A Robust and Heterogeneity-Aware Federated Learning Framework with Knowledge Distillation for Cross-Regional Load Forecasting

Zhifeng Zuo[1], Hongxing Ye[1], Jie Li[2], and Yinyin Ge[1,*]

***Abstract*—Accurate load forecasting is fundamental for power system operation and planning. While traditional single-region approaches are constrained by limited local data, cross-regional forecasting leverages larger datasets to achieve higher accuracy. Federated learning (FL) emerges as a promising solution, enabling cross-regional collaboration. However, existing FL-based approaches struggle with model, system, and statistical heterogeneity, along with security vulnerabilities. To address these issues, this paper proposes a robustness-enhanced personalized federated learning framework that integrates knowledge distillation for cross-regional load forecasting. Proxy models are utilized to enable secure knowledge transfer while preserving local model adaptability, thereby resolving model heterogeneity. Perturbed Gradient Descent (PGD) mitigates statistical heterogeneity, and a dynamic exit mechanism reduces computational costs by allowing clients to exit early upon meeting accuracy thresholds, addressing system heterogeneity. Case studies on open-access energy dataset from six European countries show that the proposed method outperforms conventional FL models, personalized FL methods, and traditional robust aggregation schemes in terms of accuracy, robustness, and model resilience.**

***Index Terms*—Federated Learning, Knowledge Distillation, Cross-Regional Load Forecasting, Model Robustness.**

I. Introduction

As the global energy sector transitions towards distributed, diversified, and sustainable energy sources, efficient energy management has become a cornerstone of achieving energy security. The growing integration of renewable sources, such as solar and wind, introduces additional complexity to load forecasting due to their inherent intermittency and variability [1]. Accurate load forecasting is therefore indispensable for optimizing energy resource distribution, ensuring grid reliability under varying demand conditions, and reducing operational expenditures attributed to generation and reserve acquisition costs [2]. In this context, cross-regional load forecasting has garnered considerable research attention due to its critical role in the planning, operation, and energy management of distributed power systems [3].

With the increasing integration of distributed energy resources, traditional load forecasting methods struggle to construct accurate load models due to the numerous complex factors influencing electricity consumption. Fortunately, the rise of artificial intelligence (AI) has shifted the focus toward data-driven approaches, making AI the primary research method for load forecasting. Moreover, advancements in intelligent energy measurement devices and communication technologies have led to the accumulation of vast amounts of data, including power load records, meteorological conditions, and geographic information, which serve as a foundation for leveraging AI, big data analysis, and other advanced technologies in load forecasting [4]. Traditional machine learning techniques, such as artificial neural networks [5], [6], support vector machines [7], random forests [8], and ensemble learning [9], [10], have been widely applied. More recently, deep learning has demonstrated superior performance by extracting hidden patterns through multi-layer nonlinear mappings, significantly improving prediction accuracy. Deep learning techniques, such as LSTM networks for short-term load forecasting [11], industry correlation models for medium-long term forecasting [12], multi-source parameter coupling using LSTM [13], multi-view neural networks for forecasting [14], and Deep Belief Networks for short-term forecasting [15], have demonstrated significant improvements in prediction accuracy.

Despite the advancements mentioned above, most existing load forecasting models are typically trained in isolation for specific regions managed by Regional Transmission Organizations, often overlooking the potential benefits of valuable supplementary datasets from other regions [16]. Moreover, issues with data acquisition—such as sensor failures or communication disruptions—can introduce noise and missing samples, leading to data corruption in model training. This few-shot problem in data-driven load forecasting inevitably results in inaccurate predictions. A natural solution would be integrating data from multiple regions to improve model integrity and robustness. FL, a specialized distributed machine learning framework, addresses this challenge by enabling multiple clients to collaboratively train models without requiring centralized data aggregation. This approach has shown promising results in multiple AI applications in power systems [17], [18]. By leveraging model coordination and adaptive aggregation strategies, FL supports knowledge sharing across regions while maintaining the autonomy of local datasets. By expanding the sample space accessible to each client, FL advances model performance and demonstrates significant potential for advancing cross-regional load forecasting.

While Federated Learning (FL) introduces a decentralized

[1] Zhifeng Zuo and Yinyin Ge are with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China (emails: zhifengzuo@stu.xjtu.edu.cn; geyinyin@xjtu.edu.cn).

[1] Hongxing Ye is with the School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an, China (email: yehxing@xjtu.edu.cn).

[2] Jie Li is with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, USA (email: lijie@rowan.edu).

*Corresponding author: Yinyin Ge (email: geyinyin@xjtu.edu.cn).

paradigm that improves system resilience by distributing the training process across multiple clients, it also exposes the learning process to new vulnerabilities—particularly from Byzantine clients, which may transmit incorrect or even malicious updates due to faults or adversarial behavior. This poses a significant threat to the robustness and reliability of FL-based systems, especially in safety-critical applications such as power system forecasting. The challenge is further complicated by the data heterogeneity inherent in FL, where client data distributions vary widely, making it difficult to distinguish truly malicious updates from legitimate but divergent ones [19]. Among various threat models, Byzantine attacks are considered the most destructive due to their arbitrary and unpredictable nature. Prior studies have shown that even a small number of Byzantine clients can effectively undermine robust aggregation mechanisms like Krum and Trimmed Mean, causing severe degradation in model performance [20]–[22]. Therefore, defending against Byzantine behaviors is essential not only for ensuring fairness and accuracy, but also for enhancing the fault tolerance and overall resilience of federated load forecasting systems.

Beyond adversarial threats, FL also faces cross-regional heterogeneity challenges, which manifest in three key aspects: model heterogeneity, due to differences in data formats and structures; system heterogeneity, caused by variations in computational resources and infrastructure; and statistical heterogeneity, stemming from Non-Independent and Identically Distributed (Non-IID) data distributions [23], [24]. These factors often lead to degraded performance or unstable convergence when training a single global model. As highlighted in [25], global models may fail to provide uniformly good performance across all clients, and in some cases, clients may derive no benefit from participating in FL at all [26].

To address such limitations, personalized federated learning (PFL) has emerged as a promising solution. PFL aims to tailor global training outcomes to the specific characteristics of each client, thus mitigating the impact of heterogeneity. Prior studies have proposed various approaches for personalization, including client clustering based on similarity [27], transfer learning via fine-tuning on local data [28], and relevance-based sample selection [29]. Recently, knowledge distillation has attracted increasing attention in the PFL context. By treating the global model as a teacher and client-specific models as students, knowledge can be transferred without direct parameter sharing [30]. Distillation-based methods such as FedMD [31] and its extensions [32] show promising flexibility in handling architectural diversity and protecting privacy. However, despite these advantages, the use of knowledge distillation for improving robustness against Byzantine threats has received limited attention. Moreover, existing personalized FL frameworks tend to focus narrowly on statistical heterogeneity, without adequately addressing the combined impact of model diversity and system-level disparities—both of which are prevalent in real-world FL applications.

To address these challenges particularly the underexplored issues of robustness and multi-dimensional heterogeneity this paper proposes a robustness-enhanced personalized federated learning (FL) framework that incorporates knowledge distillation to enable cross-regional load forecasting with improved model resilience against unreliable or adversarial participants. This framework employs a Gated Recurrent Unit (GRU) model as the base forecasting architecture and a proxy model to facilitate secure knowledge sharing across regions. Instead of directly training a global model, FL is performed on the proxy model, which is introduced to transfer globally aggregated knowledge to local models through knowledge distillation. Compared with the published literature, the main contributions of this paper are as follows:

1) To address model heterogeneity arising from inconsistent feature dimensions and potentially varied model architectures, this paper introduces a proxy-model-based federated learning framework, proposed here for the first time. The proxy model acts as an intermediate representation that bridges heterogeneous local models, mapping each local model's heterogeneous features into a feature space that is identical across all proxy models. Each proxy model receives latent features distilled from its local model and, in turn, transfers globally aggregated knowledge back to the local model. This bidirectional distillation ensures that local models benefit from cross-regional information while retaining their region-specific characteristics. Because model exchange occurs at the level of latent representations rather than direct parameter sharing, the framework avoids parameter inconsistency issues and inherently limits the propagation of poisoned updates, which substantially improves robustness against Byzantine attacks while enabling the identification and isolation of compromised clients—a capability that traditional robust aggregation methods lack.

2) The framework incorporates a Perturbed Gradient Descent (PGD) algorithm to optimize proxy models under Non-IID data conditions. Unlike conventional local Stochastic Gradient Descent (SGD), PGD explicitly aligns each proxy model's update direction with global training objectives, promoting stability and consistency across clients. This design strengthens the robustness of the distillation process, a factor often overlooked in prior federated knowledge distillation approaches facing statistical heterogeneity.

3) To accommodate varying computational capacities among clients, a dynamic exit mechanism is introduced. This mechanism allows clients with different computational capacities to participate according to their own capabilities by exiting training early based on preset thresholds, rather than requiring uniform participation as in traditional FL frameworks — a constraint that is often impractical under system heterogeneity.

The rest of the paper is organized as follows. Section II introduces the proposed FL framework with knowledge distillation, detailing the proxy model design, PGD for handling statistical heterogeneity, and the dynamic exit mechanism for addressing system heterogeneity. Section III presents the experimental setup, including dataset details, evaluation metrics, baseline comparisons, and the results analysis, demonstrating the effectiveness of the proposed approach. Finally, Section IV summarizes the key findings of the study.

II. Proposed Methodology

A. Cross-Regional Load Forecasting under Heterogeneous Federated Settings

In a regional load forecasting system, let there be N regions, each associated with a local dataset $\mathcal{D}_i = \{(X_i^j, y_i^j)\}_{j=1}^{m_i}, i = 1, 2, ..., N$, where $X_i^j \in \mathbb{R}^{d_i}$ represents the input features such as weather conditions, time of day, and regional indicators, and $y_i^j \in \mathbb{R}$ denotes the corresponding load targets. d_i is the number of input features for region i, and m_i is the number of data records. The primary objective is to train region-specific models $f_i(\theta_i)$, parameterized by θ_i, which minimize the prediction error on their respective datasets. The local task loss for region i is defined as:

$$L_{\text{task}}(\theta_i) = \frac{1}{m_i} \sum_{j=1}^{m_i} \| f_i(X_i^j; \theta_i) - y_i^j \|^2, \tag{1}$$

which measures the mean squared error (MSE) between the predicted and actual load values. However, optimizing $f_i(\theta_i)$ independently for individual regions may lean towards models that are constrained by the limitations of local data, potentially resulting in declined performance due to insufficient or poor-quality training data. Leveraging cross-regional information is expected to enhance the performance of individual models via collaborative learning. However, inherent regional differences in data distribution, computational capacities, and statistical distributions further amplify the challenges of collaborative learning.

B. Framework Overview

We propose a new FL framework for cross-regional load forecasting, utilizing a GRU model to predict future load based on historical data. The process begins with data preprocessing, where regional historical data is collected, normalized, and missing values are imputed. Each client (i.e., region) maintains two local models: a personalized local model, and a proxy model which is smaller in scale to address model heterogeneity. Only the proxy model participates in FL and transfers knowledge to the personalized model through a knowledge distillation process. As shown in Fig. 1, the proxy model interacts with the global model by aligning its updates with global model parameters during gradient descent, ensuring that the proxy model adheres to the global model's guidance while retaining other unique local characteristics. The complete framework flow is summarized in Algorithm 1.

To mitigate statistical heterogeneity, the proxy model is updated using PGD with global regularization:

$$\text{proxy}_{t+1}^k = \text{proxy}_t^k - \eta \cdot \left(\nabla \mathcal{L}(\text{proxy}_t^k) + \mu(\text{proxy}_t^k - \text{global}_t) \right), \tag{2}$$

where η is the learning rate, $\mathcal{L}(\text{proxy}_t^k)$ denotes the training loss of the proxy model at round t for client k, μ is a regularization coefficient that aligns the proxy model with the global model, and global_t represents the global proxy model aggregated at round t.

Meanwhile, the client model is refined using a combined loss:

$$\text{client}_{t+1}^k = \text{client}_t^k - \eta \cdot \nabla \mathcal{L}(\text{client}_t^k), \tag{3}$$

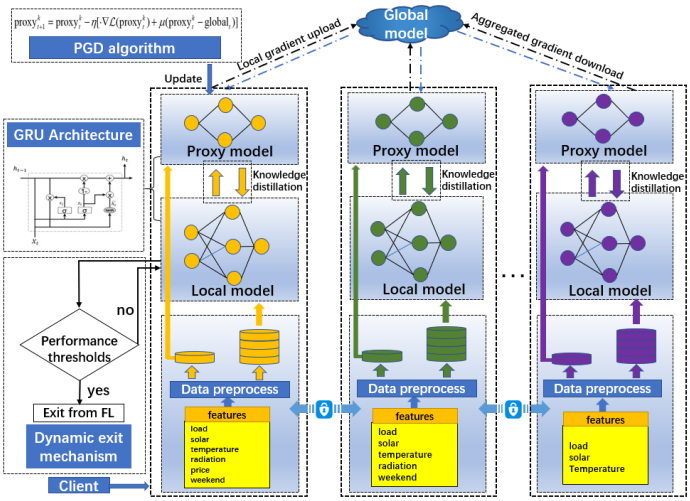

Fig. 1. The proposed FL framework.

where $\mathcal{L}(\text{client}_t^k)$ is a composite objective incorporating both local supervised loss and the distillation loss, as defined in the client update process of Algorithm 2.

Moreover, a dynamic exit mechanism is incorporated into the training process to improve overall efficiency and adapt to system heterogeneity. Specifically, each client continuously monitors its validation loss during local updates, and terminates training early once the loss falls below a predefined threshold.

C. Local and Proxy Model Architecture

In the proposed federated learning framework, both the local model $f_i(\theta_i)$ and the proxy model $p_i(\phi_i)$ adopt a two-layer Gated Recurrent Unit (GRU) [33] architecture to capture the temporal dependencies in regional load forecasting tasks. GRU is selected for its structural simplicity, computational efficiency, and strong performance on sequential data. Since clients often serve as edge devices with limited computational and memory resources, it is crucial to utilize lightweight and simple models.

Each model consists of:

- Two stacked GRU layers, each with 256 hidden units;
- A fully connected (dense) output layer to produce the final forecasted value.

As illustrated in Fig. 2, the input to each model is a sequence $X_i^j \in \mathbb{R}^{T \times d}$ representing T time steps and d input features (which may differ between clients), and the output is a predicted load value $\hat{y}_i^j \in \mathbb{R}$. The detailed architecture of the model is described below:

a) GRU Cell Dynamics.: The GRU cell is defined by the following set of equations for each time step t:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{4}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{5}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{6}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{7}$$

where:

Algorithm 1 Federated Learning with Dynamic Exit and Distillation (server-side)

Input: Client set $\mathcal{C}$, exit threshold δ, total training rounds T, distillation epochs E_d, local epochs E, local data $(\mathcal{X}_k, \mathcal{Y}_k)$

Output: Final local models $\{f_k(\theta_k^T)\}_{k\in\mathcal{C}}$ and predictions $\{\hat{Y}_k\}_{k\in\mathcal{C}}$

Notations: $\mathbf{w}_{t+1}$ is the global model after federated averaging at round $t+1$, $\mathcal{S}$ is the set of active clients in the training round, $|\mathcal{S}|$ is the number of active clients, client_{t+1}^k and proxy_{t+1}^k are the updated models from clients and proxies respectively.

Server Execution:
Initialize $\mathcal{S}$ and client_0^k, proxy_0^k
for each round $t = 1, 2, \ldots, T$ **do**
  for each client k in $\mathcal{S}$ **do**
    client_{t+1}^k,proxy_{t+1}^k←ClientUpdate(client_t^k,proxy_t^k)
    if client k meets exit condition (loss $\leq \delta$) **then**
      Mark client k for exit
      Update the set of active clients $\mathcal{S}$
    end if
  end for
  Update global model:
  global_{t+1}←Federated Average(proxy_{t+1}^k)
end for
Federated Average:
for each client k in $\mathcal{S}$ **do**

$$\mathbf{w}_{t+1} = \frac{1}{|\mathcal{S}|} \sum_{k\in\mathcal{S}} \text{proxy}_{t+1}^k$$

end for

- x_t is the input at time t;
- h_{t-1} is the hidden state from the previous time step;
- z_t and r_t are the update and reset gates, respectively;
- $\tilde{h}_t$ is the candidate activation;
- W_*, U_*, b_* are the learnable weights and biases;
- σ denotes the sigmoid function and $\odot$ denotes element-wise multiplication.

b) Model Output.: After processing the input sequence through two GRU layers, the final hidden state h_T is passed to a fully connected layer:

$$\hat{y}_i^j = \text{FC}(h_T) \tag{8}$$

where FC denotes the final dense layer.

D. Addressing Model Heterogeneity through Proxy-Based Knowledge Distillation

To address model heterogeneity, the proxy model $p_i(\phi_i)$ is introduced for each region i, which serves as intermediaries to standardize and abstract the local data representations. Both the proxy and local models share an identical model architecture, differing only in the number of input features according to their regional data. This architectural consistency ensures that knowledge distilled from the proxy to the local models preserves the underlying temporal and structural patterns, effectively reducing potential information loss.

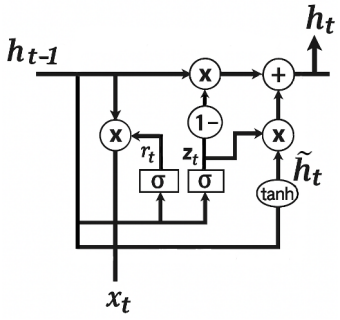

Fig. 2. Architecture of the GRU-based local and proxy models. Input sequences are encoded through stacked GRU layers, followed by a dense output layer for forecasting.

These lightweight proxy models are compact and structurally uniform, allowing seamless integration across clients with heterogeneous feature sets. By mapping all regional inputs into an identical latent dimensionality through feature masking, the proxy ensures a consistent representation space that supports coherent aggregation across heterogeneous clients. As shown in Fig. 3, a feature-masking strategy is applied to align different regional feature sets into a consistent input dimension for the proxy model. Specifically, for client i at time step j, the original input feature vector $X_i^j \in \mathbb{R}^{d_i}$, where d_i may vary across clients, is transformed into a masked vector $\bar{X}_i^j \in \mathbb{R}^{d_L}$, retaining only the essential load features:

$$\bar{X}_i^j = M_i \odot X_i^j, \quad \bar{X}_i^j \in \mathbb{R}^{d_L}, \quad d_L < d_i, \tag{9}$$

where $M_i \in \{0,1\}^{d_i}$ is the feature mask for client i, $\odot$ denotes element-wise multiplication, and d_L is the number of essential load features shared across all clients.

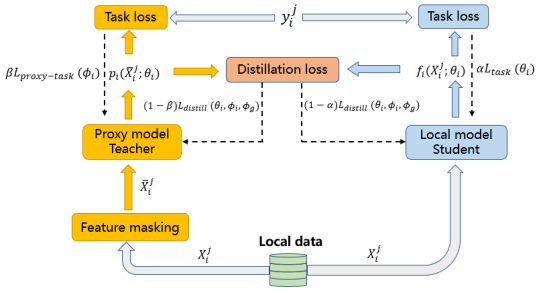

Fig. 3. Knowledge distillation process: heterogeneous client models (with different feature sets) transfer knowledge to the proxy model, which aligns client features X_i^j to a unified representation $\bar{X}_i^j$ retaining only essential load features.

The global model serves as a centralized knowledge aggregator, systematically consolidating and harmonizing regional knowledge to establish a comprehensive understanding of diverse geographical characteristics. Through the federated averaging mechanism, it iteratively updates its parameters by aggregating and synchronizing knowledge from all regional proxy models:

$$\phi_g^{t+1} = \frac{1}{N} \sum_{i=1}^{N} \phi_i^t. \tag{10}$$

where ϕ_i^t denotes the parameters of the regional proxy model for region i at iteration t,and N is the total number of regions participating in the aggregation. Since the proxy and global models share an identical architecture, the aggregation process preserves strict structural consistency. After aggregation, the proxy model further aligns its parameters with the updated global model through PGD, which is applied as the gradient update for the proxy model within the local–proxy knowledge distillation.

Knowledge transfer to local models is achieved through a distillation loss that enables each local model $f_i(\theta_i)$ to learn from its region-specific proxy model. Because the proxy model abstracts global knowledge—obtained through its parameter alignment with the aggregated global model, it provides a stable and reliable teacher for heterogeneous local models. The distillation loss is defined as:

$$L_{\text{distill}}(\theta_i, \phi_i, \phi_g) = \frac{1}{m_i} \sum_{j=1}^{m_i} w_j \| f_i(X_i^j; \theta_i) - p_i(\bar{X}_i^j; \phi_i) \|^2, \tag{11}$$

where w_j is the weight for each sample based on the similarity between the predictions of the global model $g(X_i^j; \phi_g)$ and the true labels. This weight is computed as:

$$w_j = \text{softmax}\left(-\frac{\| g(\bar{X}_i^j; \phi_g) - y_i^j \|}{\tau} \right), \tag{12}$$

where τ is a temperature parameter that controls the smoothness of the weight distribution. This reliability-aware weighting highlights samples where global model predictions are consistent with true values, suppressing unreliable teacher signals and enhancing the effectiveness of knowledge transfer.

Both local and proxy models incorporate the distillation loss in their training objectives to remain aligned, enabling mutual guidance and enhancing prediction accuracy across regions. The total loss for the local model can be written as:

$$L_{\text{local}}(\theta_i, \phi_i, \phi_g) = \alpha L_{\text{task}}(\theta_i) + (1 - \alpha) L_{\text{distill}}(\theta_i, \phi_i, \phi_g), \tag{13}$$

where $L_{\text{task}}(\theta_i)$ represents the task-specific loss, $L_{\text{distill}}(\theta_i, \phi_i, \phi_g)$ is the distillation loss, and α is a hyperparameter that balances task performance and knowledge transfer.

Similarly, the proxy model's loss can be expressed as:

$$\begin{aligned} L_{\text{proxy}}(\theta_i, \phi_i, \phi_g) = \beta L_{\text{proxy-task}}(\phi_i) \\ + (1 - \beta) L_{\text{distill}}(\theta_i, \phi_i, \phi_g), \end{aligned} \tag{14}$$

where $L_{\text{proxy-task}}(\phi_i)$ is the loss related to the task of the proxy model. $L_{\text{distill}}(\theta_i, \phi_i, \phi_g)$ is the distillation loss from the local model. β is a hyperparameter that controls the influence of the loss of distillation on the proxy model.

By leveraging proxy models and distillation, this framework ensures secure and efficient knowledge sharing while maintaining local adaptation and prediction accuracy, making it particularly suitable for the application of cross-regional load forecasting. Considering geographical characteristics of cross-regional load, we note that the proposed proxy-based framework is compatible with extensions to spatial-temporal modeling, which offers a promising direction for future improvements.

Algorithm 2 ClientUpdate with Local Training and Knowledge Distillation (client-side)

Input: Client model client_t^k, proxy model proxy_t^k, distillation epochs E_d, local epochs E, local data $(\mathcal{X}_k, \mathcal{Y}_k)$
Output: Updated $(\text{client}_{t+1}^k, \text{proxy}_{t+1}^k)$

ClientUpdate:
for each client k **do**
  Run Local Model Training:
  for each epoch $e = 1, 2, \ldots, E$ **do**
    Train local model using local data $(\mathcal{X}_k, \mathcal{Y}_k)$
  end for
  for each epoch $e = 1, 2, \ldots, E_d$ **do**
    Run Knowledge Distillation with local model and proxy model:
    Update client_{t+1}^k and proxy_{t+1}^k:
    $\text{client}_{t+1}^k = \text{client}_t^k - \eta \cdot \nabla \mathcal{L}(\text{client}_t^k)$
    $\text{proxy}_{t+1}^k = \text{proxy}_t^k - \eta[\cdot \nabla \mathcal{L}(\text{proxy}_t^k) + \mu(\text{proxy}_t^k - \text{global}_t)]$
  end for
end for

E. PGD Algorithm for Statistical Heterogeneity

To address statistical heterogeneity, the PGD algorithm is employed for updating proxy models. PGD incorporates a regularization term that aligns local proxy models with the global proxy model, thereby mitigating the impact of these data distribution differences [34]. The PGD update rule is defined as:

$$\phi_i^{t+1} = \phi_i^t - \eta \left(\nabla_{\phi_i} L_{\text{proxy}}(\phi_i) + \mu(\phi_i^t - \phi_g^t) \right), \tag{15}$$

where η is the learning rate, $L_{\text{proxy}}(\phi_i)$ represents the loss for proxy model for region i, and μ is a regularization coefficient controlling the influence of the global proxy model $g(\phi_g)$. The regularization term $\mu(\phi_i^t - \phi_g^t)$ ensures that proxy models trained on diverse regional datasets remain aligned with global knowledge, reducing the adverse effects of statistical heterogeneity, while enhancing the stability and reliability of the convergence process.

Convergence Guarantee. We analyze the convergence behavior of PGD in the presence of statistical heterogeneity. Unlike standard federated learning, PGD introduces a regularization term in each local objective to align client models with the global proxy model. The overall federated optimization objective becomes:

$$\min_{\phi} F(\phi) := \sum_{k=1}^{K} p_k \left(F_k(\phi_k) + \frac{\mu}{2} \| \phi_k - \phi_g \|^2 \right), \tag{16}$$

where $F_k(\phi_k) := \mathbb{E}_{\xi \sim \mathcal{D}_k}[f(\phi_k; \xi)]$ denotes the expected local loss on client k, and μ is the proximal regularization coefficient that controls the strength of alignment with the global proxy model ϕ_g. Here, $\xi \sim \mathcal{D}_k$ denotes a data sample drawn from the local data distribution $\mathcal{D}_k$ of client k. $\phi = \{\phi_1, \ldots, \phi_k\}$ denotes the set of local proxy model parameters for all clients, and ϕ_g is the global proxy model shared across clients.

To establish convergence, the following standard assumptions, consistent with those in the aforementioned literature, are made:

- **L-smoothness**: Each local objective F_k is L-smooth, i.e., $\|\nabla F_k(\phi) - \nabla F_k(\phi')\| \leq L\|\phi - \phi'\|$ for all $\phi, \phi' \in \mathbb{R}^d$.
- **Bounded Dissimilarity**: There exists a constant $B \geq 1$ such that $\mathbb{E}_k\|\nabla F_k(\phi)\|^2 \leq B^2\|\nabla F(\phi)\|^2$ for all ϕ.
- **Inexact Local Solution**: Each client performs K local updates per round and returns an ϵ-accurate solution ϕ_k^{t+1} such that $F_k(\phi_k^{t+1}) + \frac{\mu}{2}\|\phi_k^{t+1} - \phi_g^t\|^2 \leq \min_\phi \left(F_k(\phi) + \frac{\mu}{2}\|\phi - \phi_g^t\|^2\right) + \epsilon$.

Under these conditions, it can be shown that PGD ensures convergence in expectation. Specifically, after T global communication rounds, the average squared gradient norm satisfies:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\phi^t)\|^2\right] \leq \mathcal{O}\left(\frac{1}{T} + \frac{B^2\mu^2\epsilon^2}{K}\right), \tag{17}$$

where ϕ^t is the aggregated proxy model at round t, μ is the proximal regularization coefficient, and K is the number of local updates per round.

This bound decomposes into two terms:

- $\mathcal{O}(1/T)$: Reflects the standard optimization error that vanishes as T increases.
- $\mathcal{O}(B^2\mu^2\epsilon^2/K)$: Captures the bias introduced by statistical heterogeneity and local approximation error, scaled by the regularization strength μ.

Thus, by selecting an appropriate μ and increasing K, PGD effectively limits the influence of client drift and mitigates the impact of data heterogeneity. The proximal term $\mu\left(\phi_i^t - \phi_g^t\right)$ serves as a stabilizer, encouraging local models to remain close to the global reference and leading to smoother and more stable convergence behavior in federated settings.

F. Dynamic Exit Mechanism for System Heterogeneity and Deployment

To accommodate system heterogeneity, we introduce a dynamic exit mechanism that allows clients to exit the training process once their performance meets a predefined threshold δ. The exit condition is defined as:

$$\mathbb{I}_{\text{exit}}(i) = \begin{cases} 1, & \text{if } L_{\text{val}}(i) \leq \delta \\ 0, & \text{otherwise} \end{cases}, \tag{18}$$

where $\mathbb{I}\text{exit}(i)$ is the binary indicator for whether client i exits the training process (1 indicates the condition is satisfied), $L_{\text{val}}(i)$ is the validation loss. Once a client exits, it no longer participates in local training or global aggregation. Clients are allowed to perform a varying number of local iterations following this dynamic exit mechanism, unlike traditional FL, which requires equal epochs for all clients. The contribution of client updates is determined by whether the client is active during the training process. This is achieved by modifying the federated averaging rule:

$$\phi_g^{t+1} = \frac{\sum_{i=1}^N \mathbb{I}_{\text{active}}(i)\cdot\phi_i^t}{\sum_{i=1}^N \mathbb{I}_{\text{active}}(i)}, \tag{19}$$

where $\mathbb{I}_{\text{active}}(i)$ indicates whether client i is still active, and ϕ_i^t represents the model parameters of client i at iteration t.

To track the clients that are still participating in the current round of training, S is defined as the set of active clients which is dynamically updated each round. The global model is then updated by aggregating the models only from the active clients. The dynamic exit mechanism mitigates system heterogeneity by allowing clients to participate according to their computational capacity, even when their training progress differs.

In practical deployment, the framework operates in an offline training–online prediction mode. Local and proxy models are collaboratively trained offline through federated rounds using the dynamic exit mechanism. After training is completed, only the inference components of regional models are deployed for online forecasting. Newly arriving data are directly fed into these fixed models to generate real-time rolling predictions, without performing any online parameter updates. This design minimizes computation and communication overhead in real-world operation. To accommodate long-term distributional shifts, the entire framework can be periodically retrained offline from scratch using accumulated new data, ensuring full adaptation to evolving regional load patterns.

III. Case Study

In this case study, comprehensive experiments are conducted to evaluate the performance and efficiency of the proposed FL framework for cross-regional load forecasting. Specifically, we investigate the impact of the framework's components on model accuracy and resilience. We also analyze the effectiveness of the dynamic exit mechanism in facilitating stable and efficient collaboration among clients with diverse computational resources and data characteristics.

A. Dataset Description and Preprocessing

The dataset used for our experiments is sourced from an open-access energy system dataset from the Open Power System Data Platform [36], containing energy-related data from multiple European countries, including features such as load, temperature, solar and wind power generation, electricity price, radiation intensity and day type (i.e., whether the day is a weekend). Time series data are selected from six countries, with each country treated as an independent region and a sampling interval of one hour applied.The dataset spans two years, covering the full period from 2018 to 2019. To fully demonstrate the model heterogeneity across different regions, a varying number of dataset features are allocated to each region. The six selected countries represent a balanced distribution across northern and central-western Europe, enabling us to explore the diverse energy characteristics and conditions prevalent in these areas.

The details of the data for each region and features used in the local models are specified in Table I, while the proxy model exclusively uses load as its feature and does not incorporate any region-specific private or sensitive data.

As shown in the heatmap in Fig. 4, the load in the AT region shows strong correlations with several variables. These correlations emphasize the influence of regional characteristics on load behavior and highlight these features can be effectively leveraged for load forecasting.

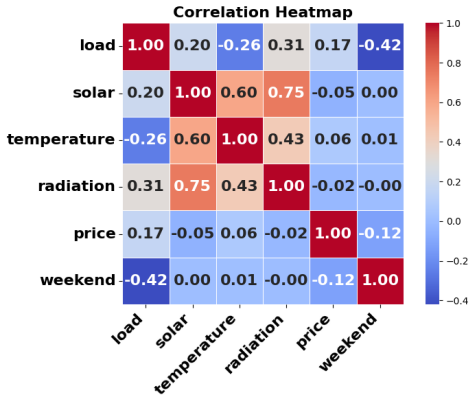

Fig. 4. Correlation heatmap of the AT region showing the relationships between load and other features.

The detailed data preprocessing procedures are summarized as follows.

a) Data Normalization and Missing Value Handling: Max-Min Normalization is applied to preprocess the input data. Each data point X is transformed using

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \tag{20}$$

where $X_{\min}$ and $X_{\max}$ are the minimum and maximum values in the dataset, ensuring that all values are scaled between 0 and 1. To address missing data, a temporal imputation method is applied by filling each missing value with the corresponding hour's data from either the previous day or the same weekday of the previous week, ensuring temporal consistency.

b) Sequence Construction for GRU Input: To prepare the data for GRU-based forecasting, a sliding window with window size of 24 hours is employed. For each time step t, the input sequence consists of the previous 24 hours:

$$X_{\text{input}}[t] = [X[t-23], X[t-22], \ldots, X[t]]. \tag{21}$$

Each $X[t]$ is composed of:

- Load values from the previous two time steps, $L[t-1]$ and $L[t-2]$, capturing short-term temporal dependencies.
- Other region-specific features at the current time step according to Table I, denoted as $F[t]$.

Thus, the GRU input at time step t is

$$X[t] = [L[t-1], L[t-2], F[t]]. \tag{22}$$

For proxy models, only historical load features are used. These are selected via the feature-masking strategy to preserve essential temporal patterns while ensuring a compact and consistent representation across regions.

The target output for each time step is the load at the next hour:

$$Y[t] = L[t+1], \tag{23}$$

allowing the model to forecast the load based on the previous 24 hours of historical data and exogenous features.

TABLE I
SELECTED REGIONAL DATASETS AND CORRESPONDING INPUT FEATURES.

Clients	Features
Austria(AT)	load,solar,temperature,radiation,price,weekend
Belgium(BE)	load,solar,temperature,radiation,weekend
Bulgaria(BG)	load,solar,temperature,radiation
Czech Republic(CZ)	load,solar,temperature,radiation,weekend
Estonia(EE)	load,solar,temperature,radiation,weekend
Finland(FI)	load,solar,temperature

c) Train-Validation Split: After preprocessing and sequence construction, the dataset is divided into training and validation sets with a 7:3 split. The split is done in temporal order to avoid information leakage, ensuring that the validation set corresponds to future time periods relative to the training set. This preserves the temporal consistency of the forecasting task and ensures reliable evaluation of the model's predictive performance.

B. Evaluation Metrics

The performance of the proposed framework is evaluated using standard metrics such as Mean Absolute Percentage Error (Mean-APE) and Maximal Absolute Percentage Error (Max-APE), which are defined as:

$$\text{Mean} - \text{APE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%, \tag{24}$$

$$\text{Max} - \text{APE} = \max_i\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%\right), \tag{25}$$

where y_i represents the true value, $\hat{y}_i$ is the predicted value, and n is the total number of prediction samples. These metrics measure the accuracy of the predictions, with lower values indicating better model performance. In this context: 1) Mean-APE represents the average error across all predictions, corresponding to the overall average accuracy of the model; 2) Max-APE reflects the worst-case error, corresponding to the minimum accuracy observed in the predictions.

C. Hyperparameter Tuning for Personalization and Robustness

Three core hyperparameters μ, α, and β control the trade-off between local adaptation and global consistency in our proposed framework:

- μ: regularization strength for aligning proxy and global models via PGD;
- α: weight of the local task loss for personalization;
- β: weight of the distillation loss for global knowledge transfer.

To determine the optimal values of key hyperparameters, we conduct a coordinate-wise grid search, where $u \in [0, 1]$ is explored with a step size of 0.1, and $\alpha, \beta \in [0, 1]$ are evaluated with a step size of 0.2. In each iteration, two parameters are

held fixed while the third is tuned to minimize the validation Mean-APE or Max-APE.

The hyperparameters α and β are particularly critical, as they control the trade-off between task-specific learning and knowledge distillation. Specifically, α modulates the contribution of the distillation loss in the training of local models, while β serves an analogous role for the proxy models. As illustrated in Fig. 5, we examine the effects of varying α and β on client EE with μ fixed at 0.5. Similar trends are observed across other clients: larger values consistently lead to lower Mean-APE, highlighting improved forecasting accuracy. To ensure a balanced integration of local objectives and global knowledge, we exclude extreme values of 0 and 1 from consideration. In particular, setting $\alpha = 1$ or $\beta = 1$ entirely removes the influence of the distillation loss, effectively disabling cross-model knowledge transfer. Based on empirical results, setting both α and β around 0.8 offers the most favorable trade-off between personalized learning and collaborative generalization.

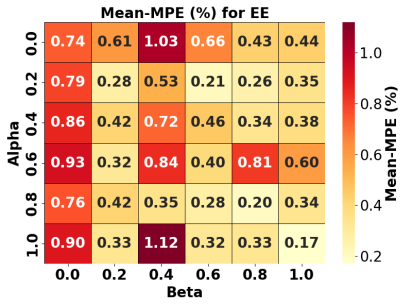

Fig. 5. Mean-APE heatmap for different α and β values, with fixed $\mu = 0.5$ for client EE.

Since statistical heterogeneity mainly impacts Max-APE, Fig. 6 presents the Max-APE results under different μ values, with both α and β fixed at 0.8 to emphasize strong personalization and knowledge transfer. Notably, each client achieves optimal performance at a distinct μ: 0.6 (AT), 0.3 (BE), 0.3 (BG), 0.6 (CZ), 0.7 (EE), and 0.3 (FI). This diversity highlights the inherent heterogeneity across clients and suggests that a universal μ setting is suboptimal. Instead, client-specific tuning enables a better balance between global guidance and local adaptation, enhancing robustness in statistically Non-IID federated environments.

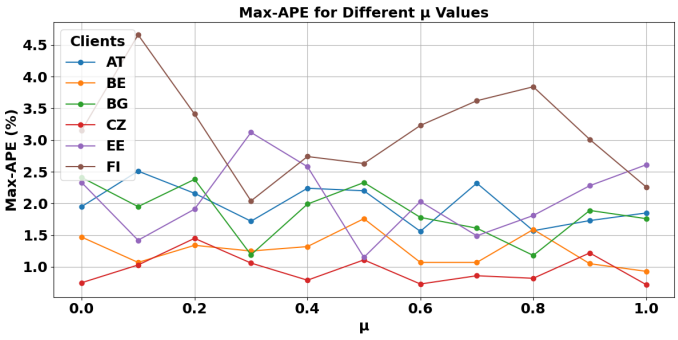

Fig. 6. Comparison of Max-APE across different μ values. Each client achieves the best performance at distinct μ.

D. Ablation Study

To validate the advantages of the proposed framework, we design four baseline frameworks based on existing literature:

- Local training: Each region trains its model independently without any knowledge sharing.
- Traditional FL: Standard federated learning is applied without personalization or knowledge distillation.
- FedProx: Clients participate in federated learning with an additional proximal-term regularization to the local objective, which encourages convergence towards a global objective while handling statistical heterogeneity and preserving some local model personalization [34].
- FedAMP: Each client aggregates other clients' models through attentive message passing, weighing more heavily those models that are similar to its own. This yields a personalized model for each client without relying on a single global model [35].

In all configurations, models use the same historical load data as input features. Differences lie in how knowledge is shared and how personalization is applied. This setup allows us to systematically evaluate the benefits of existing methods that address statistical heterogeneity and enable model personalization, highlighting the advantages of the proposed framework in these aspects. The parameters used in the experiments are summarized in Table II.

TABLE II
EXPERIMENTAL PARAMETERS.

Parameter	Value
Number of communication rounds	10
Local epochs per client	50
Knowledge distillation epochs	30
Number of clients	6
Aggregation method	FedAvg
Local model architecture	Two-layer GRU
Proxy model architecture	Two-layer GRU
Hidden layer units	256
Training-validation split	7:3
Local training learning rate	0.001
Knowledge distillation learning rate	0.001
Local Optimizer Adam's parameters	step_size = 10, gamma = 0.1
PGD Regularization parameters (μ)	[0.6, 0.3, 0.3, 0.6, 0.7, 0.3]
Local model input size (excluding load)	[5, 4, 3, 4, 4, 2]
Distillation parameters (α, β)	0.8, 0.8

The detailed performance comparison between the proposed framework and the baseline methods is summarized in Table III. It can be observed that local training, where each client learns independently without knowledge sharing, exhibits notable limitations, particularly in extreme load scenarios, as reflected by the high Max-APE values across clients. Standard FL (FedAvg) reduces the overall Mean-APE by leveraging cross-client information, demonstrating the benefit of collaborative learning. However, certain clients still experience elevated Max-APE, indicating that standard FL may be vulnerable to worst-case errors and less capable of capturing heterogeneous regional patterns. In contrast, the proposed framework, which incorporates proxy-based knowledge distillation, effectively mitigates these limitations. By enhancing generalization across diverse regional distributions and reducing overfitting to local data, it achieves substantial

TABLE III
NUMERICAL METRICS OF THE PROPOSED FRAMEWORK OVER FL.

Clients	Metrics	Local	FL	FedProx	FedAMP	Proposed
AT	Mean-APE(%)	0.32	0.12	0.13	0.12	0.09
	Max-APE(%)	6.92	7.51	7.59	6.48	1.83
BE	Mean-APE(%)	0.13	0.11	0.13	0.08	0.05
	Max-APE(%)	7.11	7.34	6.86	6.50	0.88
BG	Mean-APE(%)	0.26	0.23	0.24	0.28	0.11
	Max-APE(%)	10.94	12.30	10.98	15.76	1.21
CZ	Mean-APE(%)	0.23	0.13	0.14	0.13	0.09
	Max-APE(%)	5.18	7.00	9.10	6.06	0.63
EE	Mean-APE(%)	0.58	0.15	0.16	0.15	0.37
	Max-APE(%)	16.71	16.15	15.78	16.95	1.47
FI	Mean-APE(%)	0.16	0.13	0.12	0.17	0.13
	Max-APE(%)	5.49	6.79	6.52	7.24	3.35
Average	Mean-APE(%)	0.28	0.15	0.15	0.16	**0.14**
	Max-APE(%)	8.73	9.52	9.47	9.83	**1.56**

reductions in Max-APE while maintaining competitive Mean-APE. Consequently, even the regions with the most challenging load patterns benefit from improved robustness and more reliable prediction performance, resulting in a balanced and consistent forecasting model across all clients.

As personalized FL methods, FedProx slightly reduces Max-APE by mitigating the effects of non-IID distributions, but its reliance on aligned feature spaces and uniform model structures limits its ability to fully capture cross-regional diversity. FedAMP, by leveraging attention-based aggregation, reduces Mean-APE across most clients, yet Max-APE increases on some individual clients due to model personalization. Similar to FedProx, its effectiveness is still constrained by the need for aligned feature spaces and consistent model architectures. In contrast, the proposed framework achieves consistently lower Mean-APE and Max-APE across all clients, demonstrating superior accuracy and robustness while overcoming the limitations of feature and model heterogeneity that constrain traditional personalized FL methods.

To assess the effect of model architecture on the proposed framework, we compare four recurrent neural network variants: LSTM [37], SLSTM [38], Peephole-LSTM [39], and ChronoLSTM [40]. These models differ in gating mechanisms and computational properties. All models adopt a two-layer structure with 256 hidden units and are trained under identical settings. As shown in Table IV, which presents the performance on client AT, the GRU-based model achieves the best performance in both Mean-APE and Max-APE, confirming its effectiveness and efficiency for federated forecasting.

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT RNN VARIANTS ON CLIENT AT UNDER THE PROPOSED FRAMEWORK.

Model Architecture	Mean-APE (%)	Max-APE (%)
GRU	**0.09**	**1.83**
LSTM	0.23	3.83
SLSTM	0.42	4.09
Peephole-LSTM	0.25	2.97
ChronoLSTM	0.40	4.43

The stability benefits of PGD are evaluated by comparing its use versus Adam in updating the proxy model, with local models always trained using Adam and global aggregation unchanged. Fig. 7 shows the training and validation loss curves of client AT. Both Adam and PGD exhibit similar convergence speed; however, Adam suffers from larger oscillations during training. In contrast, PGD demonstrates smaller fluctuations and more stable convergence, indicating better robustness to Non-IID noise through regularization.

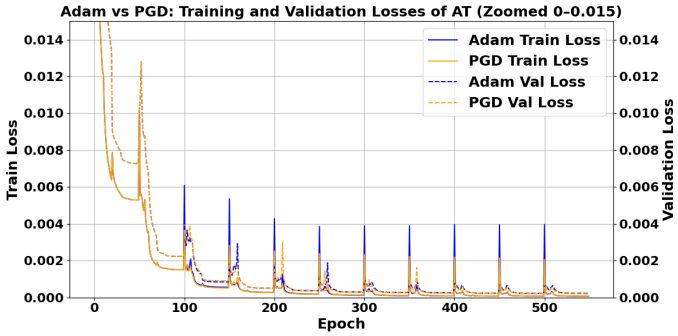

Fig. 7. Training and validation losses comparison between Adam and PGD for client AT. Both optimizers converge at similar speed, but PGD exhibits more stable convergence with smaller oscillations.

E. Model Resilience Evaluation

To further evaluate the resilience of the proposed framework against adversarial updates, we compare it with several representative robust aggregation mechanisms in federated learning, including Krum, Geometric Median (Median), and Trimmed-Mean. These robust aggregation methods are designed to mitigate the influence of Byzantine clients by either selecting updates close to the majority (Krum), computing the geometric median of all updates (Median), or trimming extreme values along each parameter dimension (Trimmed-Mean) [20], [41], [42].

The vulnerability of FL under adversarial conditions is evaluated using three representative types of Byzantine attacks,

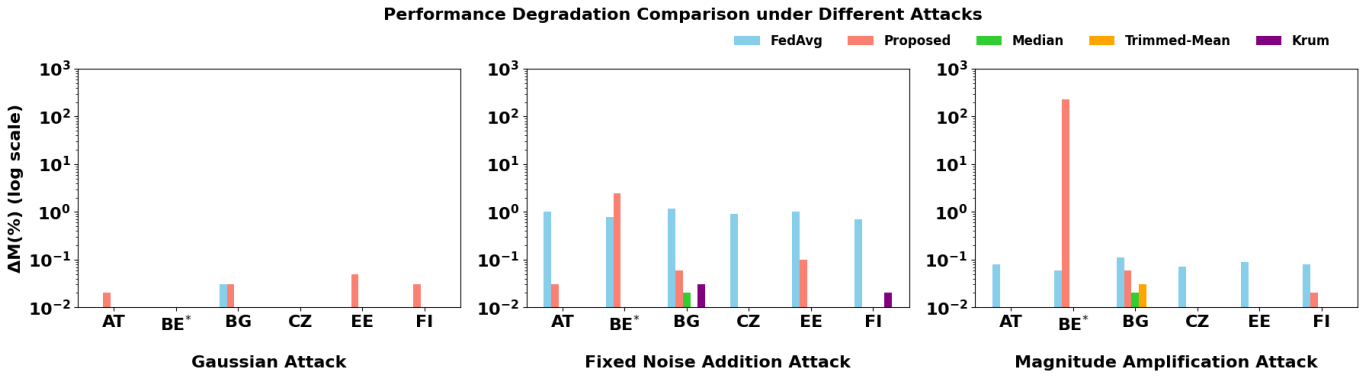

Fig. 8. Comparison of performance degradation under different attack types. Each attack (Gaussian, Fixed Noise Addition, and Magnitude Amplification) includes results from six clients.(* indicates the attacked client.)

as summarized in Table V: Gaussian attacks [43], Magnitude amplification attacks [44], and Fixed noise addition attacks. Beyond these basic attack types, three adversarial scenarios are designed to examine the robustness of the system under increasing levels of threat:

- Single-Attacker: A single client (BE*) becomes malicious in the fifth communication round and submits manipulated gradients following the attack described in Table V.
- Multi-Attackers: Two clients (BE* and BG*) simultaneously launch malicious updates in the fifth round, both employing Magnitude Amplification perturbations.
- Persistent-Attack: The client BE* continuously performs adversarial updates for five consecutive communication rounds starting from the fifth round, employing Magnitude Amplification perturbations.

TABLE V
BYZANTINE ATTACK TYPES AND PARAMETERS.

Attack Type	Perturbation Formula	Parameter
Gaussian	$\mathbf{w}_i^{\text{attacked}} = \mathbf{w}_i + \mathcal{N}(0, \sigma^2)$	$\sigma^2 = 0.01$
Fixed Noise Addition	$\mathbf{w}_i^{\text{attacked}} = \mathbf{w}_i + c$	$c = 0.5$
Magnitude Amplification	$\mathbf{w}_i^{\text{attacked}} = \mathbf{w}_i \times \lambda$	$\lambda = 100$

Validation performance is measured in terms of Mean-APE, where lower Mean-APE values and higher accuracy indicate better prediction performance. Let $\mathcal{M}_{\text{original}}$ denote the model performance (e.g., validation accuracy) before the attack and $\mathcal{M}_{\text{attacked}}$ denote the performance after introducing the perturbed model from Client BE. The relative performance degradation is computed as:

$$\Delta\mathcal{M} = \frac{\mathcal{M}_{\text{original}} - \mathcal{M}_{\text{attacked}}}{\mathcal{M}_{\text{original}}} \times 100\%. \tag{26}$$

The results in Fig. 8 first examine single-attacker across multiple attack types, comparing FedAvg, robust aggregation methods (Krum, Median, Trimmed-Mean), and the proposed framework. Under Gaussian attacks, all methods showed minimal degradation. Under more disruptive attacks such as Magnitude Amplification, FedAvg exhibited a clear performance collapse, confirming its weakness against strong adversarial manipulation. Robust aggregation methods, in contrast, remained highly effective under single-client attacks, successfully filtering malicious updates and maintaining stable performance.

Based on these results, FedAvg is excluded from further adversarial evaluations, as it consistently underperforms and provides no meaningful robustness margin. We therefore select Magnitude Amplification, the attack type with the most severe impact, as the representative adversarial setting for subsequent experiments. Under this setting, we focus on comparing robust aggregation methods with the proposed framework.

As shown in Fig. 9, for persistent attacks—where a single client repeatedly submits adversarial updates across rounds—existing robust aggregation methods remain resilient and prevent significant model degradation. However, under multi-attacker scenarios, their robustness diverges: Trimmed-Mean in particular exhibits clear performance degradation, as simultaneous extreme updates from multiple malicious clients reduce the effectiveness of its statistical filtering. In contrast, methods such as Median and Krum retain relatively stable behavior.

In comparison, the proposed framework consistently maintains robustness across all adversarial settings, including single-attacker, persistent, and multi-attacker cases. Because global knowledge is transferred through distillation rather than direct parameter aggregation, adversarial updates from local personalized models are naturally isolated and cannot corrupt the global representation. Furthermore, the proposed framework makes abnormal client behavior more distinguishable, enabling easier identification of compromised clients and supporting targeted defense in subsequent communication rounds—capabilities that traditional robust aggregation algorithms typically lack.

F. Computation-Aware Federated Training via Early Exit

To address system heterogeneity and improve training efficiency, a dynamic exit mechanism is introduced, enabling clients to stop local training once their validation performance reaches a convergence threshold. For each client i, the exit condition is defined as:

$$\mathcal{L}_t^{(i)} \leq \delta_i, \tag{27}$$

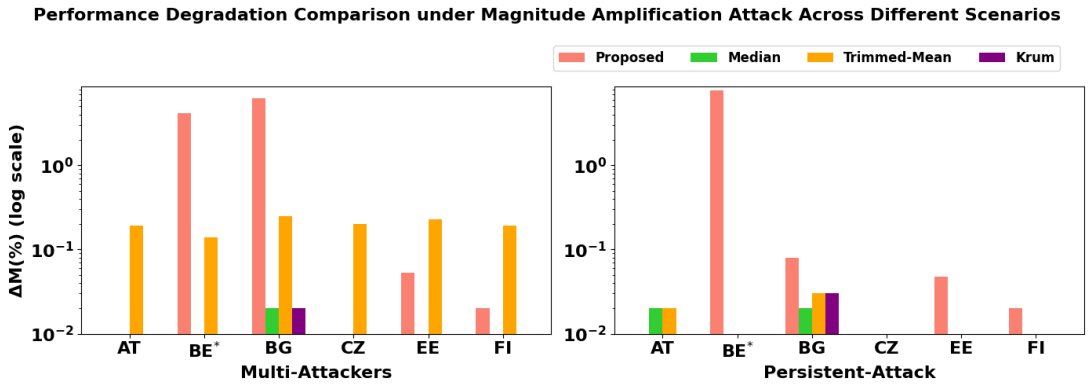

Fig. 9. Comparison of performance degradation under Magnitude Amplification Attack across different scenarios (Multi-Attackers and Persistent Attack) for four aggregation methods. (* indicates the attacked client.)

where $\mathcal{L}_t^{(i)}$ denotes the validation loss at round t, and δ_i is a client-specific threshold determined from the validation loss in the baseline training without dynamic exit. Once this criterion is satisfied, the client exits subsequent federated rounds, conserving computational resources while maintaining convergence quality.

TABLE VI
COMPARISON OF TRAINING EPOCHS WITH AND WITHOUT DYNAMIC EXIT.

Client	Without Exit	With Exit	Reduction (%)
AT	800	800	0.00%
BE	800	290	63.75%
BG	800	530	33.75%
CZ	800	770	3.75%
DE	800	800	0.00%
FI	800	370	53.75%
Average	-	-	**25.83%**

As shown in Table VI, the total number of training epochs is reduced by approximately 25% on average, demonstrating substantial computational savings. This computation-aware strategy effectively balances convergence and efficiency, enabling scalable and resource-adaptive federated learning under heterogeneous system conditions, and thus facilitates practical deployment in real-world applications.

IV. CONCLUSION

This paper presents a novel federated learning framework for cross-regional load forecasting, effectively addressing statistical, model, and system heterogeneity in traditional FL. The framework incorporates proxy models with feature-masking to align feature dimensions across heterogeneous clients, enabling secure and efficient knowledge transfer through knowledge distillation. Perturbed Gradient Descent (PGD) is employed to stabilize training across diverse regional models, while the dynamic exit mechanism allows clients to terminate training early based on computational capacity, supporting broader participation without significantly compromising accuracy. Experimental evaluations on European energy datasets demonstrate that, compared with representative personalized FL baselines, the proposed framework achieves consistently lower MAPE and significantly reduced extreme prediction errors, highlighting its effectiveness and robustness under cross-regional heterogeneity. To further assess the effectiveness of enhanced robustness of the proposed framework, three Byzantine attacks and three attack scenarios are simulated. Compared with traditional FL and robust aggregation methods, the proposed framework effectively isolates compromised clients and maintains model robustness. The proposed framework offers a viable pathway toward reliable and practical FL deployment in complex, heterogeneous energy systems.

REFERENCES

[1] Q. Hassan, S. Algburi, A. Z. Sameen, H. M. Salman, and M. Jaszczur, "A review of hybrid renewable energy systems: Solar and wind-powered solutions: Challenges, opportunities, and policy implications," *Results in Engineering*, vol. 20, p. 101621, 2023. Available: https://doi.org/10.1016/j.rineng.2023.101621.

[2] I. K. Nti, M. Teimeh, O. Nyarko-Boateng, and A. A. Belford, "Electricity load forecasting: a systematic review," *Journal of Electrical Systems and Information Technology*, vol. 7, no. 13, 2020. pp. 1-19, Available: https://doi.org/10.1186/s43067-020-00021-8.

[3] M. Gong, Y. Zhao, J. Sun, C. Han, G. Sun, and B. Yan, "Load forecasting of district heating system based on Informer," *Energy*, vol. 253, p. 124179, 2022. Available: https://doi.org/10.1016/j.energy.2022.124179.

[4] H. Hou, C. Liu, Q. Wang, X. Wu, J. Tang, Y. Shi, and C. Xie, "Review of load forecasting based on artificial intelligence methodologies, models, and challenges," *Electric Power Systems Research*, vol. 210, p. 108067, 2022. Available: https://doi.org/10.1016/j.epsr.2022.108067.

[5] A. Alhendi, A. S. Al-Sumaiti, M. Marzband, R. Kumar, and A. A. Zaki Diab, "Short-term load and price forecasting using artificial neural network with enhanced Markov chain for ISO New England," *Energy Reports*, vol. 9, pp. 4799-4815, 2023. Available: https://doi.org/10.1016/j.egyr.2023.03.116.

[6] K. A. Fakhryza, E. N. Budisusila, and A. A. Nugroho, "Application of artificial neural network for peak load forecasting in 150 kV Semarang power system," *Journal of Electrical Systems and Inf Technol*, vol. 11, p. 40, 2024. Available: https://doi.org/10.1186/s43067-024-00165-x.

[7] J. Li, Y. Lei, and S. Yang, "Mid-long term load forecasting model based on support vector machine optimized by improved sparrow search algorithm," *Energy Reports*, vol. 8, Suppl. 5, pp. 491-497, 2022. Available: https://doi.org/10.1016/j.egyr.2022.02.188.

[8] G. Dudek, "A comprehensive study of random forest for short-term load forecasting," *Energies*, vol. 15, no. 20, p. 7547, 2022. Available: https://doi.org/10.3390/en15207547.

[9] J. Wang, Q. Xing, B. Zeng, and W. Zhao, "An ensemble forecasting system for short-term power load based on multi-objective optimizer and fuzzy granulation," *Applied Energy*, vol. 327, p. 120042, 2022. Available: https://doi.org/10.1016/j.apenergy.2022.120042.

[10] J. Moon, S. Park, E. Hwang, and S. Rho, "A Hybrid Tree-Based Ensemble Learning Model for Day-Ahead Peak Load Forecasting," *2022 15th International Conference on Human System Interaction (HSI)*, Melbourne, Australia, 2022, pp. 1-6. doi: 10.1109/HSI55341.2022.9869440.

[11] S. H. Rafi, N. Al-Masood, S. R. Deeba, and E. Hossain, "A short-term load forecasting method using integrated CNN and LSTM network," *IEEE Access*, vol. 9, pp. 32436-32448, 2021. doi: 10.1109/ACCESS.2021.3060654.

[12] Y. Jiang, Q. Huang, K. Zhang, Z. Lin, T. Zhang, X. Hu, S. Liu, C. Jiang, L. Yang, and Z. Lin, "Medium-long term load forecasting method considering industry correlation for power management," *Energy Reports*, vol. 7, Supplement 7, pp. 1231-1238, 2021. Available: https://doi.org/10.1016/j.egyr.2021.09.140.

[13] B. Li, Y. Liao, S. Liu, C. Liu, and Z. Wu, "Research on short-term load forecasting of LSTM regional power grid based on multi-source parameter coupling," *Energies*, vol. 18, no. 3, p. 516, 2025. doi: 10.3390/en18030516.

[14] C. S. Lai et al., "Multi-View Neural Network Ensemble for Short and Mid-Term Load Forecasting," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 2992-3003, July 2021. doi: 10.1109/TPWRS.2020.3042389.

[15] X. Kong, C. Li, F. Zheng, and C. Wang, "Improved Deep Belief Network for Short-Term Load Forecasting Considering Demand-Side Management," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1531-1538, March 2020. doi: 10.1109/TPWRS.2019.2943972.

[16] H. Liu, X. Zhang, H. Sun, and M. Shahidehpour, "Boosted Multi-Task Learning for Inter-District Collaborative Load Forecasting," *IEEE Transactions on Smart Grid*, vol. 15, no. 1, pp. 973-986, Jan. 2024, doi: 10.1109/TSG.2023.3266342.

[17] Y. Li, J. Li, and Y. Wang, "Privacy-Preserving Spatiotemporal Scenario Generation of Renewable Energies: A Federated Deep Generative Learning Approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2310-2320, April 2022, doi: 10.1109/TII.2021.3098259.

[18] Y. Wang, I. L. Bennani, X. Liu, M. Sun, and Y. Zhou, "Electricity Consumer Characteristics Identification: A FL Approach," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3637-3647, July 2021, doi: 10.1109/TSG.2021.3066577.

[19] J. Zhang, X. He, Y. Huang, and Q. Ling, "Byzantine-Robust and Communication-Efficient Personalized FL," *IEEE Transactions on Signal Processing*, vol. 73, pp. 26-39, 2025, doi: 10.1109/TSP.2024.3514802.

[20] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 119-129, 2017. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf

[21] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. USENIX Security Symposium*, 2020, pp. 1605–1622. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/fang

[22] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2020, pp. 10467–10477. [Online]. Available: https://proceedings.mlr.press/v119/xie20a.html

[23] Q. Wu, K. He, and X. Chen, "Personalized FL for Intelligent IoT Applications: A Cloud-Edge Based Framework," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 35-44, 2020, doi: 10.1109/OJCS.2020.2993259.

[24] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of Personalization Techniques for FL," *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, London, UK, 2020, pp. 794-797, doi: 10.1109/WorldS450073.2020.9210355.

[25] Y. Qin and M. Kondo, "MLMG: Multi-Local and Multi-Global Model Aggregation for FL," *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Kassel, Germany, 2021, pp. 565-571, doi: 10.1109/PerComWorkshops51409.2021.9431011.

[26] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging FL by local adaptation," *arXiv*, 2022. Available: https://arxiv.org/abs/2002.04758.

[27] B. Wang, Y. Zhang, T. Liu, H. Chen, and J. Lin, "Multi-center FL with model decoupling," *2022 3rd International Conference on Computer Science and Management Technology (ICCSMT)*, Shanghai, China, 2022, pp. 450–455. doi: 10.1109/ICCSMT58129.2022.00101.

[28] A. Afzali and P. Shamsinejadbabaki, "PHiFL-TL: Personalized hierarchical FL using transfer learning," *Future Generation Computer Systems*, vol. 166, p. 107672, 2025. Available: https://doi.org/10.1016/j.future.2024.107672.

[29] T. Tuor, S. Wang, B. J. Ko, C. Liu, and K. K. Leung, "Overcoming Noisy and Irrelevant Data in FL," *2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, 2021, pp. 5020-5027, doi: 10.1109/ICPR48806.2021.9412599.

[30] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv*, 2015. Available: https://arxiv.org/abs/1503.02531.

[31] D. Li and J. Wang, "FedMD: Heterogeneous FL via model distillation," *arXiv*, 2019. Available: https://arxiv.org/abs/1910.03581.

[32] Alessio Mora, Irene Tenison, Paolo Bellavista, and Irina Rish, "Knowledge distillation in FL: a practical guide," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI '24)*, Article 905, pp. 8188–8196, 2024. Available: https://doi.org/10.24963/ijcai.2024/905.

[33] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST)*, Doha, Qatar, 2014, pp. 103–111.

[34] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems 2 (MLSys 2020)*, 2020. doi: https://proceedings.mlsys.org/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html.

[35] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, Y. Zhang, "Personalized Cross-Silo Federated Learning on Non-IID Data," in *Proc. 35th AAAI Conf. on Artificial Intelligence*, vol.35, no.9, pp.7865–7873, 2021. doi:10.1609/aaai.v35i9.16960

[36] F. Wiese, I. Schlecht, W.-D. Bunke, C. Gerbaulet, L. Hirth, M. Jahn, et al., "Open power system data—frictionless data for electricity system modelling," *Applied Energy*, vol. 236(C), pp. 401–409, 2019. doi: 10.1016/j.apenergy.2018.11.097.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[38] F. M. Salem, "Slim LSTMs: Parameter-Reductions within Gating Signals," in *Proc. IEEE 62nd Int. Midwest Symp. on Circuits and Systems (MWSCAS)*, 2019, pp. 235–238.

[39] F. A. Gers, N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2002.

[40] C. Tallec and Y. Ollivier, "Can recurrent neural networks warp time?," in *Proc. Int. Conf. on Learning Representations (ICLR)*, Vancouver, Canada, 2018. Available: https://openreview.net/forum?id=SJcKhk-Ab.

[41] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," *Proc. of ICML*, vol. 80, pp. 5650–5659, 2018. doi: https://arxiv.org/abs/1705.02786.

[42] X. Chen, Q. Ling, and W. Yin, "Robust aggregation for federated learning under attack," *arXiv preprint arXiv:1703.02757*, 2017. doi: https://arxiv.org/abs/1703.02757.

[43] L. Shi, S. Zhang, and Y. Sun, "Data poisoning attacks on FL by using adversarial samples," *2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, Shijiazhuang, China, 2022, pp. 158-162, doi: 10.1109/ICCEAI55464.2022.00041.

[44] X. Cao and N. Z. Gong, "MPAF: Model Poisoning Attacks to FL based on Fake Clients," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 3395–3403. Available: https://api.semanticscholar.org/CorpusID:247476326.

Zhifeng Zuo received the B.S. degree in electrical engineering and automation in 2023 from the School of Mechano-Electronic Engineering, Xidian University, Xi'an, China, and is currently working toward the Ph.D. degree with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China. His research interests include federated learning, load forecasting, data-driven power system analysis, power system resilience, and cyber–physical security of power systems.

Hongxing Ye (Senior Member, IEEE) is currently a Professor with Xi'an Jiaotong University (XJTU). He received the B.S. degree in Information Engineering and the M.S. degree in Systems Engineering from Xi'an Jiaotong University, Xi'an, China, in 2007 and 2011, respectively, and the Ph.D. degree in Electrical Engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2016. He was a visiting researcher at Argonne National Lab, Chicago. He was a Tenure-Track Assistant Professor with Cleveland State University, Cleveland, OH, USA, before joining XJTU. His research interests include renewable integration, power system planning and operation, and machine learning in cyber-physical energy system. Dr. Ye received the Sigma Xi Research Excellence Award. His research has been supported by NSFC, NSTMP, NSF, and Industry.

Jie Li (Senior Member, IEEE) is currently an Associate Professor in the Electrical and Computer Engineering Department at Rowan University. Before that, she was an Assistant Professor with the Electrical and Computer Engineering Department, Clarkson University. She received her Ph.D. degree in Electrical Engineering from Illinois Institute of Technology in 2012, M.S. and B.S. degrees in Systems Engineering and Electrical and Computer Engineering from Xi'an Jiaotong University, China, in 2006 and 2003, respectively. Dr. Li has over twenty years of experience in power and energy system planning, operation, control, security, reliability, and resilience, with a particular interest in the modeling and optimization of large-scale electricity transmission and distribution systems with a deeper penetration of distributed energy resources and large loads. She has extensive industry experience, including those with GE Energy Management and IBM Global Research.

Yinyin Ge is currently an Associate Professor in the School of Cyber Science and Engineering, Xi'an Jiaotong University (XJTU). She received the B.S. degree in Automation and the M.S. degree in Systems Engineering from Jiaotong University, Xi'an, China, in 2008 and 2011, respectively, and the Ph.D. degree in Electrical Engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2016. She served as a Research Associate at Case Western Reserve University, Cleveland, OH, USA, before joining XJTU. Her research interests focus on cyber-physical energy systems (CPES), including optimization and privacy preservation of power and energy systems, cyber-physical security, and integration of smart grid and intelligent transportation systems.