

Article

Adaptation and Learning to Learn (ALL): An Integrated Approach for Small-Sample Parking Occupancy Prediction

Haohao Qu ¹, Sheng Liu ¹, Jun Li ^{1,*}, Yuren Zhou ² and Rui Liu ³

¹ School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China; quhaoh@mail2.sysu.edu.cn (H.Q.); liush235@mail2.sysu.edu.cn (S.L.)

² Engineering Product Development, Singapore University of Technology and Design, Singapore 487372, Singapore; yuren_zhou@outlook.com

³ School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore; rui.liu@ntu.edu.sg

* Correspondence: stsljun@mail.sysu.edu.cn

Abstract: Parking occupancy prediction (POP) plays a vital role in many parking-related smart services for better parking management. However, an issue hinders its mass deployment: many parking facilities cannot collect enough data to feed data-hungry machine learning models. To tackle the challenges in small-sample POP, we propose an approach named Adaptation and Learning to Learn (ALL) by adopting the capability of advanced deep learning and federated learning. ALL integrates two novel ideas: (1) Adaptation: by leveraging the Asynchronous Advantage Actor-Critic (A3C) reinforcement learning technique, an auto-selector module is implemented, which can group and select data-scarce parks automatically as supporting sources to enable the knowledge adaptation in model training; and (2) Learning to learn: by applying federated meta-learning on selected supporting sources, a meta-learner module is designed, which can train a high-performance local prediction model in a collaborative and privacy-preserving manner. Results of an evaluation with 42 parking lots in two Chinese cities (Shenzhen and Guangzhou) show that, compared to state-of-the-art baselines: (1) the auto-selector can reduce the model variance by about 17.8%; (2) the meta-learner can train a converged model $10^2 \times$ faster; and (3) finally, ALL can boost the forecasting performance by about 29.8%. Through the integration of advanced machine learning methods, i.e., reinforcement learning, meta-learning, and federated learning, the proposed approach ALL represents a significant step forward in solving small-sample issues in parking occupancy prediction.

Keywords: small-sample prediction; federated meta-learning; reinforcement learning; knowledge transfer; parking occupancy

MSC: 68T05; 68T07; 68T09



Citation: Qu, H.; Liu, S.; Li, J.; Zhou, Y.; Liu, R. Adaptation and Learning to Learn (ALL): An Integrated Approach for Small-Sample Parking Occupancy Prediction. *Mathematics* **2022**, *10*, 2039. <https://doi.org/10.3390/math10122039>

Academic Editor: Catalin Stoean

Received: 8 May 2022

Accepted: 10 June 2022

Published: 12 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The issue of parking is one of the most important challenges faced by urban transportation systems, and the negative impacts it causes can seriously degrade the travel experience of urban residents. A recent investigation indicated that between 9 and 56 percent of traffic was cruising for parking, and the average search time was about 6.03 min [1]. It can be said that “cruising for parking” has become a common phenomenon in areas with high travel demand, which not only increases travel costs for travelers but also causes additional congestion and emissions [2]. Therefore, optimizing the allocation of parking resources is a critical and urgent issue. There have been many solutions proposed to address this problem and parking occupancy prediction (POP) plays a vital role in most of them [3,4], examples including dynamic parking charges [5,6], parking guidance [7,8], and shared parking [9].

However, these smart parking services and applications are still struggling for mass deployment due to the lack of a practical approach to small-sample parking occupancy

prediction (POP), i.e., a small-sample prediction problem [10]. Traditional data-hungry models are unable to predict with high accuracy and stability in a data-poor scenario, resulting in insufficient quality of service provided, which seriously undermines users' trust [11]. Although, with the development of Smart Cities, more and more sensing devices have been deployed for real-time data collection of all kinds, limited-sensing parking facilities still account for the majority [12], e.g., on-street and public-free parking lots. As a result, an effective and efficient POP method capable of addressing the small-sample issue is required to enable full coverage of smart parking services while reducing the cost of building sensing infrastructures.

In the current literature, there are three common ways to handle data shortages: (1) feature enhancement by introducing extra heterogeneous data [8,13,14]; (2) structure optimization by incorporating spatial-temporal data into deep neural networks [15,16]; and (3) transfer learning by pre-training the model [17,18]. However, these methods still suffer from several shortages. Feature enhancement requires a mass of data in a specific area, which is not always available in real-world scenarios. The training process for a complex model is time-consuming and laborious. Moreover, domain shift and adaptation remain challenges in transfer learning.

To overcome these shortages, this paper proposed an approach named ALL which can provide a lightweight and effective personalized model for the data-poor parking lots by integrating two novel ideas, i.e., Adaptation and Learning to Learn. As illustrated in Figure 1, the proposed algorithm consists of three modules, namely: (1) Federation: a community formed by parking lots, which adopts federated learning framework [19] to bridge data islands among the clients and support distributed computing; (2) Selector: an auto-selector used to select appropriate “guiders” automatically for the Learner module, which is trained by a Reinforcement Learning (RL) method called Asynchronous Advantage Actor-Critic (A3C) [20] with parking-related features; and (3) Learner: a meta-learner based on Federated First-order Model-agnostic Meta-learning (FedFOMAML) [21], which utilizes the aggregated gradients obtained from selected other clients (i.e., “guiders”) to update the personalized parameters of the target parking lot. In summary, the proposed approach enables the POP model to support distributed computing, customize knowledge sources and learn extra knowledge by integrating the three modules (i.e., Federation, Selector, and Learner).

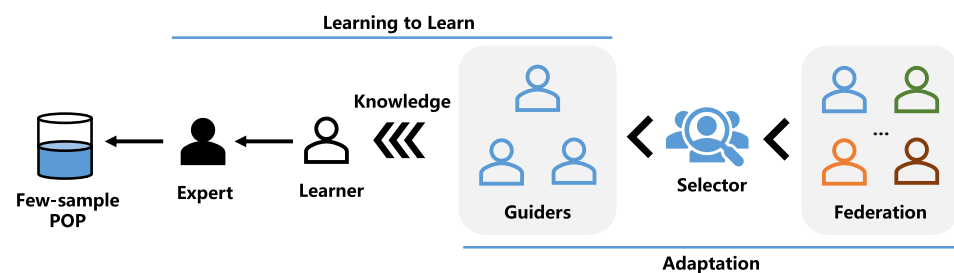


Figure 1. A neat illustration of Adaptation and Learning to Learn (ALL).

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

- Through the integration of advanced machine learning methods, i.e., reinforcement learning, meta-learning, and federated learning, the proposed approach ALL represents a significant step forward in solving small-sample issues in parking occupancy prediction;
- Different from existing approaches, ALL avoids the heavy dependence on large volume data by deploying a knowledge transfer framework. Furthermore, it employs a meta-learner based on FedFOMAML to improve domain shift via extracting gradients from multi-domains, and trains an auto-selector by A3C for source filtering to reduce negative transfer;

- We test the proposed algorithm on a real-world dataset with 42 parking lots in two Chinese first-tier cities (Shenzhen and Guangzhou). The results empirically show that: (1) the auto-selector reduces the model variance by about 17.8%; (2) the meta-learner trains a converged model $10^2 \times$ faster; (3) finally, ALL obtains the highest scores in all the four evaluation metrics, namely RMSE 0.0385, MAPE 6.82%, R2 94.49% and RAE 17.23%, and brings an extra performance improvement of nearly 29.8% compared to the state-of-the-art methods.

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

2. Related Work

While applying a prediction model for small-sample POP, several challenges are emerging and quite a few solutions are proposed.

2.1. Emerging Challenges

In general, to create an efficient and effective model for small-sample POP, four challenges are emerging, including:

- **Data Shortage:** The shortage of local data puts model training processes in a difficult circumstance that is likely to over-fit and be trapped in local optimum. POP models need access to additional information to help extrapolate future parking occupancy;
- **Knowledge Learning:** While introducing more data for model training to alleviate data shortages, extra distractions would accumulate during learning iterations called negative learning can deteriorate the performance [22]. Therefore, a knowledge learning method that can counteract misinformation is required to improve the accuracy of small-sample POP;
- **Knowledge Adaptation:** Besides knowledge learning, knowledge adaptation also remains a challenge. Boosting knowledge adaptation can not only shorten computing speed and leave considerable computing resources but also stabilize prediction variance, making POP more practical and applicable [23,24];
- **Model Scalability:** The utilization pattern may vary between different data richnesses. POP models have to face many different cases of missing data, for example, a parking lot without any parking records or with partial but insufficient occupancy. Therefore, a unified and scalable model is required, which shall manage not only complete-data cases but also empty-data scenarios effectively and efficiently [25].

2.2. Related Solutions

To tackle the aforementioned challenges, several solutions are proposed. First and foremost, in order to address data shortages, the main focus of some solutions remains on data collection and analysis to better interpret and infer the current and future parking statuses [12,13,26]. Examples include: (1) CoPASample [27] and BATF [28], which utilize heuristics-based covariance and Bayesian augmented tensor factorization, respectively, to generate synthetic samples that look close to the original data; and (2) WoT-NNs [14], which leverages the techniques of Web of Things (WoT) to collect additional information and incorporate them into neural networks. These data augmentation methods do not require significant amounts of data and can be generated by intuition or domain knowledge. However, as a knowledge learning process, the feature extraction of the generated and introduced heterogeneous data is still challenging for the traditional time-series prediction models [29]. Further, their generalization to the test data (i.e., knowledge adaptation) is usually unsatisfactory.

During the last decade, models of deep neural networks have gained huge popularity in the area of traffic modeling and prediction, and network structure optimization is emphasized to make full use of spatial and temporal data [15,16,30]. As illustrated by

MGCN-LSTM [31] which extracts discriminative features from the multi-source data, and combines the multiple-graph convolutional neural network (MGCN) and the long short-term memory (LSTM) network for capturing complex spatio-temporal correlations. Even though these models can handle fused data with high performance, they have a heavy dependence on the availability of other data sources. Moreover, creating a complex model may cause the over-consumption of computing resources and the lagging in model updates.

Therefore, methods built on the idea of transferring knowledge between entities are recently developed, which is a novel way to make the model scalable and less data-dependent. For instance, FADACS [18] is designed to implement transfer learning based on a GAN (Generative Adversarial Network). It can generate an adaptive model based on the shared patterns extracted from the mutual attack between the target and the source. Hence, FADACS can support POP in parks whose data are insufficient or even “empty” [32]. However, since FADACS transfers all the knowledge from a single source, it may suffer the over-learning issue to train a biased model. Such discrepancies are substantial in some other cases. Therefore, drawing on advanced experience in Computer Vision and Natural Language Processing, meta-learning [33] and its extensions [34,35] developed for few-shot learning can be deployed in small-sample POP to extract multi-domains knowledge and enhance knowledge learning. Furthermore, as a foundation to optimize the knowledge adaptation for transfer methods, knowledge source selection is overlooked yet [36]. Considering the large amount of parking lots in cities, an auto-selector that can automatically select suitable knowledge source domains is needed.

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., HST-GSNs) in knowledge learning and model scalability. However, the source selection of transfer methods 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., ALL, which integrates FedFOMAML and A3C to pre-train a simple but efficient POP model for data-deficient parking lots.

Table 1. Challenges and representative solutions in small-sample POP (●: Solved; ◐: Partially; ○: Not-solved).

Challenges	WoT-NNs [14] (2020)	MGCN-LSTM [31] (2021)	FADACS [18] (2021)	ALL (Proposed)
Data Shortage	◐	●	●	●
Knowledge Learning	◐	◐	●	●
Knowledge Adaptation	○	◐	◐	●
Model scalability	○	○	◐	●

3. Approach

As illustrated in Figure 2, the proposed approach consists of: (1) the Federation framework, which is deployed to bridge data islands and support distributed computing among clients and server; (2) the Learner module, which utilizes FedFOMAML to learn, integrate, and transfer prior knowledge, so that the target can finally obtain a customized well-trained network; and (3) the Selector module, which is trained by A3C and produces an appropriate client-selecting strategy to the Learner. Besides, we apply deep learning in two parts, i.e., the backbone network using recurrent neural units to extract temporal features and the auto-selector using multi-layer perception (MLP) to match heterogeneous features.

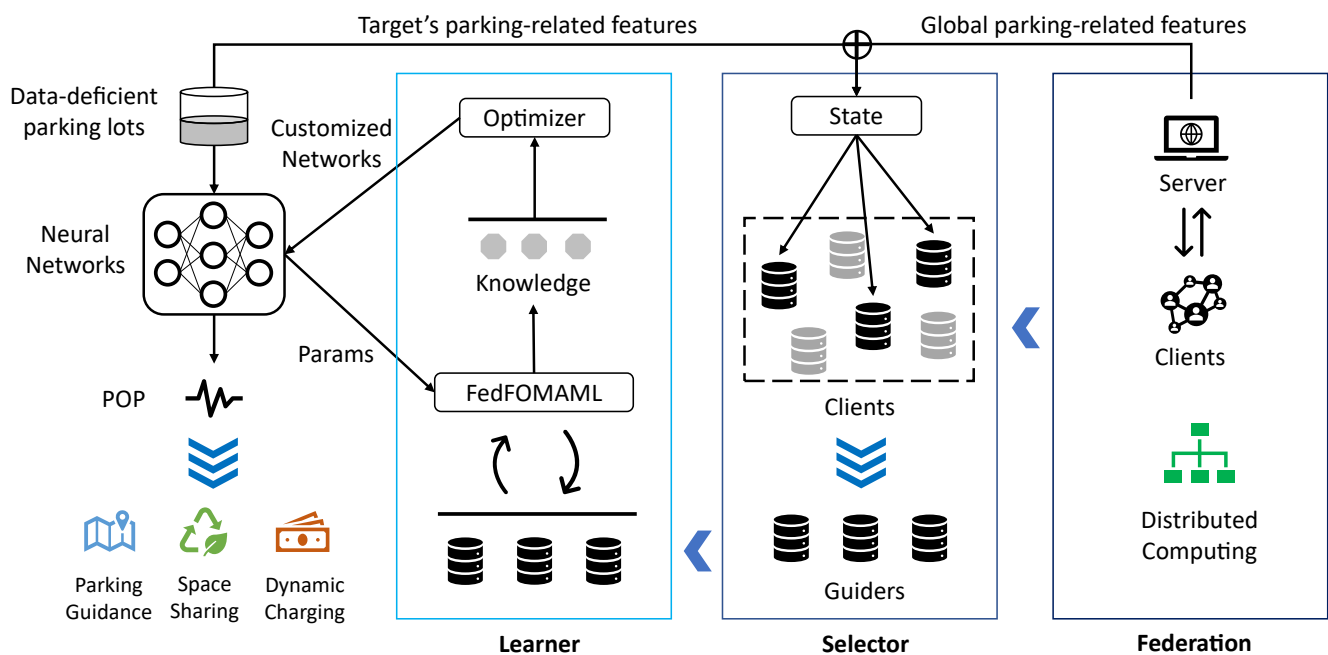


Figure 2. The architecture diagram of the proposed approach.

The definition of the small-sample problem in parking occupancy prediction will first be described, and each part of the proposed approach will be described in the following subsections. It is noted that the description of the Selector Module will come after the Learner Module for easier elaboration.

3.1. Problem Definition

The objective of the prediction problem is to minimize the difference L between the predicted value y_p^t and the actual value y_a^t at time t . Typical time-series prediction solutions learn the patterns presented in the historical data x_l and then use them to predict the future. In contrast, we solve the small-sample issues by introducing additional information (i.e., historical data x_g from other parking lots, local parking-related features f_l , global parking-related features f_g). Given the above data as input, the future parking occupancy in the target parking lot can be calculated by a function presented in (1). In general, the prediction step can be set to between 30 and 60 min, as the average travel time for an intercity trip is also in between [37].

$$\min L(y_a^t, y_p^t) \tag{1}$$

$$y_p^t = F(x_l, x_g, f_l, f_g).$$

3.2. Federation Framework

Federated Learning (FL) is an emerging concept in data management that represents a distributed computation framework for machine learning. In the context of parking occupancy prediction, collecting massive multi-source data together to train traditional machine learning models is usually impracticable since: (1) parking records are of great potential value considering data security and privacy, and both users and managers are unwilling to share their data without any economic compensation; and (2) it would bring heavy computing overloads to the server and congestion to communication channels.

To enable collaborations between different parking lots for better occupancy prediction performance, especially for those with little data, we adopt a horizontal FL framework [19,38], which is a typical client–server structure to form a parking lots federation and carry out the distributed computation. This federation can broaden sample spaces among clients (i.e.,

parking lots) by exchanging knowledge such as gradients and parameters between clients. Moreover, the distributed computation can dilute computing workloads by implementing parallel training locally. The Learner module and Selector module are embedded in this framework.

3.3. Learner Module

Meta-learning is a method proposed in the field of Computer Vision to solve the “Few-shot Learning” problem, whose essence is to increase the generalization ability of the learner in multi-task and make the knowledge be quickly adapted to new tasks. Considering the advantages of solving the small-sample issue, we adopt federated first-order model-agnostic meta-learning (FedFOMAML) in our Learner module for learning, integrating, and transferring parking-related knowledge.

The following subsections will describe the first-order model-agnostic meta-learning (FOMAML) mechanism and its integration with federated learning for small-sample parking occupancy prediction.

3.3.1. FOMAML

First-order Model-agnostic Meta-learning (FOMAML) 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 [33]. 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 (i.e., the predicting error in POP). 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 loss with θ^m in task m ; and $L(\phi)$ is the total after-training loss of initial parameter ϕ .

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

The derivative of (2) is the gradient function. To reduce computational complexity, the second derivative terms can be ignored [33]. 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).

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

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 $l^m(\theta^m)$ with respect to ϕ ; and $\nabla_{\theta} l^m(\theta^m)$ is the gradient of $l^m(\theta^m)$ with respect to θ .

3.3.2. FedFOMAML for POP

The integration is implemented through a pre-training process as illustrated in Algorithm 1 and Figure 3, where the data from clients and targets form the Train set and Test set, respectively. Furthermore, these two sets are divided into four separate portions according to the dataset partition requirements of meta-learning, 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 “small-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 the global parameter of the next epoch (ϕ_{p+1}) through the aggregation of local gradients by FedFOMAML mechanism [39]. Given that R_1^m and R_2^m represent the local Train-Support set

and Train-Query set, respectively, β denotes learning rate, and θ^m is the iterative parameter in pre-training task m , the process can be written as (4).

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

Next, this well-trained network ϕ can be transferred to the target client (parking lot) and fine-tuned to a personalized model with Test-Support set R_3 . Finally, the prediction performance can be evaluated in Test-Query set R_4 . Obtaining knowledge from all federation members is a thankless policy that spends enormous time and computing resources, which means selecting appropriate learning objects can make a big difference to the performance of the meta-learner.

Algorithm 1 Meta-Learner: FedFOMAML—pseudocode.

- Require:** batch of pre-training tasks $m = 1, \dots, M$ selected from federation
- 1: initialize ϕ , learning rate β , pre-training and fine-tuning max-epochs p, p_f
 - 2: divide data set into R_1, R_2, R_3, R_4
 - 3: **while** not done **do**
 - 4: **for** $i = 1, \dots, p$ **do**
 - 5: **for** each pre-training task m **do**
 - 6: compute $g_1^m = \nabla_{\phi} l^m(\phi, R_1^m)$
 - 7: update ϕ to θ^m with g_1^m
 - 8: obtain $g_2^m = \nabla_{\theta} l^m(\theta^m, R_2^m)$
 - 9: **end for**
 - 10: update $\phi = \phi - \beta \sum_{m=1}^M g_2^m$
 - 11: **end for**
 - 12: initiate personalized net $\phi' = \phi$
 - 13: **for** $j = 1, \dots, p_f$ **do**
 - 14: update ϕ' with R_3
 - 15: **end for**
 - 16: **end while**
 - 17: evaluate predicting performance in R_4

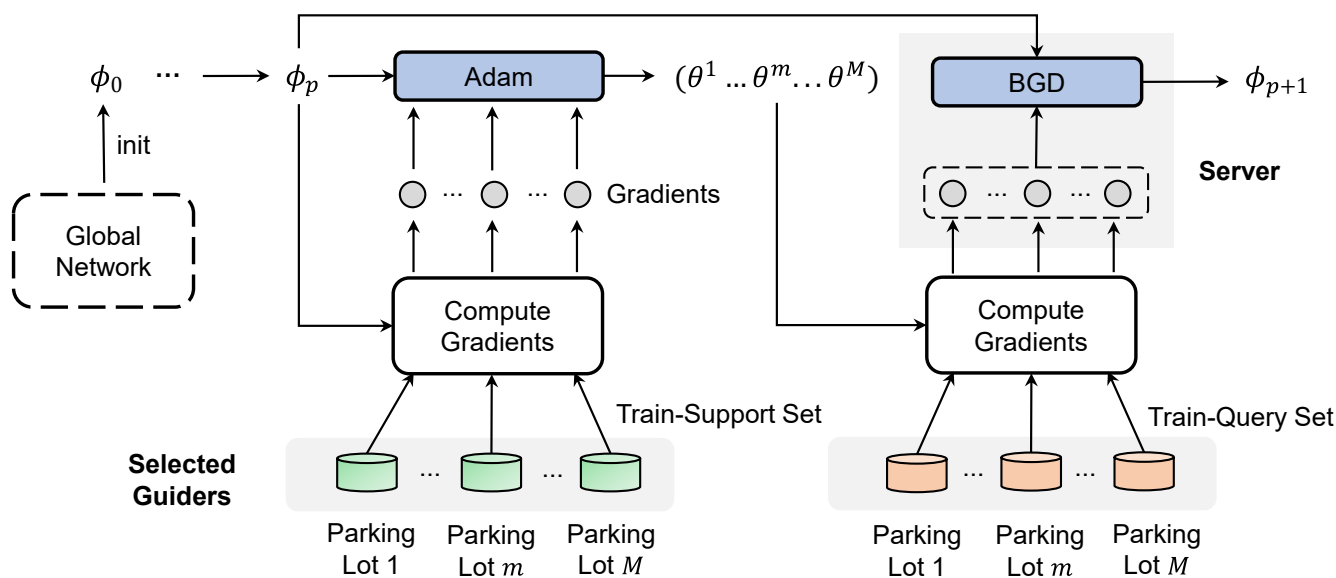


Figure 3. FedFOMAML pre-training framework in parking occupancy prediction tasks.

3.4. Selector Module

The major challenge of the client selection process is the overlarge space for states and actions. Given N parking lots and M “guiders”, the selector faces N states and C_N^M possible actions. Although the value-based RL methods using a single critic network (e.g., Deep Q Networks [40]) take the advantages of efficient computation [41], the actor-critic setup is more suitable for the scenarios with large action spaces. Therefore, we employ asynchronous advantage actor-critic (A3C) [20] that uses asynchronous gradient descent to optimize deep neural network controllers, significantly shorten the RL training time, and make the learning process stable. The RL training process is demonstrated in Figure 4.

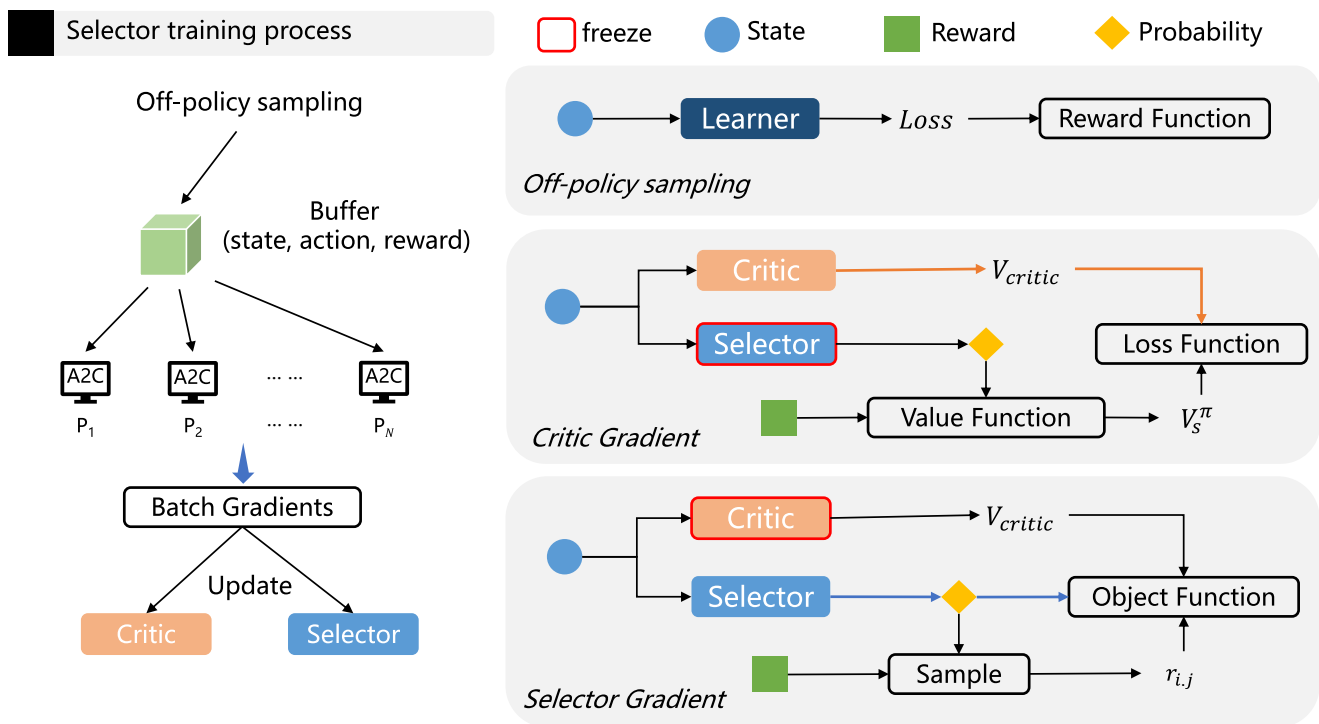


Figure 4. Selector training process based on asynchronous advantage actor-critic (A3C) framework.

Selector Training

The Selector Module consists of a Selector (3-layer MLP) and a Critic (2-layer MLP), whose structures are shown in Section 4. 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 [42] 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.

Algorithm 2 Auto-Selector Training: A3C—pseudocode.

Require: the Learner module, parking-related features

- 1: 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
 - 2: initialize global Selector and Critic net π_a and π_c ; model state s with features
 - 3: **for** episode **do**
 - 4: sampling distribution $\pi' = \pi$
 - 5: compute rewards
 - 6: sample local states, actions and rewards to the Buffer
 - 7: distribute samples (s, a, r) to corresponding member
 - 8: **for** each member m **do**
 - 9: freeze π_a
 - 10: $d\pi_c \leftarrow 0$
 - 11: compute $V_{\pi}^{s,m}$
 - 12: obtain $d\pi_c^m$ by regression
 - 13: **end for**
 - 14: update π_c , release π_a
 - 15: **for** each member m **do**
 - 16: freeze π_c
 - 17: $d\pi_a \leftarrow 0$
 - 18: obtain $d\pi_a^m$ by advantage function
 - 19: **end for**
 - 20: update π_a , release π_c
 - 21: **end for**
-

- 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):

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

where $E_{\pi}(r(\tilde{a}, \hat{s}))$ represents the expected value of the reward r for all actions \tilde{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.

We calculate the Critic gradient of state s_m through 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 Equation (7).

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

- Selector Updating

According to the idea of policy-based advantage function [20], when the reward of an action is greater than the valuation of Critic V_{critic} , 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).

$$\nabla \bar{R}_{\pi}^s = \frac{1}{T} \sum_{t=1}^T (r(a_t, \hat{s}) - V_{critic}) \nabla_{\mathbb{B}_a} \log \mathfrak{a}_{\mathbb{B}}(a_t|\hat{s}). \tag{8}$$

Similar to the updating process of Critic net, that of Selector net π_a also can leverage the aggregated local gradients as illustrated in Equation (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 selects a certain number of good “guiders” for the Learner module.

4. Performance Evaluation

In this section, the proposed method will be tested together with other state-of-the-art POP methods based on the same dataset and evaluation metrics. Moreover, the results will be analyzed to demonstrate the improvements achieved.

4.1. Evaluation Preparation

To conduct a fair comparison, we adopt the data from reliable sources, introduce several representative approaches as baselines, and set reasonable experimental conditions.

4.1.1. Data Declaration

A dataset with a minimum resolution of five minutes was collected from 42 parking lots located in two Chinese first-tier cities, Shenzhen and Guangzhou, from 1 to 30 June 2018. It is of high quality, excludes the impact of COVID-19 (earlier than 2019), and is capable of evaluation purposes. As shown by Figure 5, 30 car parks in Guangzhou (P1–P30) are set as Train tasks, while 12 car parks in Shenzhen (Target1–Target12) are Target (Test) tasks. Further, these parks are classified into six types according to the land use attributes of the area in which they are located [43]: Commercial, Hospital, Office, Residential, Recreational, and Tourism. Another important parking-related feature is the density of points of interest (POI). We calculate the density of POI using a kernel function [44] presented in (10).

$$\begin{aligned} \eta(x) &= \frac{1}{U} \sum_{u=1}^U \frac{1}{h} K\left(\frac{x - x_u}{h}\right) \\ \text{when } K(x) &\geq 0, \int K(x)dx = 1, \\ \int xK(x)dx &= 0, \int x^2K(x)dx > 0, \end{aligned} \tag{10}$$

where $\eta(x)$ is the density function of POI samples x , $K(x)$ is the kernel function adopting Gaussian distribution, h bandwidth represents the maximum acceptable walking distance in cities, and U is the total number of POI samples. Considering that the influences of different POIs are challenging to measure in the real world, we propose to simplify the calculation by setting all weights to 1, as the overestimation and underestimation errors can be partially offset in calculation.

Furthermore, the kernel density of POI and the type of parking lots are utilized to build the input of the Selector module (i.e., the “State” of reinforcement learning). The “State” matrix can be formed in the way presented in (11):

$$\begin{aligned} S_i^{2 \times N} &= \left[\eta_i^{1 \times N}, \tau_i^{1 \times N} \right] \\ \text{s.t. } \eta_i^{1 \times N} &= \lambda - (\eta_i - \bar{\eta})^2 \\ \tau_i^{1 \times N} &= \bar{\tau}_i * \Gamma, \end{aligned} \tag{11}$$

where S_i represents the state matrix of federation member i , who has a dimension of $2 \times N$, $\eta_i^{1 \times N}$ and $\tau_i^{1 \times N}$ indicate the density and type feature vectors of state respectively, η_i and $\bar{\tau}_i$ are the density scalar and type code of member i , $\bar{\eta}$ and Γ represents the density vector

and type matrix of the federation, N is the total number of members, λ denotes a positive constant, and $*$ denotes the Hadamard product.

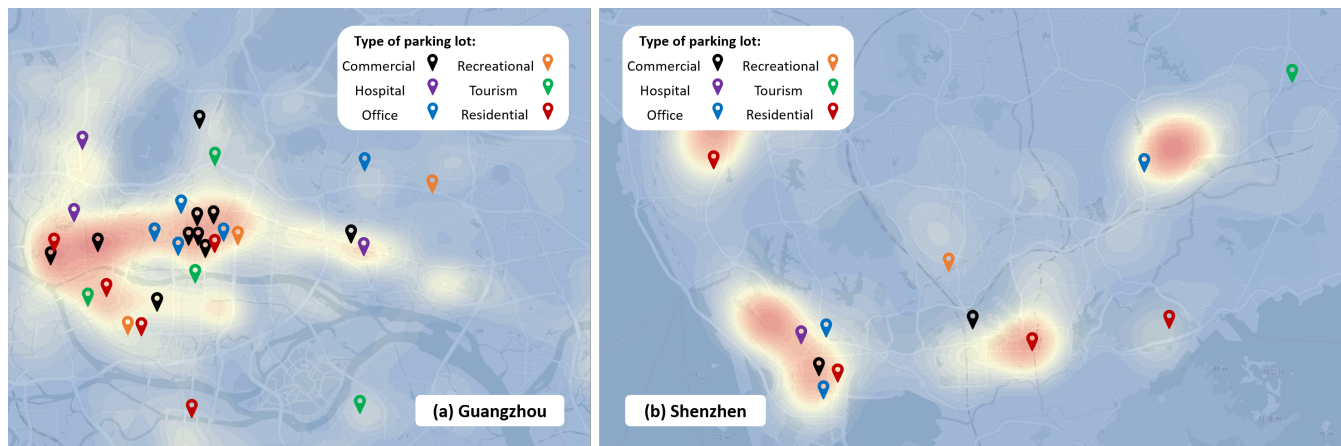


Figure 5. Spatial and category information of parking lots, (a) Train tasks; (b) Target tasks.

To reach the requirement of meta-learning, the Train and Target tasks are further divided into four parts by time. Each task in the Train set is sliced into two parts, Train-Support set (Day 1–18) and Train-Query set (Day 19–24), for obtaining meta gradients. Similarly, each task in the Test set is sliced into two parts, Test-Support set and Test-Query set. The Test-Query set is from 25 to 30 June and is used for performance evaluation. As for the Test-Support set, to test the proposed method with different target sample sizes, five different situations are designed as illustrated in Table 2: 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).

Table 2. Five categories of data richness for Test-Support set.

Data Richness	Complete	Partial	Small	Few	Empty
Date (June)	1–24	19–24	22–24	24	–
Number of days	24	6	3	1	0
Proportion (%)	100	25	12.5	4.2	0

4.1.2. Evaluation Metrics

Four common evaluation metrics are chosen to evaluate the compared models, namely Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), and Coefficient of Determination (R^2), which can be calculated as (12).

$$\begin{aligned}
 MAPE &= \frac{1}{N} \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{y_i} \\
 RMSE &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \\
 RAE &= \frac{\sum_{i=1}^N |\hat{y}_i - y_i|}{\sum_{i=1}^N |y_i - \bar{y}_i|} \\
 R^2 &= 1 - \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{\sum_{i=1}^N (y_i - \bar{y}_i)^2}
 \end{aligned}
 \tag{12}$$

where y_i and \hat{y}_i represent the actual value and the predicted value at time i ; and N is the total number of samples. Moreover, each prediction task will run ten times separately, and the averaged result is used as the final performance indicator to reduce random error.

4.1.3. Baselines and Competitive Approaches

In order to evaluate the effectiveness of the proposed method, besides several typical prediction models, we select two knowledge transfer methods as baselines. Specifically, the typical prediction models include four neural networks (NNs) and one statistical model.

- Fully Connected Neural Network (FCNN): a network widely used for function approximation and general regression problems, but it cannot distinguish between temporal and spatial features;
- Long Short-Term Memory (LSTM) [45]: a recurrent neural network that is widely used in many time series prediction tasks due to its excellent temporal feature extraction capabilities;
- Gated Recurrent Unit (GRU) [46]: a recurrent neural network with a simplified gate structure and the advantage of fast computation;
- Bi-directional Long Short-Term Memory (BiLSTM) [47]: a combination of forward and backward LSTM, often used for modeling contextual information in natural language processing tasks;
- Auto-regressive Integrated Moving Average (ARIMA) [48]: a classical statistical model for time series forecasting, which requires the series or its differentials must be stable.

The knowledge transfer approaches include two recently developed methods and two proposed methods:

- Transfer-LSTM [49]: a traffic flow prediction method using traditional transfer learning, which can also be used for occupancy prediction as it is also time series data;
- FADACS [18], a recently developed approach for parking occupancy prediction, which implements domain adaptation using adversarial learning mechanisms;
- ALL: the proposed integration of the meta-learner and auto-selector employs LSTM and GRU as a backbone network, and the models are named ALL-LSTM and ALL-GRU, respectively;
- FML: the meta-learner uses LSTM and GRU as the backbone network like ALL. Unlike ALL, its “guiders” selection strategy is “random”, as a comparison to highlight the effectiveness of our selector module.

In Table 3, the architecture of the proposed method is described. Relu [50] is the activation function that makes gradient descent and backpropagation more efficient; Res [51] denotes the residual connection, which is used to preserve the properties of the upper network layers and make the network easier to be updated. Softmax [52] is a normalized exponential function used to transform the output of the selector into a probability for each action; Dense refers to the linear neural network layer; Flatten refers to the operation of flattening a matrix into a one-dimensional vector.

Table 3. Architecture of the proposed approach.

ALL	Layer	Parameters
Feature Extractor	LSTM	input = 1 channels hidden = 8 channels output = 1 channels Sequence length = 6
	Dense	
Selector (Actor)	Dense	(30, 256)
	Relu	
	Dense	(256, 256)
	Res	
	Dense	(256, 4060)
Critic	Softmax	
	Dense	(5, 1)
	Flatten	
	Dense	(30, 1)

In Table 4, the running configurations of all comparison methods are shown, where CE stands for cross-entropy and is the loss function for selector training. All neural networks use a two-layer structure, with an encoder layer (e.g., BiLSTM, LSTM, GRU) for extracting time-series features and a decoder layer (FCNN) for outputting predicted values. Besides, the source selection strategies of the knowledge transfer methods are pretty different, and the ‘knowledge’ extracted by these methods is also different. Transfer-LSTM obtains the network structure and parameters, while FADACS generates a new network based on the shared patterns of the source and target domains. Unlike them, our proposed learner learns gradients from multiple domains to update a bespoke model, which makes the knowledge more insightful and the model more adaptive.

Table 4. Running configurations.

Model	Param	Value	Comment
*	input	(256, 6, 1)	(batch, sequence length, feature)
	output	(256, 1)	(batch, feature)
	Structure	Encoder-Decoder	two layer network structure
	Learning Rate	0.02	For all updating processes
	Max Epochs	(200, 400, 4000)	(fine-tuning, pre-training, Selector training)
	Optimizer	(Adam, BGD, Adam)	(fine-tuning, pre-training, Selector training)
	Loss Function	(MSE, MSE, CE)	(fine-tuning, pre-training, Selector training)
Knowledge Transfer	Selection Strategy	(random type)	(Transfer-LSTM, FML FADACS, ALL)
	Source Number	(1 3)	(Transfer-LSTM, FADACS FML, ALL)
	(N, M, T)	(30, 3, 4060)	the number of states, guiders, and actions
ARIMA	(p, d, q)	(2, 1, 1)	implemented on Statsmodels

* includes ALL, FML, FADACS, Transfer-LSTM, NNs.

4.1.4. Running Environment

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.

It is worth noting that, for reproductivity, the dataset and code used by this paper are shared in Github, and are downloadable from the link (<https://github.com/Quhaoh233/ALL> (accessed on 7 May 2022)).

4.2. Evaluation Results and Discussions

The performance of evaluated methods is analyzed in three aspects, namely: (1) the forecasting error to illustrate how well and scalable the model is to predict the future in different data volumes; (2) the convergence speed to demonstrate how fast the model is to be stabilized; and (3) the variance of errors to show how stable the model is to handle contexts with various parking lots and data volumes. Noted that the proposed approach ALL is a pre-training framework to empower the learning model to handle small-sample prediction issues, which is not in conflict with data enhancement. In this evaluation, we exclude the process of data enhancement to conduct a fair comparison between the compared models with and without knowledge transfer.

4.2.1. Forecasting Error

As shown in Table 5, the evaluation metrics of compared models in different data bulks (empty, few, small, partial, and complete data) are summarized. From an overall point of view, we can see the proposed approach reduces the prediction errors significantly with the highest scores in all the four evaluation metrics, namely RMSE 0.0385, MAPE 6.82%, R2 94.49%, and RAE 17.23%. Precisely, in RMSE, MAPE, R2 and RAE, (1) ALL improves accuracy over LSTM by 30.2%, 25.1%, 30.3%, 21.2%; (2) ALL outperforms FADACS with extra accuracy improvement of 22.4%, 18.1%, 28.0%, 15.7%; and (3) On the basis of FML-LSTM, the Selector module brings 4.5%, 3.7%, 23.0%, 3.5% extra improvement.

Table 5. The list of evaluation metrics of compared models in different data bulks.

Data Bulk	Empty				Few				Small				
	Metric (10^{-2})	RMSE	MAPE	R2	RAE	RMSE	MAPE	R2	RAE	RMSE	MAPE	R2	RAE
ALL-LSTM	5.40	10.34	89.81	24.08	4.52	8.18	93.25	19.94	3.78	6.32	94.75	16.55	
ALL-GRU	5.56	9.65	89.09	25.23	5.13	8.99	92.26	21.70	3.83	6.48	94.49	16.96	
FML-LSTM	5.61	10.24	90.71	22.82	5.53	9.14	91.52	22.22	4.74	7.70	92.60	19.81	
FML-GRU	6.04	11.46	88.69	24.75	5.53	9.15	91.16	22.54	4.91	8.00	92.75	19.94	
FADACS	7.72	16.32	76.81	35.63	5.76	9.66	90.79	23.22	5.00	8.11	92.49	20.12	
Transfer-LSTM	9.68	23.75	68.25	40.36	6.38	10.92	88.56	26.25	4.96	8.29	92.17	21.12	
LSTM	—	—	—	—	7.49	13.67	86.71	27.56	5.67	8.87	91.96	21.86	
BiLSTM	—	—	—	—	6.77	11.72	86.01	29.23	5.84	9.41	90.13	23.77	
GRU	—	—	—	—	9.01	15.36	69.47	39.28	5.30	9.34	87.93	26.09	
FCNN	—	—	—	—	9.38	18.04	59.37	44.54	5.99	10.19	85.29	27.58	
ARIMA	—	—	—	—	—	—	—	—	50.99	92.27	−1223.7	318.8	

Data Bulk	Partial				Complete				Average				
	Metric (10^{-2})	RMSE	MAPE	R2	RAE	RMSE	MAPE	R2	RAE	RMSE	MAPE	R2	RAE
ALL-LSTM	3.56	6.35	94.96	16.18	3.55	6.41	94.99	16.24	3.85	6.82	94.49	17.23	
ALL-GRU	3.61	6.45	94.77	16.45	3.55	6.41	94.95	16.32	4.03	7.08	94.12	17.86	
FML-LSTM	4.41	7.41	93.24	18.96	4.05	6.90	93.92	17.62	4.68	7.79	92.82	19.65	
FML-GRU	4.49	7.56	92.96	19.45	4.52	7.63	92.81	19.50	4.86	8.09	92.42	20.36	
FADACS	4.62	7.82	93.01	19.41	4.55	7.75	93.11	19.01	4.98	8.33	92.35	20.44	
Transfer-LSTM	5.49	8.05	92.87	20.14	5.51	7.78	92.85	19.95	5.59	8.76	91.61	21.86	
LSTM	5.04	7.40	93.27	19.53	4.38	6.51	94.52	16.50	5.64	9.11	91.61	21.36	
BiLSTM	5.25	7.82	93.43	19.75	5.19	8.60	91.77	21.30	5.76	9.39	90.34	23.51	
GRU	5.90	10.00	88.35	25.61	5.78	9.68	88.69	24.73	6.50	11.10	83.61	28.93	
FCNN	5.51	9.43	87.25	25.59	5.55	9.30	86.45	25.73	6.61	11.74	79.59	30.86	
ARIMA	4.62	7.03	93.75	17.69	4.63	7.02	93.74	17.70	20.08	35.44	−345.4	118.1	

Notes: “—” indicates that the result is not available; The “Average” results exclude the “Empty” cases.

In general, the methods with knowledge transfer are superior to those without due to the more information they can obtain. However, Transfer-LSTM is inferior to its backbone LSTM in the Partial and Full data cases, which reveals that the performance can be held back by the distraction from the negative transfer.

Focusing on the few-data cases, we can see that the statistical model ARIMA cannot fit the curve, which indicates the parking occupancy variation in few-data cases is unstable. These unstable samples bring the challenges of local adaptation, especially over-fitting. In this context, the deep learning methods perform poorly, e.g., GRU and FCNN have more than 30% residuals in the R2 score. By contrast, the R2 scores of the methods that utilize transfer techniques to learn prior knowledge remain over 90%.

Furthermore, to accentuate the difference in knowledge transfer methods, they are deployed on the empty-data cases. The results show that ALL can provide a reliable future parking occupancy prediction (over 88% in R2 score) even without any local fine-tuning. On the contrary, FADACS and Transfer-LSTM only get 76.81% and 68.25%, which still have lapses of knowledge adaptation.

4.2.2. Convergence Profile

The convergence profile of the ALL-LSTM is evaluated against the LSTM as shown in Figure 6, where the left vertical axis is for the orange line (i.e., ALL-LSTM) and the right one is for the blue line (i.e., LSTM). These experiments are conducted in Target1–Target4 and the few-data scenarios. The figures show that the LSTM model deployed with the ALL framework has a better prospect for loss decline than the not-deployed model. Furthermore, the errors at the beginning of the curves for ALL-LSTM are smaller than that at the 100th iteration of LSTM, indicating the advantage of ALL for knowledge extraction and propagation. As an illustrative example, although their RMSE curves in Target1 perform similar shapes, the curve of ALL-LSTM is an order of magnitude less than that of LSTM.

The former starts below 0.04 and ends at 0.032, while the latter starts over 0.5 and ends at 0.1. All these results suggest that employing the Learner module, which efficiently learns numerical valuable prior knowledge through a pre-training process, can significantly improve local adaptation and convergence speed than the non-knowledge-transfer model.

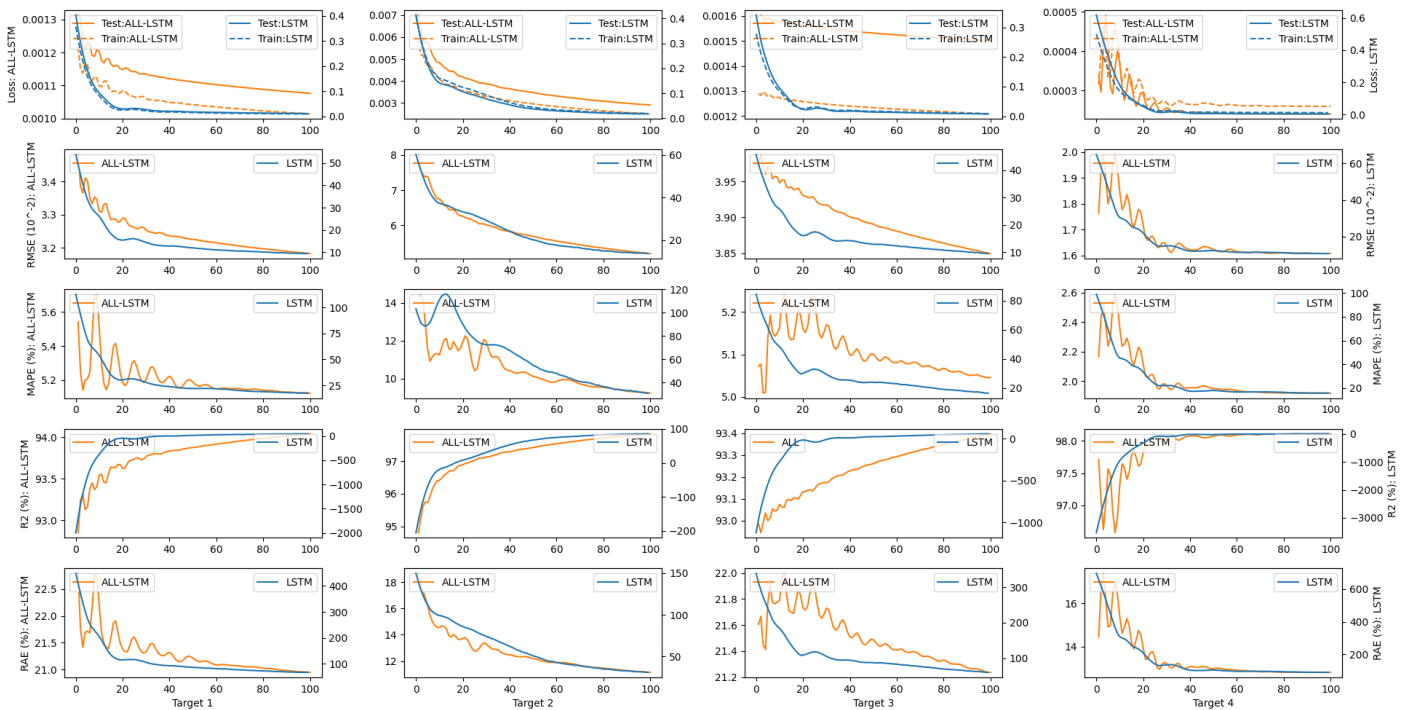


Figure 6. Convergence profiles in Targets 1–4 and few-data scenarios. The left vertical axis is for ALL-LSTM, when the right is for LSTM.

4.2.3. Error Variation

To emphasize the improvement brought by the Selector module, we give an illustration in Figure 7, where the orange, green and red shapes represent the maximum, average, and minimum MAPE of the methods in the five data richness cases. The figure shows that in all five cases, the LSTM model using the ALL framework yields smaller “boxes” than the model using FML and Transfer. To be precise, the average variance for ALL-LSTM in MAPE is 9.85×10^{-4} , which is about 17.8% less than the 16.11×10^{-4} for FML. The above results demonstrate that the Selector module can stabilize prediction, improving the application significance of the proposed approach in real-world scenarios.

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 29.8% in prediction error; (2) fast adaptation, with model adaptation and convergence speed substantially improved by 10^2 times over the model without ALL; and (3) good stability, reducing the variance of the predictions by 17.8%.

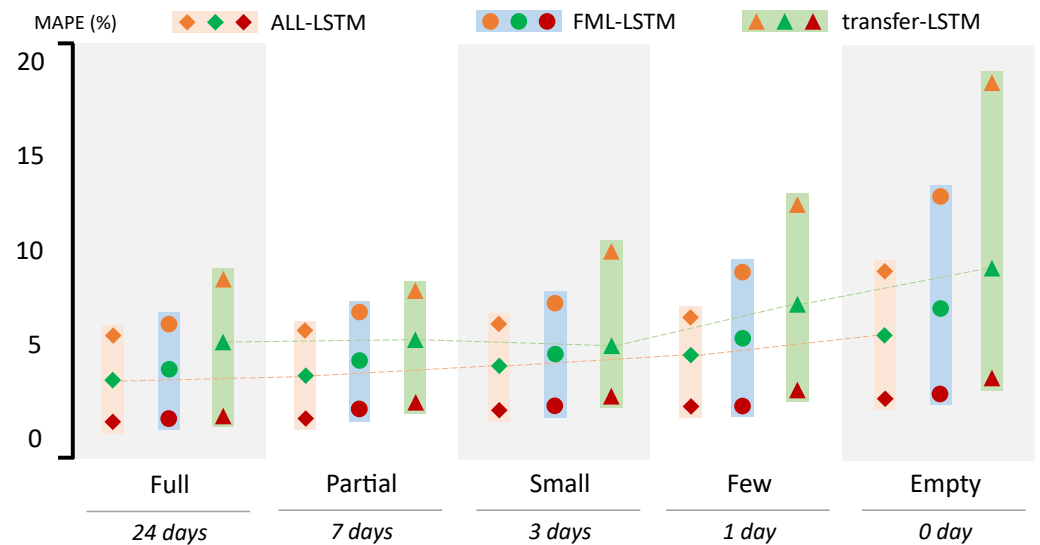


Figure 7. Result variance comparison among ALL-LSTM, FML-LSTM, Transfer-LSTM per data size.

5. Conclusions and Future Works

Parking occupancy prediction plays an important role in many real-time management scenarios, such as directing car drivers to available parking lots, adjusting parking charges, and sharing parking spaces, which can significantly improve the “smartness” of intelligent transportation systems (ITS). However, an emerging bottleneck is the poor predictive performance in data-poor parking areas, which hinders the mass acceptance of intelligent parking services associated with POP. Hence, this paper proposes a knowledge transfer approach to support small-sample parking occupancy prediction, which integrates two novel ideas: (1) adaptation: by leveraging the Asynchronous Advantage Actor-Critic (A3C) reinforcement learning technique, an auto-selector module is implemented, which can group and select data-scarce parks automatically as supporting sources to enable the knowledge adaptation in model training; and (2) learning to learn: by applying federated meta-learning on selected supporting sources, a meta-learner module is designed, which can train a high-performance local prediction model in a collaborative and privacy-preserving manner.

As the evaluation results show, the proposed method outperforms the compared methods in three respects, namely: (1) by using the ALL framework, the prediction error can be significantly reduced by approximately 29.8%; (2) by applying FedFOMAML during the model pre-training process, the convergence and adaptation speed of the model can be improved so that the loss curves of LSTM models with and without ALL maintain a 10^2 difference; and (3) by applying the Selector module, the variance can be mediated, resulting in a 17.8% improvement in the stability of the model when dealing with different data volumes.

Future work can be conducted in three directions:

- (1) To optimize source selection: In the proposed approach, data augmentation and structure optimization are two effective ways to improve source selection, which facilitates the ‘purification’ of knowledge [53]. Firstly, the structure of Selector will be replaced by other state-of-the-art structures. Secondly, a larger dataset will be used to train the Selector network and provide a more general decision strategy;
- (2) To consider data security: The knowledge transfer process incurs huge communication overhead, security issues, or privacy concerns, which is not practical for wireless networks and end-users [54]. Our federated learning framework will be revised to provide effective personalized models for each participant under device heterogeneity while ensuring differential privacy of their data solutions;

- (3) To extend ALL's applications: The ideas of adaptation and learning to learn can be applied not only to solve the small sample problem of parking occupancy prediction but also to handle other time series prediction tasks with insufficient data. We will apply our approach to other scenarios where up-to-date data are available, such as traffic flow forecasting.

Author Contributions: Conceptualization, H.Q. and J.L.; methodology, H.Q.; software, H.Q.; validation, H.Q., S.L. and J.L.; formal analysis, S.L.; writing—original draft preparation, H.Q.; writing—review and editing, J.L., Y.Z., R.L.; visualization, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (62002398).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in Github at <https://github.com/Quhaoh233/ALL> (accessed on 7 May 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Zhu, Y.; Ye, X.; Chen, J.; Yan, X.; Wang, T. Impact of Cruising for Parking on Travel Time of Traffic Flow. *Sustainability* **2020**, *12*, 3079. [CrossRef]
- Carrillo, M.; Álvarez, P.; Risso, N.; Baeza, E.; Salgado, F. Haul vehicle fuel and GHG emissions estimation using GPS data. In Proceedings of the 2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Virtual Event, 6–9 December 2021; pp. 1–7. [CrossRef]
- You, L.; Tunçer, B.; Zhu, R.; Xing, H.; Yuen, C. A Synergetic Orchestration of Objects, Data, and Services to Enable Smart Cities. *IEEE Internet Things J.* **2019**, *6*, 10496–10507. [CrossRef]
- Kelemen, M.; Polishchuk, V.; Gavurova, B.; Rozenberg, R.; Bartok, J.; Gaal, L.; Gera, M.; Kelemen, M., Jr. Model of Evaluation and Selection of Expert Group Members for Smart Cities, Green Transportation and Mobility: From Safe Times to Pandemic Times. *Mathematics* **2021**, *9*, 1287. [CrossRef]
- 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. *Phys. A-Stat. Mech. Its Appl.* **2022**, *587*. [CrossRef]
- Lin, X.; Yuan, P. A dynamic parking charge optimal control model under perspective of commuters' evolutionary game behavior. *Phys. A Stat. Mech. Its Appl.* **2018**, *490*, 1096–1110. [CrossRef]
- Liu, J.; Wu, J.; Sun, L. Control method of urban intelligent parking guidance system based on Internet of Things. *Comput. Commun.* **2020**, *153*, 279–285. [CrossRef]
- Zou, W.; Sun, Y.; Zhou, Y.; Lu, Q.; Nie, Y.; Sun, T.; Peng, L. Limited Sensing and Deep Data Mining: A New Exploration of Developing City-Wide Parking Guidance Systems. *IEEE Intell. Transp. Syst. Mag.* **2022**, *14*, 198–215. [CrossRef]
- Zhang, F.; Liu, W.; Wang, X.; Yang, H. Parking sharing problem with spatially distributed parking supplies. *Transp. Res. Part C Emerg. Technol.* **2020**, *117*. [CrossRef]
- He, J.; Wang, W.; Huang, M.; Wang, S.; Guan, X. Bayesian Inference under Small Sample Sizes Using General Noninformative Priors. *Mathematics* **2021**, *9*, 2810. [CrossRef]
- Al-Kaabi, R.; Ali, H.; Ahmed, S.; Ahmed, K. Smart parking: An investigation of users' satisfaction in the Kingdom of Bahrain. *Int. J. Serv. Technol. Manag.* **2021**, *27*, 337–350. [CrossRef]
- Pozo, R.F.; Gonzalez, A.B.R.; Wilby, M.R.; Diaz, J.J.V.; Matesanz, M.V. Prediction of On-Street Parking Level of Service Based on Random Undersampling Decision Trees. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–10. [CrossRef]
- Sun, Y.; Peng, L.; Li, H.; Sun, M. Exploration on Spatiotemporal Data Repairing of Parking Lots Based on Recurrent GANs. In Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 467–472. [CrossRef]
- Provoost, J.C.; Kamilaris, A.; Wismans, L.J.J.; van der Drift, S.J.; van Keulen, M. Predicting parking occupancy via machine learning in the web of things. *Internet Things* **2020**, *12*. [CrossRef]
- 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.* **2019**, *107*, 248–265. [CrossRef]
- Xiao, X.; Jin, Z.; Hui, Y.; Xu, Y.; Shao, W. Hybrid Spatial-Temporal Graph Convolutional Networks for On-Street Parking Availability Prediction. *Remote Sens.* **2021**, *13*, 3338. [CrossRef]
- Wang, L.; Geng, X.; Ma, X.; Liu, F.; Yang, Q. Cross-City Transfer Learning for Deep Spatio-Temporal Prediction. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Macao, China, 10 August 2019.

18. Shao, W.; Zhao, S.; Zhang, Z.; Wang, S.; Rahaman, M.S.; Song, A.; Salim, F.D. FADACS: A Few-Shot Adversarial Domain Adaptation Architecture for Context-Aware Parking Availability Sensing. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom), Pisa, Italy, 21–25 March 2021; pp. 1–10. [[CrossRef](#)]
19. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [[CrossRef](#)]
20. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.P.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016.
21. Jiang, Y.; J, K.; Rush, K.; S, K. Improving Federated Learning Personalization via Model Agnostic Meta Learning. *arXiv* **2019**, arXiv:1909.12488.
22. Gui, L.; Xu, R.; Lu, Q.; Du, J.; Zhou, Y. Negative transfer detection in transductive transfer learning. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 185–197. [[CrossRef](#)]
23. Li, J.; Lu, K.; Huang, Z.; Zhu, L.; Shen, H.T. Transfer Independently Together: A Generalized Framework for Domain Adaptation. *IEEE Trans. Cybern.* **2019**, *49*, 2144–2155. [[CrossRef](#)]
24. Li, J.; Jing, M.; Su, H.; Lu, K.; Zhu, L.; Shen, H.T. Faster Domain Adaptation Networks. *IEEE Trans. Knowl. Data Eng.* **2021**. [[CrossRef](#)]
25. Pankiv, Y.; Kunanets, N.; Artemenko, O.; Veretennikova, N.; Nebesnyi, R. Project of an Intelligent Recommender System for Parking Vehicles in Smart Cities. In Proceedings of the 16th IEEE International Conference on Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 22–25 September 2021; Volume 2, pp. 419–422. [[CrossRef](#)]
26. Balmer, M.; Weibel, R.; Huang, H. Value of incorporating geospatial information into the prediction of on-street parking occupancy—A case study. *Geo-Spat. Inf. Sci.* **2021**, *24*, 438–457. [[CrossRef](#)]
27. Agrawal, R.; Kothari, P. CoPASample: A Heuristics Based Covariance Preserving Data Augmentation. In *Machine Learning, Optimization, and Data Science*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11943, pp. 308–320. [[CrossRef](#)]
28. Chen, X.; He, Z.; Chen, Y.; Lu, Y.; Wang, J. Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model. *Transp. Res. Part C Emerg. Technol.* **2019**, *104*, 66–77. [[CrossRef](#)]
29. Cai, J.; Luo, J.W.; Wang, S.L.; Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing* **2018**, *300*, 70–79. [[CrossRef](#)]
30. Wu, X.; Ding, S.; Chen, W.; Wang, J.; Chen, P.C.Y. Short-term urban traffic flow prediction using deep spatio-temporal residual networks. In Proceedings of the 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, China, 31 May–2 June 2018; pp. 1073–1078. [[CrossRef](#)]
31. Zhao, D.; Ju, C.; Zhu, G.; Ning, J.; Luo, D.; Zhang, D.; Ma, H. MePark: Using Meters as Sensors for Citywide On-Street Parking Availability Prediction. *IEEE Trans. Intell. Transp. Syst.* **2021**. [[CrossRef](#)]
32. Yang, Y.; Zhou, D.W.; Zhan, D.C.; Xiong, H.; Jiang, Y. Adaptive Deep Models for Incremental Learning: Considering Capacity Scalability and Sustainability. In Proceedings of the KDD'19: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2019; pp. 74–82. [[CrossRef](#)]
33. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of the 34th IEEE International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1126–1135.
34. Xu, Z.; Chen, X.; Tang, W.; Lai, J.; Cao, L. Meta weight learning via model-agnostic meta-learning. *Neurocomputing* **2021**, *432*, 124–132. [[CrossRef](#)]
35. Chen, Y.; Hoffman, M.W.; Colmenarejo, S.G.; Denil, M.; Lillicrap, T.P.; Botvinick, M.; de Freitas, N. Learning to Learn without Gradient Descent by Gradient Descent. In Proceedings of the 34th IEEE International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 70.
36. Xu, J.; Wang, H. Client Selection and Bandwidth Allocation in Wireless Federated Learning Networks: A Long-Term Perspective. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 1188–1200. [[CrossRef](#)]
37. Tenkanen, H.; Toivonen, T. Longitudinal spatial dataset on travel times and distances by different travel modes in Helsinki Region. *Sci. Data* **2020**, *7*. [[CrossRef](#)]
38. Liu, S.; Chen, Q.; You, L. Fed2A: Federated Learning Mechanism in Asynchronous and Adaptive Modes. *Electronics* **2022**, *11*, 1393. [[CrossRef](#)]
39. Lin, S.; Yang, G.; Zhang, J. A Collaborative Learning Framework via Federated Meta-Learning. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 8–10 July 2020; pp. 289–299. [[CrossRef](#)]
40. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. In Proceedings of the Conference and Workshop on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 5–10 December 2013.
41. Mazouchi, M.; Naghibi-Sistani, M.B.; Sani, S.K.H. A novel distributed optimal adaptive control algorithm for nonlinear multi-agent differential graphical games. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 331–341. [[CrossRef](#)]
42. Luo, B.; Wu, H.N.; Huang, T. Off-Policy Reinforcement Learning for H-infinity Control Design. *IEEE Trans. Cybern.* **2015**, *45*, 65–76. [[CrossRef](#)]
43. Zhao, Z.; Zhang, Y. A Comparative Study of Parking Occupancy Prediction Methods considering Parking Type and Parking Scale. *J. Adv. Transp.* **2020**, *2020*. [[CrossRef](#)]

44. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [[CrossRef](#)]
45. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
46. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734. [[CrossRef](#)]
47. Schuster, M.; Paliwal, K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [[CrossRef](#)]
48. Jaynes, E.T. On the rationale of maximum-entropy methods. *Proc. IEEE* **1982**, *70*, 939–952. [[CrossRef](#)]
49. 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 Proceedings of the 23rd IEEE International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–6. [[CrossRef](#)]
50. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair. In Proceedings of the International Conference on International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010.
51. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016; pp. 770–778. [[CrossRef](#)]
52. Dukhan, M.; Ablavatski, A. Two-Pass Softmax Algorithm. In Proceedings of the 34th IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Portland, OR, USA, 30 May–3 June 2020; pp. 386–395. [[CrossRef](#)]
53. Yang, M.; Qian, H.; Wang, X.; Zhou, Y.; Zhu, H. Client Selection for Federated Learning with Label Noise. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2193–2197. [[CrossRef](#)]
54. Hu, R.; Guo, Y.; Li, H.; Pei, Q.; Gong, Y. Personalized Federated Learning with Differential Privacy. *IEEE Internet Things J.* **2020**, *7*, 9530–9539. [[CrossRef](#)]