Personalized Federated DARTS for Electricity Load Forecasting of Individual Buildings

Dalin Qin, *Student Member, IEEE*, Chenxi Wang, *Student Member, IEEE*, Qingsong Wen, *Senior Member, IEEE*, Weiqi Chen, Liang Sun, and Yi Wang, *Member, IEEE*

***Abstract*—Building-level load forecasting is becoming increasingly crucial since it forms the foundation for better building energy management, which will lower energy consumption and reduce CO$_2$ emissions. However, building-level load forecasting faces the challenges of high load volatility and heterogeneous consumption behaviors. Simple regression models may fail to fit the complex load curves, whereas sophisticated models are prone to overfitting due to the limited data of an individual building. To this end, we develop a novel forecasting model that integrates federated learning (FL), the differentiable architecture search (DARTS) technique, and a two-stage personalization approach. Specifically, buildings are first grouped according to the model architectures, and for each building cluster, a global model is designed and trained in a federated manner. Then, a local fine-tuning approach is used to adapt the cluster global model to each individual building. In this way, data resources from multiple buildings can be utilized to construct high-performance forecasting models while protecting each building's data privacy. Furthermore, personalized models with specific architectures can be trained for heterogeneous buildings. Extensive experiments on a publicly available dataset are conducted to validate the superiority of the proposed method.**

***Index Terms*—Building-level load forecasting, federated learning, differentiable neural architecture search, privacy-preserving, personalization.**

NOMENCLATURE

Indices

k	Index for all K buildings
c	Index for all C building clusters
r	Index for R_{search}/R_{train} federated communication round
i	Index for E_{pre}/E_{ft} local update iteration
t	Index for timestep

Variables

X	The model input
$\hat{Y}$	The model forecasting output
Y	The true value of the load
α	The model architecture parameter
$\bar{\alpha}$	The global model architecture parameter
ω	The model weight parameter
$\bar{\omega}$	The global model weight parameter
$o^{(i,j)}$	The operation between node i and j
η_α	The learning rate for updating α
η_ω	The learning rate for updating ω
ξ	The learning rate for a single update step in approximated optimization
f	The features of each timestep
$h_{(t)}$	The hidden state
$c_{(t)}$	The cell state
$g_{(t)}$	The intermediate state
$\mathcal{I}$	The output of input gate
$\mathcal{F}$	The output of forget gate
$\mathcal{O}$	The output of output gate
$\mathbb{K}$	The set of buildings
K	The number of buildings
$\mathbb{C}$	The set of building clusters
C	The number of building clusters
$\mathcal{D}_k$	The local dataset of the kth building
N	The total sample number of all buildings
N_k	The sample number of $\mathcal{D}_k$
E_{pre}	The number of local pre-search iterations
E_{ft}	The number of local finetune iterations
R_{search}	The number of federated search communication rounds
R_{train}	The number of federated train communication rounds
N_p	The patience of early-stop
$\mathbb{P}$	The complete set of restricted permutation matrices for calculating the adjusted error
P	The permutation matrix for calculating the adjusted error
d	The permitted displacement magnitude in calculating the adjusted error

I. INTRODUCTION

A. Backgrounds and Motivations

Buildings are responsible for a growing amount of energy consumption and CO$_2$ emissions due to rapid urbanization. According to [1], the electricity consumed by buildings accounts for nearly 50% of all electricity consumed worldwide, 35% of energy used globally, and 38% of all energy-related

Manuscript received 25 August 2022; revised 28 December 2022; accepted 3 March 2023. The work was supported in part by the National Key R&D Program of China (2022YFE0141200), in part by the National Natural Science Foundation of China (72242105), and in part by the Alibaba Innovative Research Programme. Paper no. TSG-01256-2022. (*Corresponding author: Yi Wang.*)

Dalin Qin, Chenxi Wang, and Yi Wang are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR, China, and are also with The University of Hong Kong Shenzhen Institute of Research and Innovation, Shenzhen, 518057, China (e-mail: dlqin@eee.hku.hk, cxwang@eee.hku.hk, yiwang@eee.hku.hk).

Qingsong Wen and Liang Sun are with the DAMO Academy, Alibaba Group (U.S.) Inc., Bellevue, WA 98004, USA (e-mail: qingsong.wen@alibaba-inc.com, liang.sun@alibaba-inc.com).

Weiqi Chen is with the DAMO Academy, Alibaba Group, Hangzhou, China (e-mail: jarvus.cwq@alibaba-inc.com).

CO_2 emissions in 2019. Building load forecasting, as a basis for building energy management, is therefore becoming crucial in an effort to promote renewable energy accommodation and reduce CO_2 emissions. Building load forecasting is a challenging task due to *high load volatility* and *heterogeneous consumption behaviors* [2].

When dealing with *high load volatility*, we are in a dilemma. Simple regression models may fail to sufficiently capture the characteristics of volatile load curves. If complex models are applied instead, there is a significant overfitting risk because of a shortage of data from the individual building itself. A possible solution is to make full use of data from multiple buildings. However, these methods all assume that individuals' data can be freely collected without taking privacy into consideration, which may not be the case in the real world. Governments all over the world have passed legislation governing the gathering and processing of data to preserve privacy. A comprehensive set of laws, stringent guidelines for data processing, and harsh penalties for infractions have been developed in Europe under the General Data Protection Regulation (GDPR) [3]. In China, data collection, storage, use, processing, transportation, provision, and disclosure are all regulated by the Data Security Law of the People's Republic of China [4]. A Voluntary Code of Conduct (VCC) was released in the US to address privacy issues with smart grid technologies [5]. To address the privacy issue, federated learning (FL) as a distributed machine learning technique can be applied [6], where participants' data will be kept in situ, preserving their privacy while allowing for the cooperative training of a high-performance global model to handle highly volatile time series.

To tackle the issue of *heterogeneous consumption behavior*, we need to develop adaptive models for various buildings. Recently, substantial advancements in autonomous model design have been made through auto-machine learning (AutoML). Differentiable ARchiTecture Search (DARTS) [7] is gaining popularity due to its ease of implementation and effectiveness in determining the appropriate neural network architecture to adapt to various datasets. DARTS thus opens up the possibility of adaptively constructing forecasting models for various individual buildings.

Therefore, this paper is focused on answering one question: *How to construct effective models for individual buildings with high load volatility and heterogeneous consumption patterns?* We are inspired to answer this question by leveraging the strengths of FL and DARTS and propose a personalized and privacy-preserving load forecasting model for individual buildings.

B. Literature Review

Extensive work has been done for building-level load forecasting, which can be roughly divided into two categories. The first category is statistics-based methods, such as [8], which employed the autoregressive integrated moving average (ARIMA) model for peak load forecasting of university buildings. [9] analyzed building load characteristics and performed short-term forecasting using three methods: direct curve fitting, similar day, and multiple linear regression. The second category is machine learning-based approaches. A long short term memory (LSTM) recurrent neural network was used in [10], and its prediction accuracy was confirmed to be superior to that of several other forecasting models (including multilayer perceptron, k-nearest neighbor model, extreme learning machine, and naive prediction method). In [11], a graph neural network was built for residential short-term load forecasting that considered both temporal and spatial information. According to [12], an improved deep residual neural network framework was created, and a two-stage ensemble approach was used to increase prediction accuracy. In [13], a conditional probability density function-based probabilistic residential load forecasting model was created. [14] suggested a multi-kernel transfer method to help buildings with limited data construct effective forecasting models. Actually, there are mainly two approaches to utilizing individuals' data. One is to improve the forecasting accuracy at the aggregated level by considering individual loads as subprofiles. [15] proposed to enhance aggregated residential load forecasting by adopting the practice theory in subprofiles modeling. [16] suggested ensemble the forecasting on subprofiles to improve the accuracy. The other one is to facilitate individual forecasting by incorporating data resources. [17] proposed pooling-based deep recurrent neural networks to address the overfitting issue as well as the high load volatility problem in household load forecasting by gathering historical data from multiple neighboring consumers. [18] suggested a clustering-based pooling method to avoid overfitting and proposed a multitask Bayesian deep learning approach to capture the uncertainty of household load.

Despite the effectiveness of the aforementioned techniques, two aspects can be investigated to improve building load forecasting performance even more. One is that the majority of works create forecasting models for each building but neglect the opportunity for cooperation among various buildings to fully utilize various data resources. FL is receiving a lot of attention these days as a distributed method that effectively protects privacy. Actually, the energy industry has employed several FL-based applications. Such as [19] proposed a federated Bayesian neural network for probabilistic behind-the-meter solar generation decomposition. [20] introduced FL to building heating load forecasting. [21] combined FL with clustering methods for electricity consumption pattern extraction. [22] investigated a distributed voltage control strategy for distribution networks using federated reinforcement learning. [23] achieved privacy-preserving voltage forecasting by integrating FL with differential privacy. The main challenge in the application of FL is the heterogeneous data distribution problem. [24] proposed FedProx which includes a proximal term in the objective to mitigate the heterogeneity issues. [25] suggested training the personalized model by adopting a global and local bi-level optimization approach. A meta-learning approach was proposed by [26] to construct the global model and adapt it to the local dataset to account for the heterogeneous distribution. [27] considered the heterogeneity of clients and proposed a heterogeneous FL approach by constructing global models for different client clusters and integrating transfer learning techniques. [28] proposed a gradient-based clustered federated multitask learning framework to handle incongruent client data distributions. We take inspiration from these works to incorporate FL into individual building load forecasting and investigate how buildings' heterogeneous consumption patterns will affect the performance of FL approaches.

Another issue is that current machine learning-based load forecasting models are built with a fixed structure, making them inflexible when dealing with heterogeneous consumer behaviors. DARTS, as a powerful AutoML technique, can design model architectures for a variety of tasks or datasets [7]. DARTS has gained great success in various fields. For example, [29] adopted DARTS to adaptively design model architecture for named entity recognition. [30] proposed DARTS-based AutoCTS, which supports multi-granularity for architectural search. Li et al. introduced ST-DARTS+ in [31] to adaptively decompose the spatial and temporal functionalities of the brain network. Even though many effective DARTS-related methods have been proposed in the field of computer vision, very few works integrate DARTS with load forecasting. To the best of our knowledge, only one work considered DARTS for load forecasting but with a limited analysis [32]. DARTS still has a lot of untapped potentials to adaptively design models for individual building load forecasting.

C. Contributions and Paper Organization

To address the aforementioned research gaps, this paper proposes an improved DARTS model and integrates it with FL as well as a two-stage personalization strategy to achieve privacy-preserving and personalized individual building load forecasting.

The contributions of this paper are threefold:

1) **New method:** Propose a novel Personalized Federated-DARTS (PFedDARTS) framework for building electricity load forecasting that integrates DARTS with FL, as well as a two-stage personalization strategy. Three major benefits can be achieved: (1) several high-performance model architectures are automatically generated for building clusters with diverse consumption patterns; (2) data resources from multiple buildings can be utilized to construct high-performance forecasting models in a privacy-preserving way; (3) personalized models can be effectively constructed from the perspective of both model architecture and weight parameters.
2) **New perspective:** Take privacy concerns into consideration throughout the whole procedure for building electricity load forecasting. The proposed architecture-based clustering method enables clustering without the requirement of ground-truth data from buildings. Furthermore, the clustered federated protocols to orchestra the collaborations of buildings for architecture search and model update are well designed to protect privacy.
3) **New findings:** Provide findings that the high load volatility and heterogeneous consumption patterns challenges can be effectively addressed by combing the architecture-based clustered federated approach with local model fine-tuning, which will balance the global modeling and the local modeling approach to capture both the universal load characteristics and the unique load characteristics of buildings with diverse consumption patterns. The experimental results show that the proposed method can significantly outperform the local approach or the standard FL model and have superior performance in a large-scale scenario.

The rest of the paper is structured as follows. Section II defines the problem to be solved. Section III elaborates the methodologies, including the overall framework and implementation details. Section IV reports experimental results and analysis. Section V draws the conclusions.

II. Problem Statement

As building loads are highly volatile and various buildings have heterogeneous consumption patterns, a strong and personalized forecasting model for each individual building is required. Besides, each building owns a limited amount of data, necessitating the use of data resources from multiple buildings while maintaining data privacy.

Assuming there exists a group of buildings, also denoted as clients, need to construct a personalized privacy-preserving load forecasting model with the help of a server. Each client owns historical local load profiles, which are chronologically split into three parts for training, validation, and testing separately. In order to design a personalized model, the task can be defined in the following two subtasks:

In the first subtask, the aim is to design model neural architecture α and the objective can be defined as:

$$\min_{\alpha} \ \mathcal{L}_{val}\left(\omega^{*}\left(\alpha\right),\alpha\right) \tag{1}$$

$$\text{s.t. } \omega^{*}\left(\alpha\right) = \arg\min\nolimits_{\omega}\mathcal{L}_{train}\left(\omega,\alpha\right) \tag{2}$$

where $\mathcal{L}_{val}$ and $\mathcal{L}_{train}$ is the training loss and validation loss respectively. The goal is to search optimal architecture α^{*} that minimizes the loss on the validation set $\mathcal{L}_{val}\left(\omega^{*},\alpha^{*}\right)$, where ω^{*} is optimized under the associated architecture. The approximation and solution for such a bi-level optimization problem will be detailed in the next section.

After the optimal model architecture is searched, the second subtask is to train the model by gradient descent as the usual machine learning approach. The objective is to minimize the loss function on the training dataset:

$$\min_{\omega}\mathcal{L}_{train}\left(Y\left|f_{\omega}\left(X\right)\right.\right) \tag{3}$$

Regarding privacy, buildings will not share their local data with others, but only model parameters can be shared.

III. Methodology

This section will present the overall framework of the proposed forecasting model, followed by implementation details of the improved DARTS model, Federated-DARTS model, and two-stage personalization strategy.

A. Proposed Framework

Fig. 1 depicts the overall framework of the proposed Personalized Federated-DARTS model with the indication of information flow. Several buildings of various types, including industry, lodging, education, and so on, have requested the construction of a personalized load forecasting model. First,

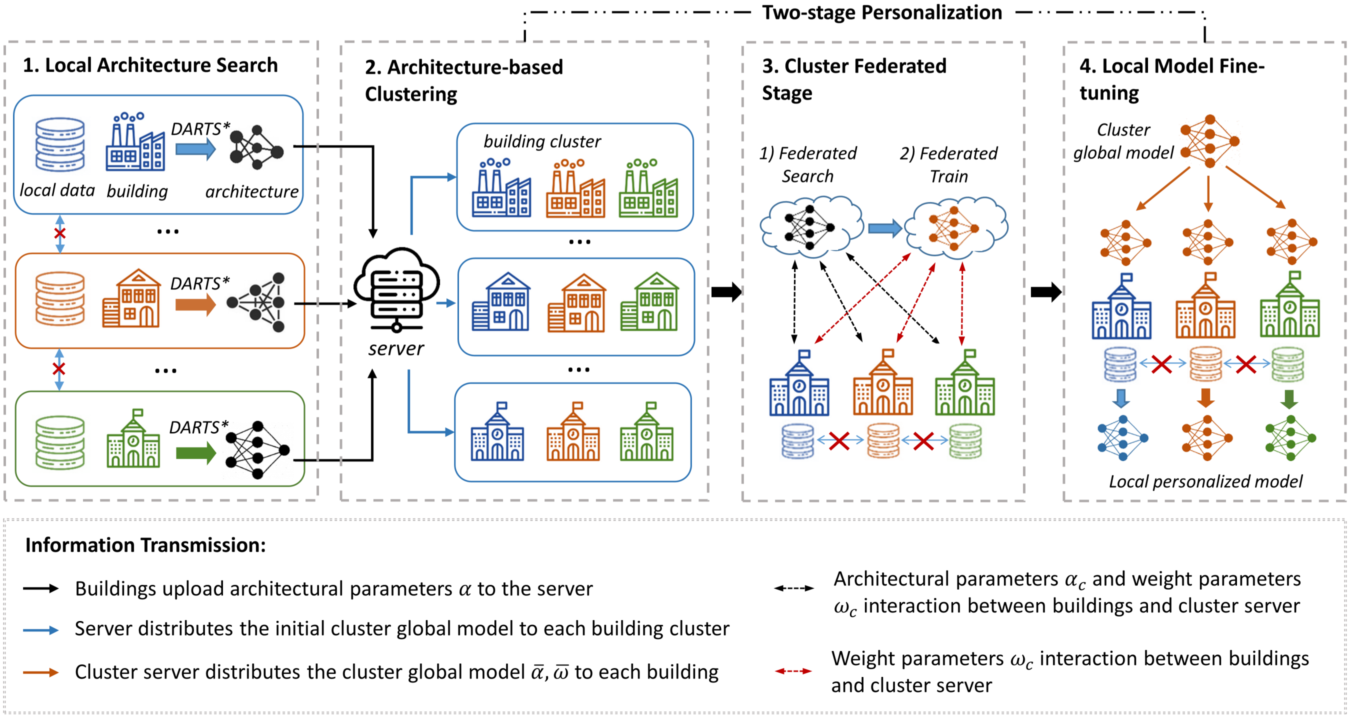

Two-stage Personalization
1. Local Architecture Search
local data
building
DARTS*
architecture
DARTS*
DARTS*
2. Architecture-based Clustering
building cluster
server
3. Cluster Federated Stage
1) Federated Search
2) Federated Train
4. Local Model Fine-tuning
Cluster global model
Local personalized model
Information Transmission:
Buildings upload architectural parameters α to the server
Server distributes the initial cluster global model to each building cluster
Cluster server distributes the cluster global model $\bar{\alpha}, \bar{\omega}$ to each building
Architectural parameters α_c and weight parameters ω_c interaction between buildings and cluster server
Weight parameters ω_c interaction between buildings and cluster server

Fig. 1. The proposed Personalized Federated-DARTS framework. Firstly, each building will locally search the model architecture for a few epochs in the pre-search stage. Then, the architectural parameters of the buildings should be uploaded to the server, and several building clusters can be formed based on the similarity of the model architecture. After the architecture-based clustering stage, buildings within each cluster will collaboratively search for a cluster global architecture and update the cluster global model in a federated manner by iteratively exchanging the architectural parameters and weight parameters with the cluster server. Finally, the cluster global model will be finetuned with local data to create a personalized model for each individual building.

an improved DARTS model called DARTS* is proposed, which enables each building to use its data to search a model architecture locally. Buildings can then be divided into various building clusters using an architecture-based clustering algorithm based on the similarity of their model architectures. By continuously exchanging training parameters, buildings within a cluster will then cooperatively search for an ideal cluster global model architecture. Based on the searched global architecture, buildings inside that cluster will then federatedly retrain the model and update models' inner weights parameters, resulting in the final cluster global model. Finally, each building can locally finetune the global model to create a personalized forecasting model.

B. Improved Differentiable Architecture Search

This section will first introduce the origin DARTS, then discuss its shortcomings and recommend improved DARTS.

1) DARTS: Liu et al. introduced DARTS to relax the search space to be continuous and search by gradient descent [7] in order to effectively search neural architectures. A cell architecture to be searched is defined by a directed acyclic graph with a series of N representation nodes $x^{(i)}$, and edges $o^{(i,j)}$ that represents operations connecting nodes $x^{(i)}$ to $x^{(j)}$. Each representation node is associated with all the predecessors and can be calculated as: $x^{(j)} = \sum_{i<j} o^{(i,j)}\left(x^{(i)}\right)$. And final output node x^{out} calculates the output by averaging all of the preceding nodes $x^{out} = \frac{1}{N}\sum x^{(i)}$.

Define the set $\mathbb{O}$ to be the candidates of operation $o(\cdot)$(for recurrent neural cell, this would be *tanh*, *relu*, *sigmoid*, *identity*, and *none*), and the architecture can then be defined by the selection of $o^{(i,j)}$. A softmax relaxation method is used to convert the search space from discrete to continuous. A mix operation $\bar{o}^{(i,j)}$, which takes into account the contributions of all operation candidates, is defined as:

$$\bar{o}^{(i,j)} = \sum_{o\in\mathbb{O}} \frac{\exp\left(\alpha_o^{(i,j)}\right)}{\sum_{o'\in\mathbb{O}}\exp\left(\alpha_{o'}^{(i,j)}\right)}\, o^{(i,j)} \tag{4}$$

where architectural parameter $\alpha_o^{(i,j)}$ is a vector of weight parameters for each candidate of $o^{(i,j)}$. Consequently, the architecture search task is reduced to determine α, which is continuous and can be updated by gradient descent.

An alternative gradient descent method is utilized to solve the objective function (1). Architectural parameters α is updated by descending $\eta_\alpha \nabla_\alpha \mathcal{L}_{val}\left(\omega^*(\alpha), \alpha\right)$, weights ω is updated by descending $\eta_\omega \nabla_\omega \mathcal{L}_{train}(\omega, \alpha)$, where η_α, η_ω is the corresponding learning rate. An approximation is applied while computing the gradient of the architectural parameter:

$$\nabla_\alpha \mathcal{L}_{val}\left(\omega^*(\alpha), \alpha\right) \approx \nabla_\alpha \mathcal{L}_{val}\left(\omega - \xi \nabla_\omega \mathcal{L}_{train}(\omega, \alpha), \alpha\right) \tag{5}$$

where the inner optimization term $\omega^*(\alpha)$ is approximated by a single update step $\omega - \xi \nabla_\omega \mathcal{L}_{train}(\omega, \alpha)$ with the learning rate ξ. After the search of α, the final discrete architecture is derived by substituting the mixed operation

$\bar{o}^{(i,j)}$ of edge (i,j) with the highest weighted operation $o^{(i,j)} = \arg\max_{o\in\mathbb{O}}\alpha_o^{(i,j)}$.

2) Improved DARTS: Despite the effectiveness of DARTS in NAS, there exist three main drawbacks. Firstly, as noted by [33], transforming the mixed architecture into a discrete one will result in a discretization gap, which will cause performance to collapse. This is due to the fact that the discretization of architecture will exaggerate the importance of a single operation while ignoring others that are just as significant but have slightly less weight. Additionally, the discretization process will eliminate the operation *none*, which stands for not performing any operation between two nodes. As a result, the connectivity information in the neural architecture is lost. Secondly, DARTS focuses on the tasks of computer vision and natural language processing. However, a dedicated DARTS approach for time series forecasting is lacking. Thirdly, a number of studies have noted that DARTS suffers from overfitting during the searching process [33] [34].

Thus, we proposed an improved version of DARTS, denoted as DARTS*. The strategy can be summarized as:

(i) ***Design a DARTS* search space specifically for load forecasting task:*** LSTM is a well-developed recurrent neural network, which shows great efficacy in time series forecasting [35]. We leverage the strength of LSTM and design DARTS* with LSTM-block as the operation candidate, which can be defined as:

$$\mathcal{I}_{(t)} = \text{sigmoid}\left(W_{xi}^T x_{(t)} + W_{hi}^T h_{(t-1)} + b_i\right) \tag{6}$$

$$\mathcal{F}_{(t)} = \text{sigmoid}\left(W_{xf}^T x_{(t)} + W_{hf}^T h_{(t-1)} + b_f\right) \tag{7}$$

$$\mathcal{O}_{(t)} = \text{sigmoid}\left(W_{xo}^T x_{(t)} + W_{ho}^T h_{(t-1)} + b_o\right) \tag{8}$$

$$g_{(t)} = o\left(W_{xg}^T x_{(t)} + W_{hg}^T h_{(t-1)} + b_g\right) \tag{9}$$

$$c_{(t)} = \mathcal{F}_{(t)} \otimes c_{(t-1)} + \mathcal{I}_{(t)} \otimes g_{(t)} \tag{10}$$

$$y_{(t)} = h_{(t)} = \mathcal{O}_{(t)} \otimes o\left(c_{(t)}\right) \tag{11}$$

where the inputs at a specific timestep are input $x_{(t)}$, hidden state $h_{(t-1)}$ and cell state $c_{(t-1)}$ from the previous timestep. The intermediate state is calculated as $g_{(t)}$, and output is $y_{(t)}$. $\mathcal{I}$, $\mathcal{F}$, $\mathcal{O}$ denotes input gate, forget gate, and output gate respectively, and $\otimes$ denotes Hadamard product. W_{xi}, W_{xf}, W_{xo}, W_{xg} are weights related to x, W_{hi}, W_{hf}, W_{ho}, W_{hg} are weights related to h, and b_i, b_f, b_o, b_g are corresponding bias terms. $o \in \{\text{sigmoid}, \text{tanh}, \text{relu}, \text{gelu}, \text{identity}, \text{none}\}$ is the operation to be searched, then the edge (i,j) in DARTS* is defined by LSTM-block with different operations, and relaxation of every edge by Eq. (4) forms the continuous search space.

(ii) ***Retain the mixing architecture and connectivity information:*** In order to prevent discretization gaps and performance collapse, we retain the mix operation $\bar{o}^{(i,j)}$, which weights the contributions of all operation candidates, including the operation *none*, which indicates connectivity in the architecture.

(iii) ***Adopt a practice of early-stopping in the searching stage:*** Zela et al. suggest an eigenvalue-based early-stopping technique to address the overfitting issue [33]. However, their approach calls for the calculation of a Hessian matrix, which involves more computation and increases the search time. In this paper, we suggest directly undertaking early-stopping of searching in accordance with $\mathcal{L}_{val}$. The experiment findings support this straightforward but efficient strategy, which doesn't necessitate heavy additional computing.

Algorithm 1: Improved DARTS (DARTS*)

```
function DARTS*($\eta_\alpha$, $\eta_\omega$, $\xi$, $E$, $N_p$):
    Generate LSTM block-based search space;
    Generate $\bar{o}^{(i,j)}$ parameterized by $\alpha^{(i,j)}$ for each
      edge $(i,j)$;
    for each search epoch $i \in E$ do
        $\alpha \leftarrow \alpha - \eta_\alpha \nabla_\alpha \mathcal{L}_{val}\left(\omega - \xi \nabla_\omega \mathcal{L}_{train}\left(\omega, \alpha\right), \alpha\right)$
          ($\xi = 0$ if first-order approximation);
        $\omega \leftarrow \omega - \eta_\omega \nabla_\omega \mathcal{L}_{train}\left(\omega, \alpha\right)$;
        if $\mathcal{L}_{val}$ doesn't decrease for $N_p$ epochs then
            break
return $\alpha$, $\omega$
```

Algorithm 1 elaborates the implementation of DARTS*, and the comparison of DARTS and DARTS* network structure is shown in Fig. 2. The original DARTS search at the micro-level, which designs DARTS-RNN cell architecture and sequentially stacks the copies of the designed cell to form the network. In contrast, DARTS* searches at the macro-level, which designs the network architecture with the LSTM block.

C. Federated-DARTS Model

As aforementioned, FL enables effective collaboration among several clients while protecting privacy. Motivated by this, we propose Federated-DARTS (FedDARTS), which is a decentralized scheme for DARTS*.

Assume K buildings, denoted as a set $\mathbb{K}$, participate in the FL scheme, each building owns local dataset $\mathcal{D}_k$ with N_k samples, and the total number of data samples from all the buildings is $N = \sum_{k=1}^{K} N_k$. The objective functions (1) are then transform into:

$$\min_{\alpha} \sum_{k=1}^{K} \frac{N_k}{N} \mathcal{L}_{\alpha}^{k}\left(\omega^*\left(\alpha\right), \alpha\right) \tag{12}$$

$$\text{s.t. } \omega^*\left(\alpha\right) = \arg\min\nolimits_{\omega} \sum_{k=1}^{K} \frac{N_k}{N} \mathcal{L}_{\omega}^{k}\left(\omega, \alpha\right) \tag{13}$$

With the coordination of a server, which is a trust-worthy third party, the problem can be solved by **Algorithm 2**. Firstly, initial architectural parameters α_0 and weights ω_0 will be shared across all the clients. Next, at each federated communication round $r \in R_{search}$, each client k will parallel update their model α_k, ω_k by local data and predefined hyperparameters (including learning rate η_α, η_ω, ξ, and number of local search epochs E) by **Algorithm 1**. After local updating, each client will return updated model parameters and the outcome of local validation result $\mathcal{L}_{val}^{k}$ to the server. Then, the server conducts parameters aggregation $\bar{\alpha} \leftarrow \sum \frac{N_k}{N} \alpha_k$, $\bar{\omega} \leftarrow \sum \frac{N_k}{N} \omega_k$ as global update and distributes to every client. The server will compute a global validation

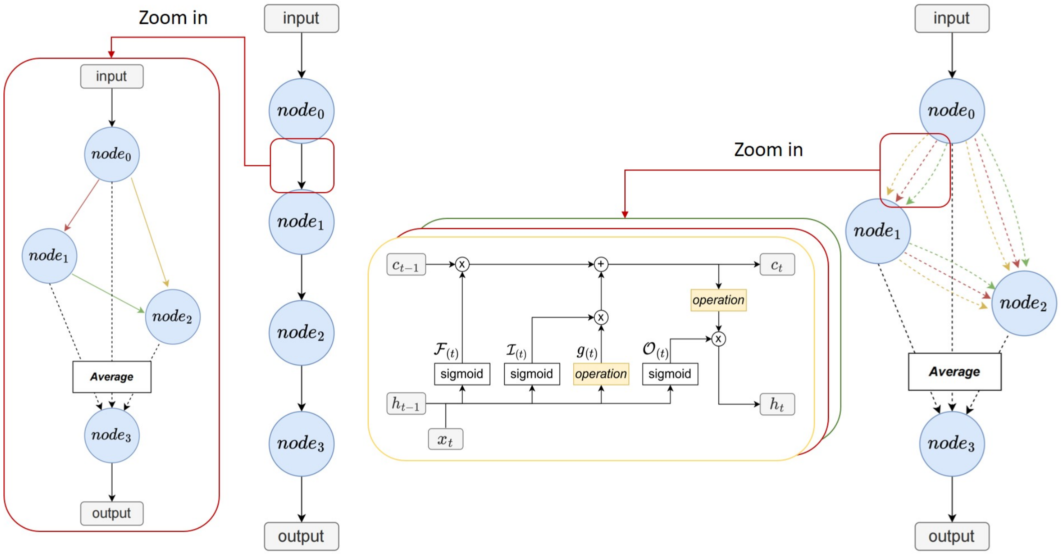


Fig. 2. The structure comparison of DARTS with DARTS*. Left: DARTS, which has a discrete architecture. Right: DARTS*, which is based on LSTM-block and retains the mixing architecture.

Algorithm 2: Federated-DARTS.

```
function FedDARTS($\eta_\alpha$, $\eta_\omega$, $\xi$, $E$, $N_p$, $R_{search}$, $\mathbb{K}$):
    Initialize: $\mathbb{K} \leftarrow \alpha_0,\ \omega_0$;
    for each round $r \in R_{search}$ do
        for each client $k \in \mathbb{K}$ do
            $\alpha_k,\ \omega_k \leftarrow$ DARTS*($\eta_\alpha$, $\eta_\omega$, $\xi$, $E$, $N_p$);
            Compute $\mathcal{L}_{val}^k$ on validation set;
            Return $\alpha_k$, $\omega_k$, $\mathcal{L}_{val}^k$ to Server;
        $\bar{\alpha} \leftarrow \sum \frac{N_k}{N} \alpha_k$;
        $\bar{\omega} \leftarrow \sum \frac{N_k}{N} \omega_k$;
        Parameters distribute: $\mathbb{K} \leftarrow \bar{\alpha},\ \bar{\omega}$;
        $\bar{\mathcal{L}} \leftarrow \sum \frac{N_k}{N} \mathcal{L}_{val}^k$;
        if $\bar{\mathcal{L}}$ doesn't decrease for $N_p$ epochs then
            break
return $\bar{\alpha}$, $\bar{\omega}$
```

loss $\bar{\mathcal{L}} \leftarrow \sum \frac{N_k}{N} \mathcal{L}_{val}^k$. If the early-stop criteria are met, which can be defined as the global validation loss $\bar{\mathcal{L}}$ not decreasing over a predetermined number of rounds, the procedure will be stopped and obtain the optimal global model architecture. Each client should preserve this searched architecture, then retrain their models by only updating the weights parameters. Clients can retrain the model in a federated fashion under the privacy-preservation scenario using **Algorithm 2**, only to communicate and update weights ω_k.

The privacy issue can be addressed by the suggested Federated-DARTS approach, which enables many buildings to cooperatively explore a high-performance neural architecture by merely exchanging the intermediate parameters without disclosing the raw data of the buildings.

D. Two-Stage Personalization

Federated-DARTS can search a global model architecture, but the ultimate objective is to develop a customized forecasting model for every single building. In order to develop a personalized model from the perspective of both architecture and inner weights parameters, we offer a two-stage personalization method, which is elaborated as follows:

Architecture-based clustering: The model architecture can be represented by the architectural parameters α after executing DARTS. Buildings can then be organized into groups depending on the similarity of their architectural features. We advise utilizing the k-means++ algorithm [36] because of its strong qualities in terms of robustness and convergence (note that the clustering method is not constrained, other approaches like k-means, and hierarchical clustering can also be taken). The number of clusters can be determined by the grid-search strategy, and the candidate with the best performance on the validation set is chosen.

Consider buildings are grouped into C clusters, denoted as set $\mathbb{C}$. Buildings in the cluster c are denoted as $\mathbb{K}_c$. Different clusters will conduct Federated-DARTS, and C distinct global model architectures $\bar{\alpha}_c$ will be searched. Model weights of each cluster will also be updated federatedly after each cluster's model architecture has been searched. For each building cluster, a cluster global model with parameters $\bar{\alpha}_c$ and $\bar{\omega}_c$ will be created by the end of the process.

The proposed architecture-based clustering method just needs the communication of architectural parameters that will not breach privacy, as opposed to requiring ground-truth data from buildings. The buildings can then be divided into various groups, and personalized models can be constructed for each building group. In contrast to [21], which suggests federated-clustering, our architecture-based approach offers novel suggestions for privacy-preserving clustering without requiring the computational-expensive federated clustering process.

Local model fine-tuning: Despite the fact that global models are personalized for each building cluster, each building in the cluster has the same model. We recommend a local model fine-tuning method that enables the model to be even more individually personalized to each building.

After receiving the clustered global model, clients within the cluster can utilize their data to locally fine-tune the model weights parameters by gradient descent. Only a small number of training epochs will be conducted for the fine-tuning phase to avoid creating a big difference between the local and global models, which could nullify the learning from the federated process. As a result, this fine-tuning strategy only requires a small amount of additional computation but exhibits high effectiveness that is supported by experiments.

E. Full Algorithm

There are three questions being considered when we design the full algorithm: (1) How to construct effective model architectures and train high-performance models through the collaboration of multiple buildings without violation of privacy? (2) How to construct effective global models in scenarios with heterogeneous consumption patterns? (3) How to conduct the personalization to account for individual building's specific load characteristics? We tackle the above challenges by integrating the two-stage personalization technique into the Federated-DARTS model and propose the Personalized Federated-DARTS model (PFedDARTS), which enables the construction of cluster-level personalized model architecture and individual-level personalized model while preserving privacy. First, all model parameters for clients are initialized. Then, during a certain number of pre-search epochs E_{pre}, each client will locally search the model architecture (see **Algorithm 1**). By obtaining information of each client's architectural parameters α_k, the server can then cluster all of the clients using an architecture-based clustering method. After the clusters $\mathbb{C}$ are formed, each cluster can run Federated-DARTS (see **Algorithm 2**) concurrently to find the best model architecture to serve as the cluster's global architecture for clients. Subsequently, the architecture will then be maintained and every client will federatedly re-update their weight parameters. After the early-stop mechanism stops the federated training process, clients will use their local data to finetune the global model and personalize it to individuals.

The FL approach and the improved DARTS are well integrated into the full algorithm to enhance each other. On the one hand, the model architecture information obtained by DARTS allows for the construction of several building clusters, and the global model for each building cluster can then be better constructed by applying the clustered federated learning approach as opposed to a vanilla global federated learning method. On the other hand, the federated learning framework enables multiple buildings to collaboratively design a more appropriate model architecture and train a better model than the local approach. Furthermore, the integration of the local finetuning approach enables the model to adapt to the local data distribution and capture individual building's load characteristics. The implementation details of the proposed PFedDARTS can be found in **Algorithm 3**.

Algorithm 3: Personalized Federated-DARTS.

```
function PFedDARTS($\eta_\alpha$, $\eta_\omega$, $\xi$, $E$, $E_{pre}$, $E_{ft}$, $N_p$, $R_{search}$, $R_{train}$, $\mathbb{K}$, $C$):
    Initialize: $\mathbb{K} \leftarrow \alpha_0,\ \omega_0$;
    Local pre-search:
    for each client $k \in \mathbb{K}$ do
        $\alpha_k,\ \omega_k \leftarrow$ DARTS*($\eta_\alpha$, $\eta_\omega$, $\xi$, $E_{pre}$, $N_p$)
        Return $\alpha_k,\ \omega_k$ to Server;
    $\mathbb{C} \leftarrow ArchBasedCluster\,(\mathbb{K}, \alpha, C)$;
    for each cluster $c \in \mathbb{C}$ do
        $\bar{\alpha}_c, \bar{\omega}_c \leftarrow$ FedDARTS($\eta_\alpha$, $\eta_\omega$, $\xi$, $E$, $N_p$, $R_{search}$, $\mathbb{K}_c$)
        Recompile: $\mathbb{K}_c \leftarrow \bar{\alpha}_c,\ \omega_0$;
        for each round $r \in R_{train}$ do
            for each client $k \in \mathbb{K}_c$ do
                $\omega_k^c \leftarrow \omega_k^c - \eta_\omega \nabla_\omega \mathcal{L}_\omega\,(\omega_k^c, \bar{\alpha}_c)$;
                Compute $\mathcal{L}_{val}^k$ on validation set;
                Return $\omega_k^c$, $\mathcal{L}_{val}^k$ to Server;
            $\bar{\omega}_c \leftarrow \sum \frac{N_k^c}{N_c} \omega_k^c$;
            Parameters distribute: $\mathbb{K}_c \leftarrow \bar{\omega}_c$;
            $\bar{\mathcal{L}} \leftarrow \sum \frac{N_k^c}{N_c} \mathcal{L}_{val}^k$;
            if $\bar{\mathcal{L}}$ does not decrease for $N_p$ epochs then
                Local fine-tune:
                for each client $k \in \mathbb{K}_c$ do
                    for each fine-tune epoch $i \in E_{ft}$ do
                        $\omega_k^c \leftarrow \omega_k^c - \eta_\omega \nabla_\omega \mathcal{L}_\omega\,(\omega_k^c, \bar{\alpha}_c)$
                break
```

IV. Case Studies

In this section, we use an open dataset BDG2 [37] to test our model for 24-hour-ahead electricity load forecasting of individual buildings. Two case studies are involved in this section. In the first case study, an experiment including 10 buildings, with two of each type of usage, is carried out to evaluate the effectiveness of the PFedDARTS model on each individual building. In the second case study, an experiment including 42 buildings of 6 different usage categories (at least 3 buildings of each type) is conducted to examine the scalability of the PFedDARTS model and its performance in scenarios with more heterogeneous consumption behaviors.

A. Experimental Setups

1) Data Description: The open dataset BDG2 is used for the case study. This dataset collects energy consumption data from 3,053 smart meters installed in 1,636 different types of buildings across North America and Europe, as well as weather data such as temperature, humidity, and precipitation. The dataset spans two years (from January 1, 2016, to December 31, 2017), with data sampled every hour. Negative outliers, large outliers, duplicates, and continuous vacancies are removed from the dataset.

In the first case study, ten buildings of five different types are chosen at random, as shown in Table I. The last month of

TABLE I
THE SELECTED BUILDINGS

Location	Type	Dataset
USA	Office	Bessie
		Napoleon
	Education	Rachael
		Madge
	Industry	Joanne
		Jeremy
	Lodging	Ora
		Shanti
	Public	Gerard
		Crystal

the dataset is designated as the test set, while the rest dataset is divided into training and validation sets with a ratio of 1:1 in the searching stage, and 8:2 in the training stage.

2) Evaluation Metrics: Firstly, three commonly used measures are used to assess the individual building load forecasting accuracy: mean absolute error (MAE), root mean square error (RMSE), and absolute mean percentage error (MAPE).

In addition, as suggested in [38], it is important to predict the load peak around the correct times in building-level forecasting. Therefore, we adopted an adjusted p-norm error E_p^d shown as (17), where P is a permutation matrix allowing the displacement of the forecasting values in time. We restrict the displacement magnitude by setting $P_{ij} = 0$ for $|i-j| > d$. Specifically, we set $d = 1$ and use 2-norm to calculate the adjusted error, which permits displacement of the forecasts by one hour ahead or delay of the original forecast time. Thus, the adjusted error is calculated by solving the minimization problem over the complete set of restricted permutations $\mathbb{P}$. The adjusted error is computed for every 24-hour forecasting, and the average adjusted error over the whole forecasting time range is computed as the final metric.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i| \tag{14}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2} \tag{15}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y}_i}{y_i}\right| \tag{16}$$

$$E_p^d = \min_{P \in \mathbb{P}} \|Y - P\hat{Y}\|_p \tag{17}$$

3) Model Setups: Tables II and Table III show the setups of local models and federated models, respectively, where local models only use local data for training.

All the models have the same feature engineering, where input $\mathbf{X} = \begin{bmatrix} f_1 & f_2 & \cdots & f_7 \end{bmatrix}$ is a series of timesteps containing a week of historical data ahead of the forecasting point. Each timestep is a vector of factors $f = \begin{bmatrix} \text{Cal} & \text{Env} & \text{Holi} & \text{Hist} \end{bmatrix}$, where calendar information is encoded as $\text{Cal} = \begin{bmatrix} \text{month} & \text{weekday} & \text{day} & \text{hour} \end{bmatrix}$, environmental factor Env represents air temperature (which is substantially influential to building loads), Holi is the one-hot encoding of whether it is a holiday, and Hist is the historical load record.

To eliminate the effect of model performance randomness on the fair comparison, the experiments are repeated ten times and the average results are presented.

B. Comparative Studies

Table IV-VII present the performance comparison in terms of MAE, RMSE, MAPE, and the adjusted error respectively. The local models that only use local data, including benchmark LSTM, origin DARTS, and proposed DARTS*, are compared in the table's left half. The table's right half compares local LSTM, Federated-LSTM (FedLSTM), and PFedDARTS. The proposed PFedDARTS model's performance improvement is highlighted in bold, and the average performance across all buildings is displayed in italics. Besides, Table VIII displays a comparison of the personalization strategies. Each building's top performance is displayed in bold, while the average performance is displayed in italics.

1) Effectiveness of DARTS:* Comparing DARTS* to the benchmark LSTM, the experiments show that DARTS* generally outperforms LSTM on each individual building in all metrics. In terms of MAE, RMSE, MAPE, and the adjusted error, DARTS* outperforms LSTM by 1.97%, 0.86%, 3.93%, and 1.39%, on average. However, DARTS* shows a minor degradation on some specific buildings, such as Joanne and Shanti, because architecture search requires extensive data, and individual buildings own limited local data, which may be insufficient for designing an appropriate model architecture.

The results also show that DARTS* outperforms the original DARTS, proving the efficacy of the strategies used to build DARTS*. Furthermore, the early-stop mechanism allows DARTS* to find optimal architecture more robustly and in less time than the origin DARTS.

Fig. 3 shows the forecasting result of Napoleon, which is an office building. Despite incorrect forecasting during the Christmas holiday due to a lack of relative historical data for model training, the DARTS* model performs better in capturing time-series characteristics, particularly peak and valley values. The superiority of DARTS* in forecasting can also be found on other buildings.

2) Effectiveness of Federated Approach: In order to compare the performance of federated models fairly, we also incorporate a local fine-tuning personalization strategy into the FedLSTM model.

The experiments show that the PFedDARTS model largely outperforms the local LSTM model by an average improvement of 6.33%, 3.12%, 10.22%, and 6.14% in MAE, RMSE, MAPE, and the adjusted error respectively. It is worth noting that after implementing PFedDARTS, the aforementioned underperforming buildings Joanne and Shanti can have comparable or even better performance than the local LSTM model, demonstrating the effectiveness of the proposed method in enabling the collaboration of multiple buildings in constructing models with better performance.

When compared to the FedLSTM model, PFedDARTS achieves an average performance of 7.81%, 3.41%, 7.93%, and 6.02% in MAE, RMSE, MAPE, and the adjusted error respectively. It can be observed that even though a local fine-tuning personalization strategy is applied with FedLSTM, the

TABLE II
LOCAL MODEL PARAMETERS

Model	Architecture Parameters				Training Parameters		
	architecture	hidden units	DARTS nodes	architecture discrete	search epochs	train epochs	optimizer
LSTM	3 LSTM layers + 1 Dense layer	10	NA	NA	NA	150(early stop)	adam
DARTS	Search space: {none, sigmoid, tanh, relu, gelu, identity}		3	yes	150	150	
DARTS*				no	150(early stop)	150(early stop)	

TABLE III
FEDERATED MODEL PARAMETERS

Model	Architecture Parameters				Training Parameters					
	architecture	hidden units	DARTS nodes	architecture discrete	pre-search epochs	clusters	fed search rounds	fed train rounds	finetune epochs	optimizer
FedLSTM	3 LSTM layers + 1 Dense layer	10	NA	NA	NA	NA	NA	150 (early stop)	150 (early stop)	adam
PFedDARTS	Search space: {none, sigmoid, tanh, relu, gelu, identity}		3	no	15	3	150 (early stop)			

TABLE IV
MODEL PERFORMANCE IN TERMS OF MAE

Dataset	Local Model					Federated Model			
	MAE			DARTS* Improvement		MAE		PFedDARTS Improvement	
	LSTM	DARTS	DARTS*	vs LSTM	vs DARTS	FedLSTM	PFedDARTS	vs LSTM	vs FedLSTM
Bessie	5.073	5.212	4.988	1.68%	4.30%	4.596	4.806	**5.28%**	-4.56%
Napoleon	11.148	10.366	9.876	11.41%	4.73%	12.077	9.016	**19.12%**	**25.34%**
Rachael	6.257	5.862	5.630	10.02%	3.96%	6.281	5.513	**11.90%**	**12.23%**
Madge	9.919	10.188	9.810	1.09%	3.71%	11.657	8.818	**11.10%**	**24.36%**
Joanne	13.255	14.046	14.032	-5.86%	0.10%	12.633	12.519	**5.55%**	**0.90%**
Jeremy	4.641	4.908	4.760	-2.56%	3.02%	5.799	4.595	**1.01%**	**20.77%**
Ora	12.834	13.042	12.668	1.29%	2.86%	14.368	12.030	**6.27%**	**16.28%**
Shanti	24.310	24.428	24.898	-2.42%	-1.93%	24.851	24.262	**0.20%**	**2.37%**
Gerard	4.550	3.739	3.740	17.81%	-0.01%	3.655	3.404	**25.19%**	**6.88%**
Crystal	16.755	17.069	16.194	3.35%	5.13%	14.574	16.900	-0.87%	-15.96%
Average	*10.874*	*10.886*	*10.660*	*1.97%*	*2.08%*	*11.049*	*10.186*	*6.33%*	*7.81%*

highly non-iid data from various buildings causes performance degradation of the federated approach. In contrast, by implementing the proposed two-stage personalization strategy, the issue of heterogeneous data can be addressed, and nearly all buildings can benefit from PFedDARTS, and an example of forecasting on an office building is shown in Fig. 4. It can be seen that the local LSTM model can only represent a rough prediction and fails to capture load volatility. The PFedDARTS model, on the other hand, can deal with load volatility better. This is due to the fact that the PFedDARTS approach uses multiple data resources for model training, which effectively improves the model's forecasting ability. Furthermore, the two-personalization strategy enables the model to accurately capture the specific characteristics of this building load.

3) Effectiveness of Personalization Approach: In order to further explore the effectiveness of each part of the personalization strategy, we compare the FedDARTS model with four different personalization configurations: 1) without clustering and local fine-tuning 2) with clustering but without local fine-tuning 3) without clustering but with local fine-tuning 4) with both clustering and local fine-tuning. The performance comparison can be found in Table VIII.

The results show that the FedDARTS model performs poorly when no personalization strategy is used. FedDARTS performance can be greatly improved by implementing one of the two personalization strategies, with the full two-stage personalization strategy providing the best results. In terms of model performance improvement, the local fine-tuning strategy outperforms the architecture-based clustering strategy. However, because only a small number of buildings are considered in this case study, the efficacy of the clustering strategy may be greater demonstrated in a scenario involving a large number of buildings, which will be shown in the second case study.

The performance of FedDARTS with different personalization strategies on building Napoleon is shown in Fig. 5. When no fine-tuning is adopted, it can be observed that the model can not perform well, especially during the peak load period. In comparison, the load curves can be better fitted if a fine-tuning strategy is adopted. In particular, when combing clustering and fine-tuning, the model performs well not only during the peak and valley periods but also during the transition period. This indicates the suggested two-stage personalization strategy is both necessary and effective.

C. Scalability Studies

We examine the model performance in a larger-scale scenario with more buildings included in order to investigate the

TABLE V
MODEL PERFORMANCE IN TERMS OF RMSE

Dataset	Local Model					Federated Model			
	RMSE			DARTS* Improvement		RMSE		PFedDARTS Improvement	
	LSTM	DARTS	DARTS*	vs LSTM	vs DARTS	FedLSTM	PFedDARTS	vs LSTM	vs FedLSTM
Bessie	7.174	7.510	7.028	2.03%	6.42%	6.744	6.834	**4.73%**	-1.34%
Napoleon	14.780	14.803	14.250	3.59%	3.74%	15.430	12.620	**14.62%**	**18.21%**
Rachael	8.057	8.061	7.873	2.28%	2.33%	8.264	7.781	**3.43%**	**5.85%**
Madge	13.018	12.944	12.819	1.53%	0.97%	14.488	11.389	**12.51%**	**21.39%**
Joanne	20.745	21.084	21.053	-1.49%	0.15%	20.525	20.482	**1.27%**	**0.21%**
Jeremy	8.681	8.628	8.579	1.17%	0.57%	9.261	8.518	**1.87%**	**8.02%**
Ora	17.197	17.501	16.979	1.27%	2.98%	18.237	16.186	**5.88%**	**11.25%**
Shanti	33.719	34.334	34.806	-3.22%	-1.37%	35.721	35.698	-5.87%	**0.06%**
Gerard	5.833	4.792	4.869	16.54%	-1.60%	4.640	4.663	**20.06%**	-0.49%
Crystal	27.050	26.504	26.653	1.47%	-0.56%	23.418	27.209	-0.59%	-16.19%
Average	*15.625*	*15.616*	*15.491*	*0.86%*	*0.80%*	*15.673*	*15.138*	*3.12%*	*3.41%*

TABLE VI
MODEL PERFORMANCE IN TERMS OF MAPE

Dataset	Local Model					Federated Model			
	MAPE			DARTS* Improvement		MAPE		PFedDARTS Improvement	
	LSTM	DARTS	DARTS*	vs LSTM	vs DARTS	FedLSTM	PFedDARTS	vs LSTM	vs FedLSTM
Bessie	8.107	8.475	7.882	2.77%	7.00%	7.230	7.769	**4.16%**	-7.46%
Napoleon	8.592	8.090	7.739	9.93%	4.34%	9.154	6.980	**18.77%**	**23.75%**
Rachael	12.722	11.801	11.241	11.64%	4.75%	12.665	10.906	**14.27%**	**13.89%**
Madge	10.727	10.919	10.460	2.49%	4.21%	12.391	9.198	**14.25%**	**25.77%**
Joanne	9.905	10.437	10.462	-5.63%	-0.24%	8.948	8.773	**11.42%**	**1.96%**
Jeremy	11.525	11.862	11.927	-3.49%	-0.55%	11.925	11.441	**0.74%**	**4.06%**
Ora	6.033	5.895	5.860	2.85%	0.58%	6.502	5.476	**9.23%**	**15.77%**
Shanti	8.754	8.768	8.920	-1.89%	-1.73%	8.445	8.638	**1.33%**	-2.29%
Gerard	10.097	8.693	8.401	16.79%	3.35%	7.988	7.479	**25.92%**	**6.37%**
Crystal	7.833	8.034	7.692	1.80%	4.26%	6.707	8.001	-2.14%	-19.28%
Average	*9.429*	*9.297*	*9.058*	*3.93%*	*2.57%*	*9.196*	*8.466*	*10.22%*	*7.93%*

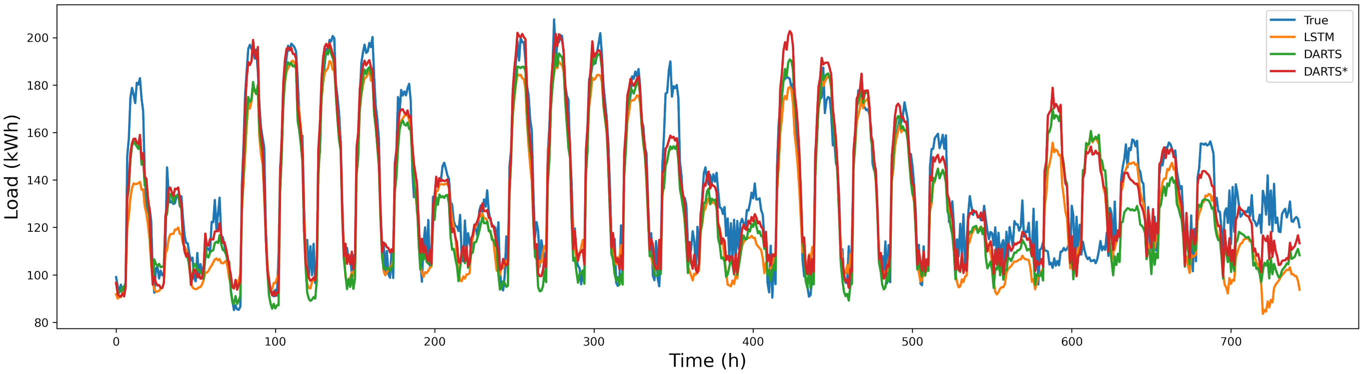


Fig. 3. The forecasting performance comparison of three local models: LSTM, DARTS, and DARTS* on building Napoleon (office).

scalability of the proposed approach. After removing buildings with a high percentage of invalid data, we included a total of 42 buildings in one site of various types in this case study, the statistics are shown in Table IX.

1) Model Performance: The average performance over 42 buildings is investigated, and the performance comparison of the PFedDARTS model with the local LSTM model is shown in Table X. The results show that the PFedDARTS model outperforms the local LSTM model by an improvement of 12.64%, 9.55%, 7.23%, and 15.58% in MAE, RMSE, MAPE, and the adjusted error respectively. Such significant improvements indicate that the PFedDARTS model is more capable of handling heterogeneous consumption patterns than the local modeling approach. An improvement of 15.58% in terms of the adjusted error demonstrates the superior ability of the PFedDARTS in capturing the volatility of building load patterns.

The performance comparison of the local LSTM model and the PFedDARTS on individual buildings of different types can be visualized in Fig. 6, where the x-axis and the y-axis represent the performance of each model in terms of RMSE respectively. The dashed line in the figure denotes that

TABLE VII
MODEL PERFORMANCE IN TERMS OF ADJUSTED ERROR

Dataset	Local Model					Federated Model			
	Adjusted Error			DARTS* Improvement		Adjusted Error		PFedDARTS Improvement	
	LSTM	DARTS	DARTS*	vs LSTM	vs DARTS	FedLSTM	PFedDARTS	vs LSTM	vs FedLSTM
Bessie	6.146	5.977	6.123	0.37%	-2.44%	5.612	5.828	**5.16%**	-3.86%
Napoleon	12.969	12.459	11.568	10.80%	7.16%	13.584	10.052	**22.49%**	**26.00%**
Rachael	7.060	6.683	6.531	7.49%	2.27%	7.048	6.400	**9.34%**	**9.19%**
Madge	11.566	11.371	11.208	3.10%	1.43%	13.043	10.152	**12.23%**	**22.17%**
Joanne	18.942	19.311	19.495	-2.92%	-0.96%	18.231	18.371	**3.01%**	-0.76%
Jeremy	7.682	7.799	7.716	-0.43%	1.07%	8.532	7.633	**0.64%**	**10.54%**
Ora	14.948	15.736	14.701	1.66%	6.58%	16.533	14.412	**3.59%**	**12.83%**
Shanti	28.362	27.732	29.800	-5.07%	-7.46%	28.550	27.651	**2.51%**	**3.15%**
Gerard	5.519	4.489	4.572	17.16%	-1.86%	4.432	4.336	**21.44%**	**2.15%**
Crystal	20.649	20.871	20.265	1.86%	2.90%	18.101	20.789	-0.67%	-14.85%
Average	*13.384*	*13.243*	*13.198*	*1.39%*	*0.34%*	*13.367*	*12.562*	*6.14%*	*6.02%*

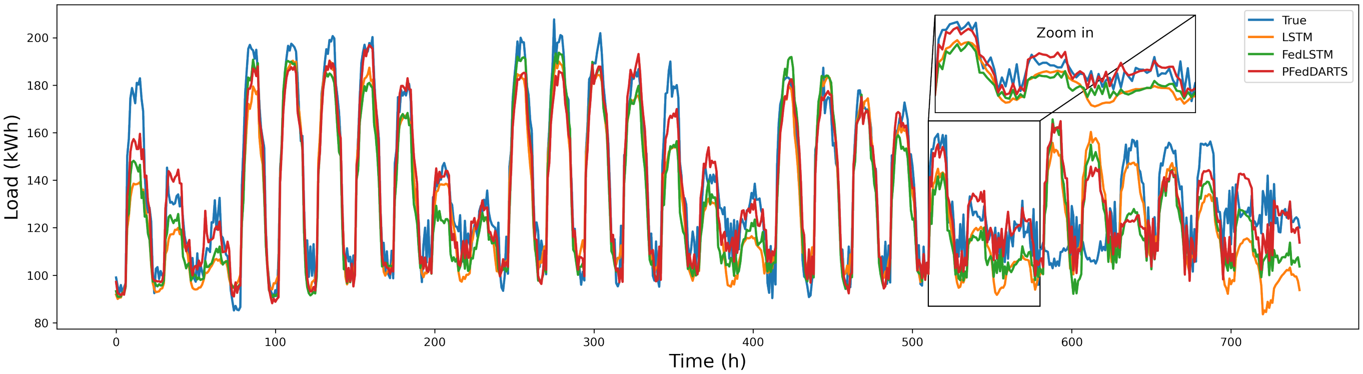


Fig. 4. The forecasting performance comparison of three models: Local-LSTM, FedLSTM, and PFedDARTS on building Napoleon (office).

TABLE VIII
COMPARISON OF FEDERATED-DARTS MODEL WITH DIFFERENT PERSONALIZATION CONFIGURATIONS

Dataset	RMSE			
	o/o	c/o	o/f	c/f
Bessie	9.807	8.216	7.559	**6.834**
Napoleon	13.066	13.334	12.752	**12.620**
Rachael	8.420	8.600	**7.647**	7.781
Madge	10.529	**10.314**	11.175	11.389
Joanne	21.587	25.149	21.588	**20.482**
Jeremy	9.460	8.981	8.692	**8.518**
Ora	22.274	16.717	**15.189**	16.186
Shanti	**31.114**	32.334	36.133	35.698
Gerard	6.152	5.704	5.790	**4.663**
Crystal	26.093	25.291	**24.873**	27.209
Average	15.850	15.464	15.140	**15.138**

[1] o/o : without clustering and fine-tuning.
[2] c/o : with clustering but without fine-tuning.
[3] o/f : without clustering but with fine-tuning.
[4] c/f : with clustering and fine-tuning.

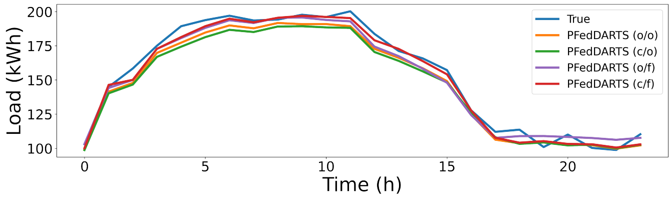


Fig. 5. The comparison of personalization strategies, including PFedDARTS(o/o) without clustering and fine-tuning, PFedDARTS(c/o) with clustering but without fine-tuning, PFedDARTS(o/f) without clustering but with fine-tuning, and PFedDARTS(c/f) with both clustering and fine-tuning, on building Napoleon(office).

TABLE IX
BUILDINGS STATISTICS

Location	Type	Number
Hog	Assembly	3
	Education	6
	Industrial	4
	Lodging	5
	Office	20
	Public	4

both models perform equally well. Buildings on which the PFedDARTS model performs better than the local LSTM model are indicated by the points below the dashed line. The fact that most of the points are below the dashed line, with some of them much below, implies that the PFedDARTS model can perform well on the majority of buildings and achieve significant improvement on a few of them.

2) Model Convergence: After the architecture-based clustering, a total of 5 building clusters are constructed in this case study. Buildings within the cluster collaboratively search for the cluster's global architecture and update it in a federated manner, which requires a number of communication rounds to reach the model convergence. We investigate the convergence process of building clusters in the federated search

TABLE X
COMPARISON OF PFEDDARTS WITH LOCAL LSTM IN A LARGE-SCALE SCENARIO

	Local_LSTM	PFedDARTS	Improvement
MAE	15.28	13.35	**12.64%**
RMSE	20.58	18.62	**9.55%**
MAPE	18.18	16.87	**7.23%**
Adjust Error	18.07	15.26	**15.58%**

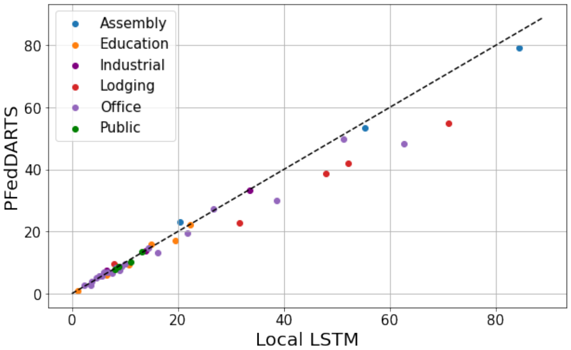


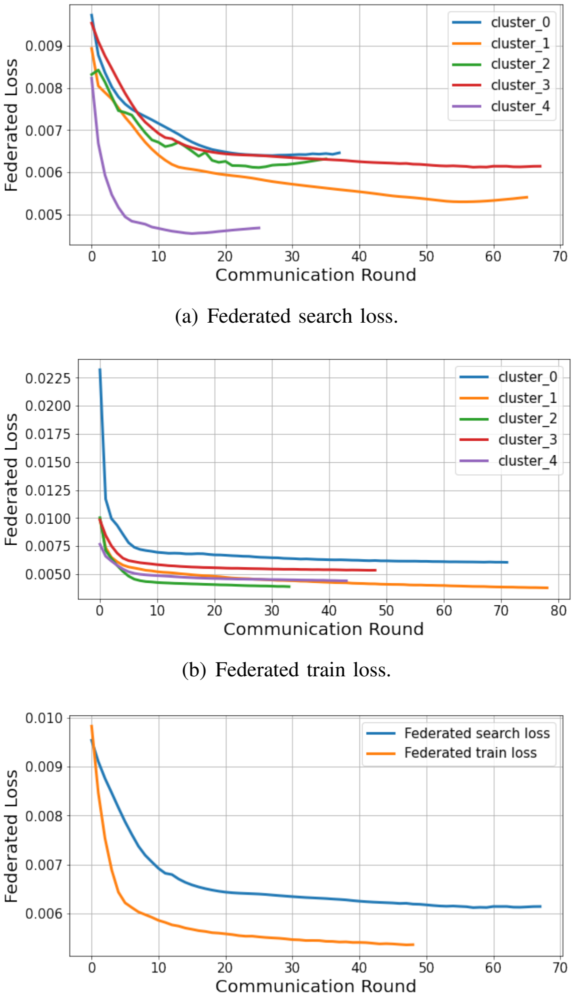
Fig. 6. Comparison of Local-LSTM and PFedDARTS in terms of RMSE. The performance of the Local-LSTM model is indicated by the x-axis, and that of the PFedDARTS model is indicated by the y-axis. Buildings on which the PFedDARTS model performs better than the Local-LSTM model are indicated by the points below the dashed line.

and federated train stage as shown in Fig. 7. The federated process will be early stopped to prevent overfitting, and it can be observed that the model can almost converge after approximately 35 communication rounds. The federated search process has a higher federated loss and requires slightly more communication rounds due to the complexity of finding an appropriate model architecture.

3) Computation Cost: The average time consumption in four stages of the proposed method is shown in Fig. 8. The computing expense is mainly concentrated in the cluster federated search and federated train stages as compared to the local pre-search and local finetune stages. However, in practical implementation, multiple building clusters can carry out federated procedures concurrently, which will be more efficient and reduce computing time. Additionally, be aware that the proposed architecture-based clustering approach may be completed quickly on the server side and does not involve any federation procedures. In conclusion, the proposed approach is feasible and can be implemented with a tolerable computing time.

V. CONCLUSIONS

This paper proposes the Personalized Federated DARTS (PFedDARTS) model to improve load forecasting accuracy on individual buildings. Extensive experiments have been conducted to test the proposed model, and the findings indicate that our model outperforms other local or federated models and greatly improves the forecasting accuracy on various individual buildings. Results suggest that our proposed DARTS* improves upon the original DARTS in terms of robustness and search speed in automatically constructing efficient model architectures. Furthermore, the proposed PFedDARTS enhances the local DARTS* model by making use of data resources from various buildings for better architecture design and model training. Buildings can benefit from the PFedDARTS approach if they fail to design effective models locally. Additionally, the PFedDARTS model is able to better handle load volatility thanks to the two-stage personalization strategy, which effectively captures the heterogeneous load patterns of various



(a) Federated search loss.

(b) Federated train loss.

(c) The convergence process of cluster 3.

Fig. 7. Convergence in cluster federated stage. (a) The Federated search loss of each building cluster. (b) The Federated train loss of each building cluster. (c) The convergence process of one building cluster.

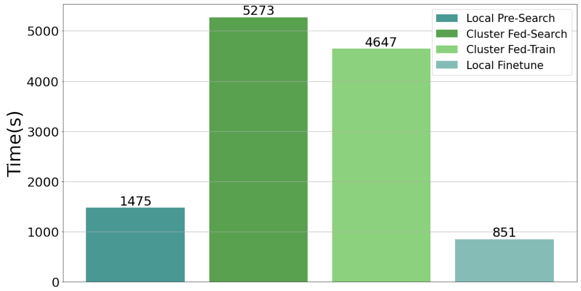


Fig. 8. Average time consumption in four stages, including Local Pre-Search stage, Cluster Fed-Search stage, Cluster Fed-Train stage, and Local Finetune stage.

individual buildings. The results of the scalability studies also indicate that the PFedDARTS model can successfully tackle the heterogeneous consumption challenges for a large number of buildings with a good convergence quality as well as tolerable computation cost.

References

[1] I. Hamilton, O. Rapf, D. J. Kockat, D. S. Zuhaib, T. Abergel, M. Oppermann, M. Otto, S. Loran, I. Fagotto, N. Steurer *et al.*, "2020 global status report for buildings and construction," *United Nations Environmental Programme*, 2020.

[2] C. Gerwig, "Short term load forecasting for residential buildings—an extensive literature review," in *International Conference on Intelligent Decision Technologies*. Springer, 2017, pp. 181–193.

[3] P. Regulation, "General data protection regulation," *Intouch*, vol. 25, 2018.

[4] Data Security Law of the People's Republic of China. [Online]. Available: https://gkml.samr.gov.cn/nsjg/bgt/202111/t20211105_336461.html

[5] A Voluntary Code of Conduct. [Online]. Available: https://www.energy.gov/oe/downloads/data-privacy-and-smart-grid-voluntary-code-conduct

[6] 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.

[7] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[8] B. Nepal, M. Yamaha, A. Yokoe, and T. Yamaji, "Electricity load forecasting using clustering and ARIMA model for energy management in buildings," *Japan Architectural Review*, vol. 3, no. 1, pp. 62–76, 2020.

[9] X. Ke, A. Jiang, and N. Lu, "Load profile analysis and short-term building load forecast for a university campus," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 2016, pp. 1–5.

[10] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2017.

[11] W. Lin, D. Wu, and B. Boulet, "Spatial-temporal residential short-term load forecasting via graph neural networks," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5373–5384, 2021.

[12] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3943–3952, 2018.

[13] M. Afrasiabi, M. Mohammadi, M. Rastegar, L. Stankovic, S. Afrasiabi, and M. Khazaei, "Deep-based conditional probability density function forecasting of residential loads," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3646–3657, 2020.

[14] D. Wu, B. Wang, D. Precup, and B. Boulet, "Multiple kernel learning-based transfer regression for electric load forecasting," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1183–1192, 2020.

[15] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1591–1598, 2015.

[16] Y. Wang, Q. Chen, M. Sun, C. Kang, and Q. Xia, "An ensemble forecasting method for the aggregated load with subprofiles," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3906–3908, 2018.

[17] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—a novel pooling deep RNN," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2017.

[18] Y. Yang, W. Li, T. A. Gulliver, and S. Li, "Bayesian deep learning-based probabilistic load forecasting in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4703–4713, 2019.

[19] J. Lin, J. Ma, and J. Zhu, "A privacy-preserving federated learning method for probabilistic community-level behind-the-meter solar generation disaggregation," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 268–279, 2021.

[20] A. Moradzadeh, H. Moayyed, B. Mohammadi-Ivatloo, A. P. Aguiar, and A. Anvari-Moghaddam, "A secure federated deep learning-based approach for heating load demand forecasting in building environment," *IEEE Access*, vol. 10, pp. 5037–5050, 2021.

[21] Y. Wang, M. Jia, N. Gao, L. Von Krannichfeldt, M. Sun, and G. Hug, "Federated clustering for electricity consumption pattern extraction," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 2425–2439, 2022.

[22] H. Liu and W. Wu, "Federated reinforcement learning for decentralized voltage control in distribution networks," *IEEE Transactions on Smart Grid*, pp. 1–1, 2022.

[23] J.-F. Toubeau, F. Teng, T. Morstyn, L. V. Krannichfeldt, and Y. Wang, "Privacy-preserving probabilistic voltage forecasting in local energy communities," *IEEE Transactions on Smart Grid*, pp. 1–1, 2022.

[24] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[25] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 394–21 405, 2020.

[26] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3557–3568, 2020.

[27] K. M. Ahmed, A. Imteaj, and M. H. Amini, "Federated deep learning for heterogeneous edge computing," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021, pp. 1146–1152.

[28] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[29] Y. Jiang, C. Hu, T. Xiao, C. Zhang, and J. Zhu, "Improved differentiable architecture search for language modeling and named entity recognition," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3585–3590.

[30] X. Wu, D. Zhang, C. Guo, C. He, B. Yang, and C. S. Jensen, "AutoCTS: Automated correlated time series forecasting–extended version," *arXiv preprint arXiv:2112.11174*, 2021.

[31] Q. Li, X. Wu, and T. Liu, "Differentiable neural architecture search for optimal spatial/temporal brain function network decomposition," *Medical Image Analysis*, vol. 69, p. 101974, 2021.

[32] G. Biju, G. N. Pillai, and J. Seshadrinath, "Electric load demand forecasting with RNN cell generated by DARTS," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 2019, pp. 2111–2116.

[33] A. Zela, T. Elsken, T. Saikia, Y. Marrakchi, T. Brox, and F. Hutter, "Understanding and robustifying differentiable architecture search," *arXiv preprint arXiv:1909.09656*, 2019.

[34] H. Liang, S. Zhang, J. Sun, X. He, W. Huang, K. Zhuang, and Z. Li, "Darts+: Improved differentiable architecture search with early stopping," *arXiv preprint arXiv:1909.06035*, 2019.

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

[36] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

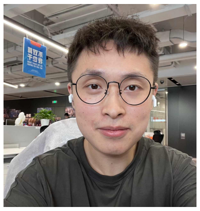
[37] C. Miller, A. Kathirgamanathan, B. Picchetti, P. Arjunan, J. Y. Park, Z. Nagy, P. Raftery, B. W. Hobson, Z. Shi, and F. Meggers, "The building data genome project 2, energy meter data from the ashrae great energy predictor iii competition," *Scientific data*, vol. 7, no. 1, pp. 1–13, 2020.

[38] S. Haben, J. Ward, D. V. Greetham, C. Singleton, and P. Grindrod, "A new error measure for forecasts of household-level, high resolution electrical energy consumption," *International Journal of Forecasting*, vol. 30, no. 2, pp. 246–256, 2014.

Dalin Qin received the B.S. degree in Electrical Engineering and its Automation from South China University of Technology, Guangzhou, China, in 2022. He is now pursuing a Ph.D. degree in Electrical and Electronic Engineering at the University of Hong Kong. His current research interests include energy forecasting and privacy-preserving data analytics in smart grids.

Chenxi Wang received the B.S. degree in Electrical Engineering and its Automation from South China University of Technology, Guangzhou, China, in 2022. He is now pursuing a Ph.D. in Electrical and Electronic Engineering at the University of Hong Kong. His current research interests include energy forecasting and data valuation.

Qingsong Wen (SM'23) is currently a Staff Engineer and Manager at DAMO Academy-Decision Intelligence Lab, Alibaba Group, working in the areas of intelligent time series analysis, data-driven intelligence decisions, machine learning, and signal processing. Before that, he worked at Qualcomm and Marvell in the areas of big data and signal processing, and received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, USA. He has published over 50 top-ranked journal and conference papers, received AAAI/IAAI 2023 Deployed Application Award, and won the First Place in 2022 ICASSP Grand Challenge Competition (AIOps in Networks). He is an Associate Editor for Neurocomputing, Guest Editor for Applied Energy, and regularly served as an SPC/PC member of the major AI and signal processing conferences including AAAI, IJCAI, KDD, ICDM, GLOBECOM, EUSIPCO, etc.



Weiqi Chen is currently an engineer at DAMO Academy Decision Intelligence Lab, Alibaba Group. He received B.S. from Xi'an Jiaotong University and M.S. from Zhejiang University in computer Science an Technology. His research interest is on time series forecasting and anomaly detection with related applications, e.g., electricity forecasting and weather prediction.

Liang Sun is currently a Senior Staff Engineer / Engineering Director at DAMO Academy-Decision Intelligence Lab, Alibaba Group. He received B.S. from Nanjing University and Ph.D. from Arizona State University, both in computer science. Dr. Sun has over 50 publications including 2 books in the fields of machine learning and data mining. His work on dimensionality reduction won the KDD 2010 Best Research Paper Award Honorable Mention, and won the Second Place in KDD Cup 2012 Track 2 Competition. He also won the First Place in 2022 ICASSP Grand Challenge (AIOps in Networks) Competition and received AAAI/IAAI 2023 Deployed Application Award. At Alibaba Group, he is working on building a data-driven decision making cycle in automated business analysis, including data monitoring, insights discovery, diagnosis and root cause analysis, action suggestion, and explainability of the cycle, with an emphasis on time series data.

Yi Wang received the B.S. degree from Huazhong University of Science and Technology in June 2014, and the Ph.D. degree from Tsinghua University in January 2019. He was a visiting student with the University of Washington from March 2017 to April 2018. He served as a Postdoctoral Researcher in the Power Systems Laboratory, ETH Zurich from February 2019 to August 2021.

He is currently an Assistant Professor with the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interests include data analytics in smart grids, energy forecasting, multi-energy systems, Internet-of-things, cyber-physical-social energy systems.