INFORMATION
Volume 29, Number 1, pp.123-130

ISSN 1343-4500 eISSN 1344-8994
©2026 International Information Institute

Representation-Based Hierarchical Federated Learning for Household Energy Forecasting

Yinzhe Guo[1], Lei Li[2]

[1]Graduate School of Science and Engineering, Hosei University, Tokyo, Japan
yinzhe.guo.3e@stu.hosei.ac.jp,

[2]Department of Computer Science, Hosei University, Tokyo, Japan

Corresponding author: Lei Li
lilei@hosei.ac.jp

Abstract

We propose a novel *Representation-based Hierarchical Federated Learning* (RHFL) framework for privacy-preserving household energy forecasting. The framework adopts a hierarchical architecture composed of lightweight Fog-level local models and a Cloud-level attention-based aggregator. Unlike traditional parameter-based federated learning approaches, RHFL requires only the transmission of semantic embeddings from Fog nodes to the Cloud. This design not only protects raw data privacy, but also enables more flexible modeling, stronger interpretability, and a globally informed modeling capability. We evaluate the framework on a synthetic smart grid dataset covering five households, and compare it against centralized models, statistical methods, and parameter-based federated learning baselines. The experimental results show that RHFL achieves competitive forecasting accuracy while strictly preserving data privacy. Moreover, it demonstrates superior generalization across heterogeneous households and offers practical advantages in flexibility, training efficiency, and global model interpretability over both centralized and parameter-based FL approaches.

Keywords: Federated Learning, Fog Computing, Smart Grid, Representation Learning, Attention, Energy Forecasting

1 Introduction

Smart grids increasingly rely on accurate energy forecasting to enable demand response, dynamic pricing, and efficient grid management. In particular, accurate household-level energy forecasting is critical for enabling personalized demand-side management and ensuring grid stability. However, traditional centralized machine learning approaches raise serious privacy concerns, as household energy data can reveal sensitive user behaviors.

Federated learning (FL) offers a promising paradigm to enable privacy-preserving learning by keeping raw data local. Yet, classical parameter-averaging FL approaches (e.g., FedAvg [1]) often struggle with non-iid data distributions and lack the ability to model global context effectively, leading to poor personalization and weak generalization. In smart grid scenarios, household behaviors exhibit significant heterogeneity, making such naive aggregation ineffective.

We propose a novel *hierarchical federated learning* architecture that decouples local learning and global modeling. Local Fog models learn per-household temporal embeddings, which are transmitted to a Cloud-level attention-based aggregator. The Cloud model integrates these embeddings with global contextual features (e.g., weather, calendar effects), enabling globally informed yet privacy-preserving forecasting. This design allows global patterns to be captured without sharing raw data or local model parameters, ensuring strong privacy guarantees.

Our contributions are:

- We design a representation-based hierarchical FL architecture tailored for smart grid energy forecasting.
- We implement an attention-based Cloud aggregator that learns to integrate heterogeneous household embeddings.
- We conduct extensive experiments comparing against centralized, statistical, and parameter-based FL baselines.
- We analyze accuracy, resource efficiency, and explainability of our method.
- Our approach enables flexible global model outputs through semantic embedding aggregation, rather than relying on parameter averaging.

2 Related Work

2.1 Federated Learning

Federated learning [1] enables collaborative model training without centralized data collection. Classical approaches aggregate model parameters (e.g., FedAvg), but struggle under data heterogeneity [2]. Recent works explore personalized FL and representation-based FL to mitigate these issues.

2.2 Energy Forecasting

Energy forecasting is a well-studied problem in smart grids. Time series models (ARIMA), tree-based methods (XGBoost), and deep learning approaches (LSTM) have been applied [3]. However, most require centralized data. Few works address privacy-preserving forecasting via FL.

Our prior work [7] explored privacy-preserving optimization-based energy management in smart grids, focusing on demand-supply planning through linear programming. In contrast, this paper focuses on the forecasting layer, developing an RHFL framework to enable flexible and privacy-preserving household energy forecasting.

2.3 Hierarchical Architectures

Hierarchical FL structures have been explored to address system heterogeneity [4]. Our work differs by focusing on representation transfer and attention-based global aggregation, tailored for energy forecasting.

Compared to prior hierarchical FL works [4], our method focuses on representation-based transfer rather than parameter averaging, enabling dynamic and explainable global modeling.

3 Problem Formulation

Given N households $\mathcal{H} = \{h_i\}_{i=1}^{N}$, each with local dataset:

$$D_i = \{(\mathbf{x}_t^{(i)}, y_t^{(i)})\}_{t=1}^{T_i}, \quad \mathbf{x}_t^{(i)} \in \mathbb{R}^d$$

and shared context $\mathbf{c}_t \in \mathbb{R}^m$ (weather, time).

Our goal is to learn a prediction function:

$$\hat{y}_t^{(i)} = f_\phi(g_{\theta_i}(\mathbf{x}_t^{(i)}), \mathbf{c}_t) \tag{1}$$

where:

- g_{θ_i} is the Fog model per household i.
- f_ϕ is the Cloud-level aggregator.

where the objective is to jointly optimize $\{\theta_i\}$ and ϕ to minimize the global loss $\mathcal{L}$. The overall objective function is defined as $\mathcal{L} = \sum_{i,t} \ell(y_t^{(i)}, \hat{y}_t^{(i)})$, where $\ell(\cdot)$ denotes the forecasting loss function (e.g., MSE).

4 Proposed Method

4.1 Fog Node Architecture

Each Fog node implements an embedding extractor and local predictor:

$$\mathbf{e}_t^{(i)} = \mathrm{MLP}_{\text{embed}}(\mathbf{x}_t^{(i)}) \tag{2}$$

$$\hat{y}_t^{(i)} = \mathrm{MLP}_{\text{pred}}(\mathbf{e}_t^{(i)}) \tag{3}$$

Only $\mathbf{e}_t^{(i)}$ and $y_t^{(i)}$ are transmitted to the Cloud.

4.2 Cloud Aggregator Architecture

The Cloud model integrates embeddings from all households and shared contextual features through an attention-based aggregation mechanism. Given the contextual feature vector $\mathbf{c}_t$, the Cloud first generates a contextual query representation:

$$\mathbf{z}_t = \mathrm{MLP}_{\text{ctx}}(\mathbf{c}_t), \tag{4}$$

where $\mathrm{MLP}_{\text{ctx}}$ is a multilayer perceptron that encodes weather, time, or other public information.

Each household embedding $\mathbf{e}_t^{(i)}$ interacts with the global context $\mathbf{z}_t$ through a bilinear attention matrix $\mathbf{U}$:

$$\alpha_i = \frac{\exp\left(\mathbf{z}_t^\top \mathbf{U} \mathbf{e}_t^{(i)}\right)}{\sum_j \exp\left(\mathbf{z}_t^\top \mathbf{U} \mathbf{e}_t^{(j)}\right)}, \tag{5}$$

where α_i denotes the relative importance of household i at timestep t. The attention weights depend jointly on the shared context $\mathbf{c}_t$ and the embedding $\mathbf{e}_t^{(i)}$.

The attention outputs are aggregated to obtain a global embedding:

$$\tilde{\mathbf{e}}_t = \sum_i \alpha_i \mathbf{e}_t^{(i)}. \tag{6}$$

Finally, the global forecast combines both the attention-weighted local predictions and the contextual global MLP head:

$$\hat{y}_t = \beta \sum_i \alpha_i \hat{y}_t^{(i)} + (1-\beta)\,\mathrm{MLP}([\mathbf{z}_t, \tilde{\mathbf{e}}_t]), \tag{7}$$

where $\beta \in [0,1]$ is a learnable mixing weight balancing local ensemble and global context refinement.

This formulation allows the Cloud to adaptively emphasize households that are contextually relevant while learning a global forecasting representation. All parameters $\mathbf{U}$ and those in $\mathrm{MLP}_{\mathrm{ctx}}$ and MLP are trained jointly with the rest of the Cloud model.

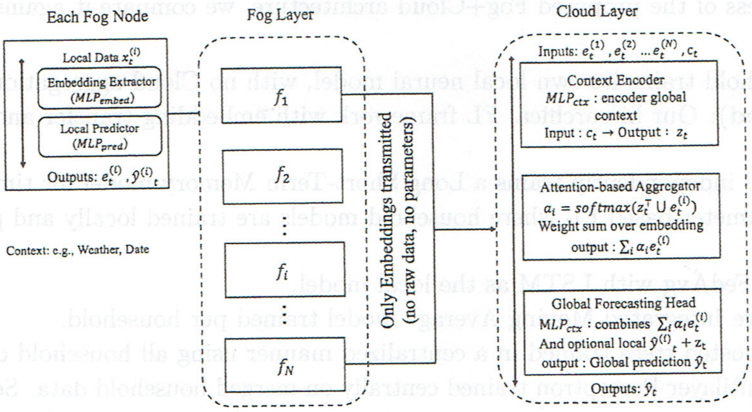

Each Fog Node
Local Data $x_t^{(i)}$
Embedding Extractor (MLP_{embed})
Local Predictor (MLP_{pred})
Outputs: $e_t^{(i)}, \hat{y}^{(i)}_t$
Context: e.g., Weather, Date
Fog Layer
f_1
f_2
f_i
f_N
Only Embeddings transmited (no raw data, no parameters)
Cloud Layer
Inputs: $e_t^{(1)}, e_t^{(2)} \dots e_t^{(N)}, c_t$
Context Encoder
MLP_{ctx} : encoder global context
Input : c_t → Output : z_t
Attention-based Aggregator
$a_i = softmax(z_t^{\top} \cup e_t^{(i)})$
Weight sum over embedding
output : $\sum_i \alpha_i e_t^{(i)}$
Global Forecasting Head
MLP_{ctx} : combines $\sum_i \alpha_i e_t^{(i)}$
And optional local $\hat{y}^{(i)}_t + z_t$
output : Global prediction $\hat{y}_t$
Outputs: $\hat{y}_t$

Figure 1: Proposed Representation-Based Hierarchical Federated Learning Architecture. Each Fog node extracts temporal embeddings from local data and sends them to the Cloud, which aggregates them using attention mechanisms combined with public contextual information. The Cloud model produces globally informed forecasts without accessing raw household data or model parameters.

The embeddings are transferred per timestep and per sample, enabling the Cloud to dynamically learn global consumption patterns through attention over temporally varying representations, rather than relying on static

4.3 Training Algorithm

Algorithm 1 Training Protocol of Fog+Cloud Architecture

```
Require: Local datasets $\{D_i\}$, global context $\{\mathbf{c}_t\}$
Initialize Fog model parameters $\theta_i$, Cloud parameters $\phi$
for each communication round $r = 1 \dots R$ do
    for each household $i$ in parallel do
        Train $g_{\theta_i}$ locally on $D_i$ for $E_{\mathrm{fog}}$ epochs
        Extract embeddings $\mathbf{e}_t^{(i)} = g_{\theta_i}(\mathbf{x}_t^{(i)})$
        Send $\{\mathbf{e}_t^{(i)}, y_t^{(i)}, \mathbf{c}_t\}$ to Cloud
    end for
    Aggregate embeddings at Cloud
    Train $f_\phi$ on $\{\mathbf{e}_t^{(i)}, y_t^{(i)}, \mathbf{c}_t\}$ for $E_{\mathrm{cloud}}$ epochs
end for
```

4.4 Training Protocol

We alternate between:

- Local Fog training on D_i to optimize θ_i.
- Cloud training on aggregated embeddings $\{(\mathbf{e}_t^{(i)}, y_t^{(i)}, \mathbf{c}_t)\}$ to optimize ϕ.

4.5 Core Differences from Parameter-Based FL

Our proposed Fog+Cloud architecture differs fundamentally from classical parameter-based federated learning methods (such as FedAvg and FedLSTM) in both information flow and modeling capability.

In parameter-based FL, local models periodically send model parameters to the Cloud, which aggregates them via simple averaging. This aggregation assumes that model parameters are semantically aligned across heterogeneous clients—a problematic assumption in highly personalized domains such as household energy consumption.

In contrast, our approach transfers temporal embeddings from each Fog node to the Cloud. These embeddings encode high-level representations of local behaviors, enabling the Cloud to perform dynamic attention-based aggregation. Furthermore, the Cloud model incorporates public context and household embeddings to explicitly model inter-household differences. This architecture allows the Cloud to learn global patterns in a semantically rich and interpretable way, while maintaining privacy since raw data and local model parameters are never transmitted.

This representation-based design provides a more flexible and powerful alternative to parameter averaging, particularly in non-iid and heterogeneous federated settings.

The overall loss is:

$$\mathcal{L} = \sum_{i,t} \ell\left(y_t^{(i)}, f_\phi(\mathbf{e}_t^{(i)}, \mathbf{c}_t)\right) \tag{8}$$

5 Baseline Models

To evaluate the effectiveness of the proposed Fog+Cloud architecture, we compare it against the following baseline models:

- **Fog Only**: Each household trains its own local neural model, with no Cloud aggregation.
- **Fog+Cloud (Proposed)**: Our hierarchical FL framework with embedding transfer and attention-based Cloud aggregation.
- **LSTM**: Each household independently trains a Long Short-Term Memory model for time series forecasting.
- **FedAvg**: Classical parameter-based FL where household models are trained locally and periodically averaged on the Cloud.
- **FedLSTM**: Variant of FedAvg with LSTM as the local model.
- **ARIMA**: Autoregressive Integrated Moving Average model trained per household.
- **XGBoost**: Gradient boosted trees trained in a centralized manner using all household data.
- **Centralized MLP**: Multilayer Perceptron trained centrally on merged household data. Serves as an upper bound.

6 Evaluation Metrics

We evaluate the forecasting performance using four standard regression metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Symmetric Mean Absolute Percentage Error (SMAPE). These metrics jointly assess both accuracy and robustness of predictions.

- **Mean Squared Error (MSE)**:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{9}$$

MSE penalizes large prediction errors more heavily and emphasizes overall stability of the forecasting model.

- **Mean Absolute Error (MAE)**:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{10}$$

MAE measures the average magnitude of prediction errors, regardless of direction.

- **Coefficient of Determination (R^2)**:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{11}$$

R^2 evaluates how well the model explains the variance of the target variable, with values closer to 1 indicating better fit.

- **Symmetric Mean Absolute Percentage Error (SMAPE)**:

$$\text{SMAPE} = 100 \cdot \frac{1}{n}\sum_{i=1}^{n}\frac{2\,|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i| + \epsilon}, \quad \epsilon = 10^{-8}. \tag{12}$$

SMAPE quantifies the relative deviation between predictions and true values, normalized by their average magnitude. A smaller SMAPE indicates higher forecasting accuracy, while the small constant ϵ prevents division by zero.

In addition, we measure:

- Training time per model.
- Average CPU utilization during training.
- Peak memory usage during training.

7 Experimental Setup

7.1 Dataset

We generate synthetic smart grid data for five households over multiple months, including:

- Daily energy consumption (target variable).
- Hourly device-level consumption.
- Weather features: temperature, precipitation, sunlight, humidity.
- Temporal features: day of week, month, weekend indicator.

The input feature space is of dimension $d = 21$.

7.2 Training Protocol

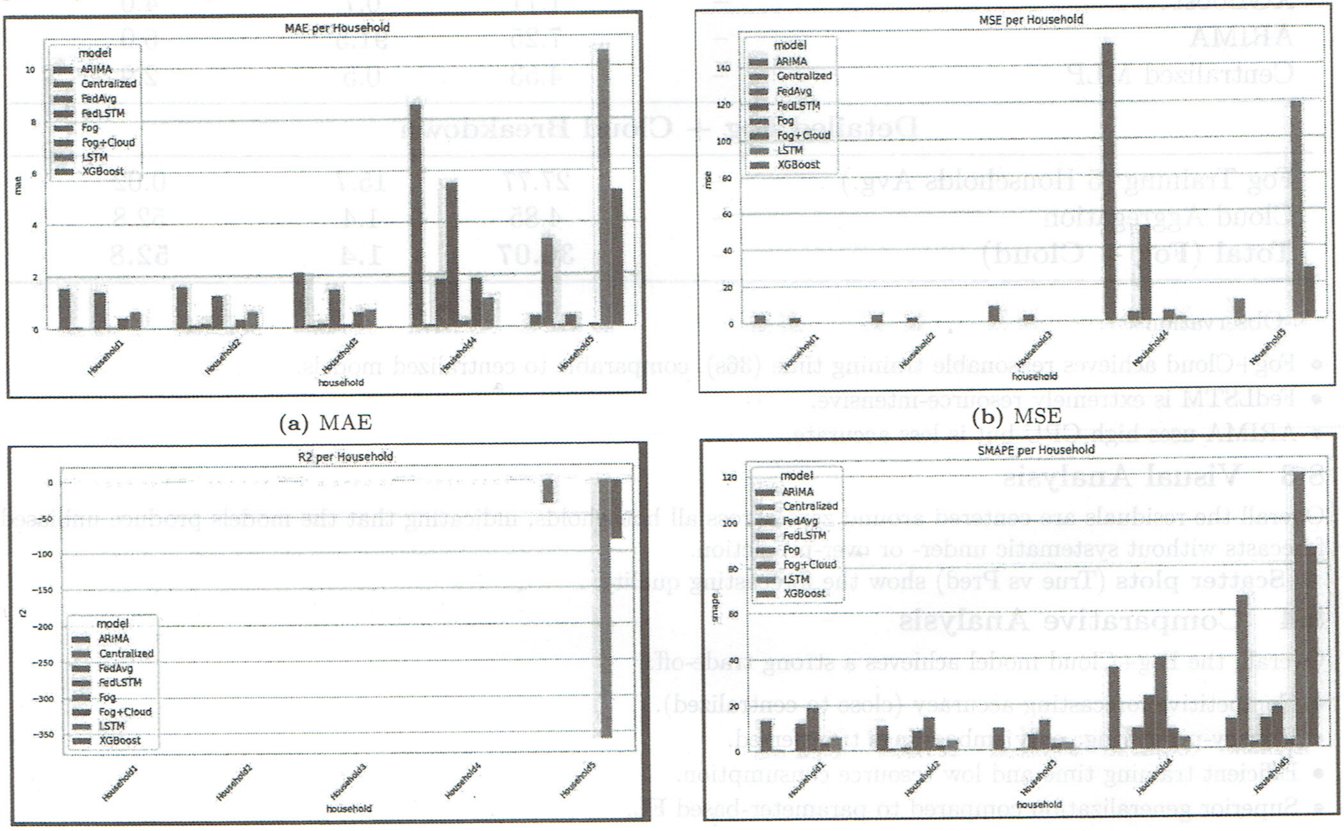
- Each Fog model is trained locally for 200 epochs.
- Fog embeddings are transferred to the Cloud for 10 communication rounds.
- Cloud model is trained with attention-based aggregation of embeddings.
- Baseline models follow their respective standard training pipelines.

We use MLPs with two hidden layers of 64 units and ReLU activations for both $\text{MLP}_{\text{embed}}$ and MLP_{pred}. The Cloud model MLP_{ctx} also uses two hidden layers of 64 units. The embedding dimension $\mathbf{e}_t^{(i)}$ is set to 64. We use Adam optimizer with learning rate 1e−3 and batch size 32 for both Fog and Cloud training. Each Fog model is trained for 200 epochs per communication round, and the Cloud model is trained for 50 epochs per round.

8 Results and Discussion

8.1 Overall Forecasting Accuracy

Across four metrics (MAE, MSE, R^2, SMAPE), RHFL ranks first or second on most households. Centralized MLP and XGBoost achieve slightly lower error in a few cases but require centralized data, whereas RHFL preserves privacy with comparable accuracy.



(a) MAE

(b) MSE

(c) R^2

(d) SMAPE

Figure 2: Comparison of forecasting performance across models.

Figures 2a–2d show that:

- The proposed Fog+Cloud model consistently outperforms Fog-only and most baseline FL models.
- Centralized MLP and XGBoost achieve slightly better raw accuracy but require centralized data.
- FedLSTM and ARIMA perform poorly under household heterogeneity.

8.2 Resource Efficiency

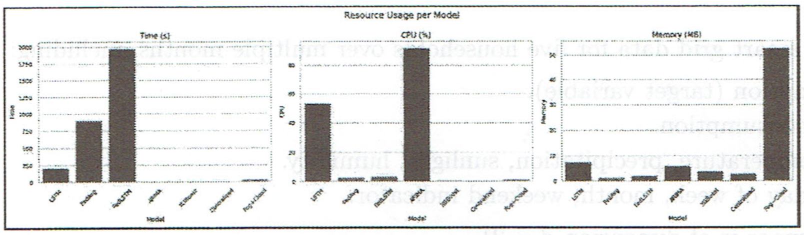


Figure 3: Resource Usage comparison across models: Training time (seconds), CPU utilization (%), and memory usage (MB).

Table 1: Comprehensive Training Resource and Time Summary Across Models

Model	**Household**	**Time (s)**	**CPU (%)**	**Memory (MB)**
Fog	Household 1	31.21	20.0	0.1
Fog	Household 2	27.17	25.0	0.0
Fog	Household 3	25.76	0.0	0.0
Fog	Household 4	26.98	33.3	0.0
Fog	Household 5	26.75	0.0	0.0
Fog+Cloud (Total)	–	36.07	1.4	52.8
LSTM	–	209.51	53.3	7.5
FedAvg	–	901.67	2.6	1.4
FedLSTM	–	1967.09	3.2	2.1
XGBoost	–	1.71	0.7	4.0
ARIMA	–	7.25	91.5	6.0
Centralized MLP	–	4.53	0.5	2.9
Detailed Fog + Cloud Breakdown				
Fog Training (5 Households Avg.)	–	27.77	15.7	0.02
Cloud Aggregation	–	4.85	1.4	52.8
Total (Fog + Cloud)	–	**36.07**	**1.4**	**52.8**

Observations:

- Fog+Cloud achieves reasonable training time (36s), comparable to centralized models.
- FedLSTM is extremely resource-intensive.
- ARIMA uses high CPU but is less accurate.

8.3 Visual Analysis

Overall the residuals are centered around zero across all households, indicating that the models produce unbiased forecasts without systematic under- or over-prediction.

Scatter plots (True vs Pred) show the forecasting quality:

8.4 Comparative Analysis

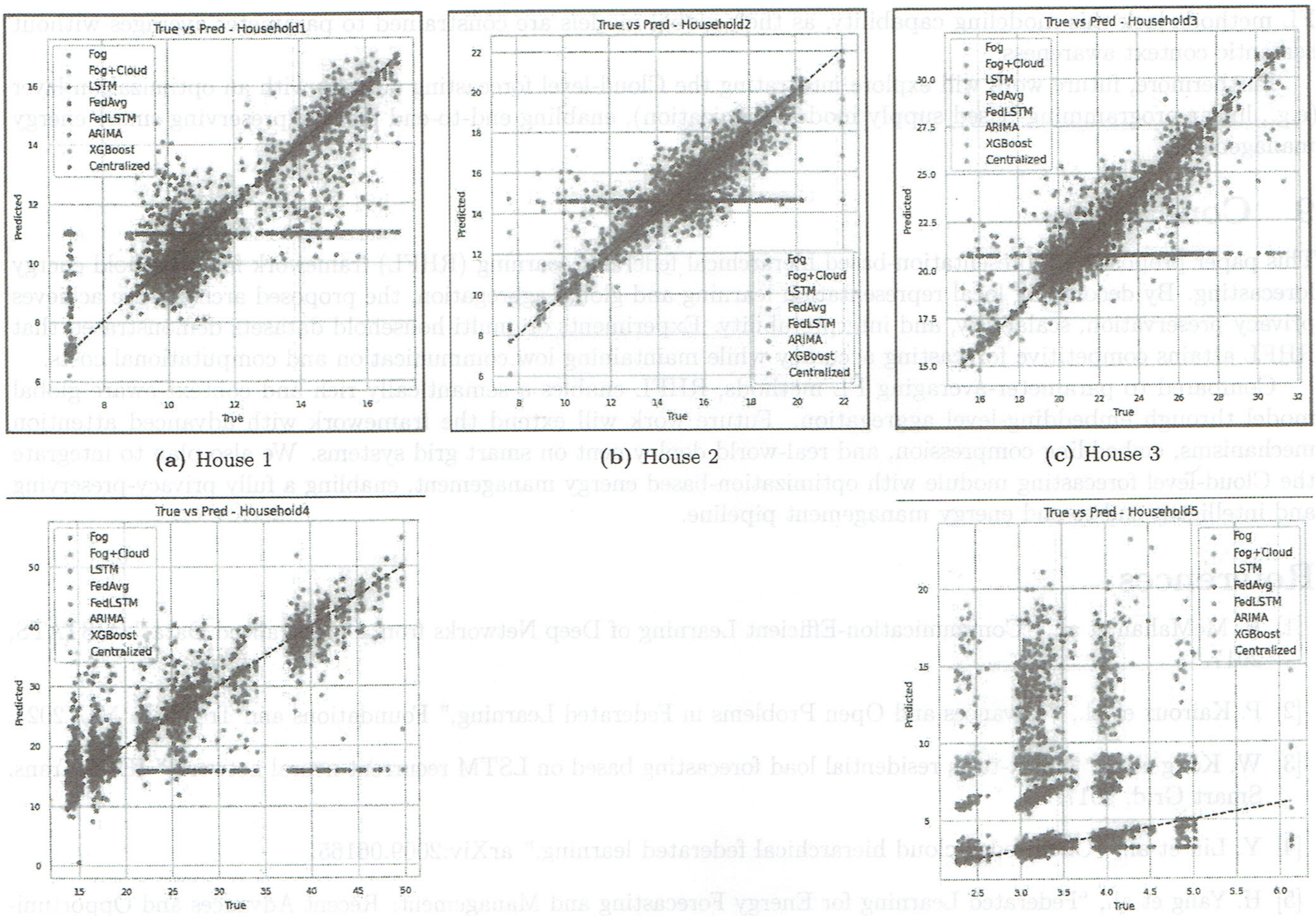
Overall, the Fog+Cloud model achieves a strong trade-off:

- Competitive forecasting accuracy (close to centralized).
- Privacy-preserving: only embeddings transferred.
- Efficient training time and low resource consumption.
- Superior generalization compared to parameter-based FL.

8.5 Limitations and Future Work

- Current Cloud model can be enhanced with more sophisticated attention mechanisms.
- Embedding compression can further reduce communication overhead.
- Real-world datasets will be evaluated in future work.

Unlike FedAvg and FedLSTM, our method enables the Cloud model to learn global patterns explicitly from the distributed embeddings, supporting interpretable and dynamically adaptable global forecasting. Parameter-based



(a) House 1

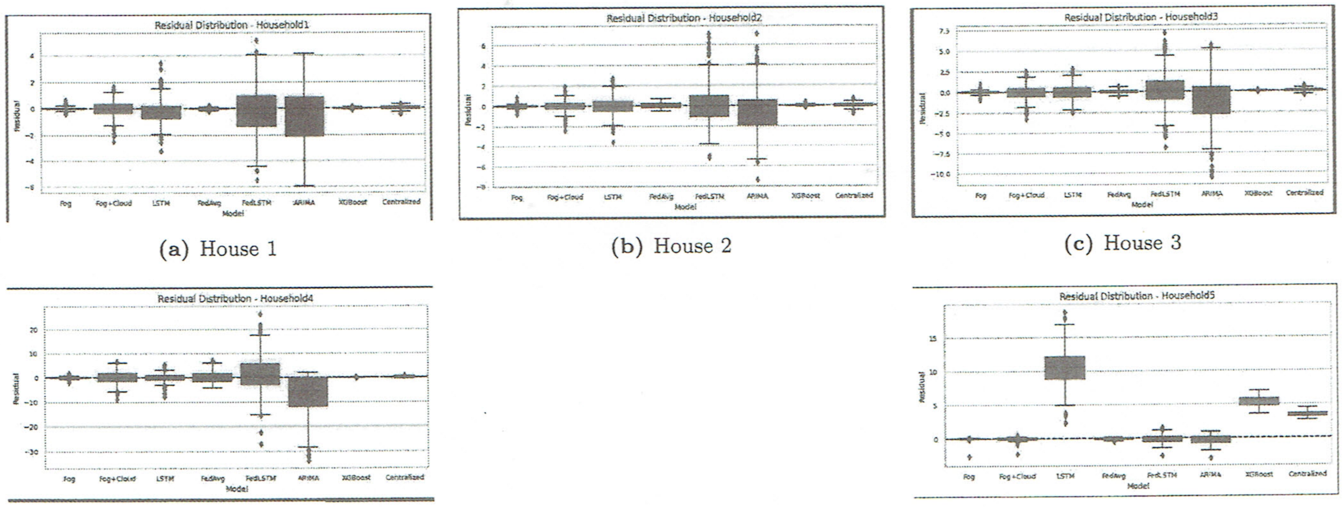
(b) House 2

(c) House 3

(d) House 4

(e) House 5

Figure 4: True vs Predicted energy consumption across five households.



(a) House 1

(b) House 2

(c) House 3

(d) House 4

(e) House 5

Figure 5: Residual distributions across five households.

FL methods lack this modeling capability, as their global models are constrained to parameter averages without semantic context awareness.

Furthermore, future work will explore integrating the Cloud-level forecasting outputs with an optimization layer (e.g., linear programming-based supply mode optimization), enabling end-to-end privacy-preserving smart energy management.

9 Conclusion

This paper proposed a representation-based hierarchical federated learning (RHFL) framework for household energy forecasting. By decoupling local representation learning and global aggregation, the proposed architecture achieves privacy preservation, scalability, and interpretability. Experiments on multi-household datasets demonstrated that RHFL attains competitive forecasting accuracy while maintaining low communication and computational costs.

Compared to parameter-averaging FL methods, RHFL enables a semantically rich and context-aware global model through embedding-level aggregation. Future work will extend the framework with advanced attention mechanisms, embedding compression, and real-world deployment on smart grid systems. We also plan to integrate the Cloud-level forecasting module with optimization-based energy management, enabling a fully privacy-preserving and intelligent end-to-end energy management pipeline.

References

[1] B. McMahan et al., “Communication-Efficient Learning of Deep Networks from Decentralized Data,” AISTATS, 2017.

[2] P. Kairouz et al., “Advances and Open Problems in Federated Learning,” Foundations and Trends in ML, 2021.

[3] W. Kong et al., “Short-term residential load forecasting based on LSTM recurrent neural network,” IEEE Trans. Smart Grid, 2017.

[4] Y. Liu et al., “Client-edge-cloud hierarchical federated learning,” arXiv:2009.06165.

[5] H. Yang et al., “Federated Learning for Energy Forecasting and Management: Recent Advances and Opportunities,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1931–1944, 2022.

[6] T. Li et al., “Federated Learning for Privacy-Preserving Smart Meter Data Analysis in Smart Grids,” *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4291–4301, 2021.

[7] Y. Guo and L. Li, “Japanese Household Electric Power Consumption and Power Supply Mode Optimization Model Based on Neural Network and Linear Programming,” in *Proc. of the 11th International Conference on Power and Energy Systems Engineering (CPESE)*, 2024, pp. 175–182. doi: 10.1109/CPESE62584.2024.10841164.