilon. \tag{11g}$$

In problem $\mathbf{P3}$, the cost in IoV includes training cost and transmission cost, thus we assign different price weights to them and set a constraint: $a_1 + a_2 = 1$, where $j \in \mathcal{B}$, and $\mathcal{B}$ is the training batch randomly from $\mathcal{D}_i$. $MSE_i(e)$ indicates that the value of TSE is more accurate, if $MSE_i(e)$ is lower. Constraints $C1$ enforces the maximum training epoch E. Constraint $C2$ enforces $\alpha_i(e)$ and $\beta_i(e)$ in set of $\{0, 1\}$ by linking equation (9), where our goal is to select decisions to reduce the cost in FedTSE. Constraint $C3$ ensures the transmission power $P_n^{tr}(e)$ between RSU and the EC server are not greater than the maximum power $P_{max}(e)$. Constraint $C4$ makes sure the channel bandwidth $B_i^{down}(e)$ and $B_i^{up}(e)$ between RSU and the EC server are not greater than the maximum bandwidth $B_{max}(e)$. Constraint $C6$ corresponds to the maximum cpu frequency $f_{max}(e)$ and maximum calculation density $\theta_{max}(e)$. Constraints $C7$ ensures the MSE must be lower than ε.

V. Proposed Solution

In this section, we will introduce how to optimize the problem $\mathbf{P3}$ to reduce the total cost in FedTSE based on deep reinforcement learning (DRL) algorithm.

The problem $\mathbf{P3}$ is a non-convex optimized problem, and the resources of computation and communication also change at each episode in FedTSE. Due to the dynamics of the environment, parameter upload and download decisions should be adaptive. To this end, DRL can be employed to solve this problem. In FedTSE, because distributed asynchronously learning models in RSUs, we need to asynchronously deploy DRL-based algorithm for learning the status. According to this idea, we adopt the asynchronous advantage actor-critic (A3C)-based [20] algorithm for solving problem $\mathbf{P3}$. A3C is an asynchronous DRL algorithm, which uses the method of multi-threading, and carries on the interactive learning with the environment in several threads at the same time. Each thread collects the learning results and saves them in a common place, namely experience pool. Based on the A3C algorithm, we define the state and action of the environment as follows:

$$\begin{aligned} &\mathcal{S} = (Pos, P_{max}, B_{max}, f_{max}, \theta_{max}), \\ &\mathcal{A} = (\alpha, \beta). \end{aligned} \tag{12}$$

Pos means the location of RSUs. P_{max} and B_{max} correspond to the resources of computation, while f_{max} and θ_{max} are for the communication resources. Hence, we need to decide the execution of actions $\mathcal{A}(e)$ according to the current state $\mathcal{S}(e)$.

We need to maximize the average reward $\mathcal{R}$ according to the problem $\mathbf{P3}$ as follows:

$$\begin{aligned} \mathcal{R} = -\Bigg\{ a_1 \Bigg\{ \frac{1}{E}\sum_{e=1}^{E}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{size(\mathcal{B})}\sum_{j=1}^{size(\mathcal{B})}(y_j - \Omega_i(x_j;\mathcal{W}_j))^2 + \\ e_i^{actual}T_i^{exe}(e)\Bigg\} + a_2\mathcal{T}\Bigg\}. \end{aligned} \tag{13}$$

The details of A3C-based solution for solving problem $\mathbf{P3}$ are in Algorithm I. First, lines 1-2 represent the global model at the EC server and local models in RSUs. Line 3 initializes the experience pool Rep of the A3C-based algorithm. Line 4 defines the life cycle of FedTSE, which is also the training episodes of the A3C-based algorithm. Line 5 gets the state $\mathcal{S}(e)$ of the current e. Line 6 indicates the A3C-based algorithm to learn the model state and communication state of IoV. Line 7 chooses the optimal action for each local RSU. Line 8 indicates that each local model updates the model parameters according to the assigned optimal decision. Line 9 shows that the EC server can aggregate parameters from local RSUs. The last line means that the environment gets the reward $\mathcal{R}(e)$ of the state $\mathcal{S}(e)$ and moves to the next state $\mathcal{S}^{'}(e)$ according to the state $\mathcal{S}(e)$ and action $\mathcal{A}(e)$. Also, we put $\left\{\mathcal{S}(e), \mathcal{A}(e), \mathcal{R}(e), \mathcal{S}^{'}(e)\right\}$ into the experience pool Rep, and then trains A3C-based algorithm in line 10.

Algorithm 1: The A3C-based algorithm for solving problem $\mathbf{P3}$.

1 Initialize parameters $[\mathcal{W}_1, ..., \mathcal{W}_N]$ of all local models $[\Omega_1, \Omega_2, ..., \Omega_N]$;
2 Initialize parameters $\mathcal{W}$ of global model in EC server;
3 Initialize replay memory Rep;
4 **for** *episode* $e = 1, 2, ...$ **do**
5   Get the state $\mathcal{S}(e)$;
6   Call **A3C-based Algorithm** under multi-threading;
7   Get the $\alpha_i(e)$, $\beta_i(e)$ according to the output of **A3C-based Algorithm** for each local RSU;
8   Train each local model, **LSTM** with uploading/downloading decision of parameters;
9   Aggregate parameters from each local model;
10   Get the $\mathcal{R}(e)$ and $\mathcal{S}^{'}(e)$ according to $\mathcal{S}(e)$ and $\mathcal{A}(e)$; Put $\left\{\mathcal{S}(e), \mathcal{A}(e), \mathcal{R}(e), \mathcal{S}^{'}(e)\right\}$ into the Rep and feed it into the **A3C-based Algorithm**;
11 **end**

VI. Simulation Results

In this section, we adopt the England Freeway Dataset for freeway TSE. Traffic states in this data set include traffic flow and vehicular speed data from January 1, 2014 to December 31, 2014, and the time interval is 15 minutes. We use the data in January and apply the weighted average method to fill missing values. In order to make the model converge faster during the training process and to ensure the original distribution of the data set, Z-score is used to standardize the data. Firstly, we introduce the parameter settings in simulations. Then, we list three communication modes of TSE. After that, we evaluate the performance of the A3C algorithm for optimizing the problem $\mathbf{P3}$ in FedTSE.

A. Parameter Settings

TABLE I
Parameter list.

Parameter	Description	Ref. Value
$\theta_{max}(e)$	Calculation density	$800 \sim 1000$ cycle/bit
$f_{max}(e)$	The maximum frequency	$0.6 * 10^9 \sim 10^9$ Hz
N_0	Noise power	10^{-13} W
$P_{max}(e)$	Maximum transmit power	$0.6 * 10^3 \sim 10^3$ W
$B_{max}(e)$	Maximum transmit bandwidth	$0.12 * 10^6 \sim 0.18 * 10^6$

We take 80% of the data as training data and use one day's data as testing data. In the comparative analysis, we set the number of RSUs to 2, 4, 8, and 16, respectively. In our local training LSTM model, we set the hidden neurons of LSTM as 100. We use Pytorch in Intel (R) Core (TM) i5-10400F CPU @ 2.9 GHz, 16GB memory for simulations. Finally, the summary of the parameter list is shown in Table I.

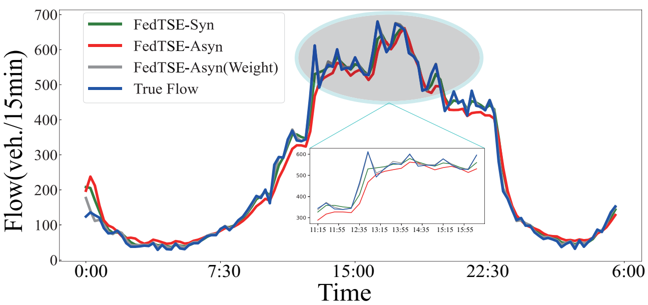


Fig. 2. Traffic flow estimation effect of different communication modes in FedTSE.

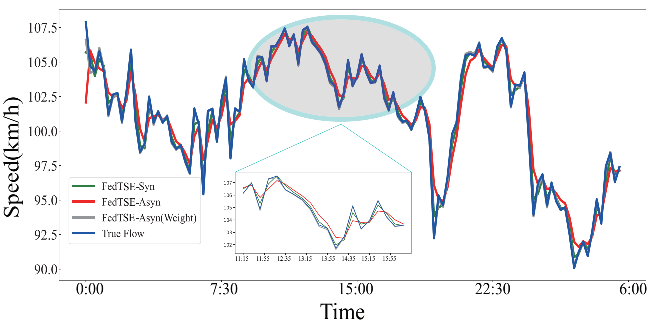


Fig. 3. Vehicular speed estimation effect of different communication modes in FedTSE.

B. Performance Comparison

Firstly, we compare the prediction performance of different communication methods for TSE in Figures 2-3. In FedTSE, we use three communication modes and set 8 RSUs for TSE as follows:

- **FedTSE-Syn**. It means that we use a synchronous communication mode for TSE. In this mode, the number of training epochs of each RSU is identical.
- **FedTSE-Asyn**. It means that we use an asynchronous communication mode for TSE. In this mode, number of training epochs of each RSU is different. Some RSUs may have fewer training epochs and others may have more training epochs.
- **FedTSE-Asyn(Weight)**. It means that we use an asynchronous communication mode for TSE and we also consider the influence of training epochs on parameters. The temporal weight factor assigned to RSUs with less FL rounds will be small.

From Figure 2, we can see the peak values of traffic flow are relatively large from 15:00 to 22:30. In **FedTSE-Syn**, the performance of traffic flow estimation is better than **FedTSE-Asyn**. The main reason is that there is no temporal lag in parameter aggregation of the global model from local models because of the same training epochs. But in reality, communication is usually asynchronous. In FedTSE, the influence of training epochs on parameter aggregation is considered, and the temporal weight factor is added. The results show that **FedTSE-Asyn(Weight)** can promote the prediction performance of LSTM. The performance of speed estimation under different communication modes is shown in Figure 3. Because of the rapid instantaneous change of vehicular speed, the time periodicity is not strong. But LSTM can effectively mine the temporal correlation of speed. Under three different communication modes, the velocity values estimated by our FedTSE basically reflect the change of the true speed. After introducing the temporal weight factor, the estimation effect of **FedTSE-Asyn(Weight)** is improved compared with **FedTSE-Asyn** and **FedTSE-Syn**, because it can reduce the occurrence of unbalanced FL.

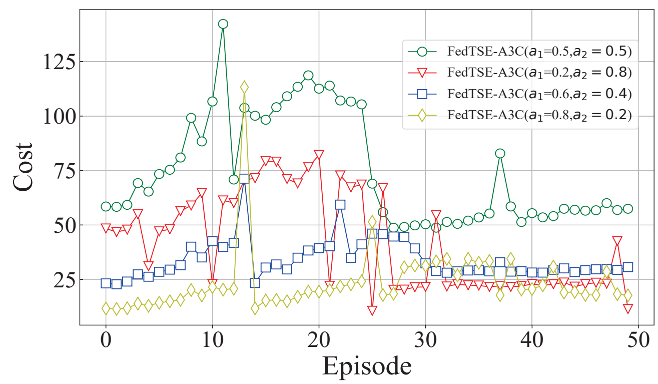
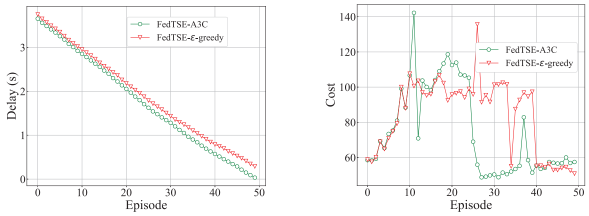


Fig. 4. The cost of different for combination of a_1 and a_2 in FedTSE-A3C.

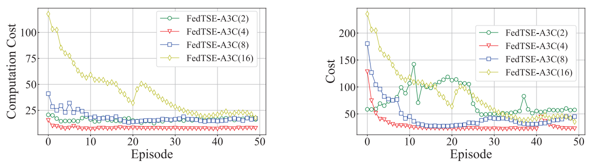
We also discuss the real-time total cost caused by A3C algorithm for solving resources problem **P3** in Figures 4 - 6. Figure 4 shows total cost in FedTSE during online training A3C-based algorithm under the different combinations of a_1 and a_2. Although the values of a_1 and a_2 are different, the value of cost converges with the increase of *Episode*. Under the restriction condition $a_1 = 0.5, a_2 = 0.5$, the FedTSE cost is worse than that in other conditions. This is because the optimization efforts of subproblem **P1** are the same as the subproblem **P2**. In fact, the computing and communication cost can not be the best at the same time. In FedTSE, most of the cost is from model training. Thus, we improve the optimization of computing resources under the restriction condition $a_1 = 0.2, a_2 = 0.8$. Under this constraint, the cost has a more significant convergence effect with the increase of *Episode* in FedTSE.

Finally, we explore the cost and latency of FedTSE over IoV. From Figure 5, we compare A3C-based algorithm with the ϵ-greedy algorithm. In ϵ-greedy, we choose the strategy with ϵ probability every time, and this method will gradually select actions that makes the agent get the maximum reward with a high probability. However, this algorithm lacks the



(a) (b)

Fig. 5. (a) The comparisons of the different solutions for delay in FedTSE. (b) The comparisons of the different solutions for cost in FedTSE.



(a) (b)

Fig. 6. (a) The computation cost of the different number of RSUs in FedTSE-A3C. (b) The cost of the different number of RSUs in FedTSE-A3C.

consideration of the dynamic state of the environment. As the A3C algorithm is asynchronous, it can be well combined with FedTSE. Therefore, in the simulation process, we can optimize problem $\mathbf{P3}$ and reduce the cost of IoV greatly. We also evaluate the performance of delay in FedTSE with different methods. By analyzing Figure 6, we can see that the cost of the system is mostly consumed in computing resources, which is consistent with our hypothesis. The A3C algorithm also can reduce the real-time average delay more than ϵ-greedy. We propose to use an A3C-based algorithm to solve problem $\mathbf{P3}$, which will reduce the system delay gradually.

VII. Conclusion

In this paper, we have developed an FL framework for privacy-preserved TSE by integrating LSTM and DRL for optimizing resource consumption with high traffic prediction accuracy. In the proposed framework, the LSTM-based local training model captures temporal information and hidden nexus between vehicular speed and traffic flow. In order to reduce the training cost of FedTSE in IoV, we have established an optimization problem to minimize the computation and communication cost of FedTSE. An A3C learning algorithm is employed to solve the problem to make parameter downloading/uploading decisions by adapting to the changing network environment. Simulation results demonstrate that FedTSE effectively reduces MSE value in prediction and the total system cost. For our future work, we will discuss how to use multi-source data to assess the accident risk in cities.

References

[1] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Vehicular Communications*, vol. 20, pp. 100 184–100 219, 2019.

[2] N. Lu, N. Zhang, N. Cheng, X. Shen, J. W. Mark, and F. Bai, "Vehicles meet infrastructure: Toward capacity–cost tradeoffs for vehicular access networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1266–1277, 2013.

[3] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.

[4] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Networks and Applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[5] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.

[6] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transportation Research Part C: Emerging Technologies*, vol. 118, pp. 102 674–102 687, 2020.

[7] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Communications Surveys Tutorials*, vol. 23, no. 2, pp. 1342–1397, 2021.

[8] Z. Du, C. Wu, T. Yoshinaga, K. A. Yau, Y. Ji, and J. Li, "Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 45–61, 2020.

[9] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "ConSTGAT: Contextual Spatial-Temporal Graph Attention Network for Travel Time Estimation at Baidu Maps," in *Proc. 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 2697–2705.

[10] Y. Ren, D. Zhao, D. Luo, H. Ma, and P. Duan, "Global-Local Temporal Convolutional Network for Traffic Flow Prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–7, 2020.

[11] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," *Proc. 35th AAAI Conference on Artificial Intelligence, Virtual Event, February 2-9, 2021*, pp. 4189–4196.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proc. 20th AISTATS Conference, Fort Lauderdale, USA,20-22 April, 2017*, pp. 1273–1282.

[13] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1–1, 2021.

[14] Y. Chen, X. Sun, and Y. Jin, "Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 4229–4238, 2020.

[15] X. Yuan, J. Chen, N. Zhang, X. Fang, and D. Liu, "A federated bidirectional connection broad learning scheme for secure data sharing in internet of vehicles," *China Communications*, vol. 18, no. 7, pp. 117–133, 2021.

[16] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[17] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain Empowered Asynchronous Federated Learning for Secure Data Sharing in Internet of Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

[18] S. Anbalagan, A. K. Bashir, G. Raja, P. Dhanasekaran, G. Vijayaraghavan, U. Tariq, and M. Guizani, "Machine-Learning-Based Efficient and Secure RSU Placement Mechanism for Software-Defined-IoV," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 950–13 957, 2021.

[19] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[20] L. Liu, J. Feng, Q. Pei, C. Chen, Y. Ming, B. Shang, and M. Dong, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor–critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2021.