2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Metaverse)

TWAFR-GRU: An Integrated Model for Real-time Charging Station Occupancy Prediction

Qiyang Chen[a], Sheng Liu[a], Haohao Qu[a], Rui Zhu[b], Linlin You[a*]
[a] School of Intelligent Systems Engineering, Sun Yat-Sen University, Shenzhen, China
[b] Systems Science Department, Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore
{chenqy87,liush235,quhaoh}@mail2.sysu.edu.cn, 911zrfelix@gmail.com, youllin@mail.sysu.edu.cn

***Abstract*—The fast-growing number of electric vehicles has resulted in a variety of charging-related issues, such as the shortage of public charging piles, the inconvenience of finding available charging piles, the additional traffic congestion caused by unnecessary cruising to search for available charging piles, etc. Since only constructing new charging piles may not significantly improve the charging efficiency, methods of predicting charging station occupancy are widely discussed as an effective means to address these issues. However, several challenges are encountered in training an efficient and effective model by utilizing distributed data, improving convergence speed, and enhancing model generalization ability. To address these issues, this paper proposes a novel mechanism, named TWAFR-GRU, which integrates Temporally Weighted Asynchronous Federated Learning (TWAFL) with Reptile and Gated Recurrent Unit (GRU). As shown by the holistic evaluation based on the charging station occupancy dataset, compared with other state-of-the-art baselines, TWAFR-GRU can 1) decrease MAE, RMSE and RAE by 19%, 15% and 17% separately and improve R^2 by 67%; 2) cut rounds for the model to converge by 75%; 3) save training time by 44%; and 4) reduce 17% in forecasting error after the personalization of the initial model to serve a specific charging station.**

***Index Terms*—Charging Station Occupancy Prediction, Asynchronous Federated Learning, Meta-learning, Gated Recurrent Unit**

I. Introduction

Electric vehicles (EVs) have become a popular product in the automobile market in recent years. Due to the popularity of low-carbon living and high oil prices, more and more people opt to purchase EVs. In this context, charging-related issues have emerged, e.g., insufficient charging piles, the difficulty in finding available charging piles, the congested traffic caused by cruising to search for idle charging piles, etc. Since the number of EVs will inevitably continue to increase, building new charging piles will not be a sustainable solution. Therefore, the method of forecasting the charging station occupancy has been widely discussed [1], which can effectively guide EVs to the nearest idle charging piles, and in return, alleviate related traffic congestion.

Given that the average travel time in the city ranges from 15 minutes to 60 minutes, the real-time charging station occupancy prediction is more useful than the long-term charging station occupancy prediction in directing EVs to the closest idle charging piles and easing traffic congestion [2]. Under this circumstance, there are three challenges in the prediction of charging station occupancy, namely: 1) the data of the charging stations are stored in their own local databases, making it difficult to centralize the data from all charging stations for training; 2) the prediction model has low convergence speed and poor generalization ability; 3) the synchronous distributed training mode is inefficient, which may cause a waste of massive computing and communication resources dispersed at the edge.

Several strategies have been put forth to address the emerging challenges, including 1) applying federated learning to allow more decentralized data to participate in model training [3]; 2) adopting meta-learning to generate an initial model with strong generalization ability, which enables the model to converge fast on different tasks [4]; and 3) increasing training efficiency by applying asynchronous training mode [5], which can flexibly utilize computing and communication resources of the clients. However, an integrated mechanism, which can efficiently generate a real-time charging station occupancy prediction model with strong generalization ability and high convergence speed in the case of distributed data, is still missing.

To fill this gap, we propose an integrated model for real-time charging station occupancy prediction, namely TWAFR-GRU, which applies 1) Gated Recurrent Unit (GRU) as the backbone of the prediction model for charging station occupancy; 2) Reptile, a first-order meta-learning model that enables the initial prediction model to have strong generalization capability; 3) Temporally Weighted Asynchronous Federated Learning (TWAFL), an asynchronous federated learning model, which allows more distributed data to be used jointly for model training with improved training performance in distributed learning scenarios.

In general, the main contributions of this paper can be summarized as follows:

- Based on Reptile, and TWAFL, an asynchronous federated meta-learning mechanism, called TWAFR, is designed, and as a general distributed learning framework, it can improve the model convergence speed and generalization ability to be cost-efficient.
- Based on GRU and TWAFR, an integrated model for charging station occupancy prediction, called TWAFR-

*Corresponding author: Linlin You, e-mail: youllin@mail.sysu.edu.cn

DOI 10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00233

GRU, is proposed, which can achieve a high prediction accuracy and training efficiency by providing a personalized model for each charging station;
- TWAFR-GRU is evaluated based on a charging station occupancy dataset and compared with other state-of-the-art baselines. As a result, the proposed model achieves the following improvements, namely 1) a reduction of 19%, 15% and 17% in MAE, RMSE, and RAE separately and an improvement of 67% in R^2; 2) a boost of 75% in convergence speed; 3) an acceleration of 44% in training time, and 4) a decrease of 17% in forecasting error after the personalization of the initial model.

The remainder of this paper is organized as follows. First, section II summarizes the related work about charging station occupancy prediction, asynchronous federated learning and federated meta-learning. Second, the proposed TWAFR-GRU is presented and evaluated in section III and IV, respectively. Finally, section V concludes the study and proposes future work.

II. Related Work

In this section, related work about charging station occupancy prediction, asynchronous federated learning (AFL), and federated meta-learning (FMeta) are summarized.

A. Charging Station Occupancy Prediction

Charging station occupancy prediction aims to predict future occupancy based on historical occupancy data, which is a type of time-series prediction. Currently, time-series prediction models can be broadly categorized into two groups, i.e., traditional machine learning models based on statistics and deep learning models based on artificial neural networks [6].

The earliest known statistics-based machine learning model dates back to 1927. The autoregressive model was first put forth by Yule et al. to investigate the sunspot activity period [7]. As the successor, Box et al. devised AutoRegressive Integrated Moving Average (ARIMA) [8], which has a significant impact on the prediction of linear time-series data. Moreover, Support Vector Regression (SVR), which can be modeled based on nonlinear functions, was proposed as a solution to the nonlinear time-series prediction problem [9]. Das et al. designed Bayesian network, which increases the precision of time-series prediction by gaining prior knowledge [10].

In recent years, thanks to the rise in computing power and the advent of the big data era, deep learning models have become a popular solution for a range of time-series prediction issues [11]. Elman et al. proposed Recurrent Neural Network (RNN) for time-series prediction [12], extracting the features of time-series data through connected neural units. To address gradient explosion in RNN, Hochreiter et al. further proposed Long Short-Term Memory (LSTM) network [13], which can remember crucial historical information through three gated structures. Based on LSTM, Cho et al. simplified the network structure and presented Gated Recurrent Unit (GRU) network [14], which reduces the scale of the model parameters while preserving the model prediction performance.

Currently, the prediction of charging demand and charging station occupancy is an emerging topic in the field of time-series prediction. Qiao et al. first applied XGBoost to the prediction of charging demand [15], which improved the model's prediction accuracy and interpretability. Inspired by the Seq2Seq model in natural language processing, Yi et al. utilized RNN to forecast the charging demand [16]. Later on, Hu et al. adopted LSTM in charging demand prediction to avoid gradient explosion in RNN [17].

B. Asynchronous Federated Learning (AFL)

With the development of Internet of Things, data become ubiquitous. However, centralized model training performed on distributed data becomes challenging, as it requires huge computational and communication resources. Therefore, federated learning was proposed to reduce the enormous computational and communication burden associated with the central server [5]. McMahan et al. first proposed Federated Averaging (FedAvg) [18], in which each client completes model training and model uploading separately and the server conducts model aggregation. To further enhance the training efficiency, Xie et al. proposed Asynchronous Federated Optimization (FedAsync) [19], where the server executes model aggregation right after receiving a model from clients. Then Chen et al. put forth Temporally Weighted Asynchronous Federated Learning (TWAFL) [20], whose weights are determined by the staleness of the model. Based on an informative client activating strategy, a multi-phase layer updating strategy, and a temporal weight fading strategy, You et al. further proposed a Triple-step AFL mechanism (TrisaFed) [21], which can significantly enhance the model's training efficiency and prediction accuracy.

In recent years, AFL has been widely applied in various distributed training scenarios. Liu et al. proposed an AFL framework for edge computing in vehicular networks so that the security and privacy of the vehicles' data can be safeguarded from cyber attack [22]. Then, Sakib et al. applied AFL with COVID-19 prediction models so as to ensure data security and save communication bandwidth [23]. Moreover, Chen et al. designed a novel AFL scheme to boost the training efficiency of heterogeneous IoT devices [24].

C. Federated Meta-learning (FMeta)

The idea of meta-learning is to enable the initial model to learn the gradient descent direction on different tasks so that the initial model can swiftly adapt to different tasks. Finn et al. first proposed the Model-Agnostic Meta-Learning (MAML) model [25]. After that, Nichol et al. further designed Reptile [26], a first-order meta-learning model, which can significantly ease the computational requirement for meta-learning. Furthermore, Federated Meta-Learning (FedMeta) was presented based on MAML and FedAvg [27].

In addition to the theoretical studies mentioned above, FMeta has now been applied to many fields. Zheng et al.

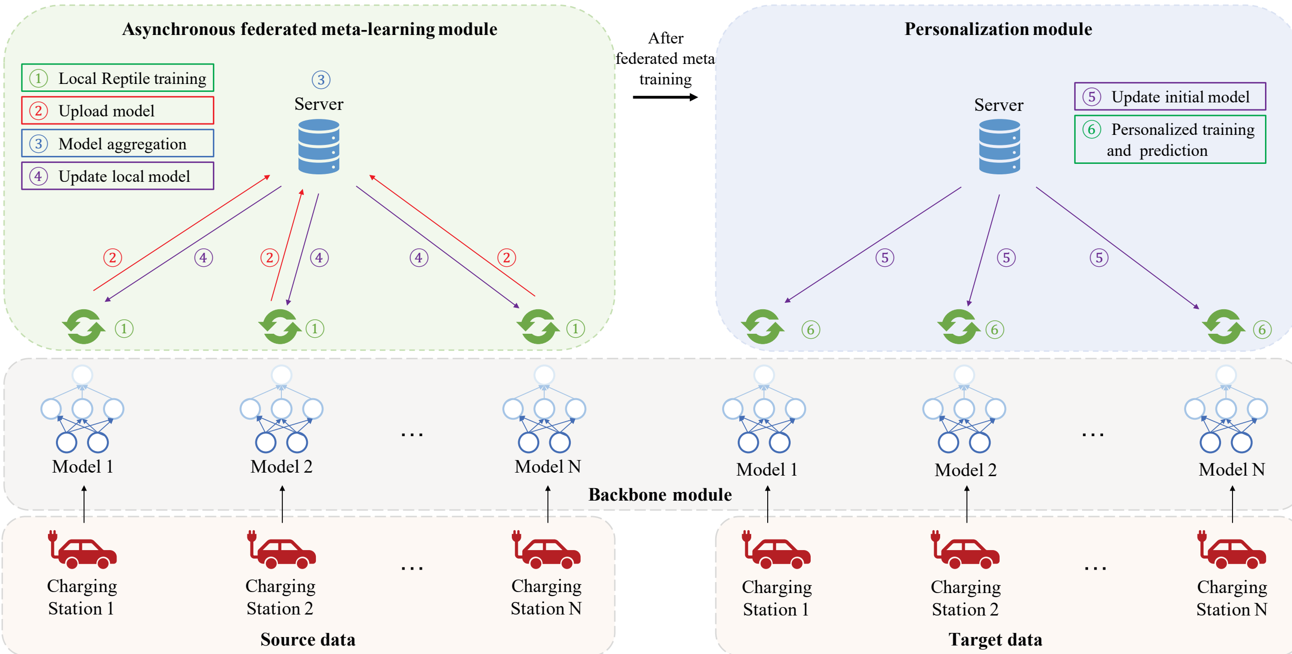

Fig. 1. Overall structure of the proposed model

combined federated meta-learning with a fraudulent credit card detection model, which brought significant improvements in detection accuracy while ensuring data security [28]. Jiang et al. introduced federated meta-learning to a fine-grained location prediction model, so as to protect the user privacy and increase the prediction accuracy [29]. More recently, Qu et al. proposed a novel model called ALL that combines first-order MAML (FOMAML) with FedAvg, which boosts the convergence speed and forecasting performance of the parking occupation prediction model [30].

III. Methodology

As depicted in Figure 1, the mechanism proposed in this paper consists of three modules, namely 1) backbone module, which determines the backbone model for predicting charging station occupancy; 2) asynchronous federated meta-learning module, which adopts AFL and meta-learning to obtain an initial model with high generalization ability; and 3) personalization module, where the initial model is separately adapted on different charging station occupancy prediction tasks. In the following sections, the details of each module will be discussed.

A. Backbone Module

It is responsible for deciding what backbone network to use for prediction. An appropriate backbone network will be helpful to achieve a high prediction performance. GRU is an improved model based on RNN and LSTM [14], in comparison to RNN, GRU can handle the gradient explosion problem during backpropagation, and compared to LSTM, it can also maintain excellent prediction performance while reducing the scale of model parameters, which can significantly save computing resources and training time.

Such that, we adopt GRU as the backbone network of the prediction model, whose structure is shown in Figure 2. In this model, at timestamp t, the vector $X = (x_{t-5}, x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t)$ is the input vector, which represents the occupancy over the previous six timestamps, and the $y_{t+\lambda}$ is the output, where λ denotes the prediction interval.

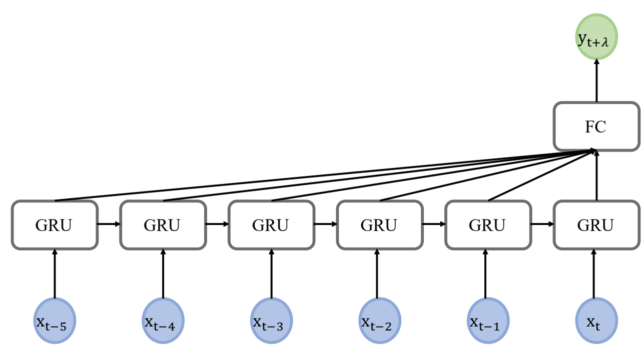

Fig. 2. The backbone model structure

Formula 1 describes the GRU calculation procedure, where γ_t and z_t stand for the reset gate vector and update gate vector respectively; n_t represents the candidate activation vector; h_t and h_{t-1} are the output vector of the current timestamp t and previous timestamp $(t-1)$, respectively; x_t is the input vector; W_r, W_u, W_n, U_r, U_u, U_n, b_r and b_u are trainable model parameters; σ and $tanh$ stand for sigmoid function and $tanh$ function severally; $\circ$ denotes the Hadamard product.

$$\begin{aligned} \gamma_t &= \sigma(W_\gamma x_t + U_\gamma h_{t-1} + b_\gamma) \\ z_t &= \sigma(W_u x_t + U_u h_{t-1} + b_u) \\ n_t &= tanh(W_n x_t + \gamma_t \circ (U_n h_{t-1})) \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ n_t \end{aligned} \tag{1}$$

B. Asynchronous Federated Meta-learning Module

AFL and meta-learning are emerging topics in deep learning. In general, AFL can bridge distributed data and computing resources more efficiently, and meta-learning can train an initial model that can be adapted rapidly for new tasks. Such that, an asynchronous federated meta-learning module is studied to inherit their benefits to not only enable efficient training by harnessing distributed data but also strengthen the generality of the model.

As illustrated in Figure 3, the module combines Reptile with TWAFL to build Temporally Weighted Asynchronous Federated Reptile (TWAFR).

1) Reptile: Given N tasks, Reptile trains the initial model by first executing k-step (k>1) Stochastic Gradient Descent (SGD) on data from each source, then utilizing $(\pi - \pi_j^{'})$ as the gradient descent direction, and finally updates the model based on gradient descent, as defined in the Formula 2, where j denotes the j^{th} task; π stands for the original model; $\pi_j^{'}$ represents the updated model of the j^{th} task; π^* denotes the updated global model; N is the number of tasks; β is the learning rate of the meta-learning training; SGD denotes the k-step SGD.

$$\begin{aligned} \pi_j^{'} &= SGD(\pi) \\ \pi^* &= \pi - \beta \sum_{j=1}^{N} (\pi - \pi_j^{'}) \end{aligned} \tag{2}$$

2) TWAFL: Due to the fact that the freshness of uploaded models may vary among each other in asynchronous federated learning, and intuitively, the model with higher freshness may be more effective and more representative, a temporally weighted strategy for AFL is proposed to aggregate local models with a weight measuring their staleness. Since the aggregation process is managed by the server, the server will assign model weights based on the models' freshness $(t_j^* - i)$ and data size n_j as defined in Formula 3, where ω_j denotes the weight of j^{th} client's updated model; i stands for the current timestamp; t_j^* represents the timestamp of j^{th} client's updated model; n_j signifies the data size of the j^{th} client; j denotes the j^{th} client; N is the number of the clients.

$$\omega_j = \frac{\frac{n_j}{\sum_{j=1}^{N} n_j} e^{-(t_j^* - i)}}{\sum_{j=1}^{N} \frac{n_j}{\sum_{j=1}^{N} n_j} e^{-(t_j^* - i)}} \tag{3}$$

After that, the server will use the weight to update the global model according to Formula 4, where θ_i denotes the updated global model at the current training round i and π_j^* represents the local model received from the j^{th} client.

$$\theta_i = \sum_{j=1}^{N} \omega_j \times \pi_j^* \tag{4}$$

3) TWAFR: Based on Reptile and TWAFL, a temporally weighted asynchronous federated reptile (TWAFR) algorithm is proposed as depicted in Algorithm 1, which runs on both the clients and the server:

- **TWAFR for the client**: First, client j receives the global model θ^{i-1} of round $(i-1)$ and the current timestamp i from the server. Second, client j clones θ^{i-1} as the initial model π_j. Third, client j samples N tasks from its local data randomly. Fourth, client j updates model π_j^* according to the Formula 2. Fifth, client j clones timestamp i as its local updating timestamp t_j^*. Finally, client j uploads the model π_j^*, the timestamp t_j^* and the data size n_j to the server.
- **TWAFR for the server**: First, the server broadcast the global model of the previous round θ^{i-1} and the current timestamp i to the clients that completed previous rounds of training. Second, the server continuously receives the local model, timestamp, and data size of each client. Third, after a predefined timer gets exceeded, the server calculates the temporal weight for each received local model according to Formula 3. Eventually, the server performs model aggregation to update the global model according to Formula 4.

Algorithm 1 The pseudocode for TWAFR

Function at the Server

for i=1,2,...,M **do**
  Transferring θ^{i-1} and i to clients
  Receive π_j^*, t_j^* and n_j from client j
  while the time limit is reached **do**
    Assign model weights ω_j according to Formula 3
    Update global model θ^i according to Formula 4
  end while
end for

Function at each client

Receiving θ^{i-1} and i from the server
Initializing the local model $\pi_j = \theta^{i-1}$
Sampling N tasks randomly
Updating the model π_j^* according to Formula 2
Updating the timestamp $t_j^* = i$
Uploading π_j^*, t_j^* and n_j to the server

In general, TWAFR can inherit the advantages of Reptile and TWAFL to train an initial model with high generality by collaborating dispersed and isolated local data of each client in a privacy-preserving and cost-efficient manner, and aggregating received parameters according to their temporal weights measuring their staleness. Therefore, the initial model

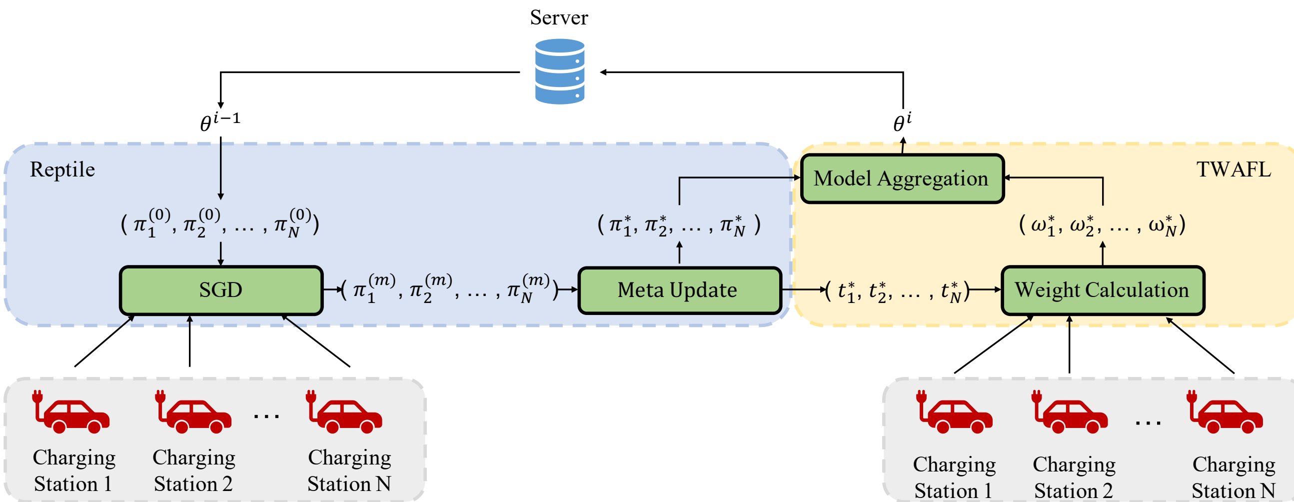

Fig. 3. The asynchronous federated meta-learning module implementing temporally weighted asynchronous federated reptile (TWAFR) by integrating TWAFL (temporally weighted asynchronous federated learning) with Reptile

can be rapidly trained with high performance as evaluated in Sections IV-B1, IV-B2, and IV-B3.

C. Personalization Module

The global initial model has a great capacity for generalization by following numerous iterations of asynchronous federated meta-learning training. However, its performance may be unsatisfactory in a specific prediction task. Therefore, before the actual usage of the initial model, the client needs to localize the initial model based on its own local data.

In particular, each client updates the initial model and conducts personalized training, as stated in the Formula 5, where θ denotes the global initial model; θ_j represents the updated initial model of the j^{th} client; θ_j^* signifies the personalized model of the j^{th} client; SGD stands for the multi-step SGD.

$$\begin{aligned} \theta_j &= \theta \\ \theta_j^* &= SGD(\theta_j) \end{aligned} \tag{5}$$

In general, after a few rounds of personalization, the localized model can achieve stable and high performance, which is evaluated in Section IV-B4.

In summary, the proposed mechanism, named TWAFR-GRU, integrates GRU as its backbone model to make charging station occupation prediction, use meta-learning as a model adaptor to train an initial model with high generality, and perform asynchronous federated learning as a training optimizer to build a global initial model by collaborating local resources of clients in a cost-efficient and privacy-preserving manner. Based on the initial model trained from TWAFR-GRU, a high-performance local model can be created through a personalization process for each station.

IV. Performance Evaluation

In this section, the experimental setting is given first, followed by analysis and discussion of the experimental results to demonstrate the merit of TWAFR-GRU.

A. Experiment Setting

It includes a common dataset, compared baselines, running configuration, and evaluation metrics.

1) Dataset Preparation: The evaluation dataset contains occupancy data of 35 charging stations in Guangzhou, China, whose geographical distribution is plotted in Figure 4. This dataset's occupancy data spans from December 10, 2021 to January 7, 2022, with a 5-minute interval. In summary, there are 8,352 occupancy records for each charging station, and thus, in total, about 0.3 million records. It's worth noting that the dataset can be downloaded from the link[1].

2) Compared Baselines: In order to evaluate the performance of the proposed model, there are two conventional machine learning and three deep learning models are used to be compared, namely:

- The two machine learning models include 1) ARIMA: Autoregressive Integrated Moving Average Model, a classic machine learning model [8], and 2) SVR: Support Vector Regression, a statistics-based machine learning model [9].
- The three deep learning models include 1) RNN: Recurrent Neural Network [12], 2) LSTM: Long Short-Term Memory, an improved model based on RNN [13], and 3) GRU: Gated Recurrent Unit, an improved model of LSTM [14].

Moreover, three federated learning methods are used, i.e., 1) FedAvg (widely used synchronous federated learning

[1] https://github.com/IntelligentSystemsLab/TWAFR-GRU

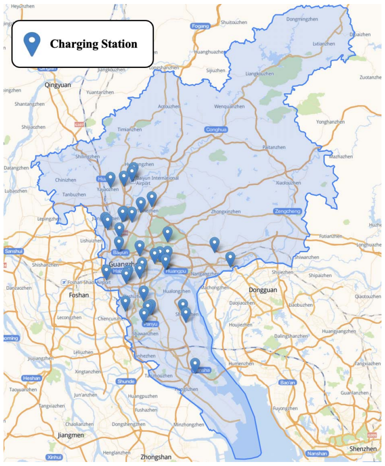

Fig. 4. Spatial distribution of the charging stations in Guangzhou, China

TABLE I
THE HYPERPARAMETERS SETTING OF THE MODELS

Model	Parameter	Value
*	total rounds of global training	500
	total rounds of personalization adaption	10
	learning rate of SGD	0.001
	learning rate of meta update	1
	loss function	MSE
	total steps in k-step SGD	5
	optimizer	Adam
ARIMA	(p,d,q)	(6,0,6)
SVR	kernel, epsilon	RBF, 0.001

* denotes models except ARIMA and SVR.

method) [18], 2) FedRep integrating FedAvg and Reptile, and 3) TWAFR (the proposed method).

Finally, the hyperparameters used by compared models are listed in Table I.

3) Evaluation Configuration: The charging station occupancy dataset is split into a training dataset and a testing dataset containing 30 and 5 charging stations, respectively. Then, the data of each charging station are subdivided into support set and query set. Data from December 10, 2021 to December 31, 2021 are used as the support set, and data from January 1, 2022 to January 7, 2022 are utilized as the query set. In this experiment, the training dataset is used to pre-train the model, followed by personalized training on the testing dataset's support set and evaluation on the testing dataset's query set. Each model is tested on four prediction intervals: 15, 30, 45, and 60 minutes. Moreover, each experiment is repeated 10 times to ensure the reliability of the evaluation.

4) Evaluation Metrics: As defined in Formula 6, where $y_i^{'}$ and y_i denote predicted value and the real value, respectively, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Relative Absolute Error (RAE) and Coefficient of Determination (R^2) are used as the evaluation metrics.

$$
\begin{aligned}
MAE &= \sum_{i=1}^{N} |y_i^{'} - y_i| \\
RMSE &= \sqrt{\frac{1}{n}\sum_{i=1}^{N}(y_i^{'} - y_i)^2} \\
RAE &= \frac{\sum_{i=1}^{N}|y_i^{'} - y_i|}{\sum_{i=1}^{N}|\bar{y}_i - y_i|} \\
R^2 &= 1 - \frac{\sum_{i=1}^{N}(y_i^{'} - y_i)^2}{\sum_{i=1}^{N}(\bar{y}_i - y_i)^2}
\end{aligned}
\tag{6}
$$

5) Running environment: The evaluation is conducted on a Windows workstation, which is equipped with two NVIDIA GeForce RTX 3090 GPU, an Intel Gold 5218R Two-Core Processor CPU, and 512 G RAM.

B. Evaluation Results

The evaluation results are analyzed from four aspects, namely forecasting error, convergence speed, training time, and model generality.

1) Forecasting Error: According to the results in Table II, the following observations can be made. First, deep learning models are superior to machine learning models in all four evaluation metrics, as more features can be extracted and modeled by the interconnected neurons.

Second, the deep learning model applying TWAFR and FedRep can significantly outperform the ones adopting FedAvg. Specifically, TWAFR-Aver models can decrease MAE, RMSE, and RAE by 19.12%, 15.43%, and 17.25% separately and improve R^2 by 67.43%. Similarly, FedRep-Aver can achieve a reduction of about 19.20%, 15.49%, and 17.33% in MAE, RMSE, and RAE, respectively, and an improvement of 67.73% in R^2.

Finally, TWAFR-GRU achieves the best performance compared with other baseline models. As summarized in Table II, MAE, RMSE, RAE and R^2 of TWAFR-GRU can reach 10.50×10^{-2}, 13.63×10^{-2}, 66.87% and 44.98% respectively, which outperforms all the compared models.

2) Convergence speed: As illustrated in Figure 5, TWAFR-GRU and FedRep-GRU can get the global model to converge faster than FedAvg-GRU. Specifically, to reach the target RMSE smaller than 0.20, FedAvg-GRU needs 200 global training rounds. However, TWAFR-GRU and FedRep-GRU only need 50 rounds. Hence, a boost of 75% in the convergence speed can be achieved. Besides that, it also can be observed that TWAFR-GRU and FedRep-GRU can eventually achieve a smaller loss compared to FedAvg-GRU, illustrating that they can train an initial model more accurately. Finally, it can be seen that a similar RMSE curve is shared between TWAFR-GRU and FedRep-GRU, indicating that compared to the synchronous method, the asynchronous method TWAFR won't deteriorate the model convergence performance even though less training time is required (as shown in Figure 6).

TABLE II
THE SUMMARY OF MODEL PERFORMANCE

Model	MAE($\times 10^{-2}$)	RMSE($\times 10^{-2}$)	RAE(%)	R^2(%)
TWAFR-GRU*	**10.50**	**13.63**	**66.87**	**44.98**
TWAFR-LSTM	10.81	13.93	68.85	42.91
TWAFR-RNN	10.68	13.86	67.94	43.36
TWAFR-Aver†	10.66	13.81	67.89	43.75
FedRep-GRU	10.76	13.94	68.63	42.74
FedRep-LSTM	10.66	13.75	67.86	44.26
FedRep-RNN	10.52	13.70	66.97	44.50
FedRep-Aver‡	10.65	13.80	67.82	43.83
FedAvg-GRU	13.05	16.28	81.15	26.55
FedAvg-LSTM	12.96	16.11	81.01	27.80
FedAvg-RNN	13.54	16.60	83.97	24.05
FedAvg-Aver§	13.18	16.33	82.04	26.13
ARIMA	15.71	19.06	99.11	-2.94
SVR	12.89	16.18	81.22	24.87

* denotes the proposed model.
† denotes the average of the TWAFR-based models.
‡ denotes the average of the FedRep-based models.
§ denotes the average of the FedAvg-based models.

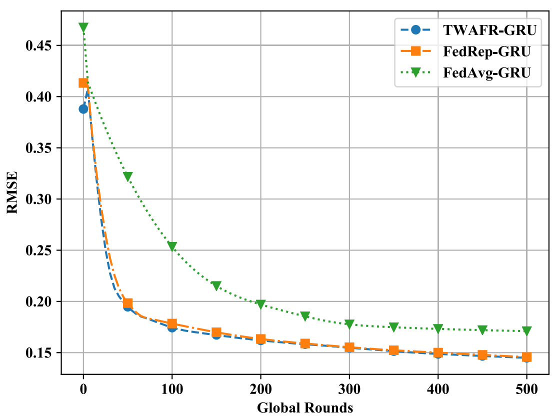

Fig. 5. The RMSE curves of global training

3) Training Time: As illustrated in Figure 6, at the end of 500 global training rounds, FedRep-GRU is slightly better than FedAvg-GRU, however, TWAFR-GRU can surpass FedRep-GRU by about 44%, even though they share a similar performance as shown in Figure 5.

4) Model generality: It can be seen from Figure 7, which plots the RMSE curves of TWAFR-GRU, FedRep-GRU and FedAvg-GRU in personalized training, that the TWAFR-GRU has the best adaptability to the new tasks. The final RMSE of TWAFR-GRU is 0.136, which is notably smaller than FedRep-GRU (0.140) and FedAvg-GRU (0.163), demonstrating that the TWAFR is beneficial for models to be adapted to new tasks. Moreover, it is worth noting that the initial RMSE of TWAFR-GRU and FedRep-GRU is also much smaller than the final RMSE of the FedAvg-GRU, greatly proving that the federated meta-learning enables the model to have a better

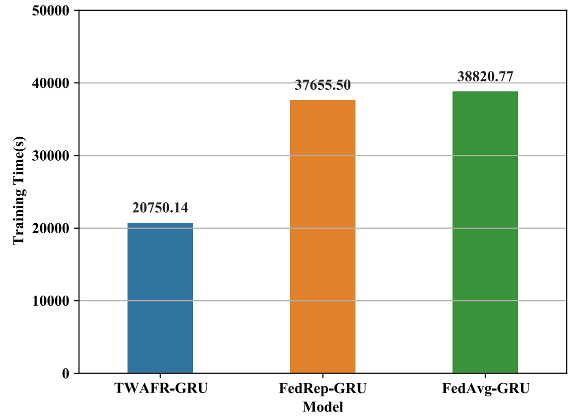

Fig. 6. The comparison of global training duration

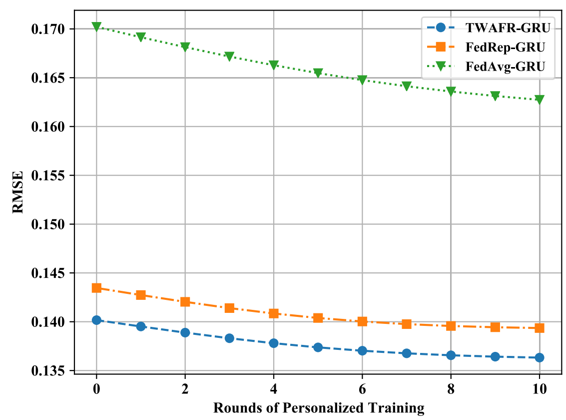

Fig. 7. The RMSE curves of personalized training

generalization ability.

C. Discussion

Based on the above experiment results, first, TWAFR-based and FedRep-based models can outperform FedAvg-based models in forecasting error, convergence speed, and generalization ability, which proves the efficiency and effectiveness of federated meta-learning in supporting charging station occupancy prediction. Second, the performance in forecasting error, convergence speed, and generalization ability between TWAFR-based models and FedRep-based models are comparable, demonstrating that the asynchronous method won't result in the degradation of model performance. Third, compared to the synchronous methods, the asynchronous method TWAFR can lead to a huge reduction in training time, proving that the TWAFR is cost-efficient. Finally, it can be seen from the whole evaluation that TWAFR-GRU is superior to all the other models with the best performance in every aspect. Therefore, the proposed method TWAFR-GRU can train an initial model with high generalization capability for each charging station to adopt it for a more accurate occupancy prediction but with a relatively low training cost.

V. Conclusions

With the penetration of EVs, the prediction of charging station occupancy becomes a crucial part of intelligent transportation systems to direct drivers to the nearest idle charging pile with unnecessary cruising for idle charging piles reduced. In this paper, a novel model, called TWAFR-GRU, is proposed for the real-time charging occupancy prediction, which integrates 1) asynchronous federated learning, enabling more distributed data to participate in training efficiently; 2) Reptile, enhancing the generalization ability and convergence speed of the initial model; 3) GRU, maintaining prediction performance while reducing the scale of the model parameters. According to the evaluation results, the proposed model outperforms other state-of-the-art baselines in every aspect, namely, prediction accuracy, convergence speed, training time, and generalization ability.

Nevertheless, TWAFR-GRU still has the potential to be improved by adopting more strategies, such as designing better backbone models, applying dynamic client selection strategies, etc. In the future, we will further design an improved model, which will be equipped with a dynamic client selection strategy as well as a multi-phase layer updating strategy to further improve its prediction performance and training efficiency.

Acknowledgment

This research was funded by the National Natural Science Foundation of China (62002398) and the Collaborative Innovation Center for Transportation of Guangzhou (202206010056).

References

[1] T.-Y. Ma and S. Faye, "Multistep electric vehicle charging station occupancy prediction using hybrid lstm neural networks," *Energy*, vol. 244, p. 123217, 2022.

[2] H. Tenkanen and T. Toivonen, "Longitudinal spatial dataset on travel times and distances by different travel modes in helsinki region," *Scientific data*, vol. 7, no. 1, pp. 1–15, 2020.

[3] H. Xu, J. Li, H. Xiong, and H. Lu, "Fedmax: Enabling a highly-efficient federated learning framework," in *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*. IEEE, 2020, pp. 426–434.

[4] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[5] S. Liu, Q. Chen, and L. You, "Fed2a: Federated learning mechanism in asynchronous and adaptive modes," *Electronics*, vol. 11, no. 9, p. 1393, 2022.

[6] Z. Liu, Z. Zhu, J. Gao, and C. Xu, "Forecast methods for time series data: a survey," *IEEE Access*, vol. 9, pp. 91 896–91 912, 2021.

[7] G. U. Yule, "Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 226, no. 636-646, pp. 267–298, 1927.

[8] G. E. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.

[9] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[10] M. Das and S. K. Ghosh, "A probabilistic approach for weather forecast using spatio-temporal inter-relationships among climate variables," in *2014 9th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 2014, pp. 1–6.

[11] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[12] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[15] F. Qiao and S. Lin, "Data-driven prediction of fine-grained ev charging behaviors in public charging stations: Poster," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, 2021, pp. 276–277.

[16] Z. Yi, X. C. Liu, R. Wei, X. Chen, and J. Dai, "Electric vehicle charging demand forecasting using deep learning model," *Journal of Intelligent Transportation Systems*, pp. 1–14, 2021.

[17] T. Hu, K. Liu, and H. Ma, "Probabilistic electric vehicle charging demand forecast based on deep learning and machine theory of mind," in *2021 IEEE Transportation Electrification Conference & Expo (ITEC)*. IEEE, 2021, pp. 795–799.

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

[19] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[20] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

[21] L. You, S. Liu, Y. Chang, and C. Yuen, "A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement," *IEEE Internet of Things Journal*, 2022.

[22] J. Liu and H. Wu, "Federated learning with dynamic staleness correction for privacy protection in vehicular networks," in *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, 2021, pp. 877–882.

[23] S. Sakib, M. M. Fouda, Z. M. Fadlullah, and N. Nasser, "On covid-19 prediction using asynchronous federated learning-based agile radiograph screening booths," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[24] Z. Chen, W. Liao, K. Hua, C. Lu, and W. Yu, "Towards asynchronous federated learning for heterogeneous edge-powered internet of things," *Digital Communications and Networks*, vol. 7, no. 3, pp. 317–326, 2021.

[25] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[26] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[27] F. Chen, M. Luo, Z. Dong, Z. Li, and X. He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

[28] W. Zheng, L. Yan, C. Gou, and F.-Y. Wang, "Federated meta-learning for fraudulent credit card detection," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 4654–4660.

[29] X. Jiang, S. Zhao, G. Jacobson, R. Jana, W.-L. Hsu, M. Talasila, S. A. Aftab, Y. Chen, and C. Borcea, "Federated meta-location learning for fine-grained location prediction," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 446–456.

[30] H. Qu, S. Liu, J. Li, Y. Zhou, and R. Liu, "Adaptation and learning to learn (all): An integrated approach for small-sample parking occupancy prediction," *Mathematics*, vol. 10, no. 12, p. 2039, 2022.