Improving Parking Occupancy Prediction in Poor Data Conditions Through Customization and Learning to Learn

Haohao Qu, Sheng Liu, Zihan Guo, Linlin You, and Jun Li(✉)

School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou, China
{quhaoh,liush235,guozh29}@mail2.sysu.edu.cn
{youlllin,stslijun}@mail.sysu.edu.cn

Abstract. Parking occupancy prediction (POP) can be used for many real-time parking-related services to significantly reduce the unnecessary cruising for parking and additional congestion. However, accurate and fast forecasting in data-poor car parks remains a challenge. To tackle the bottleneck, this paper proposes a knowledge transfer framework that can customize a lightweight but effective pre-trained network to those data-deficient parking lots for POP. The proposed approach integrates two novel ideas, namely Customization: select source domain utilizing reinforcement learning based on parking-related feature matching; and Learning to Learn: extract insightful prior knowledge from the selected sources using Federated Meta-learning. Results of a real-world case study with 34 parking lots in Guangzhou City, China, from June 1 to 30, 2018, show that compared to the baseline, the proposed approach can 1) bring approximately 21% extra performance improvement; 2) improve the model adaptation and convergence speed dramatically; 3) stabilize predictions with error minor variance.

Keywords: Knowledge-based application · Knowledge transfer · Parking occupancy prediction · Federated meta-learning · Reinforcement learning

1 Introduction

Growing applications of parking occupancy prediction (POP) have been witnessed in various parking-related smart services, such as parking guidance [1], dynamic parking charging [2] and parking spaces sharing [3], they alleviate the shortage of parking space, and the associated impact of traffic congestion [4]. However, an emerging bottleneck hinders the mass acceptance of POP services: under-performing in poor data cases. To tackle the bottleneck, this paper proposed an integrated approach, Customization and Learning to Learn (CLL), which can provide a lightweight but effective customized model for the parking lots that lack reliable historical data.

The research is supported by National Natural Science Foundation of China (62002398).

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13368, pp. 159–172, 2022.
https://doi.org/10.1007/978-3-031-10983-6_13

In the literature, data augmentation and structure optimization are two common ways to handle the small-sample prediction task in POP. They improve the performance by incorporating heterogeneous data, but the heavy data dependence on a specific area and the capacity to predict under null-data conditions remain challenges in these studies [5–7]. By contrast, knowledge transfer is a new and effective way to address these problems, using the prior knowledge extracted from the POP tasks in other parking lots (e.g., parking patterns shared between target and source domains) to pre-train a predictive model for car parks with insufficient data [8,9]. However, while transferring extra information for model training to alleviate data shortage, the extra distraction would also accumulate during learning iterations which can deteriorate the performance [10]. Therefore, what and how to transfer becomes the major topic for researchers and a knowledge learning method that can counteract misinformation is required to improve the accuracy of small-sample POP.

To fill the gap, we improve knowledge transfer by adopting two novel ideas, Customization and Learning to Learn, namely 1) Customization: select appropriate source domains (i.e., "guiders") by utilizing reinforcement learning based on feature matching to facilitate the knowledge learning; 2) Learning to Learn: find the update direction by leveraging Federated Meta-learning [11] to pre-train an effective model.

In particular, the main contributions of this paper are as follows:

- Through the integration of Customization and Learning to Learn, the proposed approach provides a new and effective way to solve the small-sample issues in parking occupancy prediction.
- Unlike existing approaches, CLL can learn more insightful knowledge and reduce negative transfer by enabling the pre-training process using federated meta-learning and the client selection using reinforcement learning.
- Experiment on a real-world dataset with 34 parking lots in Guangzhou empirically show that CLL brings nearly 21% extra performance improvement compared with several representative methods.

The remainder of this paper is structured as follows. In Sect. 2 a literature review is presented to summarize the current challenges and solutions in small-sample POP. Then, Sect. 3 introduces the proposed approach for small-sample POP, which is evaluated in Sect. 4. Finally, Sect. 5 concludes the work and discusses future research directions.

2 Related Work

While applying prediction model for the small-sample predictions of parking occupancy and other traffic conditions, several challenges are emerging and quite a few solutions are proposed.

Summary of Challenges. 1) Data shortage: The incompleteness of local parking records puts the personalized models' training in a difficult circumstance that is likely to be trapped in local optimum or over-fitting [12]. 2) Knowledge learning: Disinformation accumulates as additional data is introduced, so avoiding negative learning is vital to model training [13]. 3) Guider selection: The optimization of customization depends crucially on how the clients are selected, as they are the sources of prior knowledge [14]. 4) Generalizability: A flexible data fusion mechanism is required to be capable of handling various degrees of missing data from different car parks.

Related Solutions. Deep learning methods has achieved great success in traffic condition prediction, they emphasize the advantages of data augmentation and heterogeneous data fusion, e.g., R-GANs [15], which provide data recovery by generating samples that look closer to the original data; WoT-NNs [16], which leverages the techniques of Web of Things (WoT) to collect additional information and incorporate them into neural networks; and ToGCN [7], which uses a Topological GCN followed with a Sequence-to-sequence framework to predict future traffic flow and density with temporal correlations. However, they have heavy dependence on availability of other data sources in a specific area and their burdensome structures may cause the lagging in model updates. Differently, knowledge transfer methods provide target domains prior knowledge extracted from source domains through a pre-training process, so that the models can be trained without much historical data. An illustration is a traffic prediction method bringing about 4–13% extra performance improvements by adopting transfer learning framework [8]. However, traditional transfer methods may suffer the negative learning issue to train a biased model because of its direct "reproduction", which fosters the study of improving knowledge transfer. A representative approach is FADACS [9], a GAN-based ConvLSTM transfer learning framework that generates a parking occupancy prediction model utilizing mutual attack of target and source with graph-based patterns. But, as a foundation to optimize the knowledge adaptation for transfer methods, source selection is overlooked yet.

Table 1. Emerging challenges and representative solutions (●: Solved; ◐: Partially; ○: Not-solved)

Challenges	WoT-NNs (2020)	ToGCN (2021)	FADACS (2021)	CLL*(proposed)
Data shortage	●	●	●	●
Knowledge learning	○	◐	●	●
Client selection	○	○	◐	●
Generaliz-ability	◐	◐	◐	●

In summary, Table 1 shows the evaluation of the reviewed methods by their abilities in addressing the four challenges. The knowledge transfer methods (i.e.

FADACS) can outperform the typical deep learning models (i.e., ToGCN) in knowledge learning. However, the source selection is still inadequate, which may make them easier to be misled by the negative transfer. To fill the gap, this paper proposes a novel approach, i.e. CLL, which integrates federated meta-learning and reinforcement learning to pre-train a simple but efficient POP model for data-deficient parking lots.

3 Approach

As illustrated in Fig. 1, the proposed approach consists of A) a model training module, which employs a neural network with LSTM as the backbone; B) a model pretraining Module, which utilizes FedFOMAML (i.e., the Learner) to learn, integrate and transfer prior knowledge from "guiders", then provides the targets a customized and well-trained prediction model; and C) a client selection module, which produces an appropriate client-selecting strategy (i.e., build a Selector). Each part of the proposed approach will be described in the following subsections.

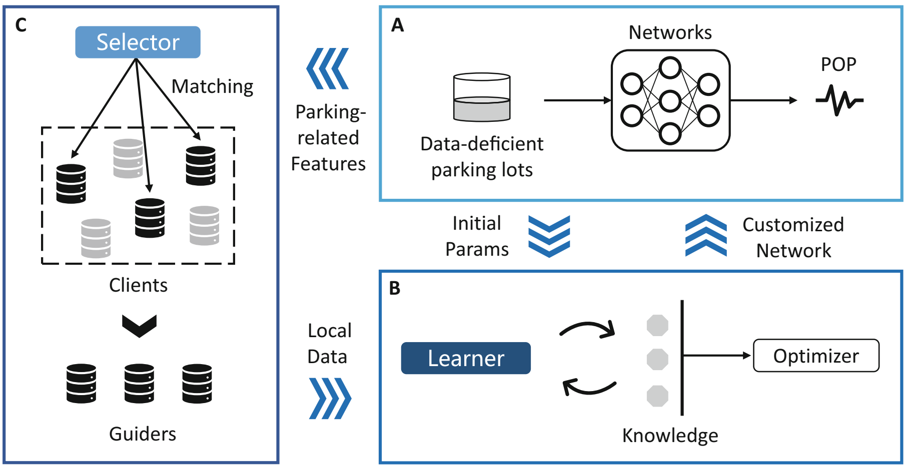

Fig. 1. Overall structure of the proposed approach: A) model training module using LSTM, B) model pretraining module applying FML, C) client selection module based on feature matching.

3.1 The Model Training Module

We utilize Recurrent Neural Network (RNN) which is a widely recognized deep learning approach to process temporal signals, and we pick LongShort-Term-Memory (LSTM) as the building block in RNN due to its success in numerous real-world applications [17]. Briefly, the state transition equation of LSTM is presented in Eq. (1).

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{1}
$$

where h_t is the hidden state at time t, c_t is the cell state at time$t$, x_t is the input at time t, h_{t-1} is the hidden state of the layer at time $t-1$ or the initial hidden state at time 0, and i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively. σ is the sigmoid function, and $\odot$ is the Hadamard product.

3.2 The Model Pretraining Module

Unlike traditional transfer learning, meta-learning takes a longer-term and comprehensive horizon, i.e., it integrates gradients from multi-domains and updates a global network instead of employing network parameters obtained from source domains, which enables the learner learn more insightful knowledge and reduce the adverse impact of negative transfer. The following subsections will describe the meta-learning mechanism and its integration with federated learning for small-sample parking occupancy prediction.

Meta-learning. We adopt a simple but effective meta-learning method, First-order Model-agnostic Meta-learning (FOMAML) [18], which is simplified based on the Model-agnostic Meta-learning (MAML) by ignoring the second derivative terms to reduce the number of gradient steps in finding the best match. Given that M defines the number of pre-training tasks, the objective function of MAML can be defined by (2), which is to find a set of initial parameters minimizing the learner loss. Thereinto, θ represents the intermediate parameter related to the initial parameter ϕ; θ^m denotes the current parameter in task m; $l^m(\theta^m)$ indicates the error with θ^m in task m; and $L(\phi)$ is the total after-training loss of initial parameter ϕ.

$$
\min L(\phi) = \sum_{m=1}^{M} l^m(\theta^m) \tag{2}
$$

The derivative of (2) is the gradient function. To reduce computational complexity, the second derivative terms can be ignored [18]. It means $\nabla_\phi l^m(\theta^m)$ can be replaced to $\nabla_\theta l^m(\theta^m)$, and the gradient function can be written as (3).

$$
\nabla_\phi L(\phi) = \sum_{m=1}^{M} \nabla_\phi l^m(\theta^m) \approx \sum_{m=1}^{M} \nabla_\theta l^m(\theta^m) \tag{3}
$$

where $\nabla_\phi L(\phi)$ is the gradient of $L(\phi)$ with respect to ϕ; $\nabla_\phi l^m(\theta^m)$ is the gradient of $l^m\left(\hat{\theta}^m\right)$ with respect to ϕ; and $\nabla_\theta l^m(\theta^m)$ is the gradient of $l^m(\theta^m)$ with respect to θ.

FedFOMAML for POP. The integration is implemented through a pre-training process as illustrated in Algorithm 1, where the clients and targets are Train set and Test set. Furthermore, these two sets are divided into four separate parts, namely 1) Train-Support R_1, for obtaining local iterative parameters; 2) Train-Query R_2, for getting local gradients; 3) Test-Support R_3, which represents the "few-sample" that target parking lots have for fine-tuning the personalized network; and 4) Test-Query R_4, for evaluating performance. Expressly, assume there are N parking lots in the federation, and M clients are selected as guiders. In a particular epoch p, local gradients are obtained from each guider via a local pre-training based on local data, then the global network parameter ϕ_p can be updated to ϕ_{p+1} the global parameter of the next epoch with the aggregation of local gradients by FedFOMAML mechanism [11]. Given that R_1^m and R_2^m represent local Train-Support set and Train-Query set, respectively, lr denotes learning rate, and θ^m is the iterative parameter in pre-training task m, the process can be written as (4).

$$
\begin{aligned}
\phi_{p+1} &= \phi_p - \frac{lr}{M}\sum_{m=1}^{M} \nabla_\theta l^m(\theta^m,\ R_2^m) \\
s.t.\ \theta^m &= \phi_p - lr\nabla_\phi l^m(\phi_p,\ R_1^m)
\end{aligned}
\tag{4}
$$

Algorithm 1. Learner: FedFOMAML - pseudocode

Require: Batch of pre-training tasks $m = 1, ..., M$ selected from federation

initialize ϕ, learning rate lr, pre-training and fine-tuning max-epochs p, p_f
divide data set into R_1, R_2, R_3, R_4
while not done **do**
  for $i = 1, ..., p$ **do**
    for each pre-training task m **do**
      compute $g_1^m = \nabla_\phi l^m(\phi,\ R_1^m)$
      update ϕ to θ^m with g_1^m
      obtain $g_2^m = \nabla_\theta l^m(\theta^m, R_2^m)$
    end for
    update $\phi = \phi - lr\sum_{m=1}^{M} g_2^m$
  end for
  initiate personalized net $\phi' = \phi$
  for $j = 1, ..., p_f$ **do**
    update ϕ' with R_3
  end for
end while
evaluate predicting performance in R_4

3.3 The Client Selection Module

Feature matching is a common way to select clients, but the weight setting of features remains challenge. Therefore, we employ asynchronous advantage actor-critic (A3C) [19] to train a Selector network to provide a selection strategy according to the predicting performance (reward) and feature matching (state). A3C is a conceptually simple and lightweight framework of deep reinforcement learning (RL) that uses asynchronous gradient descent to optimize deep neural network controllers, and it can significantly shorten the RL training time and make the learning process stable.

Selector Training. The Selector Module consists of a Selector (3-layer MLP) and a Critic (2-layer MLP). The Selector outputs actions (i.e., "guiders" selection) according to the present state modeled with parking-related features, and the Critic is used to evaluate the selection strategies. The process of Selector training using the A3C framework is presented in Algorithm 2. Firstly, we adopt an off-policy strategy for sampling and obtain the rewards through a reward function presented in (5).

$$r(a,\ S_i)\ =\ \lambda \div\ F(a,\ S_i) \tag{5}$$

The reward r is straightforwardly measured by the testing loss of the Learner module given the specific state S_i and actions a. We set a positive constant λ empirically as a threshold value.

After sampling, the coordinator collects all states, actions, and rewards into a buffer. In the case of parking occupancy prediction (POP), the number of training states is equal to the number of clients in the federation, and there will be C_N^M actions given N states and M parking lots to select. Each client utilizes Advantage Actor-Critic (A2C) to calculate local gradients of Selector and Critic, respectively, then update the global network's parameters with the aggregation of batch local gradients.

In term of Critic updating, a value-based method is employed for the Critic updating. The estimated Value V_π^s indicates the approximate expectation of rewards in a particular state s, which is defined by (6).

$$V_\pi^s = E_\pi\left(r\left(\widetilde{a}, s\right)\right) = \frac{1}{T}\sum_{t=1}^{T} r\left(a_t, s\right)\frac{\rho_\pi\left(a_t|s\right)}{\rho_{\pi'}\left(a_t|s\right)} \tag{6}$$

where $E_\pi\left(r\left(\widetilde{a}, \hat{s}\right)\right)$ represents the expected value of the reward r for all actions $\widetilde{a}$ in a specific state $\hat{s}$ using Selector π; V_π^s indicates how good the Selector could do; T is the number of actions in one state; $r(a_t, \hat{s})$ denotes the reward of action a_t in the state $\hat{s}$; $\rho_\pi(a_t|\hat{s})$ represents the probability of that the Selector takes action a_t using parameter π; and $\rho_{\pi'}(a_t|\hat{s})$ indicates the global sampling distribution that could be omitted if it were a uniform distribution.

Then, Critic gradients of state s_m can be calculated through the square error (SE) regression between the Critic's output $V_{critic}^{s_m}$ and the observed value $V_\pi^{s_m}$. Under the federation framework, we update the global Critic net π_c using the aggregated gradients as Eq. (7).

$$\pi_c' = \pi_c - \beta \sum_{m=1}^{M} \nabla_{\pi_c} (V_\pi^{s_m} - V_{critic}^{s_m})^2 \tag{7}$$

In term of Selector updating, according to the idea of policy-based advantage function [19], when the reward of an action is greater than the valuation of Critic $V_c ritic$, its probability goes up, otherwise, goes down. Then, the gradient of Actor (i.e., Selector) $\nabla \bar{R}_\pi^s$, can be written as (8).

$$\nabla \bar{R}_\pi^{\hat{s}} = \frac{1}{T} \sum_{t=1}^{T} \left(r\left(a_t, \hat{s}\right) - V_{critc}\right) \nabla_{\pi_\mathrm{a}} \log\, \rho_\pi \left(a_\mathrm{t} | \hat{\mathrm{s}}\right) \tag{8}$$

Similar to the updating process of Critic net, that of Selector net π_a also can leverage the aggregated local gradients as illustrated in Eq. (9).

$$\pi_a' = \pi_a - \beta \sum_{m=1}^{M} \nabla \bar{R}_\pi^{s_m} \tag{9}$$

Finally, after iterative updating, the Selector would be able to find a "best" policy that selecting a certain number of good "guiders" for the Learner module.

Algorithm 2. Selector Training: A3C - pseudocode

Require: the Learner module, parking-related features

```
assume that the numbers of clients and selected guiders are N and M, then the
  amount of states and actions are N and T = C_N^M respectively
initialize global Selector and Critic net π_a and π_c; model state s with features
for episode do
  sample distribution π' = π
  sample local states and actions to global buffer
  compute rewards with Learner and reward function
  distribute samples (s, a, r) to corresponding member
  for each member m do
    freeze π_a
    dπ_c ← 0
    compute V_π^{s_m}
    obtain dπ_c^m by regression
  end for
  update π_c, release π_a
  for each member m do
    freeze π_c
    dπ_a ← 0
    obtain dπ_a^m by advantage function
  end for
  update π_a, release π_c
end for
```

4 Experiments and Results

In this section, we deploy the proposed method on a real-world parking occupancy prediction case, and the predicting performance will be evaluated together with other representative forecasting methods based on the same evaluation metrics. Moreover, the results will be analyzed to demonstrate the improvements achieved.

4.1 Evaluation Preparation

Data Declaration. A shared dataset with a minimum resolution of 5 min was created based on the parking occupancy data of 34 parking lots in Guangzhou City, China, from June 1 to 30, 2018. There are four parking lots (Target 1–4) for testing and 30 (Client 1–30) for training. The parking-related features include: 1) parking-related points of interest (POI), which are collected from Gaode API, and the kernel density are clustered into 20 classes; 2) parking lot types, which are divided into six categories according to the land use, namely Commercial, Office, Residential, Hospital, Recreational, Tourism.

Dataset division: the Train-Support set and Train-Query set are (Day 1–18) and (Day 19–24) for FOMAML pre-training; the Test-Query set is (Day 25–30) for model performance evaluation. Expressly, we consider five data conditions of the Test-Support set, namely complete data (Day 1–24, 24d), partial data (Day 18–24, 6d), small data (Day 21–24, 3d), few data (Day 24, 1d), empty data (null).

Baselines and Competitive Approaches. Several representative methods for time-series prediction are compared. Including, 1) Fully Connected Neural Network (FCNN): which is widely used in function approximation and general regression problems, but it relies on feature extraction and cannot distinguish between temporal features and spatial features; 2) Long Short-Term Memory (LSTM) [17]: a recurrent-based method that is widely used in many time-series prediction tasks, which is set as the baseline; 3) Gated Recurrent Unit (GRU) [20]: a simplified LSTM structure which has advantages of rapidly computing; and 4) Bi-directional Long Short-Term Memory (BiLSTM) [21]: a combination of forward LSTM and backward LSTM, which is often used to model context information in natural language processing tasks. 5) Auto-regressive Integrated Moving Average (ARIMA) [22]: A statistical model which can be used to model time series as long as the data or the difference of the data is stationary. 6) Support Vector Regression (SVR) [23]: A typical machine learning model. 7) a traditional transfer learning method named Transfer-LSTM [24] deploys LSTM as backbone network for time-series prediction. In addition, to validate the Selector, we also evaluate the performance of FML (i.e., the Learner) which selects "guiders" randomly. The running configuration is illustrated in Table 2.

Table 2. Running configurations

Model	Param	Value	Comment
*	(N, M, T)	(30, 3, 4060)	The number of (states, guiders, and actions)
	Channel	(1, 6, 1)	(Input channel, sequence length, output channel)
	Learning rate	(0.03, 0.02, 0.05)	(Pretraining, Fine-tuning, Selector training)
	Max epochs	(200, 400, 5000)	(Pretraining, Fine-tuning, Selector training)
	Optimizer	BGD | Adam	Pretraining | Fine-tuning & Selector training
	Loss function	MSE | CE	Pretraining & Fine-tuning | Selector training
ARIMA	(p, d, q)	(2, 1, 1)	Implemented on Statsmodels
SVR	Kernel, epsilon	RBF, 0.001	Implemented on Scikit-learn (sklearn)

* CLL, FML, transfer-LSTM and NNs.

Experiment Setting. The objective is to predict the parking occupancy rate for the next 30 min (6 timesteps as accurately as possible in the last six days (25–30) of June 2018. The evaluation metrics are Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), and Coefficient of Determination (R-square). Finally, All the experiments are conducted on a Windows workstation with four NVIDIA GeForce RTX 3090 GPU, an Intel Gold 5218R Two-Core Processor CPU, and 512 G RAM. The pretraining process takes about 20 min, but the Selector training process costs around two weeks for the experiment. It is worth noting that for reproductivity, the dataset and code used by this paper are shared on Github and downloadable from the link[1].

4.2 Result and Discussions

Forecasting Error. As shown in Table 3, the average metrics of compared models in the five different data volumes (empty, few, small, partial, and complete data) are summarized. We can see the methods with a knowledge transfer learning framework are much superior to those without, and the proposed approach reduces the prediction errors significantly with the highest scores in all the four evaluation metrics, namely MAPE 4.77%, RMSE 0.0312, RAE 15.66%, and R2 96.36%. Compared to its backbone network LSTM, CLL brings nearly 21% extra performance improvements. Furthermore, the LSTM model using CLL outperforms that using FML, demonstrating the effectiveness of our auto-selector. As shown in Table 4, the selection result indicates that our selector views clients with a closer density of POI to the target as a good "guider". Moreover, it is better if the client is in the same type of parking as the target.

Convergence Profile. We compare the convergence profiles of CLL and its backbone network LSTM (i.e., the baseline), and the first-100-epochs MSE-loss curves are given in Fig. 2. We can see that the errors at the beginning of the curves for ALL-LSTM are smaller than that at the 100^{th} iteration of LSTM, indicating the advantage of ALL for knowledge extraction and propagation.

[1] https://github.com/Quhaoh233/CLL.

Table 3. List of average metrics of the compared models

Model	RMSE (10^{-2})	MAPE (%)	R2 (%)	RAE (10^{-2})
CLL-LSTM	3.12	4.77	96.36	15.66
FML-LSTM	3.36	5.05	95.94	16.41
Transfer-LSTM	3.76	5.79	95.33	17.72
SVR	3.90	5.83	94.31	19.01
LSTM	3.97	5.40	94.96	17.23
BiLSTM	4.10	6.24	93.71	20.23
GRU	5.27	8.19	88.10	26.24
FCNN	5.85	8.87	84.86	30.11
ARIMA	6.03	9.40	80.45	33.54

The metrics are averaged based on the result of 20 tasks (4 parking lots $\times$ 5 data volumes).

Table 4. Parking-related features of the targets and selected clients

Feature	Target1	Client1	Client5	Client11	Target2	Client5	Client8	Client10
Density of POI	12	8	14	10	18	14	17	18
Type of car parks	A	A	A	B	A	A	A	A
Feature	Target3	Client3	Client14	Client25	Target4	Client12	Client17	Client27
Density of POI	4	3	2	2	11	9	12	11
Type of car parks	C	A	C	E	D	B	C	F

A: Commercial; B: Hospital; C: Office; D: Residential; E: Recreational; F: Tourism.

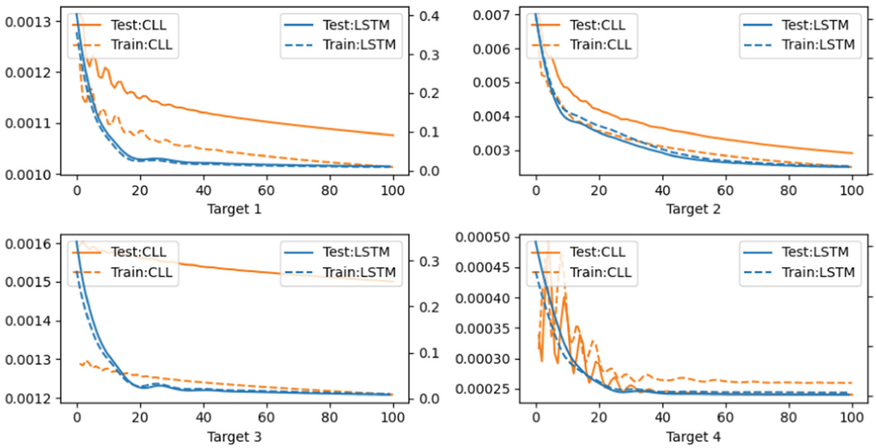

Fig. 2. Comparison of convergence profile between CLL-LSTM and LSTM

Error Variance. To emphasize the stability brought by the meta-learning pre-training and client selection, we give an illustration in Fig. 3, which reveals that using knowledge transfer (i.e., Transfer-LSTM) can lower the error variance of the learning model (i.e., LSTM). Further, using the CLL pre-training framework gives the prediction model a smaller error box than using the traditional transfer learning. The above results demonstrate that the proposed framework can stabilize prediction, improving its application significance in real-world scenarios.

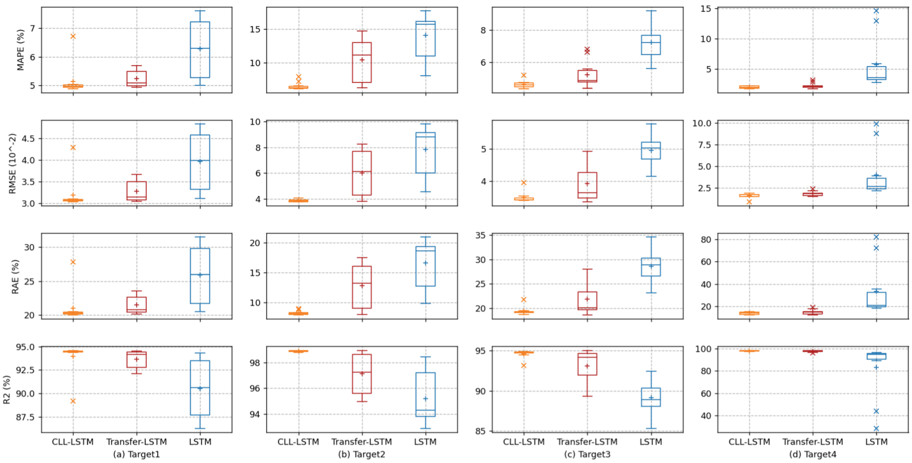

Fig. 3. Boxplot of the metrics in four target prediction tasks respectively, the baseline is LSTM

In summary, the combination of FedFOMAML in model pretraining and A3C in source selection is efficient and effective. As shown by the evaluation results, the proposed approach has the following advantages, 1) high accuracy and scalability, scoring highest in all five data groups and four evaluation metrics, with a significant reduction of 21% in prediction error; 2) fast adaptation, with model adaptation and convergence speed substantially improved by 10^2 iterations over the model without CLL; and 3) good stability, reducing the variance of the predictions.

5 Conclusions and Future Works

This paper proposed a knowledge transfer approach to support few-data parking occupancy prediction, which integrated two novel ideas, namely 1) Learning to Learn: which leveraged the Federated Meta-learning framework to transfer multi-domains knowledge; 2) Customization: which improved the performance via training a Selector for client selection. The evaluation results showed that the proposed approach CLL outperformed the compared methods in three aspects:

significant performance, fast adaptation, and more minor variances. It provided a simple but effective way to solve small-sample parking occupancy prediction.

In the future, the work will be further enhanced:

1) To extend its application: The ideas of Customization and Learning to Learn will be extended to more scenarios, e.g., traffic flow, density and speed prediction in limited-sensing roads.
2) To consider data security: The proposed approach will be improved by integrating advanced federated learning method to avoid data leakage and bridge data islands among parking facilities.

Acknowledgements. The research is supported by National Natural Science Foundation of China (62002398).

References

1. Xiao, J., Lou, Y.: An online reinforcement learning approach for user-optimal parking searching strategy exploiting unique problem property and network topology. IEEE Trans. Intell. Transp. Syst. 1–13 (2021)
2. Ding, H., Qian, Y., Zheng, X., Bai, H., Wang, S., Zhou, J.: Dynamic parking charge-perimeter control coupled method for a congested road network based on the aggregation degree characteristics of parking generation distribution. Physica A-Stat. Mech. Appl. **587**, 126481 (2022)
3. Zhang, F., Liu, W., Wang, X., Yang, H.: Parking sharing problem with spatially distributed parking supplies. Transp. Res. Part C-Emerg. Technol. **117**, 102676 (2020)
4. Bock, F., Di Martino, S., Origlia, A.: Smart parking: using a crowd of taxis to sense on-street parking space availability. IEEE Trans. Intell. Transp. Syst. **21**(2), 496–508 (2020)
5. Wang, Y., Zhang, H.: Introducing graph neural networks for few-shot relation prediction in knowledge graph completion task. In: The 14th International Conference on Knowledge Science, Engineering and Management, pp. 294–306 (2021)
6. Yang, S., Ma, W., Pi, X., Qian, S.: A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. Transp. Res. Part C Emerg. Technol. **107**, 248–265 (2019)
7. Qiu, H., Zheng, Q., Msahli, M., Memmi, G., Qiu, M., Lu, J.: Topological graph convolutional network-based urban traffic flow and density prediction. IEEE Trans. Intell. Transp. Syst. **22**(7), 4560–4569 (2021)
8. Zhang, C., Zhang, H., Qiao, J., Yuan, D., Zhang, M.: Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data. IEEE J. Sel. Areas Commun. **37**(6), 1389–1401 (2019)
9. Shao, W., et al.: FADACS: a few-shot adversarial domain adaptation architecture for context-aware parking availability sensing. In: 2021 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 1–10 (2021)
10. Gui, L., Xu, R., Lu, Q., Du, J., Zhou, Yu.: Negative transfer detection in transductive transfer learning. Int. J. Mach. Learn. Cybern. **9**(2), 185–197 (2017). https://doi.org/10.1007/s13042-016-0634-8

11. Lin, S., Yang, G., Zhang, J.: A collaborative learning framework via federated meta-learning. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), pp. 289–299 (2020)
12. Yang, M.: A survey on few-shot learning in natural language processing. In: 2021 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA), pp. 294–297 (2021)
13. Khan, A., Kim, J.-S., Kim, H.S.: Damage detection and isolation from limited experimental data using simple simulations and knowledge transfer. Mathematics **10**(1) (2022)
14. Jie, X., Wang, H.: Client selection and bandwidth allocation in wireless federated learning networks: a long-term perspective. IEEE Trans. Wireless Commun. **20**(2), 1188–1200 (2021)
15. Sun, Y., Peng, L., Li, H., Sun, M.: Exploration on spatiotemporal data repairing of parking lots based on recurrent GANs. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 467–472 (2018)
16. Provoost, J.C., Kamilaris, A., Wismans, L.J.J., Van der Drift, S.J., Keulen, M.V.: Predicting parking occupancy via machine learning in the web of things. Internet Things **12**, 100301 (2020)
17. Zhao, Z., Chen, W., Wu, X., Chen, P.C.Y., Liu, J.: LSTM network: a deep learning approach for short-term traffic forecast. IET Intell. Transp. Syst. **11**(2), 68–75 (2017)
18. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: The 34th International Conference on Machine Learning (ICML), vol. 70, pp. 1126–1135 (2017)
19. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning (ICML) (2016)
20. Cho, K., van Merrienboer, B., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encode-decoder for statistical machine translation. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014)
21. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
22. Jaynes, E.T.: On the rationale of maximum-entropy methods. Proc. IEEE **70**(9), 939–952 (1982)
23. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. **5**(4), 276–281 (2004)
24. Li, J., Guo, F., Wang, Y., Zhang, L., Na, X., Hu, S.: Short-term traffic prediction with deep neural networks and adaptive transfer learning. In: IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC) (2020)