MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Multi-Task Federated Learning for Traffic Prediction and Its Application to Route Planning

Zeng, Tengchan; Guo, Jianlin; Kim, Kyeong Jin; Parsons, Kieran; Orlik, Philip V.; Di Cairano, Stefano; Saad, Walid

TR2021-083 July 12, 2021

Abstract

A novel multi-task federated learning (FL) framework is proposed in this paper to optimize the traffic prediction models without sharing the collected data among traffic stations. In particular, a divisive hierarchical clustering is first introduced to partition the collected traffic data at each station into different clusters. The FL is then implemented to collaboratively train the learning model for each cluster of local data distributed across the stations. Using the multi-task FL framework, the route planning is studied where the road map is modeled as a time-dependent graph and a modified A* algorithm is used to determine the route with the shortest traveling time. Simulation results showcase the prediction accuracy improvement of the proposed multi-task FL framework over two baseline schemes. The simulation results also show that, when using the multi-task FL framework in the route planning, an accurate traveling time can be estimated and an effective route can be selected.

IEEE Intelligent Vehicles Symposium

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Multi-Task Federated Learning for Traffic Prediction and Its Application to Route Planning

Tengchan Zeng*†, Jianlin Guo*, Kyeong Jin Kim*, Kieran Parsons*, Philip Orlik*,
Stefano Di Cairano*, and Walid Saad†
*Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, 02139 USA
Email:{guo, kkim, parsons, porlik, dicairano}@merl.com
†Wireless@VT, Electrical and Computer Engineering Department, Virginia Tech, Blacksburg, VA, 24061 USA
Emails:{tengchan, walids}@vt.edu

***Abstract*—A novel multi-task federated learning (FL) framework is proposed in this paper to optimize the traffic prediction models without sharing the collected data among traffic stations. In particular, a divisive hierarchical clustering is first introduced to partition the collected traffic data at each station into different clusters. The FL is then implemented to collaboratively train the learning model for each cluster of local data distributed across the stations. Using the multi-task FL framework, the route planning is studied where the road map is modeled as a time-dependent graph and a modified A* algorithm is used to determine the route with the shortest traveling time. Simulation results showcase the prediction accuracy improvement of the proposed multi-task FL framework over two baseline schemes. The simulation results also show that, when using the multi-task FL framework in the route planning, an accurate traveling time can be estimated and an effective route can be selected.**

***Index Terms*—Traffic prediction, multi-task federated learning, data clustering, time-dependent graph, optimal route planning.**

I. Introduction

To facilitate the development of intelligent transportation systems (ITSs), it is imperative to have an accurate prediction for the traffic conditions, such as traffic flow and speed. This is due to the fact that, such knowledge can help drivers make effective travel decisions so as to mitigate the traffic congestion, increase the fuel efficiency, and alleviate the air pollution. These promising benefits enable the traffic prediction to play major roles in the advanced traveler information system, the advanced traffic management system, and the commercial vehicle operation that the ITSs target to achieve [1].

To reap all the aforementioned benefits, the traffic prediction must process the real-time and historical traffic data collected by traffic stations and mobile devices. For example, the inductive loop can measure the traveling speed by reading the inductance changes over time and such data can be used for the traffic speed prediction. In addition, the wide use of mobile devices (e.g., on-board global position systems and phones) enables the mobility data to be crowdsourced from the general public, further facilitating the unprecedented traffic data collection. Such emerging big data can substantially augment the data availability in terms of the coverage and fidelity and significantly boost the research interest on the traffic prediction [2]–[6].

The prior art on the traffic prediction can be mainly grouped into two categories. The first category focus on using parametric approaches, such as autoregressive integrated moving average (ARIMA) model [2] and Kalman filtering models [3]. When dealing with the traffic only presenting regular variations (e.g., recurrent traffic congestion occurred in morning and evening rush hour), the parametric approaches can achieve promising prediction results. However, due to the stochastic and nonlinear nature of the road traffic, the traffic predictions of using the parametric approaches can deviate from the actual values especially in the abrupt traffic [4]. Hence, instead of fitting the traffic data into a mathematical model as done by the parametric approach, an alternative way is using the nonparametric approaches where the machine learning (ML) based method is the most popular [4]–[6]. For example, a stacked autoencoder model is proposed to learn the generic traffic flow features for the predictions in [4]. The authors in [5] study the use of long short-term memory (LSTM) recurrent neural network (RNN) to predict the traffic flow, speed and occupancy, based on the data collected by the station and its upstream and downstream traffic stations. Along with the use of RNN, the work in [6] additionally utilizes the convolution neural network (CNN) to capture the latent traffic evolution patterns within the underlying road network.

Although the works in [4]–[6] focus on using advanced ML models for the traffic prediction, all of them study the traffic variations with a single-task learning (STL) model. However, in reality, due to the varying weather, changing road conditions (such as road work and accidents) and special events (e.g., football games and concerts), the traffic patterns on the road can vary significantly under different situations. Hence, using a STL model will be challenging to capture such diverse and complex traffic situations. Moreover, due to the limited on-chip memory available at the station, the local training data can be extremely insufficient and a promising prediction performance cannot be guaranteed. In addition, all these works assume that the data collected by the traffic stations can be shared with other stations or a centralized unit, like the traffic station accessing the data from its upstream and downstream stations in [5]. In reality, the collected data can contain the personal information, like the driving license plates captured by cameras and history trajectory of mobile phone users. In this case, directly sharing the traffic data among stations can raise the privacy issues. Meanwhile, the communication cost is another major concern.

The main contributions of this paper can be summarized as follows:

This work was done while Tengchan Zeng was working at MERL.

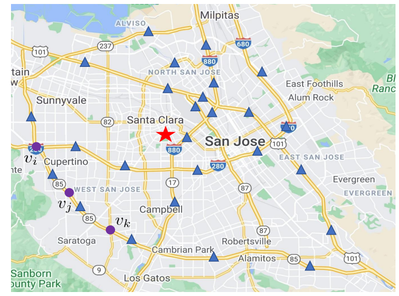
- We propose a novel multi-task federated learning (FL) framework to optimize the traffic prediction models for different traffic situations. In the framework, we first make use of the spatial-temporal dependence of collected traffic data and design a divisive hierarchical clustering to partition the traffic data at each station into a group of clusters. Then, for each data cluster distributed across stations, the FL is used where the learning model for each data cluster is collaboratively trained among all stations without sharing the actual data. In this case, as no local data is shared among stations, the proposed multi-task FL framework can protect the privacy and reduce the communication cost. The collaborative training process can also address the insufficiency of local training data.
- We implement the multi-horizon speed prediction from the multi-task FL framework into the route planning. In particular, the road network is modeled as a time-dependent graph and a modified A* algorithm is studied to obtain the optimal route with the least traveling time. We also design the interaction mechanism between the traffic prediction and route planning to ensure the promising route planning and traffic prediction under abrupt traffic situations.
- Using real data traces from Caltrans Performance Measurement System (PeMS), we validate the prediction accuracy improvement of the proposed multi-task FL framework over two baselines, i.e., single-task FL training and traditional STL schemes, in multiple criteria. We highlight the importance of performing a multi-task collaborative learning framework in the traffic prediction.
- With real traffic data collected in San Jose area and Los Angeles area and their road maps, we study the route planning performance where the multi-horizon speed predictions from the proposed framework is used in the modified A* algorithm. The simulation results show that, we can achieve more accurate traveling time estimation and route selection with less traveling time than Google map like scheme that solely accounts for the current traffic information and distance based scheme where no traffic information is considered in the planning process.

II. System Model

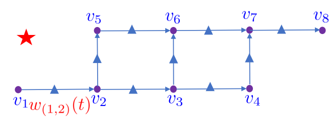
In this section, we introduce the basic multi-horizon traffic speed prediction model and route planning problem.

A. Multi-horizon Traffic Speed Prediction Model

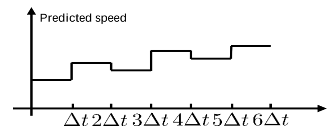
Consider a set $\mathcal{N}$ of N traffic stations where the station can be the toll station, loop detector, and camera. To capture the road traffic dynamics over time, station $n \in \mathcal{N}$ measures the average speed $x_n(t)$ at time t for all vehicles traversed in the past time interval Δt (e.g., $\Delta t = 5$ mins) and performs the speed prediction. We assume the data sample used for the prediction to be $(x_n(t+(1-l)\Delta t), x_n(t+(2-l)\Delta t), ..., x_n(t))$ with lag variable as l when the station $n \in \mathcal{N}$ predicts the future speed at time t. We also assume the multi-horizon speed



(a) Road map.



(b) Time-dependent graph.



(c) Piecewise linear speed.

Fig. 1. Traffic road map where the blue triangles, purple circles, and red pentagram, respectively, refer to the traffic stations, the vertices in the graph, and the traffic server.

prediction to be $(\hat{x}_n(t+\Delta t), \hat{x}_n(t+2\Delta t), ..., \hat{x}_n(t+h\Delta t))$ with $\hat{x}_n(\cdot)$ as the predicted speed value and h as the maximum prediction time horizon.

To guarantee that the traffic station can make accurate speed predictions, the stations use the local traffic data to train the ML model and solve the following optimization problem:

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

where S_n is the total number of training samples within the local data at station $n \in \mathcal{N}$, $\boldsymbol{x}_{n,i} = (x_{n,i}(t+(1-l)\Delta t), x_{n,i}(t+(2-l)\Delta t), ..., x_{n,i}(t))$ is the i-th input data sample, $\boldsymbol{y}_{n,i} = (x_{n,i}(t+\Delta t), x_{n,i}(t+2\Delta t), ..., x_{n,i}(t+h\Delta t))$ is the i-th target output speed data, and $f(\boldsymbol{w}, \boldsymbol{x}_{n,i}, \boldsymbol{y}_{n,i})$ is the loss function when the ML model with model parameters $\boldsymbol{w}$ is trained with data $(\boldsymbol{x}_{n,i}, \boldsymbol{y}_{n,i})$. The loss function plays a pivotal role in determining the FL performance, and the expression of the loss function is application specific. In the traffic prediction, the most common loss function is the mean squared error (MSE) [4].

For the purpose of traffic management, the stations will send the speed predictions to the traffic server over either wired network [7] or wireless network [8]. To avoid a large overhead over the traffic server, the frequency that the stations share the forecast results with the traffic server will be relatively low, e.g., 1 hour. As follows, the traffic server can broadcast the road map with traffic predictions to all the vehicles operating within its coverage. The on-board unit (OBU) inside the vehicle can then choose the optimal route from its current location to the destination with the shortest traveling time. In this next subsection, we will study how the vehicle determines the optimal route.

B. Route Planning Problem

After receiving multi-horizon speed predictions from traffic stations, the parameter server will first divide the long road segments in the road network into multiple sub-segments such that each road sub-segment is measured by a unique station. For example, in Fig. 1(a), the road connecting intersection v_i and intersection v_k has two traffic stations and therefore such road can be separated into two sub-segments (v_i, v_j) and (v_j, v_k) with v_j as the connecting point. Then, the parameter server will model the road network as a time-dependent graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, as shown in Fig 1(b). In particular, the set $\mathcal{V}$ of vertices includes the intersections and connecting points of any two adjoining road sub-segments, and the edge set $\mathcal{E}$ is thereby the road sub-segments connecting two adjacent vertices. Moreover, the weight $w_e(t) \in \mathcal{W}$ of edge $e \in \mathcal{E}$ is modeled as the traveling time at t, calculated as the ratio between the length of the road sub-segment and the predicted speed. Note that, different from the static graph where the weight associated to each edge is a constant value, the counterpart within the graph $\mathcal{G}$ is a time-varying variable due to the time-varying speed (e.g., the piecewise linear speed in Fig. 1(c)) to traverse each road sub-segment. Finally, the parameter server will broadcast the time-dependent graph to all vehicles in its communication range.

After receiving the time-dependent graph, the vehicle can determine the route $(\bar{v}_1 = s, ..., \bar{v}_i, \bar{v}_{i+1}, ..., \bar{v}_k = d)$ leaving the current location s at time t_s to the destination point d with the least traveling time as follows:

$$\arg\min_{(\bar{v}_1,...,\bar{v}_k)} t_d(s, t_s) - t_s, \tag{2}$$

$$\text{s.t.} \quad t_1 = t_s, \tag{3}$$

$$t_{i+1} = t_i + w_{(\bar{v}_i, \bar{v}_{i+1})}(t_i), i \in \{1, ..., k-1\}, \tag{4}$$

where $t_d(s, t_s)$ denotes traveling time leaving location s at time t_s to destination location d. The first constraint is due to the fact that the vehicle departures s at time t_s and the second constraint represents that the arrival time at $\bar{v}_{i+1}$ equals the sum of the departure time at $\bar{v}_i$ and the traversing time on road sub-segment $(\bar{v}_i, \bar{v}_{i+1})$ at time t_i. To obtain the optimal route, we need to tackle two key challenges. The first challenge is how to guarantee a high accuracy of multi-horizon traffic speed prediction at traffic stations. The reason is that, due to the limited on-chip memory available at the traffic stations, the local training data can be insufficient. In fact, due to the limited storage, only data pertaining to the most recent traffic, such as the traffic of past few days, can be stored where the data can be easily skewed and of poor quality. Thereby, the learning model solely trained by local data cannot guarantee an accurate speed prediction, leading to a possible failure to find the optimal route. Once we have accurate traffic predictions, we need to address the second challenge, i.e., determining the solution to the optimization problem in (2). The reason is that, different from the route planning problem in a static graph, the optimization problem focuses on a time-dependent graph where the weights are time-varying. In the following section, we will propose a multi-task FL framework and study a modified A* algorithm to address the aforementioned two challenges. We will also study the interaction mechanism between the multi-task FL framework and modified A* algorithm to further improve the route planning performance.

III. Multi-task FL Framework for the Traffic Speed Prediction and Route Planning Solution

To tackle the insufficient local training data and improve the prediction accuracy, we propose a multi-task FL framework. In particular, the stations will first use the divisive hierarchical clustering to partition their local data into different clusters. Then, the FL is used to collaboratively train the prediction model designated to each data cluster among all traffic stations.

A. Multi-task FL Framework

First of all, to capture different traffic situations existing in the collected traffic data, we use a divisive clustering method to partition the local data at each station into a group of M clusters, as shown in Fig. 2. This is motivated by the fact that the traffic data collected by the stations can have a strong spatial-temporal dependence where the same group of traffic situations exist. The criteria of divisive clustering can include the weather condition (e.g., rain, snow, and overcast), the time (e.g., weekday, weekend, and rush-hour and non-rush-hour traffic), and special events (e.g., concerts and football game). In this case, each data cluster can represent a unique traffic situation within the collected traffic data.

After the local data is grouped into different clusters, we can train the prediction models. In particular, the objective of the multi-task FL framework is to solve the following optimization problem for each data cluster [9]:

$$\arg\min_{\boldsymbol{w}_m \in \mathbb{R}} F_m(\boldsymbol{w}_m), \ \forall \ m \in \{1, ..., M\} \tag{5}$$

with

$$\begin{aligned} F_m(\boldsymbol{w}_m) &\triangleq \frac{1}{S_{(m)}} \sum\nolimits_{n \in \mathcal{N}} \sum\nolimits_{i=1}^{S_{m,n}} f(\boldsymbol{w}_m, \boldsymbol{x}_{m,n,i}, \boldsymbol{y}_{m,n,i}) \\ &\triangleq \frac{1}{S_{(m)}} \sum\nolimits_{n \in \mathcal{N}} F_{m,n}(\boldsymbol{w}_m), \end{aligned} \tag{6}$$

where $(\boldsymbol{x}_{m,n,i}, \boldsymbol{y}_{m,n,i})$ is the i-th training data sample belonging to cluster m at station n with $S_{m,n}$ as the total number of such data samples. $S_{(m)} \triangleq \sum_{n \in \mathcal{N}} S_{m,n}$ refers to the total number of training data samples belonging to cluster m across all stations and $F_{m,n} \triangleq \sum_{i=1}^{S_{m,n}} f(\boldsymbol{w}_m, \boldsymbol{x}_{m,n,i}, \boldsymbol{y}_{m,n,i})$ denotes the loss function of cluster m at station n.

To solve (5), the FL framework will use an iterative update scheme, as shown in Algorithm 1. In particular, the traffic server, operating as the parameter server, will first generate an initial global learning model with model parameters as $\boldsymbol{w}_{m,0}$ for cluster $m \in \{1, ..., M\}$ which will be sent to the stations. Then, at the first communication round, i.e., $j = 1$, all stations will use the received model parameters to update the learning models based on their own data of cluster $m \in \{1, ..., M\}$ by using the gradient descent:

$$\boldsymbol{w}_{m,j,n} = \boldsymbol{w}_{m,j-1} + \eta \nabla F_{m,n}(\boldsymbol{w}_{m,j-1}), n \in \mathcal{N}, \tag{7}$$

where η is the learning rate. As follows, the traffic stations will transmit their trained model parameters to the traffic server in the uplink. Next, the traffic server will aggregate all the received local model parameters to update the global model parameters, given by:

$$\boldsymbol{w}_{m,j} = \frac{1}{S_{(m)}} \sum\nolimits_{n \in \mathcal{N}} S_{m,n} \boldsymbol{w}_{m,j,n}, \tag{8}$$

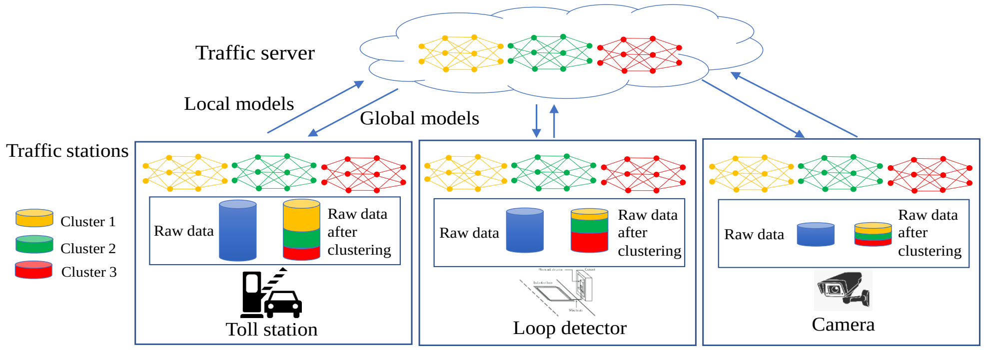


Fig. 2. Multi-task FL model for the traffic prediction.

Algorithm 1 Multi-task FL framework for the traffic speed prediction.

Input: $\mathcal{N}$, $S_{m,n}$, M, η, $m \in \{1, ..., M\}, n \in \mathcal{N}$.
Output: Speed prediction model for each traffic situation.
1. According to the clustering criteria (e.g., weather conditions, time, and special events), the divisive clustering method is used to partition collected data at each traffic station into M data clusters.
2. The traffic server generates the global learning model with model parameters $\boldsymbol{w}_{m,0}$, for data cluster $m \in \{1, ..., M\}$.
for $j = 0, ..., T-1$ **do**
  (a) The traffic server sends $\boldsymbol{w}_{m,j}, m \in \{1, ..., M\}$, to all stations.
  (b) Station $n \in \mathcal{N}$ trains the local learning models by using the gradient descent in (7) on its data clusters and obtain $\boldsymbol{w}_{m,j,n}$, which will be sent to the traffic server.
  (c) The traffic server aggregates the model parameters received from the stations and update the global learning model parameters based on (8).
end

which are then sent to all traffic stations. Each communication round will be followed by another round, and the same process will repeat among the traffic server and the stations in each round. In this case, as FL proceeds, the local and global models are sequentially updated, and the total loss function $F_m(\boldsymbol{w}_m), m \in \{1, ..., M\}$, for each cluster of local data will constantly decrease [10]. Hence, when the stations perform the speed prediction for any cluster of local data, the accuracy can be guaranteed and their insufficiency of local training data at traffic stations can be addressed. Moreover, the proposed multi-task FL framework can protect the privacy and reduce the communication cost as the stations do not share their large raw data.

B. Solution to the Route Planning

When following any route in the time-dependent graph, we assume an earlier departure will lead to an earlier arrival and a later departure will result in a later arrival. In this case, to solve the route planning problem in (2)–(4), we study a modified A* searching algorithm, as presented in Algorithms 2 and 3. Within the searching algorithm, the arrival time g_v and the heuristic total traveling time l_v, $v \in \mathcal{V}$, are initially set to infinity with the exception of the starting point s with $g_s = t_s$ and $l_s = g_s + h_d(g_s)$. Here, the heuristic total traveling time is defined as the sum of arrival time and heuristic traveling time h_d to the destination. The heuristic traveling time h_d to the destination is calculated by the ratio between the Euclidean distance to the destination and the maximum speed. Then, the searching process in the modified A* algorithm begins with the starting point s and extend to the adjacent vertices that have adjoining road sub-segments with s. For these adjacent vertices, their arrival time g will be updated by comparing the most recently assigned arrival time with the arriving time when taking the route from the starting point s. Meanwhile, their heuristic total traveling time l will be updated as well. Next, the vertex with the least heuristic total traveling time within the neighboring vertices will be selected to continue the searching process. The same process will be repeated. Finally, when reaching the destination point d, the searching process will

Algorithm 2 Modified A* algorithm for the routing planing

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, s, d, t_s, h_d.
Output: Route selection and estimated traveling time.
1. For vertex $v \in \{\mathcal{N}/v_s\}$, $g_v = \inf, l_v = \inf$; for vertex s, $g_s = t_s, l_s = g_s + h_d(s)$.
2. Let Q be a priority queue containing pair (v, l_v), $\forall v \in \mathcal{V}$, ordered by l_v in an ascending order.
3. Let P be a empty dictionary.
if Q is not an empty queue **then**
  Dequeue Q and obtain the pair $(v', l_{v'})$ and $g_{v'}$
  if $v' = v_d$ **then**
    return *OptimalRouteCalculation*(P, v') and $g_{v'}$; break.
  else
    for Adjacent vertex v connecting to v' with a road sub-segment **do**
      if $g_v \geq g_{v'} + w_{(v',v)}(g_{v'})$ **then**
        (a) $g_v = g_{v'} + w_{(v',v)}(g_{v'})$
        (b) $l_v = g_v + h_d(v)$
        (c) $P[v] = v'$
    end
    Update the queue Q.
else
  break

Algorithm 3 *OptimalRouteCalculation*(P, v')

Input: (P, v')
Output: Route R
1. $\bar{v} = v'$, $R = \emptyset$
if P is not empty **then**
    (a) Add $\bar{v}$ into the R.
    (b) $\bar{v} = P(\bar{v})$
else
    break.
2. Reverse the order of R, and **return** R.

stop and return the optimal route selection and its traveling time estimation. In contrast to the conventional Dijkstra's algorithm, the modified A* algorithm considers the heuristic distance $h_d(v)$ that approximately measures the traveling time from vertex v to the destination d. Such heuristic distance can restrict the modified A* algorithm to search the nearby area around the destination point d, reducing the searching time. Moreover, different from the traditional A* searching algorithm mainly studied in the static graph [11], the modified A* algorithm explicitly considers the time-varying traffic speed and traveling time in the time-dependent graph.

C. The Interactions between the Traffic Prediction and Route Planning

As we see in Section III-B, the multi-horizon traffic speed prediction from using the multi-task FL framework plays a significant role in modeling the time-dependent graph and solving the route planning. In fact, if it is designed properly, the interaction mechanism between the traffic prediction and route planning will in turn improve the traffic prediction accuracy. In particular, since the traffic station can only measure the traffic speed at a specific point whereas the road sub-segments can be as long as few miles, the stations cannot immediately detect the abrupt situations, like traffic accidents. Hence, the traffic speed estimation will not be updated in time, leading to inaccurate traveling time estimations and wrong route selections for vehicles. However, the vehicles can easily identify such abrupt situations by comparing the actual traffic speed and their received estimation at the current locations. Based on such fact, we design the interaction mechanism between the traffic prediction and route planning to guarantee a promising prediction accuracy and route planning solution under abrupt traffic situations in Algorithm 4.

IV. SIMULATIONS

A. Simulation Data

To show the performance of the proposed multi-task FL framework and the routing planning solution, we use the real dataset collected by PeMS[1]. PeMS provides a consolidated database of traffic data collected by traffic stations placed on the state highways across California. In the simulation, we consider the traffic data collected with time interval $\Delta t = 5$ mins for two areas, i.e., San Jose area and Los Angeles area. For San Jose area, we choose 44 traffic stations where each station measures the traffic movement for the highway sub-segments. Similarly, we consider 573 stations for the Los Angeles area. From PeMS, we use the traffic data of the first week at January 2017 as the training data and the following three days (one weekend-day and two weekdays) as the test data. To implement the multi-task FL framework, the training data is divided according to the collected time (i.e., weekday or weekend) in the divisive hierarchical clustering. Without losing of generality, we use two layers of LSTM as the learning model for both data clusters where each layer has 64 neurons. We use Tensorflow to train the learning model with learning rate $\eta = 0.001$, mean squared error as the loss function, and adam as the optimizer.

[1]http://pems.dot.ca.gov/

Algorithm 4 Interaction mechanism between the traffic prediction and route planning

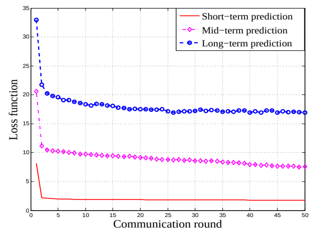
Input: Traffic speed prediction, traveling time estimation, and route selection design
Output: Updated traffic speed prediction, traveling time estimation, and route selection
1. All vehicles measure the difference between their speed with their received speed predictions at current locations. If the difference exceeds a threshold, the vehicles report the unexpected traffic event to the traffic server.
2. The traffic server collects the unexpected traffic event report from each vehicle and counts the number of reports for each road sub-segment. If the number of reports passes a threshold, the traffic server will request the traffic station to update the traffic speed estimation and send the updated information to the vehicles.
3. Vehicles will update traveling time estimation and route selection based on the modified A* algorithm.

Table. I. Traffic speed prediction performance comparison of the proposed multi-task FL framework with baseline schemes.

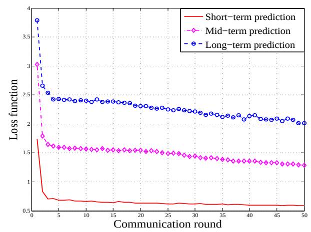
Performance criteria	AMSE	ARMSE	AMAE	AMAPE
Multi-task FL, short-term	**1.537**	**1.186**	**0.737**	**1.431%**
Single-task FL, short-term	2.114	1.401	0.894	1.771%
Traditional STL, short-term	47.872	6.438	4.015	10.855%
Multi-task FL, mid-term	**8.179**	**2.695**	**1.378**	**2.945%**
Single-task FL, mid-term	12.023	3.307	1.827	3.897%
Traditional STL, mid-term	66.700	7.632	5.079	13.108%
Multi-task FL, long-term	**19.075**	**4.118**	**2.077**	**4.704%**
Single-task FL, long-term	29.239	5.144	2.951	6.631%
Traditional STL, long-term	73.534	8.016	5.252	13.695%

B. Multi-horizon Speed Prediction

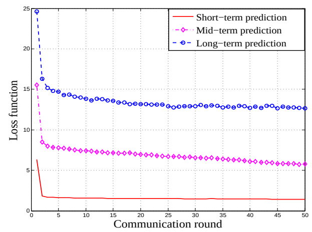
In Table I, we show the the speed prediction accuracy for three maximum prediction time horizons, i.e., the short term ($h = 1$, i.e., 5 mins), mid-term ($h = 6$, i.e., 30 mins), and long-term ($h = 12$, i.e., 60 mins) when using the proposed multi-task FL, single-task FL, and traditional STL scheme for San Jose traffic data. In particular, the single-task FL scheme uses a single-task learning model (2 layers of LSTM) to predict the weekday and weekend traffic where the learning model is trained collaboratively among the stations. In contrast, the traffic stations in the STL scheme train a single-task learning model (2 layers of LSTM) based on their own data. The performance criteria include the average mean squared error



(a) Weekday traffic.



(b) Weekend traffic.



(c) All traffic.

Fig. 3. Loss function change as the multi-task FL proceeds.

(AMSE), average root mean squared error (ARMSE), average mean absolute error (AMAE), and average mean absolute percentage error (AMAPE) as follows:

$$\text{AMSE} = \frac{1}{N}\sum_{n\in\mathcal{N}}\frac{1}{S_n}\sum_{j=1}^{S_n}\sum_{k=1}^{h}(\delta_{n,j,k})^2,$$

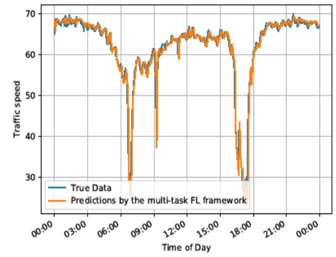
$$\text{ARMSE} = \frac{1}{N}\sum_{n\in\mathcal{N}}\sqrt{\frac{1}{S_n}\sum_{j=1}^{S_n}\sum_{k=1}^{h}(\delta_{n,j,k})^2},$$

$$\text{AMAE} = \frac{1}{N}\sum_{n\in\mathcal{N}}\frac{1}{S_n}\sum_{j=1}^{S_n}\sum_{k=1}^{h}|\delta_{n,j,k}|^2,$$

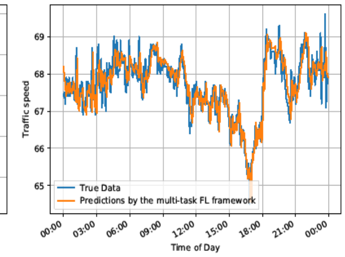
$$\text{AMAPE} = \frac{100\%}{N}\sum_{n\in\mathcal{N}}\frac{1}{S_n}\sum_{j=1}^{S_n}\sum_{k=1}^{h}\left|\frac{\delta_{n,j,k}}{y_{n,j,k}}\right|,$$

where $\delta_{n,j,k} \triangleq y_{n,j,k} - \hat{y}_{n,j,k}$ with $y_{n,j,k}$ and $\hat{y}_{n,j,k}$ respectively denoting the actual and predicted values in j-th output data for station n at k-th prediction time horizon. As shown in Table I, our proposed multi-task FL framework can improve the prediction accuracy compared to other two baselines in all three maximum prediction time horizons. In particular, compared with the traditional STL scheme, its counterpart in the FL can achieve a better traffic speed prediction, highlighting the importance of performing the collaborative training for the traffic prediction. More importantly, we can observe that, our proposed FL multi-task learning framework can obtain more accurate traffic speed predictions than the single-task FL scheme, showing the necessity of using a multi-task learning framework to further enhance the traffic speed prediction performance.

Figures 3 and 4 show more details about the traffic speed prediction performance of the multi-task FL framework. Fig. 3 shows the change of loss function for the weekday, weekend, and all traffic (including both weekday and weekend training data) in the training as the communication round increases under different maximum prediction time horizons. In particular, we can observe that, as the communication round increases, the loss functions for all three types of traffic data decrease, validating the effectiveness of the proposed multi-task FL framework. Also, we can observer that, the loss function corresponding to a smaller maximum prediction horizon is less

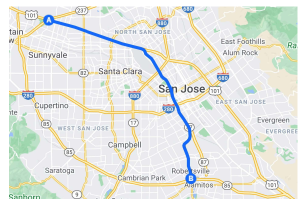


(a) Weekday traffic.

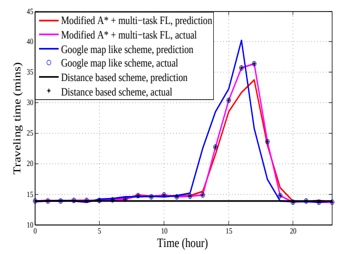


(b) Weekend traffic.

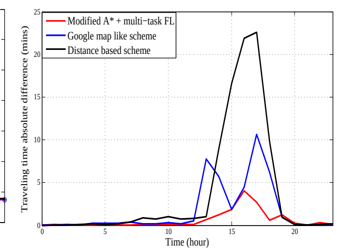
Fig. 4. Traffic predictions and the true values.



(a) Route selection.



(b) Predicted and actual traveling time when using different schemes.



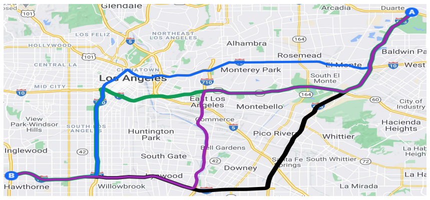
(c) Absolute difference between the actual and predicted traveling time.

Fig. 5. Route planning results and traveling time comparisons of San Jose area.

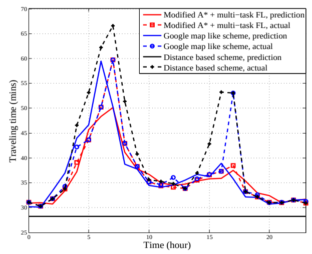
than the one for a larger maximum prediction horizon. This can be explained that, with a larger prediction horizon, we need to predict a longer traffic speed series and the traffic in a further future. Such expansion on the prediction horizon will inevitably degrade the prediction performance. Moreover, in Fig. 4, we present the short-term speed prediction for weekday and weekend traffic of two randomly selected traffic stations in San Jose area when using the proposed multi-FL framework. As we can observed, the prediction results are close to the actual speed in both rush-hour and non-rush-hour traffic in the weekday of Fig. 4(a) and clear traffic in the weekend of Fig. 4(b). It can validate that our multi-task FL framework can effectively capture the diverse traffic patterns in both weekend and weekday.

C. Route Planning

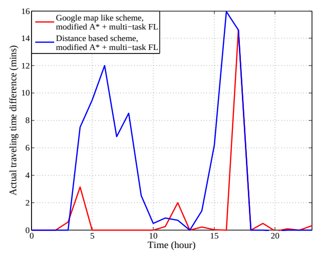
In Figures 5 and 6, we use the traffic speed predictions from the multi-task FL framework in the route planning problem of San Jose area and Los Angeles area, respectively. In particular, we compare the traveling time estimation and route selection of the modified A* algorithm using multi-horizon speed predictions from the multi-task FL framework with two baselines. The first baseline is the Google map like scheme where the route selection and traveling time estimation are determined



(a) Route selections of the modified A* algorithm when using the multi-horizon prediction (Black, green, purple, and blue routes are, respectively, selected at 12 AM, 7 AM, 5 PM, and 7 PM. The black route is also selected by Google map like and distance based schemes at 5 PM.



(b) Predicted and actual traveling time when using different schemes.



(c) Actual traveling time difference between baseline and proposed scheme.

Fig. 6. Route planning results and route selection comparisons of Los Angeles area.

based on the road map and the current traffic information. The second baseline is the distance based scheme which will always choose the route with the shortest distance without considering the traffic information on the road. In Fig. 5, we compare the traveling time estimation performance of all three schemes in San Jose area. To this end, we choose the start point as A and destination point as B in a way that the routes selected by all three schemes are the same throughout the whole day, as shown in Fig. 5(a). In Fig. 5(b), we show the traveling time estimations and actual values for all three schemes in a whole day. As observed from Fig. 5(b), since all three schemes will always choose the same route, their actual traveling time will be the same. However, when using the modified A* algorithm with the multi-horizon traffic speed prediction, the predicted traveling time is much closer to the actual value than other two baselines, verified by the absolute difference in Fig. 5(c). Hence, for the route planning, the multi-horizon traffic prediction obtained by using the proposed multi-task FL can improve the traveling time estimation.

In Fig. 6, we consider the Los Angeles area and select two points A and B so that the route selection for these schemes will vary over time and across different schemes. For example, Fig. 6(a) shows that, when using multi-horizon traffic predictions, the route selected by the modified A* algorithm is the purple route at 5 PM; whereas both Google map like and distance based schemes choose the black route. Moreover, in Fig. 6(b), we show that the traveling time estimations and actual values for all three schemes. To compare the routes selected by all three algorithm, in Fig. 6(c), we show the actual traveling time difference between each baseline scheme and the modified A* algorithm with the multi-horizon speed prediction. As we observed from Fig. 6(c), the traveling time of routes selected by Google map like and distance based schemes always exceed the one for the route chosen by the modified A* algorithm. In other words, when using the multi-horizon traffic prediction from the modified A* algorithm, we can obtain an effective route selection with less traveling time than Google map like and distance based schemes.

V. Conclusions

In this paper, we have developed a novel multi-task FL framework for the traffic prediction and have used the framework in solving the real-world problem, i.e., route planning. In particular, to capture the diverse traffic situations, we have used a divisive hierarchical clustering to partition the collected traffic data at each station into different clusters. Then, to address the insufficiency of local training data, we have integrated FL to collaboratively train each cluster of local data distributed among the stations. We have used the multi-horizon traffic speed prediction from the proposed multi-task FL model to address the route planning that effectively utilizes a modified A* algorithm. Simulation results have verified the prediction accuracy improvement by the proposed multi-task FL framework. Also, the simulation results have demonstrated that, by using the predictions of the proposed multi-task FL model, the effective route selection and accurate traveling time estimation can be achieved. As the future extension, we can implement the proposed multi-task FL framework to predict other traffic conditions, like traffic flow, and tackle other real-world problems, such as best departure time to minimize the traveling time.

References

[1] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[2] B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate short-term traffic flow forecasting using time-series analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 246–254, Jun. 2009.

[3] F. Yang, Z. Yin, H. Liu, and B. Ran, "Online recursive algorithm for short-term traffic prediction," *Transportation Research Record*, vol. 1879, no. 1, pp. 1–8, Jan. 2004.

[4] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[5] D. Kang, Y. Lv, and Y. Chen, "Short-term traffic flow prediction with lstm recurrent neural network," in *Proc. of IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Japan, Oct. 2017.

[6] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou, "LC-RNN: A deep learning model for traffic speed prediction," in *Proc. of International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, Jul. 2018.

[7] P. P. Varaiya, "Freeway performance measurement system (PeMS), PeMS 7.0," Tech. Rep., 2007.

[8] G. Lee, J. Guo, K. J. Kim, P. Orlik, H. Ahn, S. D. Cairano, and W. Saad, "Edge computing for interconnected intersections ininternet of vehicles," in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, USA, Oct.. 2020.

[9] 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.

[10] 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 *Proc. of IEEE International Conference on Communications (ICC)*, Dublin, Ireland, June 2020.

[11] A. V. Goldberg, H. Kaplan, and R. F. Werneck, "Reach for a*: Efficient point-to-point shortest path algorithms," in *Proc. of SIAM the Eighth Workshop on Algorithm Engineering and Experiments (ALENEX)*, Miami, USA, Oct.. 2006.