This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3294898

A Cloud-edge-end Collaboration Framework for Cruising Route Recommendation of Vacant Taxis

Linfeng Liu, *Member, IEEE*, Yaoze Zhou, Jia Xu, *Senior Member, IEEE*

***Abstract*—Taxis can provide convenient and flexible transportation services for citizens. The proper cruising routes should be recommended to vacant taxis, so as to help them to pick up passengers as early as possible, and thus increase their business profits. To this end, we propose a Cloud-edge-end Collaboration Framework for the Cruising Route Recommendation of vacant taxis (CCF-CRR). In CCF-CRR, each vacant taxi trains a local model based on its historical cruising route segments, and the local model parameters of the vacant taxis in the same region are periodically uploaded to an edge server for parameter aggregation. Then, the aggregated model parameters are released by the edge server to vacant taxis for their use. In addition, the future waiting time of passengers is predicted by the edge servers in different regions and is uploaded to the cloud server, and then the cloud server can measure the potential taxi demand in regions and dispatch vacant taxis among regions to achieve the taxi demand-supply equilibrium. Extensive simulations and comparisons demonstrate the superior performance of our proposed CCF-CRR, i.e., with the cloud-edge-end collaboration framework, the business profits of taxis can be significantly increased, and the pick-up distance of taxis can be largely shortened.**

***Index Terms*—vacant taxis; cruising route recommendation; federated learning; taxi demand-supply equilibrium.**

I. Introduction

Nowadays, taxis have become an integral part of the transportation system in modern cities, because they can provide convenient and flexible transportation services for citizens. Typically, taxis cruise along the roads or streets, and they pick up passengers at some locations. Then, they drive to the destinations of the passengers, as shown in Fig. 1.

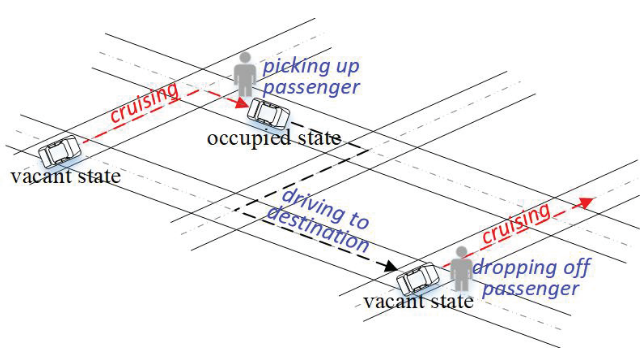


Fig. 1: A taxi picks up and drops off a passenger.

L. Liu, Y. Zhou, and J. Xu are with the School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China, and also with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing 210023, China ({liulf,1221045628,xujia}@njupt.edu.cn).

With regard to taxis and their drivers, the major concern is to increase their business profits [1], [2]. Naturally, a taxi's business profit can be increased by prolonging the occupied durations, i.e., a taxi in the vacant state should find and pick up passengers as early as possible to increase the business profit. Note that the profit increase of taxis also implies that: (i) The energy (gasoline or electricity) consumption of taxis in the vacant states is reduced; (ii) The waiting time of passengers to be served by taxis is shortened, and hence the taxi-taking experience is improved as well. Essentially, the profit maximization of taxis can minimize the useless energy consumption and maximize the efficiency of taxi resources.

Recently, the online taxi-hailing services [3] (such as Uber platform, Didi platform) have become more and more popular. Hence, vacant taxis can pick up passengers by receiving some scheduling orders from the taxi-hailing platforms or when some passengers fall into the visual ranges of the taxi drivers. However, vacant taxis are dispatched by the taxi-hailing platforms to pick up passengers only after being requested by the passengers, and the cruising routes of vacant taxis are not considered by these taxi-hailing platforms. The vacant taxis typically cruise according to the cruising tendencies of their drivers (mainly determined by the drivers' personal cognitions, such as the driving habits, pick-up experience, traffic jam experience, and familiarities with roads), which is not efficient.

To increase the business profits of taxis, the proper cruising routes (along which vacant taxis could encounter with passengers soon) should be recommended to the vacant taxis [4], [5], so as to help them to pick up passengers as early as possible. In essence, the cruising routes of vacant taxis should be determined by the potential taxi demand of passengers [6]. However, the potential taxi demand is difficult to foreknow, since the pick-up locations and drop-off locations are decided by the travel intentions of passengers. The taxi demand is quite different both temporally and spatially, e.g., the taxi demand in the downtown during rush hours is much larger than that in the suburb during dawn hours. In addition, many vacant taxis could concentrate in the same regions, leading to the imbalance of taxi demand-supply among regions. The above facts motivate us to recommend the proper cruising routes for vacant taxis, by exploiting the potential taxi demand from the historical pick-up experience of taxis and balancing the taxi demand-supply among regions.

Benefiting from the widespread application of positioning technique [7], most taxis have been equipped with

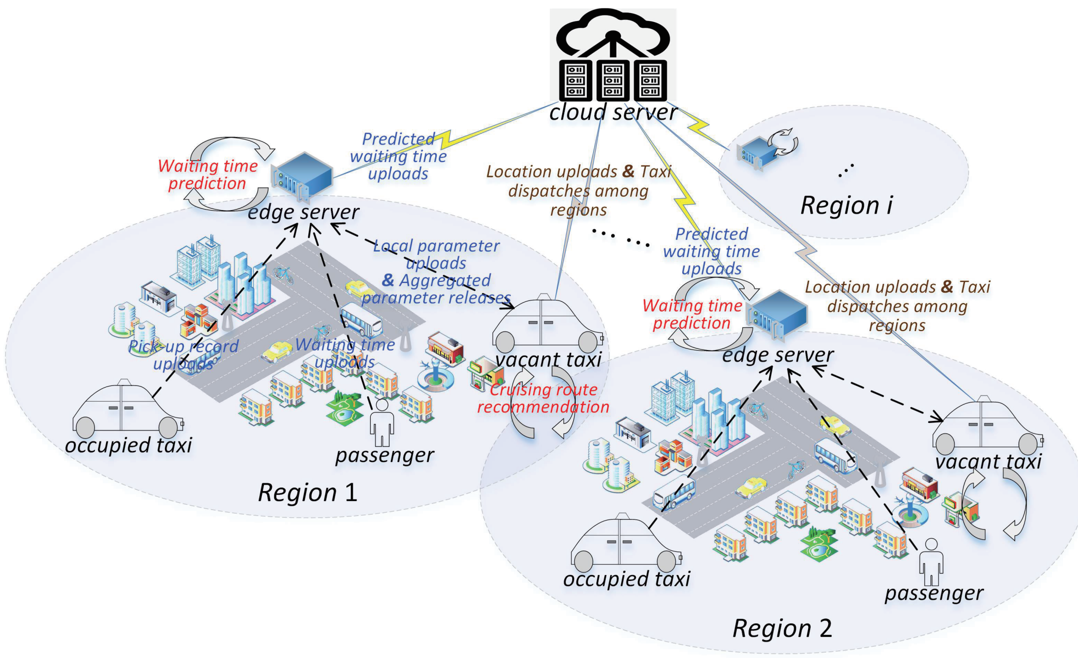


Fig. 2: A cloud-edge-end collaboration framework for cruising route recommendation of vacant taxis.

positioning devices, making their historical routes can be recorded, and these historical routes can be exploited for the cruising route recommendation. Our work focuses on the effective collaborations among different types of network entities. Specifically, the following two issues should be solved with low implementation cost (such as processing delay, computational complexity, and communication overhead) by the collaborations: (i) The proper cruising routes should be recommended to vacant taxis to pick up passengers as early as possible and increase their business profits; (ii) The taxi demand-supply should be balanced among regions by dispatching vacant taxis among regions. To this end, we propose a Cloud-edge-end Collaboration Framework for the Cruising Route Recommendation of vacant taxis (CCF-CRR), as illustrated in Fig. 2, where ”cloud” denotes the cloud server, ”edge” denotes the edge servers, and ”end” denotes the taxis and passengers.

Considering that the pick-up locations of passengers cannot be foreknown by taxis before passengers make the hailing requests, the cruising routes of vacant taxis can be decided by a learning framework which exploits the future taxi demand from the historical cruising routes of taxis. Moreover, a centralized learning scheme is not feasible for the cruising route recommendation of vacant taxis due to the extremely long training time, and a distributed learning scheme is more preferable. As a promising framework in distributed learning, federated learning is adopted in our proposed cloud-edge-end collaboration framework.

In our proposed CCF-CRR, the city is divided into several regions, and each region has an edge server. An edge server and the vacant taxis in the same region adopt a federated learning manner [8], [9], i.e., each vacant taxi trains a local learning model based on the historical cruising route segments before arriving at the pick-up locations, which contain valuable information about the potential pick-up locations of passengers. The local model parameters of vacant taxis in the same region are periodically uploaded to the edge server for parameter aggregation, and then the aggregated model parameters are released by the edge server to vacant taxis to recommend the cruising routes for them. Besides, each edge server predicts the future waiting time of passengers in the region and upload it to the cloud server, and thus the cloud server can measure the potential taxi demand in different regions and dispatch vacant taxis among regions to achieve the taxi demand-supply equilibrium [10].

The merits of our proposed framework are summarized as follows:

- The federated learning manner allows each vacant taxi to train a local learning model in parallel, and hence the training time is largely shortened.
- The edge servers and the cloud server do not require the historical routes of vacant taxis, and hence the privacy of vacant taxis is protected.
- The local training on vacant taxis differentiates the historical route segments during different time intervals and/or in different regions, and thus the spatio-temporal variations in future taxi demand can be embodied in the cruising routes which are recommended to vacant taxis.
- The potential taxi demand in each region is measured by the future waiting time of passengers, and is predicted by the edge server according to the pick-up records of occupied taxis and the historical waiting time of passengers. Accordingly, the vacant taxis in different regions can be dispatched by the cloud server to balance the taxi demand-supply among regions. This mechanism can further increase the business

profits of taxis.

The remainder of this paper is organized as follows: Section II briefly surveys some existing related studies. Section III provides a system model and formulates the problem of cruising route recommendation. Section IV provides the details of CCF-CRR. Section V covers some analyses on CCF-CRR, including the complexity, convergence, expected pick-up distance, and expected profit of a taxi. Simulation results for performance evaluation of CCF-CRR are reported in Section VI. Finally, Section VII concludes the paper.

II. Related Work

A. Route Recommendation for Taxis

Some relevant research has been conducted on the route recommendation of taxis, such as [11], where a solution is proposed to improve the seeking strategies and optimize the global transportation situation by mining the movements of taxis. The weights of road segments are dynamically adjusted based on the number of vacant taxis passing through these road segments, thus avoiding the conflict between local optimization and global optimization. Considering the complexity of urban traffic scenario and the uncertainty of traffic events, the objectives of route recommendation for vacant taxis could be much different, such as the routes with the shortest cruising distance, the most-profitable routes, or the most-familiar driving routes. For example, a route recommendation engine with the aim of Minimizing Distance through Monte carlo tree search (MDM) is developed in [12]. MDM employs a continuous learning platform where the underlying model for predicting the future passenger requests is dynamically updated.

Likewise, in [13], the concept of urban traffic Coulomb's law is applied to model the relationship between taxis and passengers, and then a route recommendation scheme is proposed. By calculating the traffic attraction and traffic force, the cruising taxis can be routed to the optimal road segments to pick up the desired passengers. Besides, [14] designs a fairness-aware vehicle displacement system for electric taxis, and the overall profit efficiency and the profit fairness of electric taxi fleets are improved through jointly considering the taxi demand of passengers and the charging demand of taxis.

We can draw some valuable insights from the above works for exploring the cruising route recommendation of vacant taxis. However, the issue of the taxi demand-supply equilibrium among regions is vital to the recommendation of cruising routes, and it has not been carefully investigated in the above works.

B. Learning Methods in Route Recommendation

The historical information regarding taxis can be exploited for the cruising route recommendation by some learning methods, and recent years have witnessed the great advantages of some learning methods in solving the problem of route recommendation. For example, in [4], a real-time route recommendation system is provided for the drivers of electric taxis. The travel knowledge (including the probability of picking up passengers and the distribution of passengers' destinations) can be learned from the GPS trajectories. In [15], a deep reinforcement learning based approach is proposed to dispatch vacant taxis to the areas with high taxi demand. While collectively optimizing the fleet objectives, this approach distributes the inference at each taxi. The passengers need to upload their information to the platform for a centralized learning. Nevertheless, a centralized learning method is not suitable for the route recommendation of vacant taxis, due to the extremely large computational complexity and communication overhead.

The privacy protection [16] is another vital issue for taxis. Federated learning is a privacy-preserving distributed learning method, and has been applied in some existing works. In [17], a Semi-Supervised Federated Learning (SSFL) framework is formulated to identify the travel model without the raw trajectories of users. Specifically, SSFL adopts a grouping-based aggregation scheme and a data flipping augmentation scheme. Moreover, [18] proposes a trajectory prediction model utilizing federated learning to make full use of the distributed multi-source datasets. A relation-sequence-aware search strategy is provided to realize the automatic design of the trajectory prediction model. In [19], a bidirectional trajectory prediction model called BTPM is proposed to model the mobility behaviors of users, and a data pre-processing phase with short execution time is introduced. [20] gives the DeepFM-based taxi pick-up area recommendation, which is employed to mine the combined relationship between the grid features. Thereby, the most suitable grid area can be recommended to taxi drivers. Qu $et\ al.$ present a profitable taxi route recommendation method named ASER, which predicts the pick-up probability and the capacity of locations by Kalman filtering method [21]. In [22], a Spatio-Temporal Digraph Convolutional Network (STDCN) model is proposed, and the pick-up locations and drop-off locations are modeled into a directed spatio-temporal graph as the model input. STDCN performs well in the dynamic spatio-temporal feature extraction.

To solve the route recommendation problem by a learning method, two issues must be considered: (i) Route recommendation should be promptly made for vacant taxis. To shorten the training time and reduce the computational complexity, vacant taxis should train their local models distributedly and parallelly. (ii) The taxi drivers are typically not willing to share their private information (such as historical routes, occupied durations, and earned profits), and therefore the private information of each taxi should not be shared with others (other taxis, edge servers, and cloud server).

C. Motivation of our Work

In our proposed cloud-edge-end collaboration framework, the federated learning is executed between the vacant taxis and their edge servers to recommend the cruising routes which are with high taxi demand. The edge servers predict the future waiting time of passengers which helps the cloud server to dispatch the vacant taxis among regions.

Essentially, the cloud-edge-end collaboration framework properly offloads the recommendation tasks to taxis, edge servers, and cloud server, thus reducing the computational complexity and processing delay significantly. Besides, the amount of communications in this framework is small, through reducing the unnecessary data exchanges (e.g., the potential taxi demand in regions is predicted by exploiting the historical waiting time of passengers rather than the historical routes of taxis). Hence, the communication overhead is reduced, while the privacy of taxis can be protected.

III. System Model and Problem Formulation

We first describe the problem of cruising route recommendation of vacant taxis. TABLE I shows the list of main notations.

TABLE I: Main notations

Parameter	Description
$\mathcal{R}$	Set of regions
$\mathcal{V}$	Set of taxis
R_i	The i-th region
E_i	Edge server in the i-th region
N_i	Number of taxis in the i-th region
$Prf(v_k)$	Business profit of taxi v_k
$Income(v_k, t')$	Business income of taxi v_k paid by passengers at the t'-th time slot
$Cost(v_k, t')$	Cost of v_k spent on travelling at the t'-th time slot
$D_p(v_k)$	Pick-up distance of taxi v_k
l_m	The m-th trajectory point in a route segment
$(\bar{l}_{m+1}, \cdots, \bar{l}_{m+n})$	A recommended cruising route
$\omega_i(v_k)$	Local model parameters uploaded by taxi v_k in the i-th region
$\widetilde{\omega}_i$	Aggregated model parameters in the i-th region
$W_i^{(t)}$	Actual average waiting time of passengers in the i-th region
$\overline{W}_i^{(t)}$	Predicted average waiting time of passengers in the i-th region
$\overline{\mathbf{W}}_{RIS}^{(t)}$	Set of predicted average waiting time of passengers in the regions with insufficient taxi supply
$\overline{\mathbf{W}}_{RSS}^{(t)}$	Set of predicted average waiting time of passengers in the regions with sufficient taxi supply

Time is divided into discrete time slots with an equal length of ω_s, and some relevant definitions are given as follows:

A. Edge Servers and Cloud Server

The city is divided into I regions, and the division can be obtained by clustering the historical pick-up locations of passengers. We provide an example in Fig. 3, which is based on a dataset comprised of the trajectories and occupied records of taxis [23] in Shuangliu district, Chengdu city. The set of regions is denoted by $\mathcal{R}$:

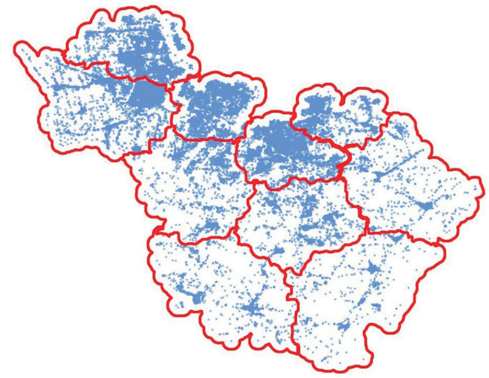


Fig. 3: Division of regions in Shuangliu district, Chengdu city.

$$\mathcal{R} = \{R_1, \cdots, R_i, \cdots, R_I\}, \tag{1}$$

where R_i denotes the i-th region. Each region has an edge server, and the edge server deployed in R_i is denoted by E_i. In our proposed CCF-CRR, E_i is responsible for aggregating the local model parameters uploaded by the vacant taxis in R_i and predicting the future waiting time of passengers in R_i.

There is a cloud server deployed in the city. The cloud server receives the predicted waiting time of passengers from edge servers and dispatches vacant taxis among regions to balance the taxi demand-supply.

B. Taxis and Local Training Datasets

The set of N taxis is denoted by $\mathcal{V}$, and the number of taxis in the region R_i is denoted by N_i, and there is $\sum_{i=1}^{I} N_i = N$. At the t-th time slot, the business profit of a taxi v_k is denoted by $Prf(v_k)$ which is calculated as the sum of net incomes of v_k:

$$Prf(v_k) = \sum_{t'=1}^{t} Income(v_k, t') - \sum_{t'=1}^{t} Cost(v_k, t'), \tag{2}$$

where $Income(v_k, t')$ denotes the business income paid by passengers at the t'-th time slot. $Cost(v_k, t')$ denotes the cost of v_k which is spent on travelling (such as the electricity consumption or gasoline consumption) at the t'-th time slot.

The local training dataset of each taxi v_k is comprised of the historical cruising route segments of v_k. Note that each cruising route segment of a taxi exhibits a progress from seeking passengers to picking up passengers, and the cruising route segments contain rich information regarding the potential pick-up locations of passengers.

C. Pick-up Distance

When a passenger is hailing taxis. The taxis falling into the hailing range S_H of the passenger can receive the hailing request, and the nearest vacant taxi will be selected to pick up this passenger. The travel distance from the location of the selected vacant taxi to the location of the passenger is referred to as the *pick-up distance*, as shown in Fig. 4.

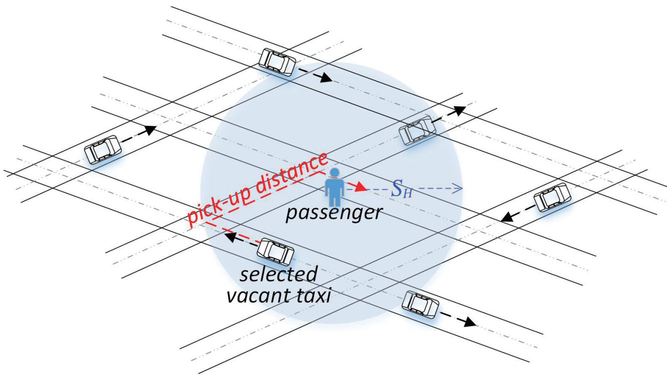


Fig. 4: Pick-up distance.

The shorter pick-up distance of taxis indicates the shorter waiting time of passengers and less non-profitable energy consumption of taxis, and thus the pick-up distance of taxis can also measure the quality of the cruising routes recommended to vacant taxis.

D. Objective Functions

To increase the business profits of taxis and shorten the pick-up distance of taxis, proper cruising routes should be recommended to vacant taxis, and vacant taxis should be dispatched among regions to achieve the taxi demand-supply equilibrium. Therefore, the problem objectives are formally presented as follows:

$$\begin{cases} \max \sum_{v_k \in \mathcal{V}} Prf(v_k), \\ \min \sum_{v_k \in \mathcal{V}} D_p(v_k), \end{cases} \tag{3}$$

where $\max \sum_{v_k \in \mathcal{V}} Prf(v_k)$ indicates that the sum of business profits of all taxis needs to be maximized, i.e., the recommended cruising routes should help vacant taxis to increase their business profits as much as possible.

$\sum_{v_k \in \mathcal{V}} D_p(v_k)$ denotes the sum of pick-up distance of all taxis, and $\min \sum_{v_k \in \mathcal{V}} D_p(v_k)$ indicates that the recommended cruising routes should make vacant taxis move close to the potential passengers (the areas with high taxi demand).

The objective of the maximization of profits enables taxis to spend as much time as possible on carrying passengers, also implying that the pick-up distance of taxis should be minimized. These two objectives can reduce the useless energy consumption of taxis and promote the efficiency of taxi resources.

IV. Cloud-Edge-End Collaboration Framework for Cruising Route Recommendation

As introduced in Section I, the cloud server (”cloud”) is responsible for dispatching the vacant taxis among regions according to the future waiting time of passengers which is predicted and uploaded by the edge servers (”edge”). In each region, for predicting the future waiting time of passengers, the occupied taxis (”end”) upload the pick-up records and the passengers (”end”) upload the historical waiting time to the edge servers; The vacant taxis (”end”) train local models for cruising route recommendation, and the edge servers aggregate the local model parameters for vacant taxis in the same regions.

The objectives of the cruising route recommendation of vacant taxis are to increase the business profits of taxis and shorten the pick-up distance of taxis. In essence, the vacant taxis should pick up the passengers as early as possible and avoid the pick-up conflicts as much as possible. To achieve these objectives, in our proposed CCF-CRR there are some collaborations and competitions among taxis (and passengers). Some vacant taxis could intend to pick up the same passengers, i.e., the vacant taxis could compete for the same passengers. Moreover, the vacant taxis (in the same region) train their local models collaboratively, and the occupied taxis (in the same region) upload the pick-up records and the passengers upload the historical waiting time to the edge server for measuring the potential taxi demand collaboratively.

A. Local Learning on Vacant Taxis

To characterize the historical cruising route segments of each vacant taxi, a stacked Long Short-Term Memory (stacked-LSTM) [24] model is taken as the encoder. Stacked-LSTM has a preferable generalization ability, and it is suitable for processing a long-term sequence. Compared with the single layer LSTM, the stacked-LSTM performs better in learning the trajectory points [25]. Besides, the LSTM-based decoder receives the characterized route segments from encoder and generates the recommended cruising routes for vacant taxis.

For training the local model on each vacant taxi, the $m+n$ trajectory points (historical cruising route segment) before the pick-up location in each historical cruising route are selected. The spatial-temporal vectors $((l_1, time, dow), \cdots, (l_m, time, dow))$ are input into the stacked-LSTM model, as illustrated in Fig. 5, where l_m denotes the m-th trajectory point, $time$ and dow denote the time and the day of week when recording the trajectory point. After obtaining the output of the encoder h_m^2, we train the decoder with the last n trajectory points $(l_{m+1}, l_{m+2}, \cdots, l_{m+n})$.

In Fig. 5, the symbols $<Start>$ and $<End>$ denote the start label and the end label of decoder, respectively. Each vacant taxi independently trains the local model with its historical cruising route segments, and then the local model parameters are uploaded to the edge server (in the same region) for parameter aggregation.

By the stacked-LSTM model, the cruising route (n trajectory points) recommended to a vacant taxi is obtained by:

$$\begin{cases} \bar{l}_{m+1} = LSTM_{decoder}\left(h_m^2, <Start>\right), \\ \quad\vdots = \vdots \\ \bar{l}_{m+n} = LSTM_{decoder}\left(h_{m+n-1}, \bar{l}_{m+n-1}\right), \end{cases} \tag{4}$$

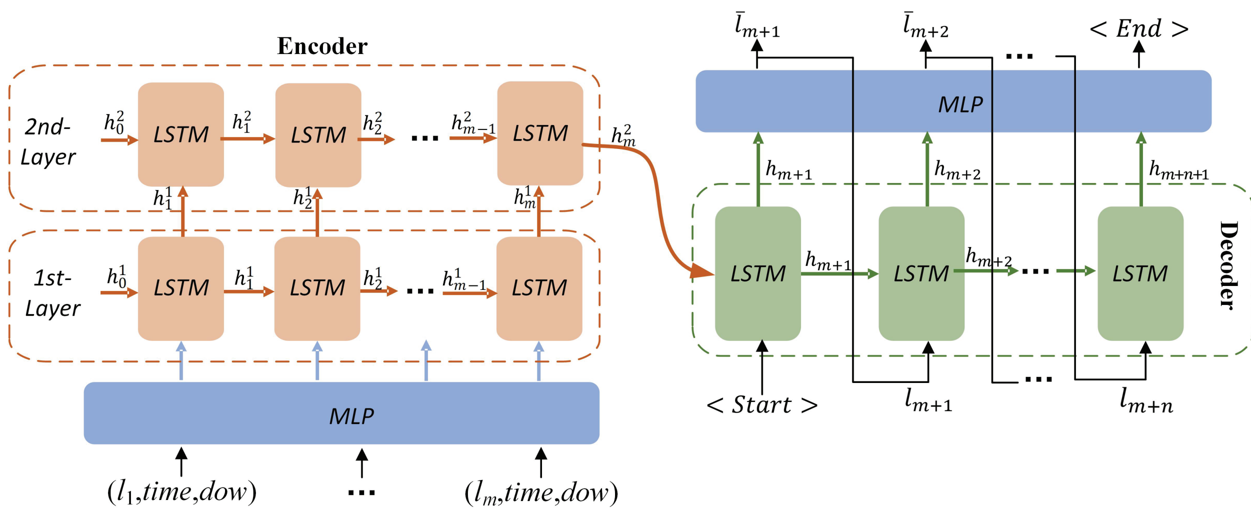


Fig. 5: A stacked-LSTM model for cruising route recommendation.

and the decoder yields the trajectory points until $< End >$ is generated.

B. Cloud-edge-end Collaboration

In our proposed cloud-edge-end collaboration framework, some collaborations are implemented among the network entities: cloud server (cloud), edge servers (edge), taxis (end), and passengers (end):

(i) The vacant taxis (in the same region) train their local models collaboratively, and a federated learning manner is adopted. The local model parameters of vacant taxis are periodically uploaded to the edge server (in the same region) for parameter aggregation, and then the aggregated model parameters are released to vacant taxis for recommending the cruising routes. As shown in Fig. 6, $\omega_i(v_1), \omega_i(v_2), \cdots, \omega_i(v_n)$ denote the local model parameters uploaded by the vacant taxis in the region R_i, and $\widetilde{\omega}_i$ denotes the aggregated model parameters. Besides, in each region, the synchronization of time slots can be guaranteed according to Greenwich time which is easily available to taxis and the edge server.

(ii) The occupied taxis upload the pick-up records and the passengers upload the historical waiting time to the edge server (in the same region), and thus the potential taxi demand in each region can be measured by the edge server. Specifically, the potential taxi demand in a region is measured by an MLP-based model through predicting the future waiting time of passengers, as illustrated in Fig. 7. Typically, the average waiting time of passengers in a region is continuously varied over time. In addition, the duration of a time slot is quite short, and thus the real-time information is hard to quickly obtain for promptly predicting the future average waiting time of passengers. Therefore, we use the information at the last time slot (the $(t-1)$-th time slot) to predict the average waiting time of passengers at the current time slot (the t-th time slot).

TABLE II illustrates the input metrics and their correlations with the output in our proposed MLP-based model.

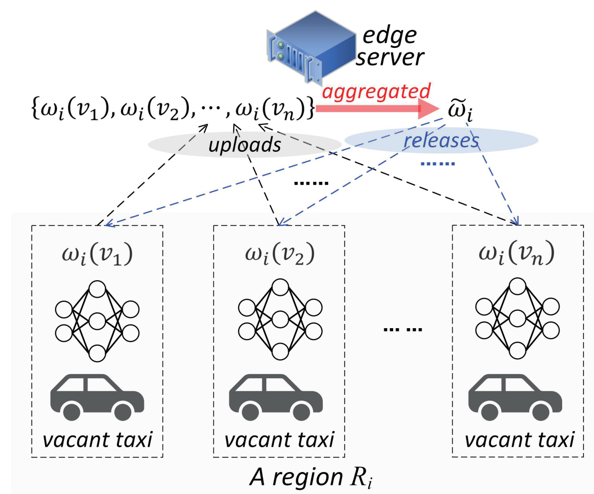


Fig. 6: Federated learning between vacant taxis and edge server in a region.

Suppose in the i-th region R_i (the area of R_i is denoted by A_i) at the $(t-1)$-th time slot, there are $N_{pax}^{(t-1)}$ passengers to be served by taxis, and the proportion of vacant taxis to all taxis is $P_{vac}^{(t-1)}$. $N_{pax}^{(t-1)}$ and $P_{vac}^{(t-1)}$ can be extracted from the pick-up records and vacant records uploaded by taxis. The prediction interval is ∇t. The input of the MLP-based model is expressed as:

$$Input = \nabla t \bigotimes (t-1) \bigotimes A_i \bigotimes N_{pax}^{(t-1)} \bigotimes P_{vac}^{(t-1)}, \quad (5)$$

and the future average waiting time of passengers is predicted by:

$$\overline{W}_i^{(t)} = W_2 \cdot \sigma\left(W_1 \cdot \left(\delta\left(W_0 \cdot Input + b_0\right) + b_1\right) + b_2. \quad (6)$$

The matrices $W_0 \in \mathbb{R}^{8\times|Input|}$, $W_1 \in \mathbb{R}^{4\times 8}$, and $W_2 \in \mathbb{R}^{1\times 4}$, which can be learned by the MLP-based model. b_0, b_1, and b_2 are three biases to be learned. $\sigma(\cdot)$ and $\delta(\cdot)$ denote the Sigmoid activation function and ReLu activation function, respectively.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3294898

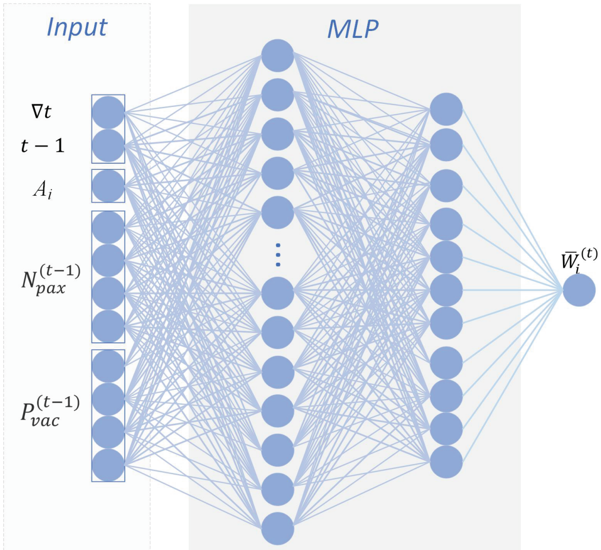


Fig. 7: An MLP-based model for predicting the average waiting time of passengers.

TABLE II: Input metrics and their correlations with output in MLP-based model

Input metric	Description	Correlation with output
∇t	Prediction interval	Difference of average waiting time of passengers in different time intervals
$t-1$	Serial number of last time slot	Temporal correlation in average waiting time of passengers between the $(t-1)$-th time slot and the t-th time slot
A_i	Area of region R_i	For the measurement of distribution density of passengers
$N_{pax}^{(t-1)}$	Number of passengers to be served by taxis at the $(t-1)$-th time slot	Indication of taxi demand
$P_{vac}^{(t-1)}$	Proportion of vacant taxis to all taxis at the $(t-1)$-th time slot	Indication of redundant taxi supply

Moreover, the loss function is defined by:

$$Loss = \arg\min \left| W_i^{(t)} - \overline{W}_i^{(t)} \right|, \quad (7)$$

where $W_i^{(t)}$ denotes the actual average waiting time of passengers which is calculated by the historical waiting time uploaded by passengers.

(iii) The edge servers upload the predicted average waiting time of passengers to the cloud server at an time interval of ∇t. Note that the average waiting time of passengers in a region reflects the regional taxi demand-supply. Then, the cloud server dispatches vacant taxis among regions to achieve the taxi demand-supply equilibrium.

Let $\overline{\boldsymbol{W}}^{(t)} = \left\{ \overline{W}_1^{(t)}, \cdots, \overline{W}_i^{(t)}, \cdots, \overline{W}_I^{(t)} \right\}$ denotes the set of predicted average waiting time of passengers in all regions at the t-th time slot. For a region R_i, if the predicted average waiting time $\overline{W}_i^{(t)}$ is larger than those in adjacent regions, and then R_i is taken as a Region with Insufficient taxi Supply (RIS); Otherwise, R_i is taken as a Region with Sufficient taxi Supply (RSS).

A gradient descent method is adopted by the cloud server to determine the dispatching of vacant taxis among regions. The gradient descent method is depicted in Fig. 8, where $d_{i,j}$ denotes the number of vacant taxis dispatched from R_i to R_j, and $\overline{\boldsymbol{W}}_{RIS}^{(t)}$ and $\overline{\boldsymbol{W}}_{RSS}^{(t)}$ denote the sets of predicted average waiting time of passengers in RISs and in RSSs (at the t-th time slot), respectively. With regard to the i-th row and the j-th column of the matrix, there are $r_i = \sum_{R_j \in \overline{\boldsymbol{W}}_{RIS}^{(t)}} d_{i,j}$ and $c_j = \sum_{R_i \in \overline{\boldsymbol{W}}_{RSS}^{(t)}} d_{i,j}$.

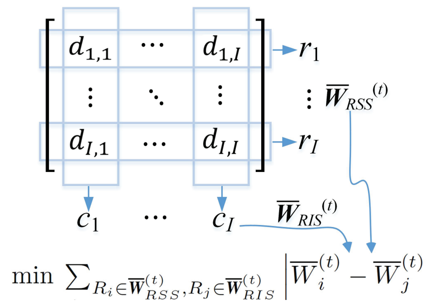


Fig. 8: Gradient descent method.

By the gradient descent method, several rounds of iterations could be executed to minimize the expression $\sum_{R_i \in \overline{\boldsymbol{W}}_{RSS}^{(t)}, R_j \in \overline{\boldsymbol{W}}_{RIS}^{(t)}} \left| \overline{W}_i^{(t)} - \overline{W}_j^{(t)} \right|$. In each round of iteration, both $N_{pax}^{(t-1)}$ and $P_{vac}^{(t-1)}$ are updated.

Specially, the situation that a massive number of taxis move into the same region can be avoided, due to the following two reasons: (i) According to the predicted average waiting time of passengers (the potential taxi demand of passengers), the cloud server dispatches vacant taxis among regions to achieve the taxi demand-supply equilibrium, and this mechanism will not dispatch many vacant taxis to the same regions. (ii) The cruising route (obtained by the stacked-LSTM model) recommended to each vacant taxi is related to the current location of the taxi, which is quite different from the cruising routes recommended to other vacant taxis with different locations, even if these vacant taxis are in the same regions.

V. Theoretical Analysis of CCF-CRR

A. Complexity

With regard to the communication complexity: during each training epoch of the federated learning, the vacant taxis in the same region upload the local model parameters to an edge server for parameter aggregation, and then the edge server releases the aggregated parameters, indicating that the communication complexity reaches $O(N)$ in the worst case. Besides, the historical waiting time and pick-up records are uploaded by passengers and occupied taxis (in the same region) to the edge server for predicting the future waiting time of passengers, and the number of communications is up to $O(N+N_{pax})$, where N_{pax} denotes the total number of passengers. For the cloud server, it receives the predicted average waiting time of passengers from edge servers and dispatches vacant taxis among regions, and thus the number of communications reaches $O(N+I)$. Therefore, the communication complexity of CCF-CRR is of $O(N+I+N_{pax})$.

With regard to the computational complexity: each taxi trains a stacked-LSTM model. The computational complexity of training the stacked-LSTM model is mainly related to the number of parameters, which is approximatively written as $O(N_{layer} \cdot (N_{input} \cdot N_h + N_h^2 + N_h))$, where N_{layer} denotes the number of layers in stacked-LSTM, N_{input} denotes the dimension of input trajectory points, and N_h denotes the dimension of hidden layers. Thus, by training the local models with a federated learning manner, the total amount of computations is expressed as $O(N \cdot N_{layer} \cdot (N_{input} \cdot N_h + N_h^2 + N_h))$. N_h is much larger than N_{input}, and thus the stacked-LSTM model can capture the features from the input trajectory points as more as possible. In addition, the edge servers predict the future waiting time of passengers, and the cloud server determines the dispatching of vacant taxis, which could result in $O(I)$ computations and $O(I^2)$ computations, respectively. Therefore, the computational complexity of CCF-CRR is of $O(N \cdot N_{layer} \cdot N_h^2 + I^2)$.

B. Convergence

The convergence of the federated learning in CCF-CRR can be proven by Lemma 1.

Lemma 1: The federated learning in CCF-CRR can converge to the optimal model parameters.

Proof: Similar to [26], suppose the optimal model parameters $\widetilde{\omega}^*$ are obtained after k_1 local training epochs and k_2 aggregations, and the model parameters after k aggregations are denoted by $\widetilde{\omega}_k$. When the loss function is β-smooth and convex, we have that:

$$\begin{aligned}
&\|\widetilde{\omega}_k - \widetilde{\omega}^*\| \le G_c(k,\eta) \le G_c(k_1 \cdot k_2, \eta) \\
&= h(k_1 \cdot k_2, \Delta, \eta) + \frac{1}{2}(k_2^2 + k_2 - 1) \cdot (k_1+1) \cdot h(k_1, \vartheta, \eta) \\
&= \frac{\Delta}{\beta} \cdot \left[(\eta \cdot \beta + 1)^{k_1 \cdot k_2} - 1\right] - \eta \cdot \beta \cdot k_1 \cdot k_2 + \\
&\frac{1}{2}(k_2^2 + k_2 - 1) \cdot (k_1+1) \cdot \left\{\frac{\vartheta}{\beta} \cdot \left[(\eta \cdot \beta + 1)^{k_1} - 1\right] - \eta \cdot \beta \cdot k_1\right\}.
\end{aligned} \tag{8}$$

Besides, we have $\Delta = \vartheta = 0$, because the trajectory data is typically independent and identically distributed. Thus, the deviation of weights is expressed as:

$$G_c(k_1 \cdot k_2) = h(k_1 \cdot k_2 - (\tau - 1) \cdot k_1, \Delta + \vartheta, \eta) = 0, \tag{9}$$

where τ denotes the aggregation interval.

(9) indicates that the model parameters can converge to the optimal model parameters. □

C. Expected Pick-up Distance and Expected Profit of a Taxi

Suppose at the t-th time slot, the number of vacant taxis $N_{vac}^{(t)}$ is larger than the number of passengers $N_{pax}^{(t)}$. Typically, $N_{pax}^{(t)}$ obeys a Poisson distribution $P(\lambda_t)$ [27], and the trip distance of passengers obeys a Gamma distribution $Ga(\alpha_t, \beta_t)$ [28]. Then, the expected pick-up distance of a taxi is written as:

$$\begin{aligned}
\mathbb{E}(D_p) &= \sum_{n_p=1}^{\infty} \frac{\mathbb{E}(d_p) \cdot P\left(N_{pax}^{(t)} = n_p\right)}{n_p} \\
&= \sum_{n_p=1}^{\infty} \frac{\mathbb{E}(d_p) \cdot \frac{\lambda_t^{n_p}}{n_p!} \cdot \exp\{-\lambda_t\}}{n_p},
\end{aligned} \tag{10}$$

where $\mathbb{E}(d_p)$ denotes the expected distance between a vacant taxi and a passenger (the vacant taxi receives a hailing request from the passenger) when there are n_p passengers. When the number of vacant taxis is large, and the roads are densely distributed in the region (with the area A), $\mathbb{E}(d_p)$ is approximatively expressed as:

$$\mathbb{E}(d_p) \approx \sum_{k=1}^{\frac{S_H}{s}} \frac{k^2 - (k-1)^2}{S_H^2} \cdot k \cdot s, \tag{11}$$

where s denotes the average length of a road segment.

For a taxi, the income is obtained by carrying passengers, and the expected net income of a taxi carrying a passenger from the pick-up location to the drop-off location is written as:

$$\begin{aligned}
r_i \cdot \mathbb{E}(d_{trip}) &= r_i \cdot \int_0^{\infty} x \cdot \frac{\beta_t^{\alpha_t}}{\Gamma(\alpha_t) \cdot x^{\alpha_t} \cdot \exp\{-\beta_t \cdot x\}} \mathrm{d}x \\
&= r_i \cdot \frac{\alpha_t}{\beta_t},
\end{aligned} \tag{12}$$

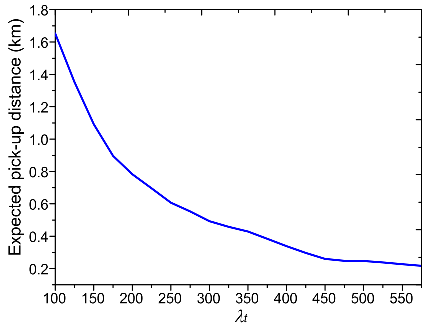
where r_i denotes the net income of a taxi carrying passengers per kilometer.

The cost of a taxi includes two parts: (i) the cost for the taxi cruising along roads; (ii) the cost for the taxi moving to pick up the passenger when it has approved the hailing request. Therefore, the expected profit of a taxi is expressed as:

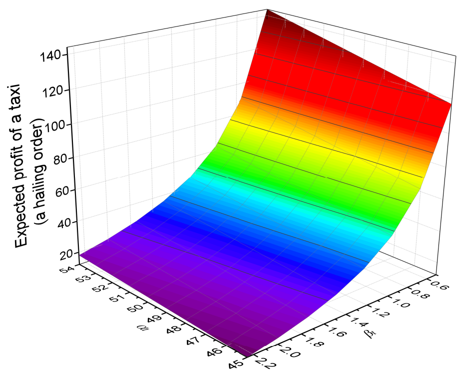
$$\begin{aligned}
\mathbb{E}(Prf) = & r_i \cdot \frac{\alpha_t}{\beta_t} - r_c \cdot \sum_{n_p=0}^{\infty} \mathbb{E}(d_p) \cdot \frac{\lambda_t^{n_p}}{n_p!} \cdot \exp\{-\lambda_t\} \\
& - r_c \cdot \phi \cdot \sum_{n=0}^{\left\lfloor \frac{A}{\phi \cdot S_H} \right\rfloor} \frac{(n+1) \cdot n}{\sum_{n=0}^{\left\lfloor \frac{A}{\phi \cdot S_H} \right\rfloor} (n+1)},
\end{aligned} \tag{13}$$

where r_c denote the travel cost of a taxi (without carrying passengers) per kilometer, and ϕ denotes the travel distance between two adjacent trajectory points. In (13), $\frac{(n+1)}{\sum_{n=0}^{\left\lfloor \frac{A}{\phi \cdot S_H} \right\rfloor} (n+1)}$ denotes the probability of a vacant taxi cruising along n recommended trajectory points until receiving and approving a hailing request (a hailing order).

We provide some numerical results regarding the expected pick-up distance and the expected profit in Fig. 9. The expected pick-up distance is shortened when λ_t is increased (when there are more passengers). The expected profit is increased when α_t becomes large or β_t becomes small (when the trip distance of passengers is longer).



(a) Expected pick-up distance



(b) Expected profit of a taxi (a hailing order, $\lambda_t = 200$)

Fig. 9: Expected pick-up distance and expected profit.

VI. Performance Evaluations

In this section, we provide comprehensive performance evaluations on our proposed CCF-CRR, and we make comparisons with some related training models and related methods. The simulations are conducted on a dataset released by Didi company (https://github.com/BigBrotherChou/trajectories-data). This dataset consists of a collection of trajectory points of taxis and taxi states from October 1, 2017 to October 10, 2017, in Chengdu city, China. In this dataset, the taxi ID is used to indicate the unique identifier of each taxi; *date* represents the timestamps when the trajectory points are recorded; *lng* and *lat* denote the longitudes and latitudes of taxis, respectively; *angle* and *speed* denote the travel directions and speed of taxis, respectively; $load$ denotes the states of taxis, i.e., the taxis are occupied or vacant.

We develop a simulator using Python language. The main parameter settings are provided in TABLE III.

An example of CCF-CRR is given in Fig. 10 with three taxis, where the taxis travel along the cruising routes recommended by CCF-CRR until they pick up passengers. After dropping off the passengers, the taxis travel along the recommended cruising routes again. The recommended cruising routes are denoted by solid lines, and the routes when taxis carrying passengers are denoted by dashed lines.

TABLE III: Simulation parameters

Parameter	Description	Value
N	Number of taxis	200
I	Number of regions	10
S_H	Hailing range of passengers	3 km
ω_s	Duration of each time slot	30 s
N_{layer}	Number of LSTM layers in the encoder of stacked-LSTM model	2
m	Number of input trajectory points	10
n	Number of output trajectory points	3
l_r	Learning rate in MLP-based model	0.01
∇t	Prediction interval of MLP-based model	30 min
Pr	Proportion of vacant taxis uploading parameters for aggregation in each epoch	80%
r_i	Net income of a taxi carrying passengers per kilometer	1.5 CNY/km
r_c	Travel cost of a taxi without carrying passengers per kilometer	0.95 CNY/km
ϕ	Travel distance between two adjacent trajectory points	0.5 km

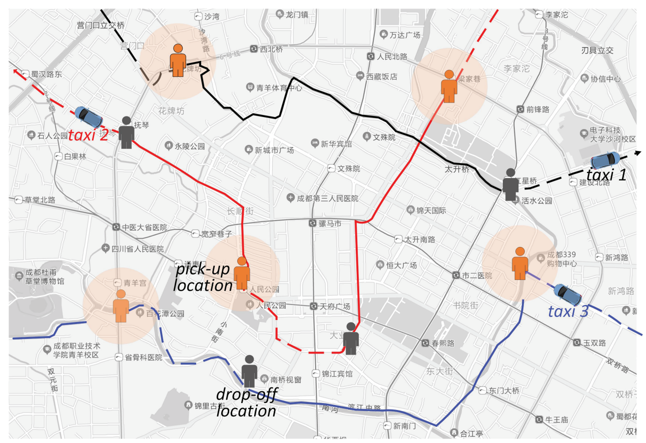


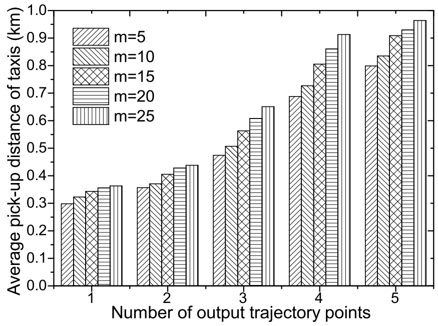
Fig. 10: An example of three taxis' routes.

A. *Comparisons among Different Training Models*

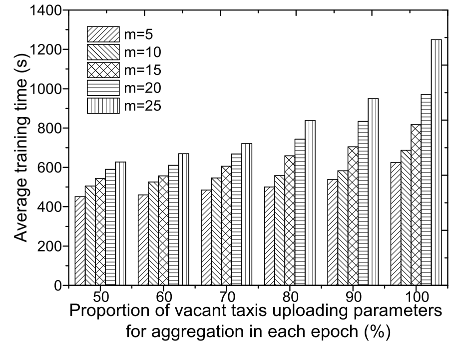
We first use two metrics (average cruising distance and average pick-up distance) to measure the performance of CCF-CRR. The cruising distance of a taxi denotes the travel distance of the taxi in the vacant state, and the pick-up distance of a taxi denotes the travel distance from the location where the taxi receives a hailing request of a passenger to the location where the taxi picks up the passenger. We observe the average cruising distance of taxis and the average pick-up distance of taxis among different training models (stacked-LSTM, stacked-GRU [29], and stacked-RNN [30]), as shown in Fig. 11.

Fig. 11(a) indicates that the average cruising distance of taxis (in an hour) obtained by three training models ascends with the increase of the number of output trajectory points n, and the curve of stacked-LSTM is the lowest among these curves, due to the following reasons: (i) The recommendation results of these training models are worsened when these models output more trajectory points; (ii) Compared with stacked-GRU and stacked-RNN, the stacked-LSTM has a deeper network to fit the input

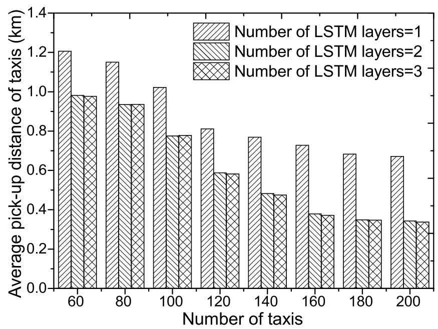
This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3294898



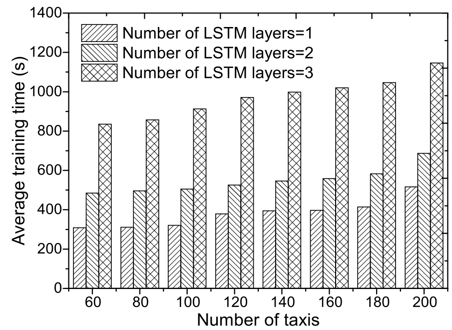
(a) Average pick-up distance of taxis vs. m and n



(b) Average training time vs. m and Pr



(c) Average pick-up distance of taxis vs. N_{layer} and N



(d) Average training time vs. N_{layer} and N

Fig. 12: Average pick-up distance of taxis and average training time.

trajectory points, and has more sophisticated forget gates to solve the problem of vanishing gradient. Thus, the stacked-LSTM can deal with the input trajectory points better than other two models and is adopted in our CCF-CRR for the cruising route recommendation of vacant taxis.

In Fig. 11(b), we compare these models under different values of Pr (the proportion of vacant taxis uploading parameters for aggregation in each epoch). When Pr is varied from 50% to 100%, the average pick-up distance of taxis obtained by three models is shortened, because when more vacant taxis participate in the federated learning, better aggregated model parameters can be obtained and released to vacant taxis for their use.

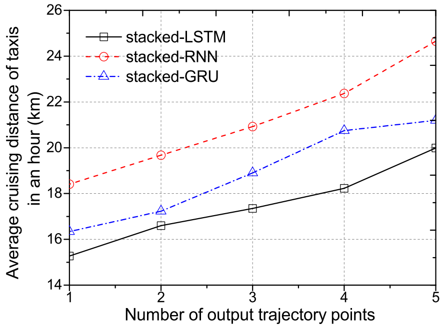
Note that the average cruising distance of taxis is related to the business profits of taxis, since taxis can increase their business profits by shortening their cruising distance. Besides, the average pick-up distance of taxis is taken as an important metric to measure the taxi-taking experience of passengers, i.e., the passengers have better taxi-taking experience when the pick-up distance of taxis is shorter.

B. Average Pick-up Distance of Taxis and Average Training Time

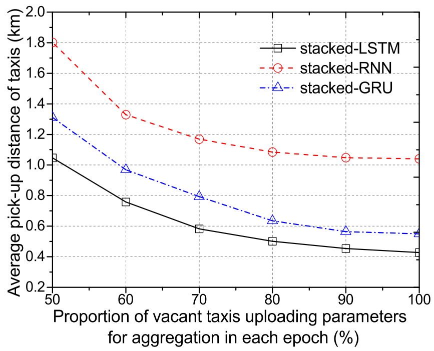
From Fig. 12(a), we obtain two observations as follows: (i) With the increase of the number of input trajectory points m or the number of output trajectory points n, the average pick-up distance of taxis is increased (Fig. 12(a)). With a larger m, more previous trajectory points are input into the stacked-LSTM model, and more noise could be included in the input, which makes the gradient propagated back through the network either decay or grow exponentially, and hence the proper cruising routes cannot be exploited from the too old trajectory points.

Similar to the phenomenon provided in Fig. 11(a), the average pick-up distance of taxis rises up with the increase of the number of output trajectory points. Specially, when $m = 25$ and $n = 5$, we obtain the longest average pick-up distance of taxis which is equal to 0.964 km.

(ii) The training time is comprised of three parts: the time consumed for training the local models on vacant taxis, the communication delay for uploading and releasing the local model parameters, and the time for edge servers aggregating the model parameters. Our workstation for simulations is equipped with i5-10400 CPU, 16G DRAM, and NVIDIA GeForce RTX3060-12G GPU. The average training time is prolonged with the increase of m or Pr (Fig. 12(b)), which is attributed to the facts that: the time consumed for training the local models is increased when more trajectory points are input into the stacked-LSTM model (with a larger m, the structure of the encoder for processing the input trajectory points is much more complex than that of the decoder for outputting the n trajectory points). Besides, the time for each edge server aggregating the model parameters



(a) Average cruising distance of taxis



(b) Average pick-up distance of taxis

Fig. 11: Comparisons among different training models.

is increased when the local model parameters of more vacant taxis should be uploaded to the edge server for parameter aggregation.

The number of LSTM layers in the encoder (N_{layer}) affects the average pick-up distance of taxis and the average training time, as shown in Fig. 12(c) and Fig. 12(d). The shorter average pick-up distance of taxis is yielded by CCF-CRR with a larger N_{layer}, while the average training time is increased due to the dimension increase of the hidden layers in the encoder. Naturally, the average training time is sharply increased with the increase of N_{layer} (Fig. 12(d)).

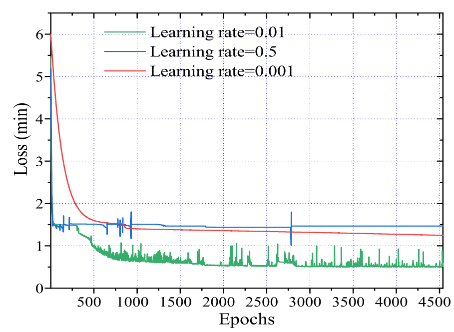
However, when the number of LSTM layers is larger than 2, the average pick-up distance of taxis is decreased slightly with the further increase of N_{layer} (Fig. 12(c)), e.g., the average pick-up distance of taxis with $N_{layer} = 2$ is very close to that with $N_{layer} = 3$. This phenomenon implies that N_{layer} should be properly set to make an appropriate tradeoff between the training effect and the training cost.

With the increase of N, the average pick-up distance of taxis is largely reduced, since the taxis could be nearer to the passengers when the taxis are distributed on roads more densely. Obviously, the average training time is increased when there are more taxis participating in the federated learning.

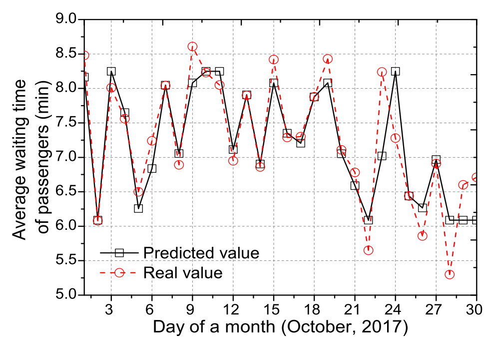
C. Prediction Accuracy of MLP-based Model

In CCF-CRR, an MLP-based model is applied to predict the future waiting time of passengers. To measure the prediction accuracy of the MLP-based model, we observe the loss value, and compare the predicted value with the real value.

In Fig. 13(a), the convergence of the MLP-based model with learning rate 0.5 is faster than others, and that with learning rate 0.001 is more stable than others. The MLP-based model with learning rate 0.01 achieves the smallest loss, implying that the learning rate should be appropriately set for the MLP-based model, because when the learning rate is set too large or too small, the problem of local optimization or convergence failure could be caused. In Fig. 13(b), the predicted waiting time of passengers is close to the real value, which indicates that the MLP-based model has a preferable prediction accuracy.



(a) Loss value



(b) Average waiting time of passengers

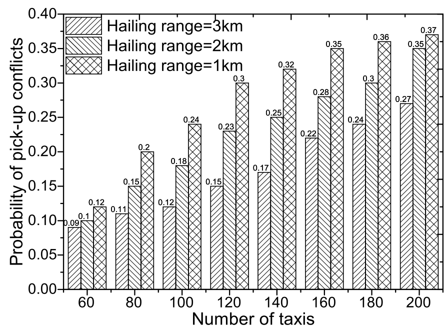
Fig. 13: Prediction accuracy of MLP-based model.

D. Probability of Pick-up Conflicts and Average Pick-up Time of Taxis (with Traffic Congestion)

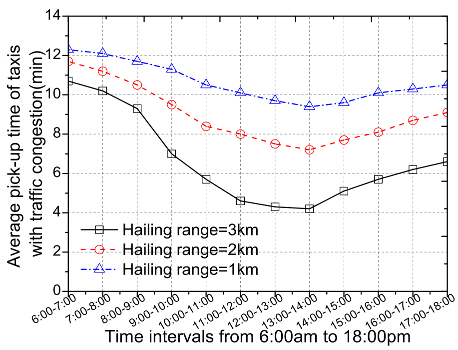
Several vacant taxis could cruise towards the same locations and intend to pick up the same passengers, and hence the late-arriving taxis fail to pick up the passengers, which is referred to as the pick-up conflicts of taxis. The probability of pick-up conflicts of taxis is provided in Fig. 14(a), which indicates that the probability of pick-up

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3294898

conflicts is increased with the increase of the number of taxis and the decrease of the hailing range of passengers. Because when more taxis compete for the passengers, the pick-up conflicts are intensified. When the hailing range of passengers becomes larger, vacant taxis could receive more hailing requests from more passengers, implying that more passengers can be selected by vacant taxis, and thereby the pick-up conflicts are alleviated.



(a) Probability of pick-up conflicts



(b) Average pick-up time of taxis (with traffic congestion)

Fig. 14: Probability of pick-up conflicts and average pick-up time of taxis (with traffic congestion).

The pick-up time of a taxi is calculated as the period from the taxi receives the hailing request to it picks up the passenger. In this simulation, the influence of traffic congestion is considered, and the speed of taxis with traffic congestion refers to [31]. As shown in Fig. 14(b), the average pick-up time of taxis is the smallest during the time interval 13:00pm-14:00pm, due to the following two facts: (i) the taxi demand is high during this time interval; (ii) the traffic congestion is significantly relieved during this time interval. Besides, a larger hailing range of passengers gives rise to shorter average pick-up time of taxis, because in CCF-CRR the vacant taxis with shorter pick-up distance are preferentially assigned to pick up the passengers when there are more vacant taxis falling into the hailing range of passengers (with a larger hailing range).

In the real taxi-hailing scenario, a passenger should be picked up by a vacant taxi with shorter pick-up time rather than that with shorter pick-up distance when the traffic congestion is considered. For this issue, in CCF-CRR we can use the logical pick-up distance instead of the pick-up distance. The logical pick-up distance is obtained by jointly considering the pick-up distance and the traffic congestion on roads.

E. Comparisons among Different Collaborations

CCF-CRR adopts a cloud-edge-end collaboration. To verify the advantages of the cloud-edge-end collaboration, we compare the cloud-edge-end collaboration with other collaborations (cloud-end collaboration, edge-end collaboration, and end-only). In the cloud-end collaboration, the local model parameters of vacant taxis in different regions are uploaded to the cloud server rather than the edge servers, and the cloud server releases the aggregated model parameters to all vacant taxis. In the edge-end collaboration, the local model parameters of vacant taxis are uploaded to edge servers for parameter aggregation, while the dispatching of vacant taxis among regions is not considered. In the end-only manner, vacant taxis train their local stacked-LSTM models without parameter aggregation. TABLE IV illustrates these collaboration manners.

TABLE IV: Different collaboration manners

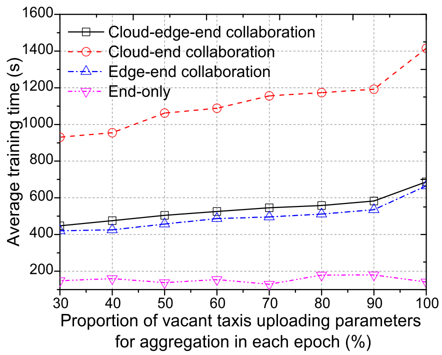
Collaboration manner	Cloud server	Edge server	Parameter aggregation	Dispatching of vacant taxis
Cloud-edge-end collaboration	Yes	Yes	By edge servers	Yes
Cloud-end collaboration	Yes	No	By cloud server	Yes
Edge-end collaboration	No	Yes	By edge servers	No
End-only manner	No	No	No	No

In Fig. 15(a), the end-only manner achieves the shortest average training time, because it does not require the parameter uploads and parameter releases. The curve of the cloud-edge-end collaboration is slightly higher than that of the edge-end collaboration, because in the cloud-edge-end collaboration the edge servers are responsible for predicting the future waiting time of passengers.

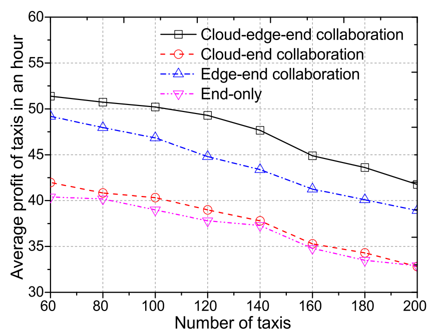
The average training time of the cloud-end collaboration is the longest, and the reason is that the local model parameters of vacant taxis must be uploaded to the cloud server, and large delay is produced due to the asynchronous communications of vacant taxis. Moreover, compared with the trajectory data collected from the whole city, the division of the city allows the vacant taxis to train their local models with local trajectory data which is independent and identically distributed. Thus, the cloud-end collaboration requires more epochs to reach the convergence of the federated learning.

Fig. 15(b) indicates the average profit of taxis obtained by the cloud-edge-end collaboration is evidently larger than others, because the federated learning is carried out to exploit the potential pick-up locations of passengers, and vacant taxis can be dispatched among regions to balance the taxi demand-supply. By these mechanisms, the vacant

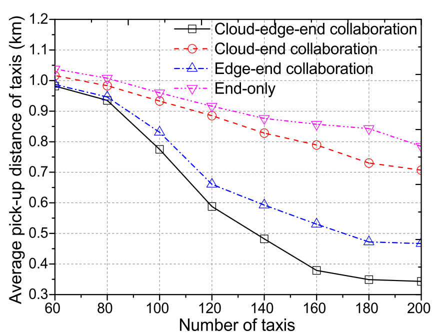
This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3294898



(a) Average training time



(b) Average profit of taxis in an hour



(c) Average pick-up distance of taxis

Fig. 15: Comparisons among different collaborations.

taxis can find and pick up passengers quickly. Accordingly, the average pick-up distance of taxis of the cloud-edge-end collaboration remains the shortest (Fig. 15(c)).

Fig. 16 illustrates the loss value after 453 epochs, which indicates that a better convergence can be obtained by the division of regions and the employment of edge servers.

The above results demonstrate that the division of regions can improve the performance of learning models and shorten the training time. Furthermore, we conduct a simulation to observe the impacts of division granularity (the city is divided into 6, 8, and 10 regions, respectively), as illustrated in Fig. 17. A larger I implies that the city is divided into more regions. With a larger I, the average profit of taxis is increased, and the average waiting time of passengers

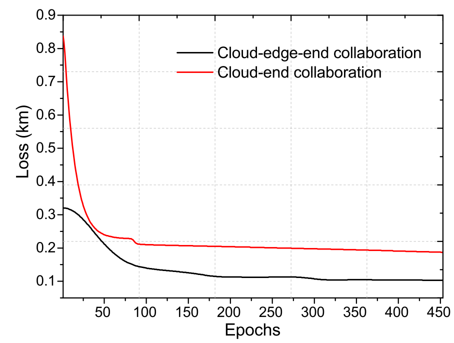
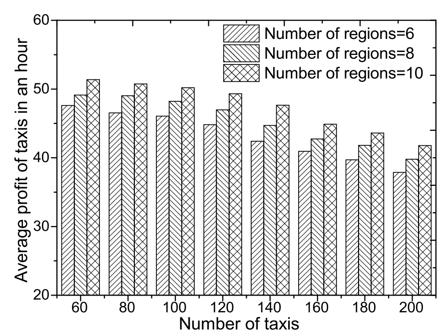
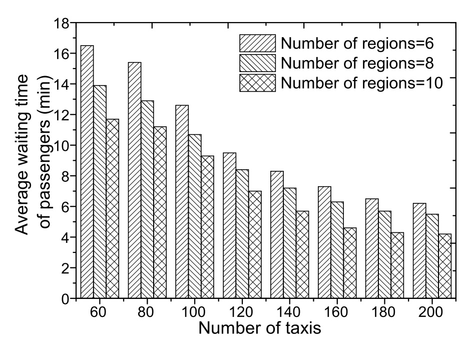


Fig. 16: Loss value of cloud-edge-end collaboration and cloud-end collaboration (for stacked-LSTM model).

is shortened, and this is because the better balance of taxi demand-supply among regions can be achieved when vacant taxis are dispatched among more small-size regions. However, with a larger I, more edge servers should be deployed, and the computation of the gradient descent method becomes more complex.



(a) Average profit of taxis

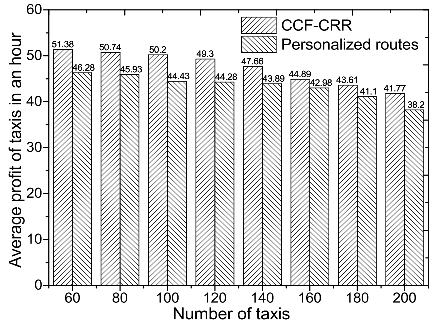


(b) Average waiting time of passengers

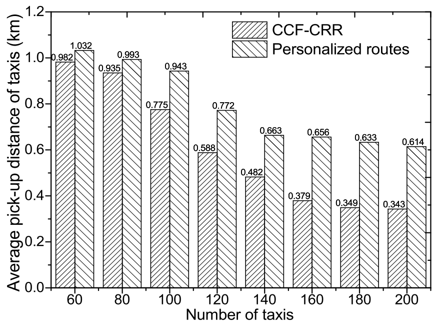
Fig. 17: Impacts of division granularity.

F. Comparisons among Different Methods

Firstly, we compare the performance between CCF-CRR and personalized routes. Each vacant taxi obtains the



(a) Average profit of taxis



(b) Average pick-up distance of taxis

Fig. 18: Comparisons between CCF-CRR and personalized routes.

personalized routes by independently learning its historical cruising routes (rather than the cruising route segments before the pick-up locations), and the parameter aggregation among taxis is not required. The simulation results are provided in Fig. 18, which indicates that the cruising routes recommended by CCF-CRR outperforms the personalized routes in terms of average profit of taxis and average pick-up distance of taxis, because in CCF-CRR the potential taxi demand can be exploited from the historical pick-up experience of taxis and historical waiting time of passengers, and the potential taxi demand is used to recommend the cruising routes for vacant taxis and dispatch vacant taxis among regions.

To further verify the merits of CCF-CRR, we compare CCF-CRR with some related methods, including Random Walk (vacant taxis cruise randomly), BTPM [19], TPR_SI [20], ASER [21], and STDCN [22]. CCF-CRR obtains superior results and outperforms these methods, in terms of average cruising distance of taxis, proportion of vacant taxis, average profit of taxis, and average pick-up distance of taxis.

Compared with BTPM, the stacked-LSTM model in CCF-CRR has stronger ability to learn the historical trajectory data. Besides, both TPR_SI and ASER divide the city into some equal-size grids, ignoring the difference in the spatial distribution of taxi demand. Specially, in TPR_SI, vacant taxis are dispatched among grids based on the prediction of pick-up probability without considering the capacity of grids.

In CCF-CRR, the edge servers predict the future waiting time of passengers in different regions, by which vacant taxis can be dispatched among regions. Thus, CCF-CRR obtains shorter cruising distance of taxis than other methods (Fig. 19(a)), while the proportion of vacant taxis is smaller, i.e., more taxis are occupied by passengers, as shown in Fig. 19(b). Random Walk method always performs the worst among these methods, implying that the random movements of vacant taxis are not reasonable, and the proper cruising routes are essential to vacant taxis.

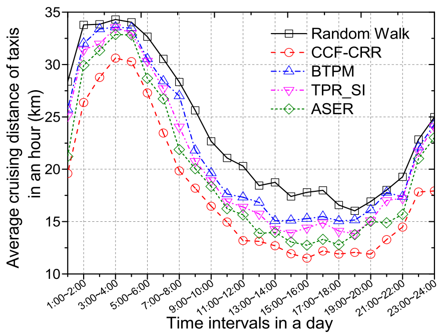
CCF-CRR performs better than STDCN, because in STDCN the traffic data of the city is analyzed and trained by the cloud sever, and then a recommendation model is yielded and released to all edge servers for their use. However, the difference of taxi demand in different regions is not considered.

Similar to the phenomenon in Fig. 19(b), the taxis can prolong their occupied durations by CCF-CRR, and hence the average profit of taxis is larger than others (Fig. 19(c)). When the vacant taxis travel along the cruising routes recommended by CCF-CRR (these cruising routes are generated by exploiting the potential pick-up locations of passengers), they can find and pick-up passengers as early as possible, and therefore the average pick-up distance of taxis obtained by CCF-CRR is shorter than others (Fig. 19(d)).

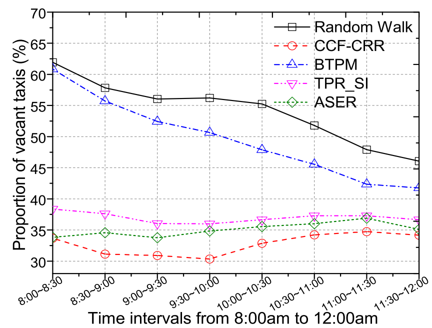
VII. Conclusion

We have studied the problem of cruising route recommendation of vacant taxis, and a Cloud-edge-end Collaboration Framework for the Cruising Route Recommendation of vacant taxis (CCF-CRR) has been introduced. In CCF-CRR, each vacant taxi trains a local model based on its historical cruising route segments, and the local model parameters of vacant taxis in the same region are periodically uploaded to the edge server for parameter aggregation. Then, the aggregated model parameters are released to vacant taxis to recommend the cruising routes for them. The potential taxi demand in regions is measured by the future waiting time of passengers. The edge servers are responsible for predicting the future waiting time of passengers which is then uploaded to the cloud server, and thus the cloud server can dispatch vacant taxis among regions to achieve the taxi demand-supply equilibrium. CCF-CRR can increase the business profits of taxis and shorten the pick-up distance of taxis effectively. In CCF-CRR, the private information of each taxi will not be revealed to edge servers, cloud server, and other taxis.

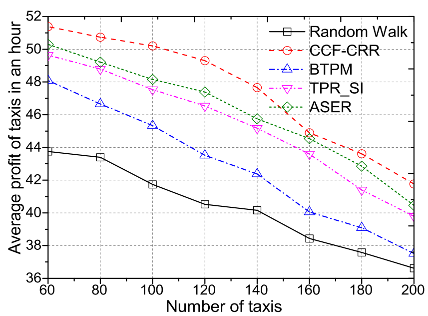
In CCF-CRR, the pick-up conflicts of taxis can be avoided as much as possible, due to the following two reasons: (i) The cruising routes recommended to vacant taxis are related to the current locations of vacant taxis, and thus the recommended cruising routes are personalized; (ii) The taxi demand of passengers is quite different both



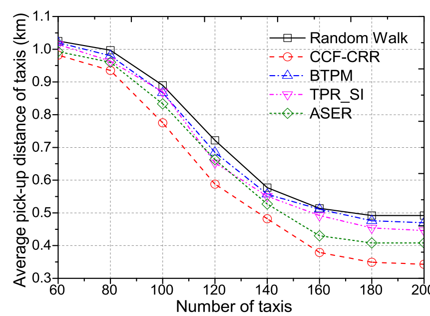
(a) Average cruising distance of taxis



(b) Proportion of vacant taxis



(c) Average profit of taxis



(d) Average pick-up distance of taxis

Fig. 19: Comparisons among different methods.

temporally and spatially, and the recommended cruising routes typically guide vacant taxis to the nearby areas with high taxi demand. Besides, the personal cognitions of taxi drivers (such as the driving habits, pick-up experiences, traffic jam experiences, and familiarities with roads) can be specially investigated to further alleviate the pick-up conflicts of taxis. For the cruising route recommendation, some factors are not considered in this work, such as the impacts of weather, seasons, and festivals. If multi-modal information is taken as the input, a multi-modal learning model will be suitable for learning the multi-modal information and generating the recommended cruising routes. If some real-time information (the real-time number of passengers and the real-time number of vacant taxis in a region) is available, and then the prediction results of the MLP-based model could be more accurate.

In the real taxi-hailing scenario, the waiting time of passengers could be affected by some traffic events such as the traffic congestion. With regard to passengers, the long waiting time implies the taxi supply in the regions is insufficient, no matter the long waiting time is caused by the insufficient vacant taxis in the regions or the traffic congestion. However, with regard to vacant taxis, they could not willing to be dispatched to the regions where the roads have heavy traffic congestion, i.e., some vacant taxis could not agree with the dispatching of the cloud server. A pricing mechanism can be adopted to incent these vacant taxis to accept the dispatching of the cloud server. Moreover, if vacant taxis can obtain the pick-up locations and destinations of passengers in advance, then they are prone to move towards and pick up the passengers with longer trip distance (i.e., the destinations are farther away from the pick-up locations). In such case, the taxi-taking fairness of passengers becomes a vital issue. The above issues will be investigated in our future work.

Acknowledgments

This research is supported by National Natural Science Foundation of China under Grant Nos. 62272237, 61872191, 61872193.

References

[1] S. Ji, Z. Wang, T. Li, *et al.*, "Spatio-temporal feature fusion for dynamic taxi route recommendation via deep reinforcement learning," *Knowledge-Based Systems*, vol. 205, 2020.

[2] H. P. Hsieh, F. Lin, N. Y. Chen, *et al.*, "A decision framework to recommend cruising locations for taxi drivers under the constraint of booking information," *ACM Transactions on Management Information Systems*, vol. 13, no. 3, pp. 1–30, 2022.

[3] E. Liang, K. Wen, W. H. K. Lam, *et al.*, "An integrated reinforcement learning and centralized programming approach for online taxi dispatching," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4742–4756, 2022.

[4] W. Tu, K. Mai, Y. Zhang, *et al.*, "Real-time route recommendations for e-taxies leveraging GPS trajectories," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3133–3142, 2021.

[5] T. Wang, Z. Shen, Y. Cao, *et al.*, "Taxi-cruising recommendation via real-time information and historical trajectory data," *IEEE Transactions on Intelligent Transportation Systems*, DOI: 10.1109/TITS.2021.3093207, 2021.

[6] C. Zhang, F. Zhu, X. Wang, *et al.*, "Taxi demand prediction using parallel multi-task learning model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 794–803, 2022.

[7] J.-H. Lim, K. H. Choi, L. Kim, *et al.*, "Land vehicle positioning in urban area by integrated GPS/BeiDou/OBD-II/MEMS IMU," *IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Singapore, 2016.

[8] Q. Yang, Y. Liu, T. Chen, *et al.*, "Federated machine learning: concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, Article 12, 2019.

[9] S. Wang, T. Tuor, T. Salonidis, *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[10] Z. Liu, J. Li, and K. Wu, "Context-aware taxi dispatching at city-scale using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1996–2009, 2022.

[11] H. Rong, S. Huo, Q. Zhang, *et al.*, "GREEN: A global optimization scheme for transportation efficiency by mining taxi mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1596–1606, 2022.

[12] N. Garg and S. Ranu, "Route recommendations for idle taxi drivers: find me the shortest route to a customer!," *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1425–1434, London, UK, 2018.

[13] Y. Lai, Z. Lv, K.-C. Li, *et al.*, "Urban traffic coulomb's law: a new approach for taxi route recommendation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 3024–3037, 2019.

[14] G. Wang, S. Zhong, S. Wang, *et al.*, "Data-driven fairness-aware vehicle displacement for large-scale electric taxi fleets," *IEEE 37th International Conference on Data Engineering (ICDE)*, Chania, Greece, 2021.

[15] K. Manchella, M. Haliem, V. Aggarwal, *et al.*, "PassGoodPool: joint passengers and goods fleet management with reinforcement learning aided pricing, matching, and route planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3866–3877, 2022.

[16] Y. Kweon, B. Sun, and B. B. Park, "Preserving privacy with federated learning in route choice behavior modeling," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2675, no. 10, pp. 268–276, 2021.

[17] Y. Zhu, Y. Liu, J. J. Q. Yu, *et al.*, "Semi-supervised federated learning for travel mode identification from GPS trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2380–2391, 2022.

[18] C. Wang, X. Chen, J. Wang, *et al.*, "ATPFL: automatic trajectory prediction model design under federated learning framework," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6563–6572, 2022.

[19] N. Bahra and S. Pierre, "A bidirectional trajectory prediction model for users in mobile networks," *IEEE Access*, vol. 10, pp. 1921–1935, 2021.

[20] X. Wang, Y. Liu, and Z. Liao, "DeepFM-based taxi pick-up area recommendation," *International Conference on Pattern Recognition (ICPR)*, pp. 407–421, 2021.

[21] B. Qu, W. Yang, G Cui, *et al.*, "Profitable taxi travel route recommendation based on big taxi trajectory data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 653–668, 2020.

[22] Y. Zhang, G. Shen, X. Han, *et al.*, "Spatio-temporal digraph convolutional network-based taxi pickup location recommendation," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 394–403, 2023.

[23] X. Lv, Z. Wang, Y. Ren, *et al*, "Traffic network resilience analysis based on the GCN-RNN prediction model," *International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE 2019)*, Zhangjiajie, China, 2019.

[24] Q. Gao, F. Zhou, K. Zhang, *et al*, "Identifying human mobility via trajectory embeddings," *26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, pp. 1689–1695, 2017.

[25] Y. Xu, Z. Piao, and S. Gao, "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, 2018.

[26] L. Liu, J. Zhang, S. H. Song, *et al*, "Client-edge-cloud hierarchical federated learning," *2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020.

[27] R. Zhang and G. Roger, "Demand, supply, and performance of street-hail taxi," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4123-4132, 2020.

[28] C. Shi, Q. Li, S. Lu, *et al*, "Exploring temporal intra-urban travel patterns: An online car-hailing trajectory data perspective," *Remote Sensing*, vol. 13, no. 9, 2021.

[29] P. Sun, A. Boukerche, and Y. Tao, "SSGRU: A novel hybrid stacked GRU-based traffic volume prediction approach in a road network," *Computer Communications*, vol. 160, pp. 502–511, 2020.

[30] M. Wang, Y. Zhang, and G. Cui, "Human motion recognition exploiting radar with stacked recurrent neural network," *Digital Signal Processing*, vol. 87, pp. 125–131, 2019.

[31] Y. Chen, C. Chen, Q. Wu, *et al*, "Spatial-temporal traffic congestion identification and correlation extraction using floating car data," *Journal of Intelligent Transportation Systems*, vol. 25, no. 3, pp. 263–280, 2021.

AUTHOR BIOGRAPHY

Linfeng Liu received the B. S. and Ph. D. degrees in computer science from the Southeast University, Nanjing, China, in 2003 and 2008, respectively. At present, he is a Professor in the School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, China. His main research interests include the areas of vehicular ad hoc networks, wireless sensor networks and multi-hop mobile wireless networks. He has published more than 80 peer-reviewed papers in some technical journals or conference proceedings, such as IEEE TMC, IEEE TPDS, IEEE TIFS, IEEE TITS, IEEE TVT, IEEE TSC, ACM TAAS, ACM TOIT, Computer Networks, Elsevier JPDC. He has served as the TPC member of Globecom, ICONIP, VTC, WCSP.

Yaoze Zhou received the B. S. degree in computer science from the Nanjing University of Posts and Telecommunications in 2021. At present, he is a master student of Nanjing University of Posts and Telecommunications. His current research interest includes the areas of Internet of electric vehicles and vehicular ad-hoc networks.

Jia Xu received the Ph. D. Degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in Jiangsu Key Laboratory of Big Data Security and Intelligent Processing at Nanjing University of Posts and Telecommunications. His main research interests include crowdsourcing, edge computing and wireless sensor networks. Prof. Xu has served as the PC Co-Chair of SciSec 2019, Organizing Chair of ISKE 2017, TPC member of Globecom, ICC, MASS, ICNC, EDGE, and has served as the Publicity Co-Chair of SciSec 2021.