2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 5-8, 2025. Vienna, Austria

2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 979-8-3315-3358-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/SMC58881.2025.11342616

Privacy-Preserving Estimated Time of Arrival Prediction with Lightweight Multi-Task Federated Learning

Jiahui Zhai[1], Jing Bi[1], Haitao Yuan[2], Ziqi Wang[3], Hongyao Ma[1], Chen Wang[1] and Jia Zhang[4]

[1]College of Computer Science, Beijing University of Technology, Beijing, 100124, China
[2]School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, China
[3]School of Software Technology, Zhejiang University, Ningbo 315100, China
[4]Dept. of Computer Science, Southern Methodist University, Dallas, TX 75275, USA

***Abstract*—Accurate estimated time of arrival (ETA) prediction for long vehicular trips remains challenging in intelligent transportation systems (ITS) due to heterogeneous traffic patterns and limited local data availability. While federated learning (FL) addresses privacy concerns by decentralizing data training, traditional FL frameworks often struggle with high computational costs and poor adaptability to multi-task scenarios. To overcome these limitations, this paper proposes a Lightweight Multi-task Federated Learning (LMFL) framework for efficient and privacy-preserving ETA prediction. LMFL integrates a novel SE-CIFG, combining a Squeeze-Excitation (SE) attention module to prioritize critical spatio-temporal features and a Coupled Input and Forget Gate (CIFG) to simplify long-term traffic dependency modeling. Additionally, LMFL employs a Federated Gradient Compression Algorithm (Fed-GCA) to reduce communication overhead between edge and cloud using adaptive thresholding and sparse tensor encoding. Real-world traffic simulation dataset demonstrates that LMFL achieves significantly higher predictive accuracy compared to existing methods, achieving an average 17.1% improvement in prediction precision while reducing training time by 4.1%.**

***Index Terms*—Estimated time of arrival, travel time prediction, federated learning, vehicular networks, intelligent transportation systems.**

I. Introduction

With the rapid advancement of technologies such as artificial intelligence and edge computing, intelligent transportation systems (ITS) focus has shifted from infrastructure construction to the overall optimization of system operations [1], [2]. In this context, analyzing and applying large-scale real-time traffic data have become particularly crucial. Among various research directions, estimated time of arrival (ETA), the technique of predicting the travel time of a vehicle[1] required from an origin to a destination at a specific time has emerged as a core topic in ITS research [3]. Accurate and efficient ETA prediction enhances traffic management systems' effectiveness and significantly improves the travel experience.

Machine learning (ML) methods have been successfully employed in multiple transportation domains to achieve high-precision ETA predictions [4]–[6]. Novak *et al.* [4] introduce a recursive ML framework for 15-minute-interval urban travel time forecasting. Li *et al.* [5] present the integration of crowdsourced speed data with general transit feed specification and interaction networks to model transit-traffic interactions in urban roads. Huang *et al.* [6] propose a graph transformer framework for travel time estimation to optimize transportation systems and enhance urban mobility. However, the above studies pose significant risks of data breaches and associated damages. To address data accessibility and privacy challenges, federated learning (FL), a decentralized training framework, has gained significant attention as a privacy-preserving solution that enables secure modelling processes without centralized data sharing [7], [8]. However, traditional FL often suffers from high computational costs and limited adaptability to multiple tasks, restricting their efficiency and practicality in real-world applications.

Unlike the above-mentioned studies, this work proposes a Lightweight Multi-task Federated Learning (LMFL) to jointly achieve privacy-preserving, efficient, and versatile collaborative learning. Specifically, LMFL leverages a lightweight ETA prediction model named SE-CIFG, which integrates Squeeze-Excitation (SE) attention module and Coupled Input and Forget Gate (CIFG) to enhance prediction performance. LMFL leverages the SE attention module to automatically prioritize influential spatial-temporal features in traffic data by smartly adjusting the weight of different data channels. Subsequently, it integrates CIFG to simplify information processing through unified gating mechanisms, enabling efficient capture of long-term traffic pattern dependencies. Moreover, it employs a Federated Gradient Compression Algorithm (Fed-GCA) to reduce the data transmission cost between edge and cloud for improving the efficiency of ETA prediction using adaptive thresholding and sparse tensor encoding. A

This work was supported by the National Natural Science Foundation of China under Grants 62173013 and 62473014; in part by the Beijing Natural Science Foundation under Grants L233005 and 4232049; in part by Beihang World TOP University Cooperation Program; and in part by the 2023 International Cooperation Training Program for Innovative Talents ("Double First-class" Construction Special Program-"Artificial Intelligence + Internet of Things") of the China Scholarship Council (CSC). (Corresponding author: Jing Bi)

[1]ETA can also refer to the time needed for aircraft, ships, and computer files to reach their destinations. However, this paper focuses on vehicle movement.

dataset collected from real-world simulated environments is utilized to evaluate LMFL. Experimental results demonstrate that LMFL outperforms other comparative algorithms, confirming its capability to achieve more accurate ETA predictions.

II. Proposed Methodology

This section presents the problem definition and an overview of LMFL. Three components, including the lightweight ETA prediction model, gradient compress algorithm, and multi-task FL framework, are integrated into LMFL to enhance the prediction accuracy for ETA in distributed ITS.

A. Problem Definition

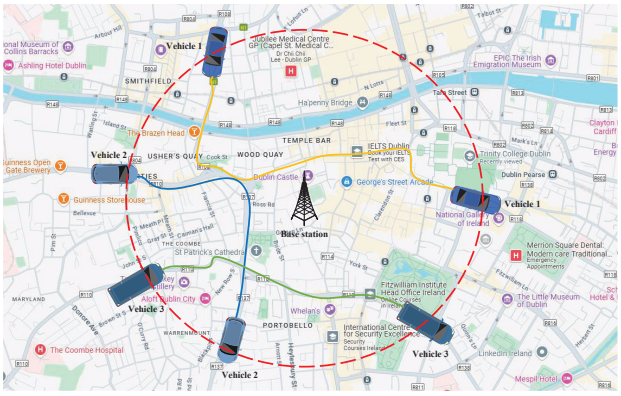

Fig. 1. Multi-vehicle data collection within a BS's coverage.

Fig. 1 illustrates multiple vehicles traversing a networked region through distinct trajectories. As each vehicle enters and exits the coverage area (marked by a red circle) of the centrally located base station (BS), it records and transmits corresponding timestamps and location data. BS serves as the primary data aggregation point. Each vehicle travels a unique path through the networked region. This enables analysis of coverage duration and data exchange events. Let $\mathcal{J}$ denote a set of traffic BSs, and J represents the number of traffic BSs. To characterize real-time road traffic dynamics, each BS j ($1 \leq j \leq J$) calculates the ETA $x_j(t)$ at time t for all vehicles that traversed its coverage area during the preceding time interval Δt (*e.g.,* 1 sec.). BSs subsequently conduct ETA predictions with historical data. Specifically, we define $(x_j(t+(1-l)\Delta t), x_j(t+(2-l)\Delta t), ..., x_j(t))$ as the input data sequence for BS j when predicting future ETA values at time t. l represents the temporal lag parameter. We define $(\hat{x}_j(t+\Delta t), \hat{x}_j(t+2\Delta t), ..., \hat{x}_j(t+h\Delta t))$ as the corresponding multi-horizon prediction outputs. $\hat{x}_j(\cdot)$ denotes predicted ETA values and h indicates the maximum prediction horizon.

To ensure accurate ETA predictions, each traffic BS trains ML models using locally collected traffic data and solves the following optimization problem:

$$\arg\min_{\boldsymbol{w}} \sum_{i=1}^{S_j} f(\boldsymbol{w}; \boldsymbol{x}_{j,i}, \boldsymbol{y}_{j,i}), \tag{1}$$

where S_j denotes the total number of training samples in the local dataset of BS j. The input sequence i is defined as $\boldsymbol{x}_{j,i} = (x_{j,i}(t+(1-l)\Delta t), x_{j,i}(t+(2-l)\Delta t), \ldots, x_{j,i}(t))$, and its corresponding multi-horizon target output is $\boldsymbol{y}_{j,i} = (x_{j,i}(t+\Delta t), x_{j,i}(t+2\Delta t), \ldots, x_{j,i}(t+h\Delta t))$. Here, $f(\boldsymbol{w}; \boldsymbol{x}_{j,i}, \boldsymbol{y}_{j,i})$ represents the loss function of the ML model with parameters $\boldsymbol{w}$. It is evaluated on the training pair $(\boldsymbol{x}_{j,i}, \boldsymbol{y}_{j,i})$. $f(\cdot)$ critically influences the FL performance and depends on the application context.

B. Lightweight ETA Prediction Model

Long short-term memory (LSTM) achieves high accuracy when predicting time series data [9]. As a variant of the LSTM model, CIFG reduces the structural complexity of LSTM without compromising prediction accuracy [10]. Unlike standard LSTM that employs separate input and forget gates, CIFG simultaneously uses a single gate to control both the input and the cell state's recursive connection. In practice, CIFG reduces training time by about 20%. Table I shows the corresponding formulas of each structure in the CIFG unit.

TABLE I
CORRESPONDING FORMULA OF EVERY STRUCTURE IN CIFG UNIT

Structure	Equation
Input gate	$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$
Forgetting gate	$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$
Cell state of current input	$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_C)$
Current cell state	$C_t = f_t \times C_{t-1} + (1-f_t) \times \tilde{C}_t$
Output gate	$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$

Furthermore, we choose the squeeze-excitation (SE) module because of its lightweight design and ease of integration into various neural networks to enhance the prediction accuracy of CIFG [11]. As shown in Fig. 2, SE operates in three main steps. First, it extracts the feature map from the input and applies a squeeze operation. Second, it introduces an excitation function to generate a weight matrix. Finally, it multiplies the weight matrix with the feature map to complete the attention operation. Therefore, we design a lightweight ETA prediction model named SE-CIFG.

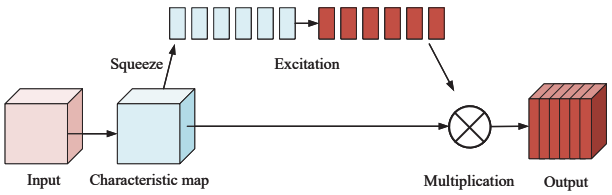

Fig. 2. Structure of SE attention module.

C. Gradient Compress Algorithm

In the horizontal FL framework, each BS needs to send updated parameters to the cloud during each training round to complete parameter aggregation and obtain the aggregated gradient parameter of each training round. Aggregating parameters calculated by all BSs result in massive

transmission and time costs. To solve this problem, some BSs are usually selected as the dataset source of training, and the appropriate GCA calculates the parameter results of each round. We propose Fed-GCA to reduce the number of parameter transmissions in the training process and improve the efficiency of gradient aggregation calculation. Fed-GCA is expected to improve the overall efficiency of distributed training without losing training information. Different from standard horizontal FL algorithms, such as FedAvg [12], FedProx [13], *etc.*, Fed-GCA carries out local gradient accumulation on the gradient generated by each round of training through threshold judgment in BSs, then uploads the qualified gradient to the cloud to complete the gradient descent process to obtain the result of each round of training. Fed-GCA operates in each BS, aiming to reduce the gradient to zero. Let $\boldsymbol{G}_r^j$ denote the gradient generated by BS j in the r training round. $\boldsymbol{G}_r^j$ is updated as:

$$\boldsymbol{G}_r^j = \boldsymbol{G}_r^j + \frac{1}{J}\nabla \sum_{i=1}^{S_j} f(\boldsymbol{w}; \boldsymbol{x}_{j,i}, \boldsymbol{y}_{j,i}). \tag{2}$$

The gradient clipping function is introduced to clip the obtained results, which is defined as:

$$\boldsymbol{G}_r^j = \text{Local\_gradient\_clipping}(\boldsymbol{G}_r^j). \tag{3}$$

Fed-GCA sets a threshold T for local gradient accumulation on each parameter $\boldsymbol{w}$. If gradients exceed T, they are transmitted immediately. If they fall below T, they are accumulated until they reach T, and then they are transmitted. Fed-GCA reduces the number of gradient transmissions and the total amount of transmitted data. To address the lag caused by local gradient accumulation, we make T dynamic. If more than n gradients consistently fall below the current T, we lower T to the highest value among these gradients. Conversely, if more than n gradients exceed the current T, we raise T to the lowest value among them. Thus, T is updated as follows:

$$T = \begin{cases} \min\{|g| \in \boldsymbol{G}_r^j || g| > T\}, & \text{if} |\{|g| \in \boldsymbol{G}_r^j || g| > T\}| > n, \\ \max\{|g| \in \boldsymbol{G}_r^j || g| < T\}, & \text{if} |\{|g| \in \boldsymbol{G}_r^j || g| < T\}| > n, \\ T, & \text{otherwise.} \end{cases} \tag{4}$$

Finally, Fed-GCA obtains the aggregation gradient $\boldsymbol{G}_r$ and the parameter $\boldsymbol{w}$, which are expressed as:

$$\boldsymbol{G}_r = \frac{1}{J}\sum_{j=1}^{J} \boldsymbol{G}_r^j, \tag{5}$$

$$\boldsymbol{w}_{r+1} = \boldsymbol{w}_r + \eta\, \mathbb{E}[\boldsymbol{G}_r], \tag{6}$$

where η represents the learning rate.

D. Multi-Task FL Framework

We propose LMFL to address the challenge of limited local training data and to enhance prediction accuracy. Specifically, BSs first apply divisive hierarchical clustering to split their local data into multiple clusters. Then, LMFL is used to collaboratively train a prediction model for each data cluster across all traffic BSs. As shown in Fig. 3, LMFL employs a

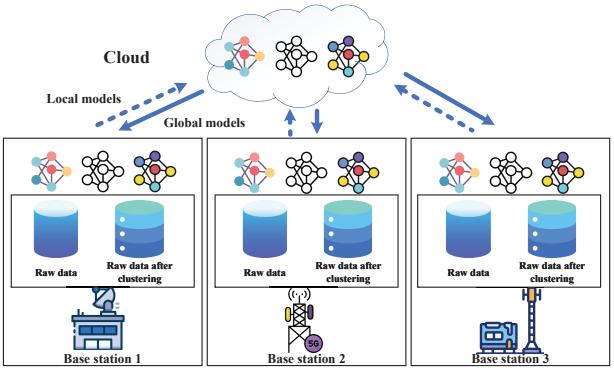

Fig. 3. LMFL model for ETA prediction.

divisive clustering method at each BS to partition the local data into M clusters. LMFL addresses the strong spatio-temporal correlations often observed in BS-collected traffic data, where similar traffic tasks frequently recur. Factors that are used to distinguish these tasks include road types (*e.g.*, highways, city streets, rural roads), vehicle categories (*e.g.*, passenger cars, trucks, public transport), and temporal factors (*e.g.*, peak hours, off-peak hours, seasonal variations). As a result, each cluster represents a distinct traffic task with the collected traffic data. Once the local data is divided into distinct clusters, LMFL trains the ETA prediction models. Specifically, LMFL aims to solve the following optimization problem for each data cluster [14]:

$$\arg\min_{\boldsymbol{w}_m \in \mathbb{R}} F_m(\boldsymbol{w}_m), \forall m \in \{1, \ldots, M\}, \tag{7}$$

$$\begin{aligned} F_m(\boldsymbol{w}_m) &\triangleq \frac{1}{S_{(m)}} \sum_{j \in \mathcal{J}} \sum_{i=1}^{S_{m,j}} f(\boldsymbol{w}_m; \boldsymbol{x}_{m,j,i}, \boldsymbol{y}_{m,j,i}) \\ &\triangleq \frac{1}{S_{(m)}} \sum_{j \in \mathcal{J}} F_{m,j}(\boldsymbol{w}_m), \end{aligned} \tag{8}$$

where $(\boldsymbol{x}_{m,j,i}, \boldsymbol{y}_{m,j,i})$ denotes the training data sample i belonging to cluster m at BS j. $S_{m,j}$ is the total number of such samples. Furthermore, $S_{(m)} \triangleq \sum_{j \in \mathcal{J}} S_{m,j}$ represents the total number of training data samples in cluster m across all BSs. Lastly, $F_{m,j} \triangleq \sum_{i=1}^{S_{m,j}} f(\boldsymbol{w}_m; \boldsymbol{x}_{m,j,i}, \boldsymbol{y}_{m,j,i})$ denotes the loss function for cluster m at BS j.

To solve (8), LMFL uses an iterative update scheme, as illustrated in Algorithm 1. First, cloud initializes a global learning model with parameters $\boldsymbol{w}_{m,0}$ for each cluster $m \in \{1, ..., M\}$ and distributes these parameters to the BSs. Then, each BS applies stochastic gradient descent using its local data from cluster $\tilde{m} \in \{1, ..., M\}$ to update the learning models with the received global parameters at the first communication round $r=1$, which is expressed as:

$$\boldsymbol{w}_{m,r+1,j} = \boldsymbol{w}_{m,r,j} + \eta \nabla F_{m,j}(\boldsymbol{w}_{m,r,j}), j \in \mathcal{J}. \tag{9}$$

Afterwards, each traffic BS transmits its updated model parameters to the cloud via the uplink. Cloud then aggregates

all received local parameters to update the global model parameters, as follows:

$$\boldsymbol{w}_{m,r} = \frac{1}{S_{(m)}} \sum_{j \in \mathcal{J}} S_{m,j} \boldsymbol{w}_{m,r,j}. \tag{10}$$

The updated global model parameters are then sent to all traffic BSs. This marks the completion of one communication round. In each new round, the cloud and the BSs repeat the same process. During LMFL, both local and global models undergo iterative refinements, causing the total loss function $F_m(\boldsymbol{w}_m)$ for each data cluster ($m \in \{1, ..., M\}$) to decrease continuously [15]. As a result, BSs achieve higher accuracy in ETA predictions for any local data cluster and effectively address the problem of limited local training data. In addition, because the BSs do not transmit large raw datasets, LMFL safeguards privacy and reduces communication costs.

Algorithm 1: LMFL for the ETA prediction

Input: Traffic BSs set ($\mathcal{J}$), total number of data samples (S), loss function (F), number of the clusters (M), number of communication rounds (R), learning rate (η)

Output: ETA prediction model for different traffic tasks

/* **Initialization process** */

According to the chosen clustering criteria (*e.g.*, road types, vehicle categories, temporal factors), each traffic BS applies a divisive clustering method to split its collected data into M clusters;

Cloud initializes a global learning model with parameters $\boldsymbol{w}_{m,0}$ for each data cluster $m \in \{1, ..., M\}$;

/* **Fed-GCA process** */

for $r \leftarrow 0$ **to** $R-1$ **do**

  Cloud sends $\boldsymbol{w}_{m,r}$ to all BSs;

  BS j trains SE-CIFG by (9) on its data cluster m, obtaining $\boldsymbol{w}_{m,r,j}$, which is then sent to the cloud;

  Cloud aggregates the model parameters $\boldsymbol{w}_{m,r,j}$ obtained from the BSs and updates the global learning model parameters $\boldsymbol{w}_{m,r}$ according to (10);

end

III. Performance Evaluation

This section presents the simulation settings and datasets. Then, LMFL is compared with its state-of-the-art peers and ablation experiments to predict ETA in distributed ITS.

A. Simulation Settings

This work presents a comprehensive set of ablation and comparison experiments to demonstrate the performance of LMFL. We compare LMFL with LSTM and FedAvg. Furthermore, SE and CIFG serve as state-of-the-art benchmark peers for conducting ablative experiments. All algorithms are independently repeated 10 times, and the average-performing ones are selected through a testing set to yield the final comparison results. This work aims to predict the ETA for the following short-term (5 mins.), mid-term (15 mins.), and long-term (30 mins.). All experiments use PyTorch on a computer with an Intel® Xeon® Gold 6152 CPU with 10-core 2.1GHz processors, 30GB of memory, and an NVIDIA GeForce RTX 3090 GPU. The simulation platform is SUMO version 1.16.02, one of the most widely used open-source microscopic traffic simulators. This SUMO-generated map depicts four distinct regions in Fig. 4, each reflecting a typical road configuration for a specific context.

Fig. 4. Simulation map.

B. Datasets

This work focuses on predicting ETA, and we utilize the simulation dataset collected by SUMO. The training data is divided according to the collected time in the divisive hierarchical clustering to implement LMFL. In the simulation, we consider the traffic data collected with time interval Δt=1 sec., over a total simulation duration of a weekday with 24 hours. This work employs a synthetic data generation procedure to model vehicular mobility across four heterogeneous BSs within a 500-unit coverage area, resulting in 10,080 unique vehicles, each assigned a single origin-destination pair. We record vehicle positions with periodic GPS updates, including information at the trip's start and during the journey. Each record contains a vehicle ID, current coordinates, the timestamp when the vehicle entered the BS, the current timestamp, and the BS affiliation. We preserve time consistency from trip initiation through intermediate updates to final arrival. This ensures robust modeling for ETA prediction.

C. Experimental Results

Table II summarizes the comparison results of all algorithms on the dataset. There is the prediction task, which utilizes all features from the dataset as input to predict the target sequence. We predict the ETA given the dataset. To evaluate the performance of LMFL and peer algorithms, root mean squared error (RMSE), mean absolute error (MAE),

mean absolute percentage error (MAPE), and coefficient of determination (R^2) are used as metrics. Table II shows that LMFL significantly outperforms other algorithms, obtaining all the best results. Specifically, LMFL exhibits an average reduction of 17.1% in MAE compared to the existing methods.

TABLE II
PERFORMANCE COMPARISON OF ALL METHODS WITH THE DATA SET

Methods	RMSE	MAE	MAPE	R^2
LSTM+FedAvg	183.4663	148.7035	0.7734	-0.0027
CIFG+FedAvg	183.5888	148.7462	0.7717	-0.0041
SE-LSTM+FedAvg	177.1814	136.8470	0.7037	0.0651
SE-CIFG+FedAvg	180.5738	144.2538	0.7230	0.0115
LMFL	**148.5816**	**119.7455**	**0.6001**	**0.3423**

Note:
1) Boldfaced results in each partition typically show the lowest RMSE, MAE, MAPE, and R^2 values.
2) LMFL training completes global model convergence in a single training session.

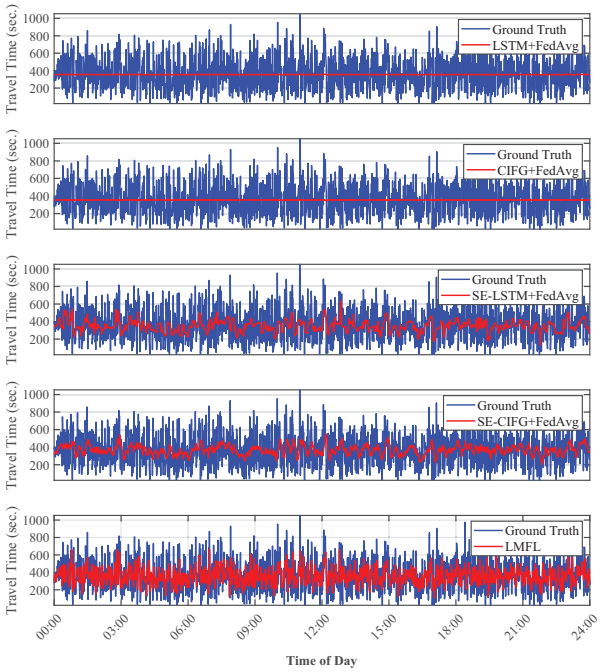

Fig. 5. Prediction results of different algorithms on the test dataset.

Fig. 5 illustrates the prediction results of five algorithms for the entire test set of the dataset. All algorithms accurately capture the daily travel time trends without overfitting. Both LSTM and CIFG exhibited relatively poor fitting performance, failing to capture the trend variations in the sequence. SE-LSTM+FedAvg closely follows the ground truth for peak periods (*e.g.,* morning and evening rush hours). SE-CIFG+FedAvg performs better in modeling short-term fluctuations, such as abrupt changes during midday, aligning well with ground truth variations. LMFL exhibits the highest alignment with ground truth data, particularly excelling in capturing complex temporal variations during rush hours. These results highlight that SE, CIFG, and Fed-GCA all enhance the prediction performance of LMFL.

TABLE III
PERFORMANCE COMPARISON OF LMFL WITH SINGLE-TASK FL

Methods	Terms	RMSE	MAE	MAPE	R^2
Single-task FL	Short-term	177.0951	136.7670	0.7151	0.0660
LMFL		**148.5816**	**119.7455**	**0.6001**	**0.3423**
Single-task FL	Mid-term	189.5041	153.7106	0.8093	-0.0691
LMFL		**179.6178**	**144.8890**	**0.7553**	**0.0395**
Single-task FL	Long-term	188.0920	152.7019	0.8098	-0.0528
LMFL		**183.1629**	**148.6059**	**0.7827**	**0.0016**

Table III shows ETA prediction accuracy for three maximum prediction time horizons, *i.e.,* short-term (5 mins.), mid-term (15 mins.), and long-term (30 mins.) compared with single-task FL for the dataset. As shown in III, LMFL improves prediction accuracy compared to single-task FL in all three maximum prediction time horizons. In particular, compared with the traditional single-task FL, its counterpart in the FL can achieve a better ETA prediction, highlighting the importance of performing collaborative training for traffic prediction. More importantly, LMFL obtains more accurate ETA predictions than the single-task FL, showing the necessity of using a multi-task learning framework to enhance the ETA prediction performance further. Specifically, LMFL exhibits an average reduction of 7.0% in MAE compared to the single-task FL.

TABLE IV
ABLATION STUDIES OF LMFL WITH THREE METHODS

Methods	RMSE	MAE	MAPE	R^2
w/o SE	152.4238	123.0060	0.6185	0.3078
w/o CIFG	150.7497	121.5366	0.6120	0.3229
w/o Fed-GCA	180.5738	144.2538	0.7230	0.0115
LMFL	**148.5816**	**119.7455**	**0.6001**	**0.3423**

Table IV shows the ablation studies of LMFL with three methods. The addition of each method can improve the prediction. Specifically, SE improves the prediction performance by 2.7% (123.0→119.7) in MAE, CIFG and Fed-GCA enhances the prediction performance by 1.5% (121.5→119.7) and 17.0% (144.3→119.7) in MAE. The reason is that SE dynamically amplifies critical spatial-temporal patterns in traffic data by reweighting channel-wise features. CIFG simplifies learning through unified gates, effectively modeling long-term traffic dependencies with lower computational overhead. Fed-GCA contributes the most significant gain by compressing transmitted gradients through adaptive thresholding and sparse encoding, reducing communication overhead while maintaining prediction reliability. LMFL optimizes feature extraction efficiency and collaborative learning performance, ultimately minimizing prediction errors.

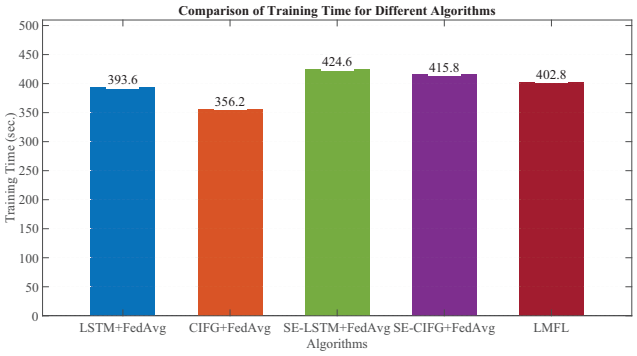

Fig. 6. Training time of five algorithms.

Fig. 6 compares the training time of five algorithms given the dataset. It is observed that the training time of LMFL does not significantly increase with adding more features. This indicates that LMFL's training efficiency (the prediction accuracy ratio to training time) is greatly improved compared to the other algorithms. Additionally, LMFL's training time is lower than SE-LSTM+FedAvg and SE-CIFG+FedAvg by 4.1% on average. The reason is that Fed-GCA reduces communication overhead between edge and cloud using adaptive thresholding and sparse tensor encoding.

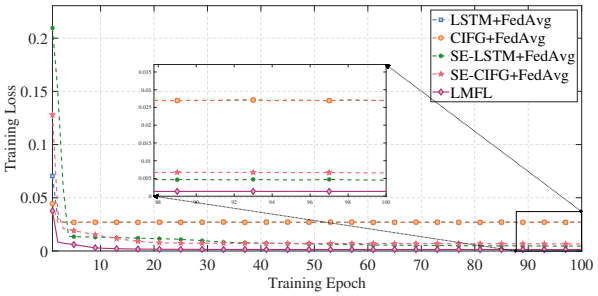

Fig. 7. Training loss values of different methods.

Fig. 7 shows the loss values of different methods. After iteration 5, it is evident that LMFL's loss values are comparatively smaller than those of other models. This demonstrates that LMFL possesses superior modeling capabilities compared with other variants. Consequently, LMFL outperforms other benchmark methods given the same setting.

IV. Conclusion

Driven by recent advancements in artificial intelligence and edge computing, federated learning (FL) is increasingly employed in privacy-sensitive intelligent transportation systems. However, accurately predicting the estimated time of arrival (ETA) for long vehicular trips spanning multiple tasks remains challenging due to heterogeneous traffic patterns and insufficient local data. This work proposes a Lightweight Multi-task Federated Learning (LMFL) framework to address the critical challenges of privacy protection and multi-task adaptability in ETA prediction. By integrating the Squeeze-Excitation (SE) attention module and Coupled Input and Forget Gate (CIFG) with Federated Gradient Compression Algorithm (Fed-GCA), LMFL achieves privacy-preserving collaborative learning while maintaining computational efficiency and adaptability to diverse traffic tasks. SE enables dynamic identification of critical spatial-temporal features, while CIFG simplifies long-term dependency modeling. Fed-GCA further reduces communication overhead through adaptive gradient compression, making LMFL suitable for resource-constrained edge devices. Experiments on real-world simulated datasets confirm LMFL's accuracy and efficiency superiority over existing methods. Specifically, it improves the prediction accuracy and training efficiency by 17.1% and 4.1% on average compared with existing methods, respectively.

References

[1] S. Li, J. Li, Y. Liang, H. Zhang, S. Wu, S. Wang, and L. Cheng, "Td-sas: A trust-aware and decentralized speed advisory system for energy-efficient autonomous vehicle platoons," *IEEE Trans. Intell. Veh.*, pp. 1–16, 2023.

[2] J. Zhai, J. Bi, and H. Yuan, "Collaborative computation offloading for cost minimization in hybrid computing systems," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2022, pp. 1772–1777.

[3] H. Hong, Y. Lin, X. Yang, Z. Li, K. Fu, Z. Wang, X. Qie, and J. 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*, 2020, pp. 2444–2454.

[4] H. Novak, F. Bronić, A. Kolak, and V. Lešić, "Data-driven modeling of urban traffic travel times for short-and long-term forecasting," *IEEE Trans. Intell. Transp. Syst.*, 2023.

[5] X. Li, A. Cottam, and Y.-J. Wu, "Transit arrival time prediction using interaction networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 3833–3844, 2023.

[6] Y. Huang, X. Song, S. Zhang, L. Li, and J. J. Yu, "Gt-tte: Modeling trajectories as graphs for travel time estimation," *IEEE Internet of Things Journal*, 2024.

[7] R. Zhu, M. Li, J. Yin, L. Sun, and H. Liu, "Enhanced federated learning for edge data security in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13 396–13 408, 2023.

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

[9] Z. Wang, X. Wu, J. Bi, H. Yuan, J. Zhang, and M. Zhou, "Long-term water quality prediction based on intelligent optimization and seasonal-trend decomposition," in *2024 IEEE 20th International Conference on Automation Science and Engineering*. IEEE, 2024, pp. 264–269.

[10] H. Yadav and A. Thakkar, "Noa-lstm: An efficient lstm cell architecture for time series forecasting," *Expert Systems with Applications*, vol. 238, p. 122333, 2024.

[11] J. Jin, X. Wu, I. Daly, W. Chen, X. He, X. Wang, and A. Cichocki, "Squeeze and excitation-based multiscale cnn for classification of steady-state visual evoked potentials," *IEEE Internet of Things Journal*, 2024.

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

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

[14] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[15] T. Zeng, O. Semiari, M. Mozaffari, M. Chen, W. Saad, and M. Bennis, "Federated learning in the sky: Joint power allocation and scheduling with uav swarms," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.