Federated Domain Separation for Distributed Forecasting of Non-IID Household Loads

Nan Lu, Shu Liu, Qingsong Wen, *Senior Member, IEEE*, Qiming Chen, Liang Sun, Yi Wang, *Member, IEEE*

***Abstract*—Household load forecasting is increasingly essential since it enables various demand-side management applications. The federated learning approach is becoming popular for its advantages in fully using different households' load data with privacy preservation. However, due to the non-independent and identically distributed (non-IID) characteristic of each household's local data, the knowledge acquired by local training may have a strong bias. It can introduce contamination and make the global model vulnerable if locally trained models are simply aggregated as traditional FL methods do. To this end, we develop a novel framework that integrates federated domain separation to alleviate the negative effects caused by non-IID data. Specifically, we divide the acquired knowledge into the useful part and potentially contaminating part. By acquiring the former and removing the latter through a well-designed algorithm, a more anti-contamination and more personalized FL model can be expected. Compared to current post-processing personalization methods, the proposed framework can avoid global knowledge forgetting, thus achieving more comprehensive knowledge utilization to give more accurate results. Extensive comparison experiments with benchmarking methods are conducted on a publicly available dataset to validate the superiority of the proposed framework, while a variety of ablation experiments prove the effectiveness of all inner components.**

***Index Terms*—Household load forecasting, federated learning, Non-IID data, domain separation, personalization**

I. Introduction

A. Background and Motivations

Accurate short-term load forecasting (STLF) plays an indispensable role in the efficient operation of smart grids [1]. Household-level STLF is a fundamental tool for developing various demand-side management applications, e.g., electricity retail pricing and microgrid operation [2]. In recent years, advanced metering infrastructure coupled with large-scale smart meter deployments allows utilities to record and collect accurate historical data from customers, thus enabling a vast array of data-driven load forecasting approaches [3].

However, household-level STLF is a challenging task. In the normal course of events, using limited data owned by an individual household to train a reliable model that can effectively capture volatility in load data is extremely hard. A possible solution is to make full use of load data from multiple households. For example, in [4], [5], smart meter data from multiple households are transferred to a central server to perform joint training, thus achieving a high-performance forecasting model.

These approaches require full access to each household's local data, which can impose security and privacy concerns and make compliance with strict data regulations (such as the EU General Data Protection Regulation) difficult [6]. To fully utilize data resources while preserving privacy, a distributed training strategy named federated learning (FL) is proposed [7]. In FL, each client receives a copy of a deep learning-based model and trains it with the local data. Then, the central server performs an aggregation of local models to transform local knowledge into global knowledge. Since the parameters of the model are transferred rather than each client's private data, security and privacy concerns can be avoided.

One of the main concerns of FL-based load forecasting approaches is the presence of non-independent and identically distributed (non-IID) household load data [8]. It means that knowledge acquired by each local training exhibits a certain bias, which can introduce contamination to the global model in the server aggregation process [9], [10]. Moreover, because of the strictly limited data access in the FL scenario, this bias cannot be easily eliminated by direct knowledge processing, which makes accurate household STLF even more challenging.

Up to now, how to alleviate the negative effects imposed by non-IID load data is still under investigation. To address this issue, current works mainly focus on properly utilizing personalized knowledge to make the model adapt to each household's local distribution (e.g., local fine-tuning in [11]). However, there are two factors that may limit their accuracy improvement: unremoved contaminating knowledge caused by knowledge bias during the FL process and inevitable forgetting during the personalized knowledge acquisition stage [12]. In other words, there is still room to enhance accuracy by designing a knowledge utilization method that can tackle these limitations.

Accordingly, this work is focused on one question: *How to make the FL model achieve more accurate forecasts even when dealing with non-IID data?* We are inspired to answer this question by introducing a novel knowledge utilization method: federated domain separation, which can make the model acquire comprehensive useful knowledge while removing potentially contaminating knowledge.

The work was supported in part by the National Key R&D Program of China (2022YFE0141200) and in part by the Research Grants Council of the Hong Kong SAR (HKU 27203723). (*Corresponding author: Yi Wang.*)

Nan Lu, Shu Liu, 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: lnan@connect.hku.hk, u3582420@connect.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).

B. Literature Review

Extensive work has been done on household load forecasting, which can be mainly classified into statistics-based methods and deep learning-based methods. Commonly used statistics-based methods include exponential smoothing models [13], multiple linear regression [14], autoregressive integrated moving average (ARIMA) [15]. They have simple structures and offer operational simplicity. Compared to statistics-based methods, deep learning-based methods can utilize more complex networks to capture volatility and non-stationarity and thus achieve higher accuracy. In [16], a long short-term memory (LSTM) model is proposed for load forecasting, and its superiority has been proven by comprehensive experiments. In [17], multiple models such as LSTM and XGBoost are combined together to improve the robustness.

In recent years, an increasing number of studies have noticed the opportunity to utilize multiple data resources to achieve more accurate results. FL has been one of the most popular methods among them for its unique advantages in cooperation among multiple households, full utilization of data resources, and privacy preservation. Up to now, according to distribution characteristics of data in feature and sample spaces, three FL scenarios have been extensively studied: horizontal FL (same feature space but different sample spaces), vertical FL (different feature spaces but same sample space), and transfer learning FL (different feature spaces and different sample spaces) [18], [19]. Since this paper aims to explore the potential for accuracy and reliability improvement by fully utilizing high-quality samples from different sources, horizontal FL is used in households' load forecasting here.

Conventional horizontal FL-based load forecasting methods can be classified into two categories [20]. The first category is federated stochastic gradient descent (FedSGD), such as [21], [22], in which the server receives each client's gradient to jointly update a global model. The second category is federated averaging (FedAvg), in which the server calculates the weighted average of locally trained model parameters, such as [9], [23], [24]. However, due to households' various electricity consumption habits, load data often exhibits significant non-IID characteristics. This brings challenges to both conventional FedSGD and FedAvg methods since their aggregation process is relatively simple and may introduce contamination (caused by knowledge bias) to global knowledge. Moreover, this data imbalance cannot be easily eliminated by household-to-household communications due to FL privacy-preserving requirements [25].

As an increasing number of researchers have noticed the negative effects imposed by non-IID data, many efforts have been devoted to making full utilization of personalized knowledge to create adaptive FL models. In [11], [26], a local fine-tuning step is designed to participate in an FL-based load forecasting task. By retraining the partial model with the local data, a personalized model that is more applicable to the local distribution can be created. In [27], a domain adaptation method that was often adopted in the transfer learning field [28] is first introduced to the FL-based load forecasting approach. The authors align each client's local data distribution by minimizing the Maximum Mean Discrepancies (MMD) distance [29] in the local fine-tuning process to alleviate the non-IID data's negative influence. A similar method is also seen in [30], which changes to employ correlation alignment [31]. In [32], a domain augmentation method is adopted to improve the model adaptability. In addition, researchers in [33] abandon the idea of training a single global model and instead seek an explicit trade-off between the central model and the personalized model to fit divergent data.

Despite the progress made by the aforementioned methods, their utilization of knowledge is still relatively simple and insufficient. Therefore, two aspects can be investigated to further improve the accuracy. The first one is to remove the potentially contaminating part of global knowledge. The authors in [34] point out that not all knowledge acquired in other households can work for the current household's forecasting task. To address this issue, they first employ alignment-based domain separation networks to realize the contamination removal in a two-domain scenario. However, in such an FL load forecasting task with multiple households, the alignment, which plays an irreplaceable role in domain separation, is hard to perform. [35] proposes a possible way to multi-domain alignment, in which the alignment loss is defined as the summation of the MMD distance between each two domains, which raises concerns about high communication and calculation costs. Since current separation-based methods are not able to work effectively, how to achieve federated domain separation is one of the keys to tackling the non-IID issue.

Another current limitation is that the post-processing-based personalization methods (e.g., [11], [27]) suffer from global knowledge forgetting [12]. This is because the local fine-tuning rewrites the network parameters and somehow nullifies part of the learning from the federated process. Although some of them, like [26], [36], [37] adopt methods such as freezing, reducing the fine-tuning epochs, and clustering to mitigate the forgetting, it is still inevitable. Consequently, personalizing the model while avoiding global knowledge forgetting is also an untapped potential for improving forecasting performance.

C. Contributions and Organization of This Paper

This paper is therefore strongly incentivized to develop a load forecasting framework with a federated domain separation process, aiming at addressing the aforementioned research gaps: unremoved contaminating knowledge in the FL process and inevitable forgetting in the personalization process. The contributions of this work can be concluded as follows:

1) Propose an advanced FL-based household load forecasting framework combined with federated domain separation. It can comprehensively acquire useful knowledge from all households while excluding potentially contaminating parts, thus giving more accurate forecasts even in the presence of non-IID load data.
2) Define a knowledge reference block for knowledge interaction with other households while avoiding the invasion of households' privacy. It also enables a communicationally and computationally efficient multi-domain alignment method.

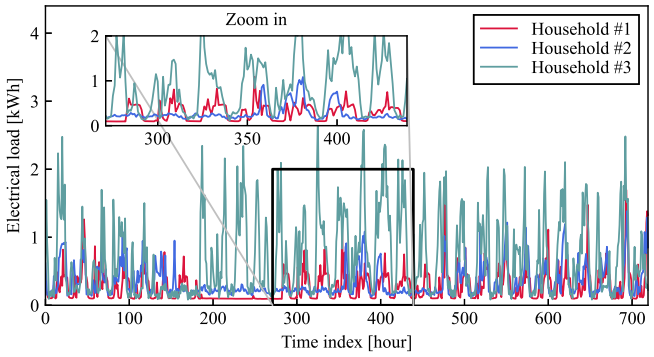


Fig. 1. The load profiles of three different households.

3) Develop a personalization strategy based on a synchronous utilization of information supplement rather than post-network rewriting (e.g., local fine-tuning) to avoid any forgetting of knowledge in the FL process. A personalization module with an attention-based collaboration strategy is additionally constructed to achieve this objective.

The rest of this paper is organized as follows: Section II briefly describes the problems to be solved; Section II introduces the methodology, including a specialized framework and its implementation details; Section IV provides the experimental results and analysis; and Section V concludes our work and points out the possible future work.

II. Problem Statement

This section provides a brief overview of the dataset, states the issues to be addressed, and gives the main objective of this work.

A. Dataset Description

This work is performed on an open smart meter dataset in London, which is gathered under the Low Carbon London project and provided by UK Power Networks [38]. The dataset mainly collects smart meter data of 5567 households, each one containing half-hourly data for several months to three years. As an example, the hourly energy consumption profiles of three selected households are illustrated in Fig. 1. It is observed that the load patterns and distributions vary significantly among households, which makes the data exhibit an obvious non-IID characteristic (especially for load data from the 200-th to 300-th hour). This makes accurate load forecasting extremely challenging for traditional FL-based approaches. In addition to load data, additional hourly data, including humidity, temperature, precipitation, weather, etc., are also provided by darksky API [39], which will be considered in our work.

In the commonly seen FL scenario, each household owns a limited and insufficient amount of local data [2], which means the data from other households is also required to train a strong and reliable model. Besides, to preserve the privacy of the households, the data owned by each cannot be accessed by others or the central server throughout the process. In this work, data from September 1 to November 31 was used for model training and validation, while the data in December was for performance testing.

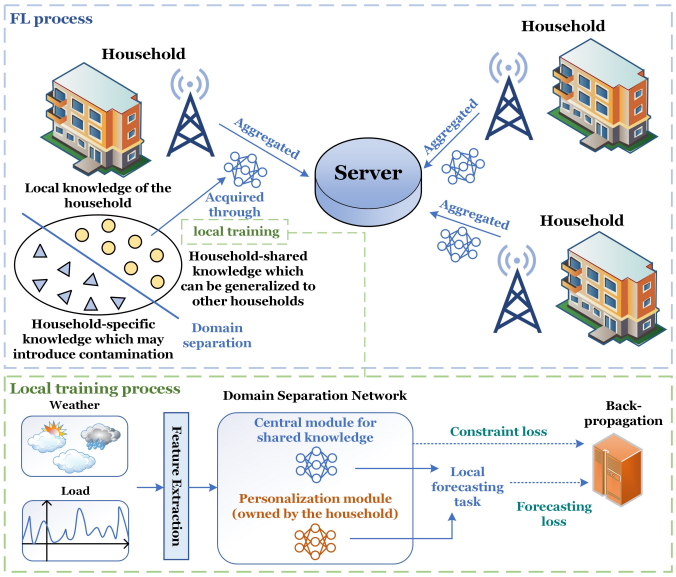


Fig. 2. Description of the main objectives in this work: to make the central module acquire all households' shared knowledge in the FL process and utilize a personalization module to supplement the current household's knowledge.

B. Main Objective

As the load data are non-IID, the knowledge acquired by the model in an individual household may not be fully useful for the others' load forecasting tasks. Thus, the simple model aggregation in the current FedAVG may introduce contamination to global knowledge. In view of the above, we separate the knowledge of each household (i.e., the hidden representations extracted by the neural network) into two types:

1) Household-shared type: it is distributed similarly so that it can be generalized to other households and thus help their load forecasting tasks.
2) Household-specific type: it is distributed discrepantly from the shared type and is only helpful for local forecasting tasks. This may introduce contamination to other households' forecasting tasks.

On this basis, we aim at building a model for each household that can acquire the **household-shared knowledge shared among all households** and **household-specific knowledge only from the current household**. A practical technical strategy is proposed to achieve this, illustrated in Fig. 2:

1) First, we design a FL process in which the knowledge can be separated through a collaboration of multi-domain alignment and decoupling. Then, a central module is constructed for each household to extract local household-shared knowledge. At the end of each communication round, part of each household's central module is transferred to a central server for household-shared knowledge aggregation.
2) Second, in each local training process, a personalization module is also constructed to compensate for the filtered

household-specific knowledge and create a more personalized model. It is noted that each personalization block is owned locally, and there is no constraint between the knowledge produced by it and other blocks; therefore, it remains relatively more personalized.

III. Methodology

In response to the aforementioned objectives (as shown in the caption of Fig. 2), this section provides a practical framework, as is shown in Fig. 3, and elaborates on its implementations.

A. Knowledge Extraction

First of all, our approach is directed at the acquired knowledge, which can be reflected in the block's ability to transform the original dataset into hidden representations. Therefore, an effective feature selection strategy and an advanced block structure are required for knowledge extraction.

Feature selection greatly affects the final accuracy, and there have been many researches in this investigation. Inspired by [4] and [1], multiple features highly correlated to electrical load are taken into account in this work, which can be categorized into two types: 1) temporal features including historical load and temperature sequences, which are sampled by a sliding window; and 2) non-temporal features including one-hot encoded time index (hour of the day, day of the week, holiday mark, etc.), precipitation (rain or snow) and weather (breezy, mostly cloudy, clear, foggy, etc.) at the target time point.

To deal with both of the two types of features and convert them into hidden representations, a hybrid-LSTM neural network [40], which has been proven to be a good candidate for such a load forecasting task, is employed here as the basic structure of each block. The structure details and the interfaces to the selected features are shown in Fig. 3 (c). More specifically, the network consists of an LSTM layer to establish the complex temporal relationship among multiple time steps of temporal features and a multi-FC layer to handle non-temporal features.

Given the domain separation approach in the later section works only on the distributions of hidden representations, we extract each type of knowledge using blocks with the same structure. In addition, they are made separately with different constraints to intervene in the distributions, which will be described in the following parts.

B. Federated Domain Separation Network

To realize the domain separation and accurate knowledge acquisition in Section II-B, a federated domain separation network is proposed here. As shown in Fig. 3, several blocks with different functions and their corresponding fully connected (FC) layers are constructed in the network:

1) **Knowledge reference block**: It serves as a bridge that allows interaction with other households without accessing their data.
2) **Local alignment and separation blocks**: They separate the original hidden representations into two differently distributed components (shared and specific) by MMD loss constraint and Orthogonal loss constraint.
3) **Personalization block**: It provides a household-specific information supplement for model personalization.

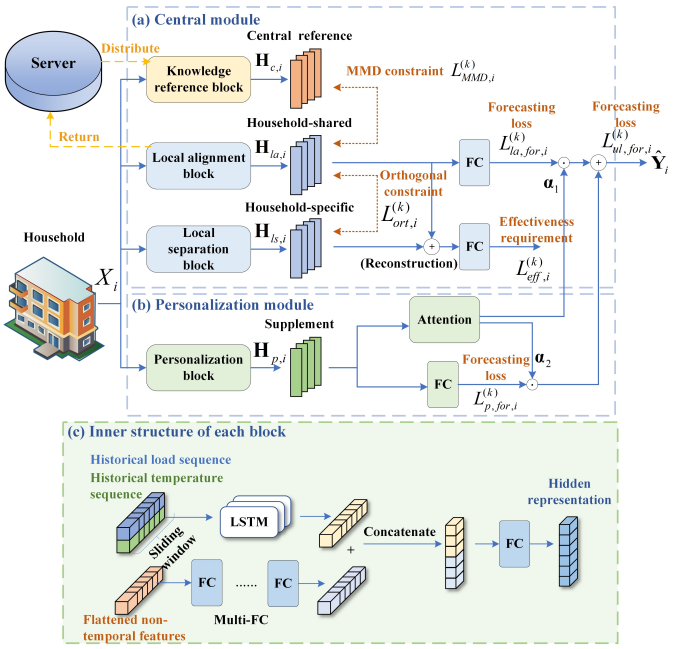


Fig. 3. Details of the proposed FL-based household load forecasting framework. (a) and (b) show the implementation details of the blocks in the central module and personalization module, respectively; (c) shows the inner structure of each block in it.

Note that only part of the framework (knowledge reference block and local alignment block) is transferred between the household and the server, so inferring each household's load data from it can be hard.

Implementation details of each of them will be illustrated in the rest of the context in this subsection. Let $\{X_i\}_{i=1}^N$ denote the input set, $\{\mathbf{Y}_i\}_{i=1}^N$ denote the corresponding label set, N denote the number of households, $M(\cdot)$ denote the mapping function of each block, and $F(\cdot)$ denote the mapping function of the FC layer. For a clearer illustration, the i-th household in the k-th communication round is set as an example here.

1) Knowledge Reference Block: Since only model parameters can be directly accessed by the server or other households in FL, the knowledge interaction must rely on a block that can be exchanged between the server and each household, i.e., the knowledge reference block. Considering the main knowledge to be interacted with in the FL process is the other households' shared knowledge, we aggregate all local alignment blocks to achieve the knowledge reference block. The updating process of the knowledge reference block parameters $\mathbf{W}_c^{(k)}$ at the k-th communication round can be represented as:

$$\mathbf{W}_c^{(k)} = \sum_{i=1}^{N} \mathbf{W}_{la,i}^{(k-1)} \tag{1}$$

where $\mathbf{W}_{la,i}^{(k-1)}$ denotes the parameters of the local alignment block of the i-th household at the previous communication

round. Then the hidden representations $\mathbf{H}_{c,i}$ produced by it can be calculated by

$$\mathbf{H}_{c,i} = M\left(X_i \mid \mathbf{W}_c^{(k)}\right) \tag{2}$$

This block only serves as a reference of the global knowledge aggregated from all households' shared knowledge and does not participate in load forecasting throughout the whole process. Thus, its parameters are set as frozen. Based on the knowledge reference block, we can adjust the parameters of other blocks to achieve interaction with other households' shared knowledge.

2) Local Alignment and Separation Blocks: The next step is to split the knowledge into the desired household-shared and household-specific types, which can be achieved by constraining the generation of the output hidden representations of local alignment and separation blocks, i.e., $\mathbf{H}_{la,i}$ and $\mathbf{H}_{ls,i}$. On the one hand, we partition a mutually separate subspace for each of them to remove knowledge coupling; on the other hand, each household's $\mathbf{H}_{la,i}$ and $\mathbf{H}_{c,i}$ are expected to be aligned to ensure that the local alignment block can extract more generalized household-shared knowledge [41]. The hidden representations of these two blocks are first calculated by

$$\mathbf{H}_{la,i} = M\left(X_i \mid \mathbf{W}_{la,i}^{(k)}\right) \tag{3}$$

$$\mathbf{H}_{ls,i} = M\left(X_i \mid \mathbf{W}_{ls,i}^{(k)}\right) \tag{4}$$

where $\mathbf{W}_{l,i}^{(k)}$ denotes the parameters of the local separation block.

Before the official separation, it is essential to ensure the effectiveness and integrity of the original knowledge (i.e., the unseparated hidden representations are actually the abstract features mapped from the original inputs and correlated to the forecasting task). In [34], a reconstruction loss is defined to satisfy this requirement. However, it is hard for the proposed framework to reconstruct the input because the input consists of both temporal and non-temporal components. As an alternative, this paper turns to only reconstructing hidden representations and uses them to perform the forecasting task to satisfy the effectiveness requirement. Therefore, the loss function $L_{eff,i}^{(k)}$ for this constraint can be calculated as:

$$L_{eff,i}^{(k)} = \left\| \mathbf{Y}_i - F\left(\mathbf{H}_{la,i} + \mathbf{H}_{ls,i} \mid \boldsymbol{\theta}_{ls,i}^{(k)}\right) \right\| \tag{5}$$

where $\boldsymbol{\theta}_{la,i}^{(k)}$ denotes the parameters of the FC layer corresponding to the local alignment block, and $\|\cdot\|$ denotes the norm (we utilize 1-norm in this work to reduce the effects of anomaly).

After satisfying the effectiveness requirement, we encourage these two blocks to produce more decoupling hidden representations to separate the knowledge. An orthogonal constraint loss function $L_{ort,i}^{(k)}$ is defined here to guide the generation process of $\mathbf{H}_{ls,i}$ and $\mathbf{H}_{la,i}$:

$$L_{ort,i}^{(k)} = \left\|\mathbf{H}_{ls,i}^T \mathbf{H}_{la,i}\right\|_F^2 / \left(\left\|\mathbf{H}_{ls,i}\right\|_F \cdot \left\|\mathbf{H}_{la,i}\right\|_F\right) \tag{6}$$

where $\|\cdot\|_F$ denotes the Forbenius norm.

After decoupling, we expect all households' shared knowledge to be acquired by the local alignment block, and its generalization can be ensured. To achieve this, we encourage the local alignment block and the knowledge reference block to produce more aligned hidden representations. In view of this, an MMD constraint is added here, which measures the distribution discrepancy between $\mathbf{H}_{la,i}$ and $\mathbf{H}_{c,i}$. First, a linear combination of multiple radial basis function (RBF) kernels is adopted to map the hidden representations to a Reproducing Kernel Hilbert Space. Then, the calculation of the MMD constraint loss can be represented as:

$$\begin{aligned} L_{MMD,i}^{(k)} = \frac{1}{N_h^2} \Bigg\{ & \sum_{p,q=0}^{N_h} k\left(\mathbf{h}_{c,p}, \mathbf{h}_{c,q}\right) + 2 \sum_{p,q=0}^{N_h} k\left(\mathbf{h}_{c,p}, \mathbf{h}_{la,q}\right) \\ & + \sum_{p,q=0}^{N_h} k\left(\mathbf{h}_{la,p}, \mathbf{h}_{la,q}\right) \Bigg\} \end{aligned} \tag{7}$$

where $\mathbf{h}_{c,p}, \mathbf{h}_{c,q} \in \mathbf{H}_{c,i}, \quad \mathbf{h}_{la,p}, \mathbf{h}_{la,q} \in \mathbf{H}_{la,i}, \quad N_h$ denotes the total number of the vectors in the hidden representation matrix (often equal to batch size), and $k(\cdot,\cdot)$ denotes the combination of kernels, which can be calculated by

$$k\left(\mathbf{h}_{*,p}, \mathbf{h}_{*,q}\right) = \sum_{d=1}^{n} \exp\left(-\frac{1}{2\sigma_d} \left\|\mathbf{h}_{*,p} - \mathbf{h}_{*,q}\right\|_2^2\right) \tag{8}$$

where n denotes the number of kernels and σ_d denotes the standard deviation of the d-th kernel.

Then, we expect to only utilize household-shared knowledge to perform the forecasting task, thus removing the contamination of household-specific knowledge. Therefore, An FC layer is utilized to transform $\mathbf{H}_{la,i}$ into the forecasted load. The calculation of the forecasting loss $L_{la,for,i}^{(k)}$ can be represented as:

$$L_{la,for,i}^{(k)} = \left\| \mathbf{Y}_i - F\left(\mathbf{H}_{la,i} \mid \boldsymbol{\theta}_{la,i}^{(k)}\right) \right\| \tag{9}$$

where $\boldsymbol{\theta}_{la,i}^{(k)}$ denotes the parameters of the FC layer corresponding to the local alignment block.

3) Personalization Block and Attention-based Collaboration strategy: By using the aforementioned blocks, the forecasting model can acquire household-shared knowledge from all households, but the ultimate objective of this work is to create a customized model that can better fit the local data distribution. Therefore, a personalization block is additionally constructed for each household to compensate for the filtered household-specific knowledge. The hidden representations and the forecasting loss function of the personalization block can be respectively calculated by

$$\mathbf{H}_{p,i} = M\left(X_i \mid \mathbf{W}_{p,i}^{(k)}\right) \tag{10}$$

where $\mathbf{W}_{p,i}^{(k)}$ denotes the parameters of the personalization block. To guide the personalization block training, a forecasting loss function is defined here:

$$L_{p,for,i}^{(k)} = \left\| \mathbf{Y}_i - F\left(\mathbf{H}_{p,i} \mid \boldsymbol{\theta}_{p,i}^{(k)}\right) \right\| \tag{11}$$

where $\boldsymbol{\theta}_{p,i}^{(k)}$ denotes the parameters of its corresponding FC layer.

Then an attention-based collaboration strategy is proposed to fully utilize the central blocks and the personalization block. Therefore, the next step is to utilize the self-attention

mechanism to calculate the attention weights $\boldsymbol{\alpha}_{1,i}^{(k)}$ and $\boldsymbol{\alpha}_{2,i}^{(k)}$ based on the hidden representations, which can be represented as:

$$\boldsymbol{\alpha}_{1,i}^{(k)} = sigmoid\left(F\left(\mathbf{H}_{p,i} \mid \boldsymbol{\theta}_{att,i}^{(k)}\right)\right) \tag{12}$$

$$\boldsymbol{\alpha}_{2,i}^{(k)} = \mathbf{1} - \boldsymbol{\alpha}_{1,i}^{(k)} \tag{13}$$

where $\boldsymbol{\theta}_{att,i}^{(k)}$ denotes the parameters of the FC layer which calculates the attention weights. Now the ultimate forecasted load $\hat{\mathbf{Y}}_i^{(k)}$ which is obtained from both central blocks and the personalized block and the corresponding forecasting loss function $L_{ul,for,i}^{(k)}$ can be calculated by

$$\hat{\mathbf{Y}}_i^{(k)} = \boldsymbol{\alpha}_{1,i}^{(k)} \odot F\left(\mathbf{H}_{p,i} \mid \boldsymbol{\theta}_{p,i}^{(k)}\right) + \boldsymbol{\alpha}_{2,i}^{(k)} \odot F\left(\mathbf{H}_{la,i} \mid \boldsymbol{\theta}_{la,i}^{(k)}\right) \tag{14}$$

where $\odot$ denotes the element-wise product. To guide the effective generation of attention weights and achieve accurate ultimate results, a forecasting loss function is defined:

$$L_{ul,for,i}^{(k)} = \left\| \mathbf{Y}_i - \hat{\mathbf{Y}}_i^{(k)} \right\| \tag{15}$$

Note that the personalization is synchronous with the FL process, so it avoids knowledge forgetting in current post-processing methods, which use local data to rewrite part of the network.

4) Optimization Objective: Based on the analysis of block implementations, the loss functions that guide the model training (which have been clearly visualized in Fig. 3) can mainly be categorized into three types: 1) effectiveness requirement satisfaction, i.e., $L_{eff,i}^{(k)}$; 2) knowledge generation constraint, i.e., $L_{ort,i}^{(k)}$ and $L_{MMD,i}^{(k)}$; and forecasting losses, i.e., $L_{la,for,i}^{(k)}$, $L_{p,for,i}^{(k)}$ and $L_{ul,for,i}^{(k)}$. They are required to be minimized in each local training process. To simplify the illustration, we let $\mathbf{W}_i^{(k)} = \left\{\mathbf{W}_{la,i}^{(k)}, \mathbf{W}_{ls,i}^{(0)}, \mathbf{W}_{p,i}^{(k)}, \boldsymbol{\theta}_{la,i}^{(k)}, \boldsymbol{\theta}_{ls,i}^{(k)}, \boldsymbol{\theta}_{p,i}^{(k)}\right\}$ denote the set of the parameters to be updated. The optimization objective can be represented as:

$$\begin{aligned}\min_{\mathbf{W}_i^{(k)}} \Big\{ & L_{MMD,i}^{(k)} + L_{la,for,i}^{(k)} + L_{ort,i}^{(k)} + L_{eff,i}^{(k)} \\ & + L_{p,for,i}^{(k)} + L_{ul,for,i}^{(k)} \Big\}\end{aligned} \tag{16}$$

It is worth noticing all the components in (16) are calculated from each household's local data, and there is no data sharing throughout the whole process. Therefore, household privacy can be preserved.

C. Full Algorithm

After introducing the architecture and functions of the proposed framework, the next thing to be considered is how to design a practical process for the privacy-preserving distributed load forecasting task. There are three main challenges to be tackled: 1) how to initialize the parameters, especially for the parameters in the knowledge reference block; 2) how to transfer, preserve, and load the parameters of each component in the proposed framework; and 3) how the central server and each household act in each communication round. Taking into account these concerns, we design the full algorithm for the proposed framework, which consists of two parts: the federated initialization process and the federated training process.

Algorithm 1: Federated Initialization Process

Data: inputs $\{X_i\}_{i=1}^N$ and labels $\{\mathbf{Y}_i\}_{i=1}^N$ owned by the households

Server execution:
- **for** *each household* $i = 1, 2, \cdots$ **do**
  - $\mathbf{W}_{la,i}^{(0)} \leftarrow$ *LocalInitialize* $(X_i, \mathbf{Y}_i)$
- $\mathbf{W}_{c,i}^{(1)} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathbf{W}_{la,i}^{(0)}$
- Preserve $\mathbf{W}_{la,i}^{(0)}$

Household execution:
- **Procedure** *LocalInitialize* $(X_i, \mathbf{Y}_i)$**:**
  - Randomly initialize $\mathbf{W}_i^{(0)}$
  - **for** *each local epoch* **do**
    - $\mathbf{W}_{la,i}^{(0)} \leftarrow \mathbf{W}_{la,i}^{(0)} - \eta \nabla_{\mathbf{W}_{la,i}^{(0)}} L_{la,for,i}^{(0)}$
  - Preserve $\mathbf{W}_i^{(0)}$ locally
  - Return $\mathbf{W}_{la,i}^{(0)}$

1) Federated Initialization Process: As shown in **Algorithm** 1, all the parameters of the model except for $\mathbf{W}_c^{(0)}$ are randomly initialized by the household. Then, the local alignment block and its corresponding FC layer of each household will be instructed to perform a simple training (in this case, it can be equated to an individual hybrid LSTM forecasting network). The parameters of local alignment blocks $\mathbf{W}_{la,i}^{(0)}$ will in turn be transferred and aggregated by the central server and set as the initial parameters for the knowledge reference block $\mathbf{W}_c^{(1)}$.

2) Federated Training Process: As shown in **Algorithm** 2, all the blocks will be operational to calculate the required components in the optimization objective. The local alignment block, which is utilized to extract desirable household-shared knowledge, will be transferred to a central server after each communication round. According to each household's latest local alignment block, the server will perform knowledge aggregation. At the same time, the personalization block can learn how to use local household-specific knowledge to supplement the forecasting task from the local training. After multiple federated learning rounds, a forecasting model that is more adept at non-IID data can be expected.

D. Complexity Analysis

Since current multi-domain alignment methods [35] require high computational and communicational costs, which makes them hard to be applied in FL, we investigate the reduction offered by the proposed framework. Define the controlling parameters: size of mini-batch N_h, number of local iterations Q, number of communication rounds G, and proportion of participants λ.

Algorithm 2: Federated Training Process

Data: inputs $\{X_i\}_{i=1}^{N}$ and labels $\{\mathbf{Y}_i\}_{i=1}^{N}$ owned by the households

Server execution:
- **for** *each communication round* $k = 1, 2, \cdots$ **do**
  - Randomly select a percentage of households as participants
  - **for** *each household* $i = 1, 2, \cdots$ **do**
    - **if** *the household is selected* **then**
      - Transfer $\mathbf{W}_c^{(k)}$ to the i-th household
      - $\mathbf{W}_{la,i}^{(k)} \leftarrow$ *LocalTrain* $\left(X_i, \mathbf{Y}_i, \mathbf{W}_c^{(k)}\right)$
    - **else**
      - $\mathbf{W}_{la,i}^{(k)} \leftarrow$ local alignment block parameters preserved by the server
    - Preserve $\mathbf{W}_{la,i}^{(k)}$
  - $\mathbf{W}_c^{(k+1)} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathbf{W}_{la,i}^{(k)}$

Household execution:

Procedure *LocalTrain* $\left(X_i, \mathbf{Y}_i, \mathbf{W}_c^{(k)}\right)$**:**
- Load locally preserved parameters as $\mathbf{W}_i^{(k)}$
- Load $\mathbf{W}_c^{(k)}$
- **for** *each local epoch* **do**
  - $\mathbf{W}_i^{(k)} \leftarrow \mathbf{W}_i^{(k)} - \eta\nabla_{\mathbf{W}_i^{(k)}} \left\{L_{MMD,i}^{(k)} + L_{la,for,i}^{(k)} + L_{ort,i}^{(k)} + L_{eff,i}^{(k)} + L_{p,for,i}^{(k)} + L_{ul,for,i}^{(k)}\right\}$
- Return $\mathbf{W}_{la,i}^{(k)}$

1) Computational Complexity: The computational cost mainly results from the floating-point operations (FLOPs) of each propagation (denoted by T) and MMD loss computation (denoted by M). For the proposed framework, each local iteration requires $3T$ for training, T for validation, and M for MMD loss computation. Therefore, the computational complexity is $O\left(\lambda NQG\left(4T + M\right)\right)$. Current methods define the multi-domain alignment loss function as the average of the MMD distance between each two domains. So their computational complexity in FL is calculated as $O\left(QG(4\lambda NT + (\lambda N)!M)\right)$.

2) Communicational Complexity: For the proposed framework, the computational cost mainly results from the burden of transferring parameters of two blocks. Let P denote the number of all model parameters. Usually, the deep learning model parameters are stored in 16-bit floating point format, so each of them requires 16 bits for communication. Therefore, its communicational cost is calculated as $O\left(2 \times 2 \times \frac{1}{4}P\lambda NG\right) = O\left(\lambda NPG\right)$. For current methods, the burden of transferring hidden representations $\mathbf{H}_{la,i}$ (which are usually stored in 32-bit format) is also taken into account. Let D denote the number of hidden nodes, then their communicational cost is $O(\lambda NG(P + 2QN_hD))$.

TABLE I
CONFIGURATION DETAILS

Parameter	Configuration
Dataset	- Provided by UK Power Networks - Number of households: $N = 20$ - Training: From Sep. 1 to Nov. 30, 2013 - Test: From Dec. 1 to Dec. 30, 2013 - Validation: 10% of training set - Width of the sliding window: 168h - Forecasted period: Ahead 1h
Architecture	- LSTM layers: 1 - Multi-FC layers: 2 - Hidden nodes of each layer: $D \in \{8, 16, \mathbf{32}, 48\}$
Training	- Maximum communication rounds: 200 - Early stopping patience: 15 - Early stopping minimum delta: 1×10^{-4} - Proportion of participants $\lambda \in \{0.3, 0.4, \mathbf{0.5}\}$ - Number of local epochs $\in \{3, 4, \mathbf{5}\}$ - Batch size: $N_h = 256$ - Initial learning rate: 1×10^{-4} - Optimizer: Adam - Total number of kernels: $n \in \{10, 15, \mathbf{20}\}$ - Kernel standard deviation: $\sigma_d = d$

IV. EXPERIMENTS

This section mainly conducts comprehensive experiments to validate the effectiveness of the proposed load forecasting framework.

A. Experimental Setups

The experiments in this work are all conducted in a virtual environment with Python 3.8.8 and Pytorch 1.13.1, where the model is trained on a single NVIDIA GTX 4080 TI GPU with a computing power of 317656.35 million FLOPs per second for 32-bit floating-point data. To simulate the FL scenario, this work assumes that active households in each communication round can perform training in parallel and all households share the same communication network with a bandwidth of 1 Mbyte per second.

Considering that the LSTM neural network is sensitive to the data scale, min-max normalization is adopted for the load and temperature data, and one-hot encoding is adopted for other category data. To avoid overfitting, 10% of the training dataset is set as the validation set, and early stopping is adopted in the federated learning process.

For hyperparameter selection, this work applies a two-stage optimization strategy. Firstly, considering the hybrid-LSTM structure is the basis of our proposed framework, the number of hidden nodes is selected by evaluating the performance of a single hybrid-LSTM model. Secondly, the FL-related hyperparameters (proportion of participants, number of local epochs, and total number of kernels) are selected by utilizing a simple grid search method. The values or search spaces of all crucial parameters are illustrated in the TABLE I, where the selected hyperparameters are presented in bold font.

To measure the quality of forecasting performance, the mean absolute error (MAE) and root mean squared error (RMSE)

are adopted here:

$$\text{MAE} = \frac{1}{|T|}\sum_{j\in T}|y_j - \hat{y}_j| \tag{17}$$

$$\text{RMSE} = \sqrt{\frac{1}{|T|}\sum_{j\in T}(y_j - \hat{y}_j)^2} \tag{18}$$

where T is the set of forecasted data, y_i and $\hat{y}_i$ denote the actual and forecasted electrical loads, respectively. The mean absolute percentage error (MAPE) is not considered for the reason load values approaching zero occur frequently, as shown in Fig. 1.

B. Comparison with Benchmarking Methods

To verify the effectiveness of the proposed framework, several commonly seen methods are set as benchmarking methods. Brief descriptions of these mentioned methods are provided as follows:

1) Method 1 (Persistence for an hour) [42]: Hourly loads are predicted as the loads at the previous hour. This is a blank control used to validate the effectiveness of machine learning models.
2) Method 2 (Localized learning) [4]: A separate hybrid-LSTM model for each household, which is trained in isolation using only the data available to that household.
3) Method 3 (FedAVG) [6]: A fraction of households are selected in each communication round to train a model, and then the parameters are returned to a central server for aggregation to obtain a joint model.
4) Method 4 (FedAVG with local fine-tuning) [26]: The trained FedAVG model is sent to each client and performs simple training so that the model can better adapt to each household's local distribution.

The model performances over the 20 households are presented in TABLE II and TABLE III. For a clearer illustration, two examples of Household #1 and Household #2 are visualized in Fig. 4. The results indicate the superiority of the proposed framework with the best performances in most cases. Generally, it can reduce average MAE/RMSE by 4.64% /4.33% when compared to the best benchmarking method, i.e., Method 4, and 9.75%/8.66% when compared to the localized learning. It can be seen that localized learning is insufficient to train a reliable model with a small amount of data held by each household (Method 2 almost always yields poorer results, especially around the 400-th hour). In contrast, FL-based methods are able to fully utilize all households' data resources, thus improving forecasting accuracy while preserving their privacy. However, the performance of current FL-based methods still shows a certain gap compared to the proposed framework that is more proficient in dealing with non-IID load data in this work.

C. Effectiveness of Domain Separation

In order to further explore the effectiveness of domain separation, the performances of several invariants of the proposed framework, which lack some crucial components, are also provided. The invariants of the proposed framework are briefly described as follows:

1) Method 5 (proposed framework without domain separation): no decoupling or alignment is adopted in the proposed framework.
2) Method 6 (proposed framework without alignment): No alignment is adopted in the proposed framework, but the original knowledge is still separated into two decoupling components.
3) Method 7 (proposed framework without decoupling): No decoupling is adopted in the proposed framework, but the distribution of the knowledge is still aligned with the central reference.

The comparison of the forecasting results can be found in TABLE II and TABLE III. It can be observed that the removal of any components of domain separation can contribute to lower forecasting accuracy. More specifically, the removal of domain separation/alignment/decoupling can increase the MAE by 5.99%/1.63%/6.93% and RMSE by 5.43%/1.51%/6.30%. According to the results, the removal of domain separation can contribute to an increase in error to varying degrees on most household datasets, which indicates the negative effects of conflicting knowledge. Specifically, except for household #14, the contaminating knowledge can increase the forecasting error by 2.84% to 14.43%. To our surprise, Method 7, which only removes decoupling, performs even worse than Method 5, which removes the whole domain separation. This anomaly indicates the alignment is insufficient to make the model acquire household-shared knowledge, and the negative effects caused by the knowledge contamination are still obvious. Compared to them, decoupling the two components can effectively improve the performance, but a certain gap is still seen when compared to the proposed framework. Only the collaboration of decoupling and alignment can both acquire household-shared knowledge and remove the contamination, thus further improving the accuracy.

D. Effectiveness of the Knowledge Supplement

To investigate the effectiveness of our proposed personalization strategy, i.e., household-specific knowledge supplement, two FL models with different personalization strategies are considered: Method 8 (proposed framework without knowledge supplement) and Method 2. The comparison results in terms of MAE and RMSE are presented in TABLE II and TABLE III, respectively. It can be observed that the knowledge supplement can significantly improve performance. More specifically, it can reduce the MAE/RMSE by 4.15%/4.66%. Although the previous context proves that the current local fine-tuning process can make more accurate results based on FedAvg, it still performs worse than the proposed framework on the majority of households.

Another thing worth mentioning is that even though Method 8 fails to achieve the best results on all datasets, it still outperforms Method 4 on average. This indicates that under some circumstances, local fine-tuning can obviously improve accuracy by making the model adapt to the local distribution,

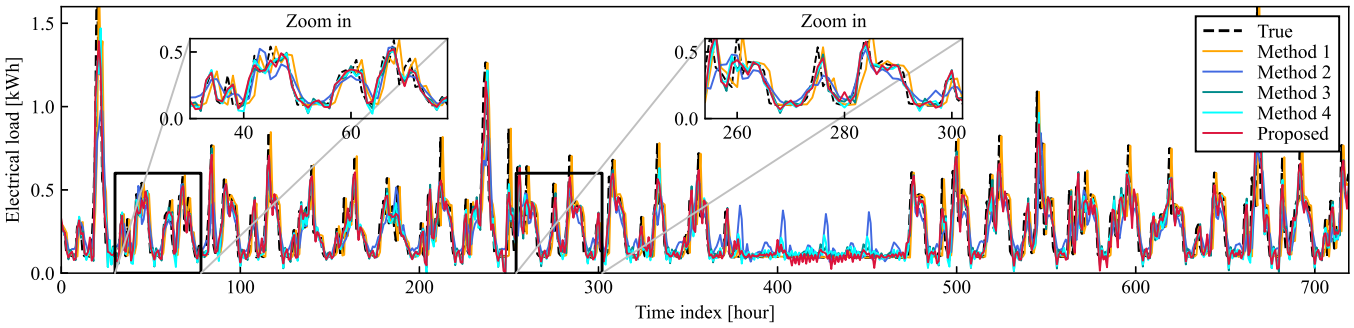


Fig. 4. The forecasting results given by the proposed framework and benchmarking methods in Household #1.

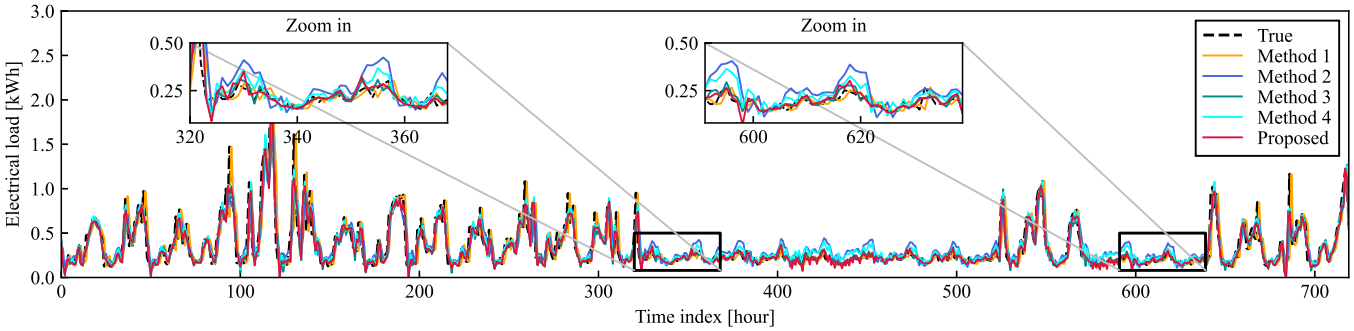


Fig. 5. The forecasting results given by the proposed framework and benchmarking methods in Household #2

TABLE II
MODEL PERFORMANCE OF THE PROPOSED FRAMEWORK AND BENCHMARKING METHODS IN TERMS OF MAE [WH]

Household	MAE [Wh]										
No.	Proposed	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6	Method 7	Method 8	Method 9	Method 10
#1	**55.44**	89.78	84.70	60.33	58.76	58.08	55.89	58.68	56.57	95.18	111.32
#2	**56.38**	94.66	86.56	60.71	68.76	60.54	57.92	61.14	59.07	87.74	83.15
#3	**145.86**	265.69	159.75	165.27	146.75	155.59	148.83	157.00	154.27	203.49	175.04
#4	**121.86**	223.93	144.78	130.76	135.00	128.24	126.83	129.17	127.39	180.62	138.14
#5	**61.39**	114.39	67.93	66.58	64.24	65.22	63.09	65.98	65.19	95.69	61.17
#6	120.51	216.68	125.67	135.52	**118.45**	127.62	125.39	129.93	126.42	156.37	135.89
#7	104.39	175.08	103.71	103.52	103.44	108.78	102.08	111.24	103.29	131.31	**102.97**
#8	**86.22**	162.93	94.25	97.07	94.71	92.70	87.81	93.40	92.81	122.49	87.90
#9	103.05	181.77	104.68	108.30	104.57	108.74	102.87	108.27	107.13	107.66	**100.03**
#10	71.94	127.16	74.39	77.67	72.12	75.61	72.74	76.50	74.90	88.37	**71.20**
#11	**90.60**	158.85	109.34	102.85	103.50	98.93	92.34	99.16	98.80	122.78	96.92
#12	34.85	65.89	42.73	37.62	36.91	38.76	36.00	38.92	37.02	42.03	**34.62**
#13	99.98	174.97	103.73	118.32	106.17	109.68	101.64	111.09	107.21	110.51	**94.75**
#14	25.30	36.78	22.02	27.98	21.31	26.77	26.23	26.57	22.98	26.17	**21.97**
#15	96.19	192.07	109.46	113.84	98.06	107.48	99.93	111.03	104.83	130.50	**91.08**
#16	**76.46**	136.87	84.81	86.13	84.61	82.64	78.46	83.37	81.31	113.05	78.14
#17	**89.03**	154.88	91.32	96.43	98.46	92.82	89.92	93.89	91.89	116.23	90.31
#18	34.93	57.89	37.92	38.85	**33.78**	37.49	35.32	37.36	34.85	39.77	34.34
#19	**74.16**	132.28	79.14	78.74	83.04	77.89	75.37	77.26	77.03	78.70	74.29
#20	45.31	76.22	43.46	47.60	43.02	45.97	45.46	46.45	43.96	47.58	**42.76**
Average	**79.69**	141.94	88.52	87.70	83.78	84.98	81.21	85.82	83.35	104.81	86.30

but the negative effects of knowledge forgetting cannot be ignored.

E. Comparison with State-of-the-Art Methods

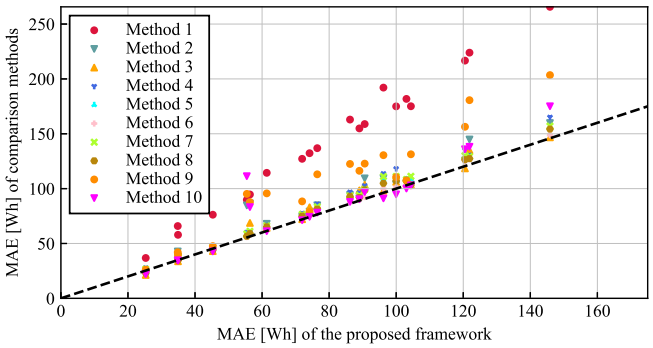
To further present the superiority of the proposed framework, two state-of-the-art methods are investigated, which are described as:

1) Method 9 (Neural Basis Expansion Analysis for Time Series) [43]: An interpretable time-series forecasting method with multiple FC-based stacks and blocks to continuously learn the hidden knowledge in the residue from the last unit. The hyperparameters of this method are set as follows: the number of stacks $\in \{1, 2, 3\}$, the number of blocks $\in \{1, 2, 3\}$, and hidden nodes $\in \{8, 16, 32\}$.
2) Method 10 (Impactnet) [44]: This method includes multiple residual convolutional units and FC layers to handle different types of features. The hyperparameters of this method are set as follows: the number of convolutional channels $\in \{8, 16, 32\}$ and hidden nodes $\in \{32, 96\}$.

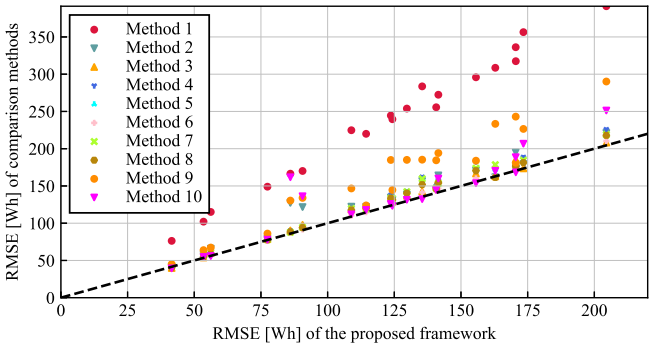
The performances of these two methods are provided in

TABLE III
MODEL PERFORMANCE OF THE PROPOSED FRAMEWORK AND BENCHMARKING METHODS IN TERMS OF RMSE [WH]

Household	RMSE [Wh]										
No.	Proposed	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6	Method 7	Method 8	Method 9	Method 10
#1	**86.04**	166.48	127.25	90.04	90.34	87.75	87.87	87.81	88.34	130.37	161.65
#2	**90.61**	170.11	121.62	95.09	97.84	93.66	91.56	94.49	94.07	133.98	136.17
#3	**204.50**	391.24	221.09	226.11	208.27	217.37	210.32	218.80	217.88	290.18	251.22
#4	**170.52**	336.22	195.05	179.42	185.46	177.55	175.84	178.38	176.52	243.05	188.56
#5	**108.90**	224.79	122.22	120.41	118.32	118.75	110.26	119.56	117.85	146.66	111.87
#6	**173.41**	356.44	179.42	188.87	173.59	180.97	180.70	184.55	181.38	226.54	206.69
#7	**162.87**	308.59	161.80	163.09	162.86	171.85	164.10	178.53	161.42	233.34	170.34
#8	**123.70**	244.52	134.50	138.14	134.99	132.53	126.91	133.62	133.59	184.88	125.81
#9	170.56	317.46	173.86	176.97	174.10	177.39	170.47	176.76	176.53	181.06	**168.24**
#10	124.29	239.57	131.44	133.87	128.73	132.21	124.60	133.33	131.54	144.59	**123.28**
#11	**141.52**	272.43	164.16	158.21	157.53	153.17	146.52	153.44	154.45	194.21	159.92
#12	56.19	115.10	65.84	59.56	60.28	60.88	56.58	60.82	59.07	67.33	**55.91**
#13	155.62	295.86	170.68	182.74	166.67	170.58	160.11	173.77	170.57	183.91	**153.73**
#14	41.54	76.29	40.63	43.50	39.89	43.59	41.22	43.33	40.45	45.05	**39.34**
#15	135.45	283.66	156.61	162.29	142.79	155.47	141.41	159.64	151.83	185.25	**132.26**
#16	**129.71**	253.86	138.34	143.19	136.92	141.45	133.71	142.16	140.38	185.00	131.44
#17	**140.71**	255.61	144.81	150.13	151.64	145.96	143.32	146.98	146.01	184.44	144.83
#18	53.48	102.06	57.58	56.95	54.01	56.69	**53.07**	56.74	54.80	63.87	54.67
#19	**114.43**	219.98	121.98	120.92	122.97	119.72	114.86	119.12	120.53	123.86	117.85
#20	**77.44**	149.01	79.31	79.27	78.56	78.22	78.15	78.27	77.52	86.07	77.87
Average	**123.07**	238.96	135.41	133.44	129.29	130.79	125.58	132.00	129.74	161.68	135.58



(a)



(b)

Fig. 6. Comparison of the proposed framework and all aforementioned methods in terms of (a) MAE and (b) RMSE in each household. The performance of the proposed framework is indicated by the x-axis, and the performances of other methods are indicated by the y-axis. The point above the black dashed line denotes a better performance of the proposed framework.

TABLE II and TABLE III. It can be observed the proposed framework has a significantly lower average MAE/RMSE than Method 9 and Method 10. It is worth noticing that Method 10, as an advanced and effective forecasting model, outperforms the proposed framework in several households. However, because of the much lower accuracy on the first several households, the average error of Method 10 is still obviously higher than the proposed framework.

TABLE IV
MODEL PERFORMANCES IN TERMS OF MAE[WH] WITH DIFFERENT NOISE INJECTIONS

Perturbed Households	25%		50%	
Noise Injection	2.5%	10.0%	2.5%	10.0%
Proposed	81.07	86.52	81.04	87.26
Method 2	87.77	92.65	88.94	93.17
Method 3	87.51	89.20	89.47	91.97

F. Robustness Investigation

Although the proposed framework has been proven to perform well in differently distributed load forecasting tasks, we still wish to explore its performance under a perturbing environment. Therefore, the robustness of the proposed framework, Method 2, and Method 3. In each test, a proportion of households' load data are perturbed by an injected noise. The results in terms of MAE [Wh] are provided in TABLE IV. Despite a certain growth in error as the degree of disturbance increases, the investigated methods can still give relatively accurate results. Moreover, the federated domain separation will not influence the robustness compared to localized learning or conventional federated learning methods, which proves its good usability.

G. Convergence Investigation

The FL-based methods require some communication rounds for convergence. To investigate the convergence of the proposed framework and other aforementioned FL-based methods, their validation loss curves are provided in Fig. 7. Although Method 3 has the fastest convergence rate and the smoothest convergence curve, it is unable to reduce the value of the validation loss below 0.007, which contributes to poorer performance. The proposed framework and its variants (i.e., Methods 5, 6, and 7) have relatively lower convergence rates.

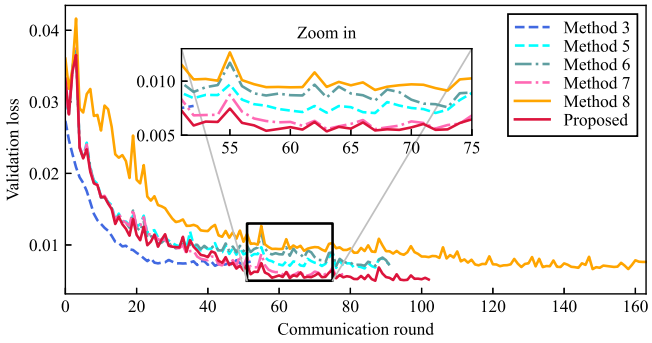


Fig. 7. The convergence curves of validation loss of the proposed framework and other FL-related methods.

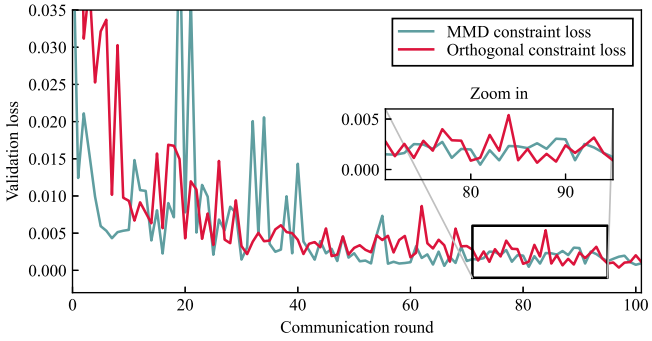


Fig. 8. The convergence curves of validation MMD constraint loss and orthogonal constraint loss of the proposed framework.

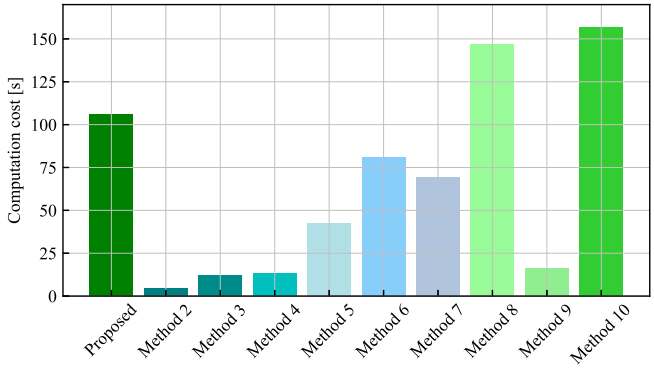


Fig. 9. The computational costs of the methods in this work.

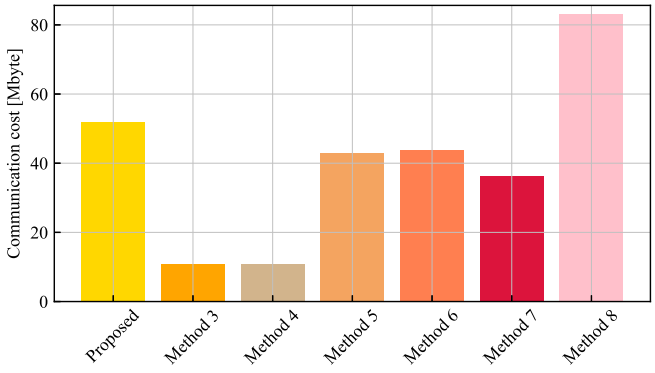


Fig. 10. The communicational costs of the FL-based methods in this work.

Specifically, the validation loss of the proposed framework can converge to the lowest among all of them. It is worth noticing that the removal of household-specific knowledge supplements can result in a much lower convergence rate.

Moreover, we have investigated the convergence curves of the two constraint losses of our proposed framework, which have been provided in Fig. 8 It can be observed that both of them can converge to a low level, which indicates households' local alignment blocks can produce similarly distributed hidden states and local separation blocks can effectively separate the conflicting part.

H. Computational and Communicational Costs Investigation

Lastly, we investigate the computational and communicational costs of the proposed framework and other methods, which have been provided in Fig. 9 and Fig. 10. Specifically, our proposed framework consists of 0.05 million parameters and requires about 2105.62 million FLOPs for propagation and 100 million FLOPs for an MMD loss computation. Considering the dataset size and batch size, each local epoch consists of 8 iterations. According to the complexity analysis in Subsection III-D and hyperparameters selection in TABLE I, the whole training process requires $0.5 \times 20 \times 40 \times 103 \times (4 \times 2105.62 + 100) = 3.46 \times 10^8$ million FLOPs and $0.5 \times 20 \times 0.05 \times 103 = 51.50$ million communication bytes. In the 10-household parallelism scenario, the theoretical computational cost is 109.89s. It can be observed that the theoretical costs are consistent with the tested costs.

To validate the real-time usability of the proposed framework, we consider a real-time updating circumstance, i.e., retraining the model once before each use. According to Fig. 9 and Fig. 10, it can be seen that the proposed framework requires about 150 seconds to complete an FL process, which is acceptable in such an STLF task. Despite a relatively higher training time in comparison with localized learning, the forecasting accuracy can be significantly improved. In addition, Method 6 requires less computational cost (around 110 seconds) and gives relatively accurate forecasting results (according to TABLE II and III), so it can be used as a substitute in the case of insufficient computing power.

V. Conclusion and Future Work

This paper proposes an FL-based load forecasting framework based on federated domain separation, which is more expertise in dealing with non-IID smart meter data. The comparison experiments with several benchmarking methods indicate the superiority of our federated domain separation strategy. In addition, extensive ablation experiments also prove the effectiveness of multi-domain alignment, decoupling, and knowledge supplement, respectively, which are crucial components of the proposed framework. Despite a relatively higher computation cost (which is still acceptable even in an STLF

scenario), the proposed framework can be a good candidate for such a privacy-preserving household load forecasting task.

Our proposed framework has many hyperparameters. Although we have utilized a simple grid search for some hyperparameters selection, there is still room for accuracy improvement by more refined adjustments. Future work will be concentrated on combining it with more advanced search methods.

REFERENCES

[1] N. Kim, H. Park, J. Lee, and J. K. Choi, "Short-term electrical load forecasting with multidimensional feature extraction," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2999–3013, 2022.

[2] Y. He, F. Luo, M. Sun, and G. Ranzi, "Privacy-preserving and hierarchically federated framework for short-term residential load forecasting," *IEEE Transactions on Smart Grid*, pp. 1–1, 2023.

[3] X. Liu, Z. Zhang, and Z. Song, "A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: From classical data mining to deep learning," *Renewable and Sustainable Energy Reviews*, vol. 119, p. 109632, 2020.

[4] 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, 2019.

[5] 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, 2018.

[6] M. N. Fekri, K. Grolinger, and S. Mir, "Distributed load forecasting using smart meter data: Federated learning with recurrent neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 137, p. 107669, 2022.

[7] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[8] H. Liu, "Reliability of a load-sharing k-out-of-n:g system: non-iid components with arbitrary distributions," *IEEE Transactions on Reliability*, vol. 47, no. 3, pp. 279–284, 1998.

[9] N. Gholizadeh and P. Musilek, "Federated learning with hyperparameter-based clustering for electrical load forecasting," *Internet of Things*, vol. 17, p. 100470, 2022.

[10] C. Xu, G. Chen, and C. Li, "Federated learning for interpretable short-term residential load forecasting in edge computing network," *Neural Computing and Applications*, vol. 35, no. 11, pp. 8561–8574, 2023.

[11] C. Briggs, Z. Fan, and P. Andras, "Federated learning for short-term residential load forecasting," *IEEE Open Access Journal of Power and Energy*, vol. 9, pp. 573–583, 2022.

[12] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2018.

[13] E. M. de Oliveira and F. L. C. Oliveira, "Forecasting mid-long term electric energy consumption through bagging arima and exponential smoothing methods," *Energy*, vol. 144, pp. 776–788, 2018.

[14] 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.

[15] G. Lowry, F. U. Bianeyin, and N. Shah, "Seasonal autoregressive modelling of water and fuel consumptions in buildings," *Applied energy*, vol. 84, no. 5, pp. 542–552, 2007.

[16] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1087–1088, 2018.

[17] W. Yang, J. Shi, S. Li, Z. Song, Z. Zhang, and Z. Chen, "A combined deep learning load forecasting model of single household resident user considering multi-time scale electricity consumption behavior," *Applied Energy*, vol. 307, p. 118197, 2022.

[18] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[19] K. Wei, J. Li, C. Ma, M. Ding, S. Wei, F. Wu, G. Chen, and T. Ranbaduge, "Vertical federated learning: Challenges, methodologies and experiments," *arXiv preprint arXiv:2202.04309*, 2022.

[20] H. Liu, X. Zhang, H. Sun, and M. Shahidehpour, "Boosted multi-task learning for inter-district collaborative load forecasting," *IEEE Transactions on Smart Grid*, pp. 1–1, 2023.

[21] Y. He, F. Luo, G. Ranzi, and W. Kong, "Short-term residential load forecasting based on federated learning and load clustering," in *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2021, pp. 77–82.

[22] J. Lin, J. Ma, and J. Zhu, "Privacy-preserving household characteristic identification with federated learning method," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1088–1099, 2022.

[23] J. D. Fernández, S. P. Menci, C. M. Lee, A. Rieger, and G. Fridgen, "Privacy-preserving federated learning for residential short-term load forecasting," *Applied energy*, vol. 326, p. 119915, 2022.

[24] R. Wang, H. Yun, R. Rayhana, J. Bin, C. Zhang, O. E. Herrera, Z. Liu, and W. Mérida, "An adaptive federated learning system for community building energy load forecasting and anomaly prediction," *Energy and Buildings*, p. 113215, 2023.

[25] S. Shen, T. Zhu, D. Wu, W. Wang, and W. Zhou, "From distributed machine learning to federated learning: In the view of data privacy and security," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 16, p. e6002, 2022.

[26] D. Qin, C. Wang, Q. Wen, W. Chen, L. Sun, and Y. Wang, "Personalized federated darts for electricity load forecasting of individual buildings," *IEEE Transactions on Smart Grid*, pp. 1–1, 2023.

[27] Y. Shi and X. Xu, "Deep federated adaptation: An adaptative residential load forecasting approach with federated learning," *Sensors*, vol. 22, no. 9, p. 3264, 2022.

[28] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2272–2281.

[29] Q. Zhu and Z. Wang, "An image clustering auto-encoder based on predefined evenly-distributed class centroids and mmd distance," *Neural Processing Letters*, vol. 51, no. 2, pp. 1973–1988, 2020.

[30] Y. Zhang, G. Tang, Q. Huang, Y. Wang, K. Wu, K. Yu, and X. Shao, "Fednilm: Applying federated learning to nilm applications at the edge," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 2, pp. 857–868, 2023.

[31] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," *Domain adaptation in computer vision applications*, pp. 153–171, 2017.

[32] A. Balint, H. Raja, J. Driesen, and H. Kazmi, "Using domain-augmented federated learning to model thermostatically controlled loads," *IEEE Transactions on Smart Grid*, pp. 1–1, 2023.

[33] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," *arXiv preprint arXiv:2002.05516*, 2020.

[34] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," *Advances in neural information processing systems*, vol. 29, 2016.

[35] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5400–5409.

[36] J. Li, C. Zhang, Y. Zhao, W. Qiu, Q. Chen, and X. Zhang, "Federated learning-based short-term building energy consumption prediction method for solving the data silos problem," in *Building Simulation*, vol. 15, no. 6. Springer, 2022, pp. 1145–1159.

[37] S. Zhao, J. Liu, G. Ma, J. Yang, D. Liu, and Z. Li, "Two-phased federated learning with clustering and personalization for natural gas load forecasting," in *International Workshop on Trustworthy Federated Learning*. Springer, 2022, pp. 130–143.

[38] UK Power Networks. Smartmeter energy consumption data in london households. [Online]. Available: https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households

[39] Weather data for london area. [Online]. Available: https://support.apple.com/en-us/102594

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

[41] X. Yao, C. Huang, and L. Sun, "Two-stream federated learning: Reduce the communication costs," in *2018 IEEE Visual Communications and Image Processing (VCIP)*, 2018, pp. 1–4.

[42] X. Lü, T. Lu, C. J. Kibert, and M. Viljanen, "A novel dynamic modeling approach for predicting building energy performance," *Applied energy*, vol. 114, pp. 91–103, 2014.

[43] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," *arXiv preprint arXiv:1905.10437*, 2019.

[44] W. Zhang, S. Liu, O. Gandhi, C. D. Rodríguez-Gallegos, H. Quan, and D. Srinivasan, "Deep-learning-based probabilistic estimation of solar PV soiling loss," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 4, pp. 2436–2444, 2021.

Nan Lu received the B.S. degree in Electrical Engineering and Automation from Nanjing University of Aeronautics and Astronautics in 2022, and the M.Sc. degree in Electrical and Electronic Engineering from the University of Hong Kong in 2023. 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.

Shu Liu received his B.Eng degree in energy engineering from the University of Hong Kong in 2024. He is currently pursuing a Ph.D. degree in the Department of Electrical and Electronic Engineering at the University of Hong Kong. His research interests include data analytics and forecasting in power systems.

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.

Qiming Chen is currently an algorithm engineer in DAMO Academy-Decision Intelligence Lab, Alibaba Group, Hangzhou, China. In 2022, he received Ph.D. from the State Key Laboratory of Industrial Control Technology, Zhejiang University. His main research interests include control system performance evaluation and fault diagnosis, signal decomposition and time-frequency analysis, industrial big data causality analysis and knowledge mining, intelligent system modeling and tuning. Dr. Chen has over 30 publications in the fields of signal processing and control engineering.

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.