Computing and Informatics, Vol. 44, 2025, 580–611, doi: 10.31577/cai_2025_3_580

PERSONALIZED FEDERATED LEARNING BASED ON HYPERNETWORKS AND ATTENTION MECHANISM ENSEMBLES FOR INTERNET OF THINGS

Lu Liu, Huiqi Zhao, Fang Fan

College of Intelligent Equipment
Shandong University of Science and Technology
271000 Taian, China
e-mail: {liulu, zhaohq, fangfan}@sdust.edu.cn

Sibo Qiao*

School of Software
Tiangong University
300387 Tianjin, China
e-mail: siboqiao@126.com

Zhihan Lyu

Department of Game Design, Faculty of Arts
Uppsala University
75236 Uppsala, Sweden
e-mail: lvzhihan@gmail.com

Abstract. As the demand for data privacy protection continues to grow and the concept of collaborative modeling gains traction, federated learning has emerged as a pivotal distributed learning paradigm in the Internet of Things (IoT) domain. However, the client data held by different institutions often varies significantly in sources and characteristics, which can hinder the efficiency of federated learning model training and increase the risk of personal privacy breaches. To

* Corresponding author

address the challenges of model accuracy degradation and privacy exposure when federated learning is applied to multi-source heterogeneous data, we propose a personalized federated learning strategy that integrates hypernetworks with attention mechanisms. This strategy involves transforming labeled data at the source to protect personal privacy while employing hypernetworks and Transformer-based mechanisms to focus on the personalized information of clients from various institutions. Our proposed approach supports handling heterogeneous data, thereby better meeting the personalized needs of different institutions. Experimental results demonstrate that this framework not only effectively safeguards data privacy but also significantly enhances the performance and generalization capability of federated learning on heterogeneous data. This research offers a novel perspective for developing more adaptable personalized federated learning models, facilitating cross-institutional collaborative research, and providing an innovative model training solution for various IoT devices, balancing the dual requirements of data privacy protection and multi-institutional data sharing.

Keywords: Data privacy, data protection, hypernetwork, personalized federated learning, transformer

Mathematics Subject Classification 2010: 68-T05

1 INTRODUCTION

Federated learning has emerged in distributed learning with the increasing awareness of users to protect their personal privacy. As an emerging distributed machine learning paradigm, the core advantage of federated learning lies in its ability to utilize data distributed across different clients for learning while protecting data privacy, which provides a new approach for systems to protect clients' privacy and data security. However, existing federated learning approaches are often limited by the lack of data quality and quantity when dealing with huge systems, especially the performance degradation problem when dealing with non-independent and identically distributed data, and traditional federated learning models face some challenges when dealing with personalized information. In order to overcome the problem that global models are not applicable to all clients due to data heterogeneity, personalized federated learning has emerged.

In federated learning environments, when clients have non-independent identically distributed (non-IID) data, traditional federated learning methods lead to high communication overhead and low training efficiency [1]. Furthermore, the vastness, complexity, and variability of heterogeneous data present significant challenges to machine learning, particularly when such data includes sensitive personal information [2]. Existing distributed semi-supervised learning (DSSL) algorithms have challenges in dealing with data uncertainty and computational communication overhead [3]. The problem of non-independent homogeneous distribution of data

is a thorny problem that needs to be solved for federated learning applied in the medical field [4], now many scholars have noticed this aspect and there have been many approaches to solve the problem of different distribution of client data. In recent times, numerous researchers have focused on this issue, and various methods have been developed to address the challenge of differing client data distributions. For example, with the development of Transformer, many scholars have noticed that Transformer can use the attention mechanism to learn the global interactions of the inputs during the training of the client model, which has better performance on highly heterogeneous data features [5].

Many studies have also shown that privacy issues in federated learning also face serious threats, where an attacker can access the client's training data based on the stolen model parameters [6]. There are also multiple ways to infer users' private information in federated learning environments, including data leakage vulnerabilities [7] and membership inference [8]. Attackers can use the benign data obtained from inference to train the generation of adversarial network models [9], which generate new malicious data to escape security identification, and then go on to attack operations.

Recently, privacy-preserving methods for data based on personalized federated learning have been evolving, where data containing clients' private data are retained in the local client for model training, and replaced by the exchange of model parameters for interactions between clients. Through the personalized federated learning framework with privacy preserving function, on the one hand, it can realize the function of protecting privacy data, and on the other hand, it can also solve the problem of degradation of the accuracy of federated learning model due to the different characteristics of data from different system devices, as shown in Figure 1.

To address the above problem, we propose a personalized federated learning framework that combines hypernetworks and Transformer models. Our approach improves the performance and adaptability of federated learning while preserving data privacy. Our main contributions in this paper are as follows:

1. We propose a privacy-preserving mechanism based on data and label transformation, where local user data and labels are transformed before model training in the local client, which achieves privacy preservation from the data source while ensuring the availability of local data;
2. We propose a personalized federated learning framework (pFedHT) based on hypernetworks and attention mechanisms, which introduces a hypernetwork architecture on the server side to dynamically generate personalized weights from the attention layer for each client. By using the hypernetwork to generate unique embedding vectors customized for each client's attention mechanism, the performance degradation problem in facing non-independent and identically distributed data models is solved by capturing the personalized data features among clients more effectively;
3. In this paper, we experimentally analyze the impact of different levels of data heterogeneity on the performance of the algorithms and validate the effectiveness

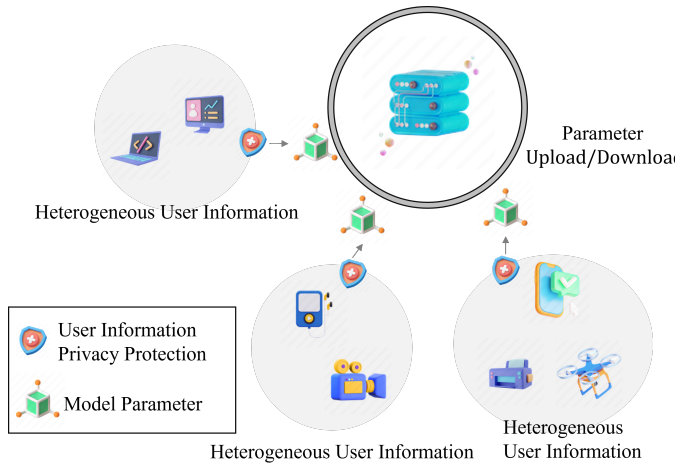


Figure 1. Schematic diagram of personalized federated learning with privacy protection mechanism. Data from different sources are processed by the privacy-preserving mechanism and then the personalized federated learning framework achieves the protection of user privacy data and solves the problem of heterogeneity of data.

of this paper's framework on a variety of non-independent and identically distributed experimental datasets. Comparison experiments with a large number of mainstream federated learning algorithms show that the framework proposed in this paper has better performance in the face of data heterogeneity while fully protecting user data privacy.

Next, the article will be presented according to the following sections, the second section is part of the previous studies on federated learning, hypernetworks, and Transformer; the third section presents an overview of the framework proposed in this paper, along with the details of its implementation process. The fourth section covers the dataset setup and provides an analysis of the experimental results. The final section offers a summary of our findings and discusses future research directions. Additionally, we will evaluate our model using several public datasets and perform a comparison with existing federated learning techniques to demonstrate the effectiveness and advantages of our approach.

2 RELATED WORK

2.1 Federated Learning

In the face of growing data and increasing user demands for privacy protection, federated learning, a new distributed paradigm, is gradually becoming the method of choice. And with the era of increasing data categories, the problem of data hetero-

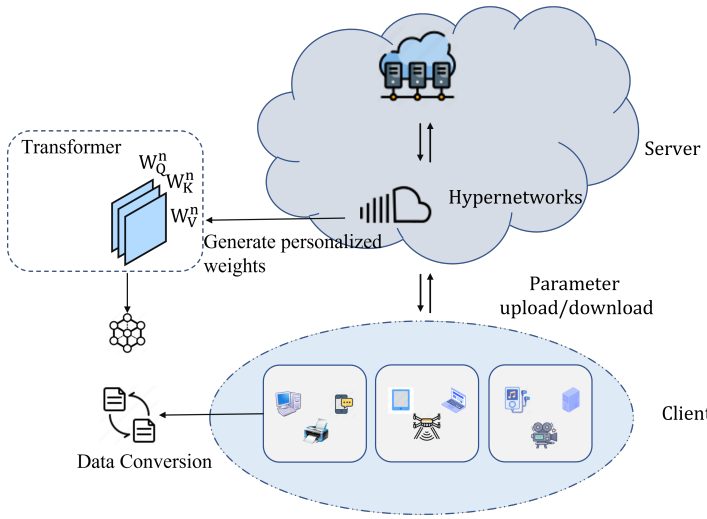


Figure 2. Network architecture diagram. Clients are trained with personalized federated learning models using personalized weights generated by the Hypernetwork after a privacy-preserving mechanism based on data and label transformation.

geneity is gradually being realized. In many recent researches, many personalized federated learning parties have been proposed to solve the heterogeneity problem caused by data heterogeneity. There are mainly solutions for both data and model perspectives.

2.1.1 Data-Based Approaches

In their article, Zhao et al. [10] examine the effect of Non-IID data on Federated Learning and suggest a strategy to enhance training by creating a globally shared data subset. This approach reduces communication costs by utilizing local data for training, yet it still encounters challenges such as the potential decline in model accuracy due to Non-IID data and the need to address the issue of weight divergence. In article [11], Wu et al. proposed a cloud edge-based personalized federated learning framework for home health monitoring and used the Generative Convolutional Auto Encoder (GCAE) technique to generate category-balanced datasets to address the imbalanced and non-IID distributions of user health monitoring data. Most of the above approaches are implemented by means of data augmentation, but in federated learning scenarios, data augmentation usually requires some form of data sharing or construction of proxy datasets, which makes the research very challenging.

The other approach is to select the clients and thus achieve an even distribution of data. Li et al. [12] proposed an adaptive FL framework FedSAE in their paper, which automatically adapts the training task based on the device's history of training tasks and actively selects participants to mitigate performance degradation. Zheng

et al. [13] in their paper use a tier-based FL system to solve the personalization problem by grouping clients into different tiers based on the training performance and selecting clients from the same tier in each round of training. This approach significantly improves the training performance by reducing the "procrastinator" problem due to resource and data heterogeneity and maintains the test accuracy comparable to conventional FL through an adaptive tier selection strategy, but it does not focus on dynamic tier management and client selection. Both of these approaches depend on the client's historical training data to forecast future training behavior.

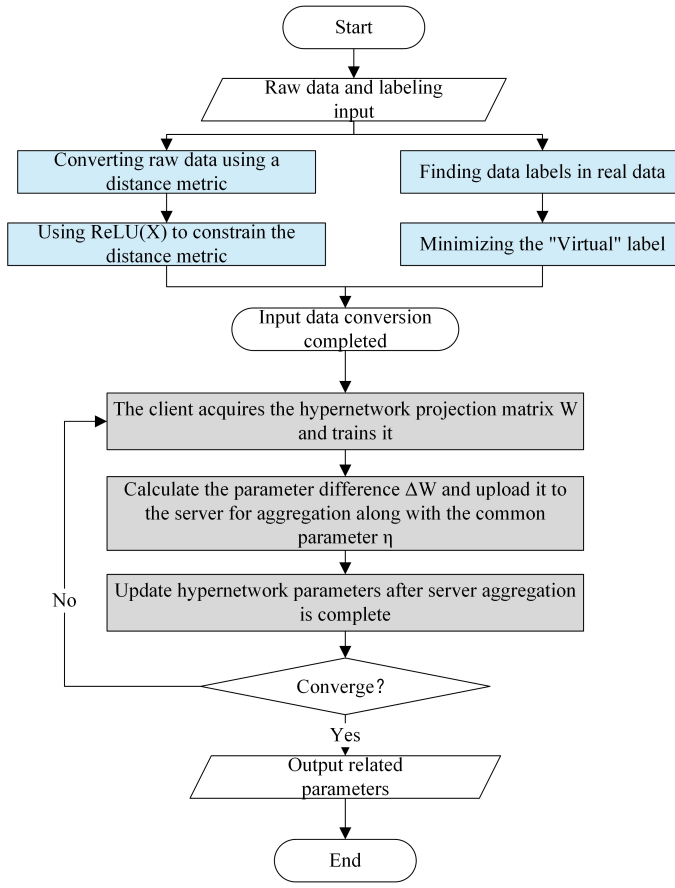


Figure 3. Framework flowchart. Blue is the client's privacy protection process, gray is the acquisition of personalized parameters under the Hypernetwork.

2.1.2 Model-Based Approaches

The main approach for the model side is to restrict the architecture of the models to ensure that the server is aligned when aggregating the models. The article [14] proposes an algorithm for FedMD that combines the ideas of model distillation and transfer learning to allow different models to exchange knowledge by sharing output category scores without the need to share data or model architectures and utilizes a large public dataset to train models in order to address the problem of small private dataset sizes. In the article, Arivazhagan et al. [15] proposed a new federated learning framework named FedPer. The framework addresses the statistical heterogeneity of data by dividing the deep neural network model into a shared base layer and a personalized layer. The base layer is globally updated through federated averaging, while the personalization layer is trained locally. McMahan et al. proposed the Federated Averaging Algorithm (FedAvg) [16] in 2016, which is a federated learning method based on iterative model averaging to reduce communication costs by computing updates locally at the client and then aggregating them by the server. However, the algorithm needs to fully utilize the local data of each client to ensure the quality of the global model. Li et al. [17] in their paper better dealt with system heterogeneity by introducing a proximal term to the FedAvg method that allows each device to perform a different amount of local work depending on its system resources and named the method FedProx. The method provides theoretical convergence guarantees and has been shown in practice to have more stable convergence behavior than FedAvg when dealing with system heterogeneity. Chen et al. [18] proposed a federated learning framework called FED-ROD, which explicitly decouples the model's dual responsibilities by using two prediction tasks: one for the generic prediction task and the other for the personalized prediction task, and which is able to maintain the performance of the global model while significantly improving the performance of the personalized model, and achieves fast adaptation to new clients through the use of hypernetworks, but requires the user to specify complex training strategies.

2.2 Federated Learning Privacy Protection Study

Federated learning was proposed with the intention of solving the privacy protection problem of users, but recent studies have shown that updating the model with some of the user data embedded in the model can also lead to privacy leakage problems, and the sharing of the model and gradient during the training process can also expose the federated learning model to inference attacks, such as attribute inference attacks [19] and model inversion attacks [20]. Sun et al. [21] consider a scenario where the federated learning server is malicious, aiming to reconstruct the client's private data from the device's model parameters. The experiments focus on the observation that class data representations of each device's data are embedded in shared local model updates, and such data representations can be inferred to perform model inversion attacks. The authors provide an analysis to reveal how

data representations are embedded in model updates and propose an algorithm to infer class data representations to perform model inversion attacks. However, experimental studies have shown that the correlation between data representations inferred using the algorithm and real data representations during local training is as high as 0.99, which clearly poses a serious privacy concern for systems under the federated learning framework.

Geiping et al. [20] assume that the attacker is fully aware of the model architecture of federated learning and then uses an exact extraction attack to perfectly reconstruct individual training samples, but this approach is not common in concrete life scenarios. They propose an attack in their paper to reconstruct the user's data by utilizing the information in the model gradient. The authors show that high resolution input images can be recovered from gradient information even in deep and non-smooth network architectures. A new data reconstruction attack is demonstrated in article [22] that allows an active and dishonest server-side to efficiently extract user data from received gradients. The attacker amplifies the data leakage in the model gradient by introducing "trap weights" in the shared model weights sent to the user without significantly changing the model performance.

Nowadays, the main methods to defend against attacks in federated learning are using differential privacy and gradient perturbation. Differential privacy protects data privacy by introducing controlled noise or randomness in the data processing. The introduction of noise makes the query results ambiguous and uncertain, thus preventing the attacker from accurately reconstructing the original data through multiple query results. The gradient perturbation is also a differential privacy technique, which will mainly modify the gradient before updating the gradient to the server to protect the gradient information during the training of the model, so as to defend against attackers, and thus ensure that the personal information in the system will not be disclosed.

2.3 Hypernetwork

Hypernetworks, introduced by Ha et al. in 2016 [23], are a deep learning architecture consisting of two interconnected networks: a primary network and a secondary hypernetwork. The main function of the hypernetwork is to generate the weights of the main network, and this design allows the model to dynamically adapt its behavior to different tasks or data. Its structure is also well suited to be configured in a federated learning framework, especially when dealing with tasks that require a large amount of personalization, where the distribution of data may be different across participants, the hypernetwork is able to generate model weights for each participant, which in turn helps the global model to better adapt to this data heterogeneity. Shamsian et al. [24], on the other hand, proposed in their paper a personalization using hypernetwork method named pFedHN. The method uses a hypernetwork to generate a personalized model for each client, and then the hypernetwork outputs model weights for each client through an embedding vector as input. Ma et al. [25] proposed a new federated learning framework pFedLA in 2022, which identifies the

contributing factors of each layer among different clients by introducing a specialized hypernetwork for each client at the server side and updating the aggregated weights of the layers. This leads to more accurate model personalization.

2.4 Transformer

Transformer is a deep learning model based on the self-attention mechanism, while the attention mechanism can directly establish dependencies between arbitrary positions within a sequence, helping the model to better understand the contextual relationships in the sequence, and since its operation does not depend on the positional information in the sequence data, Transformer can process the data in parallel, which greatly improves the training efficiency. In the scenario of data heterogeneity, the data distribution on each device participating in FL may be different, which leads to the model facing convergence difficulties and performance degradation during the training process. In the article [5], the authors employ the Transformer architecture to address the issue of catastrophic forgetting in Federated Learning on heterogeneous data. They leverage the Transformer's robustness to data distribution shifts to mitigate this problem. The article also found that Transformer is actually more suitable for data heterogeneous scenarios than CNN. The FedPerfix algorithm [26] utilizes plugins to deliver information from the aggregation model to the local client for personalization, but the introduction of plugins also brings new requirements for the users.

3 OUR PROPOSED FRAMEWORK

3.1 Framework Design

In this section, we present the design of the pFedHT framework, as shown in the network architecture diagram in Figure 2. In order to solve the problem of degradation of model accuracy in federated learning in the face of heterogeneous data, we design a personalized federated learning framework based on the combination of hypernetwork and attention mechanism. Personalized federated learning, by design, allows clients to have a model tailored to their unique data characteristics, which is crucial for handling the diversity in data sources in IoT applications. This approach also prevents the "one-size-fits-all" issue of traditional federated models, where a single global model may perform poorly across clients due to data variation. And hypernetworks allow the model to adapt dynamically to different clients by generating personalized model weights, improving the model's ability to generalize across varied datasets. The attention mechanism, specifically Transformer-based mechanisms, is employed to focus on the most relevant and personalized aspects of the client's data, thus reducing noise and ensuring privacy. Together, these mechanisms enable better performance and privacy preservation compared to standard federated learning models. Therefore we chose to use that method to solve the problem.

The framework has three main parts: the client, the server, and the hypernetwork set on the server side. The client mainly includes a large number of IoT devices, which collect a large amount of data from clients and then train the model based on the model downloaded from the server. The server mainly initializes the global model and then receives the trained model parameters from different clients and aggregates them to get a new global model. The hypernetwork, on the other hand, dynamically generates the personalized weights of the self-attention layer for each client, so that the model can effectively capture the personalized data characteristics of the clients in each system. The flowchart is shown in Figure 3, which mainly includes the two processes of client privacy protection and acquisition of personalized parameters under the hypernetwork.

3.2 Privacy Protection Mechanisms Based on Data and Label Transformation

Algorithm 1 Data Conversion Pseudocode

Input: Raw data and labels: (X, x), Weights: (ω, θ), Clients: N;
Output: Transformed data and labels (X', x') and associated gradients

Stochastic initialization
$X' \to U(0,1)$
$x' \to U(0,1)$
Calculate the raw gradient

$$\frac{\partial F(\theta, X, x)}{\partial \theta} \to gradient(X, x)$$

for $i = 1$ to N **do**
  Calculation of "virtual" gradients

$$\frac{\partial F(\theta, X', x')}{\partial \theta} \to gradient(X', x')$$

  $argmin_{X',x'} ReLU(h - ||X' - X||_2) + |x'_{\min} - x'|$
  L_2 metrics constrain transformation
  $argmin_{X',x'} ReLU(\left|\begin{matrix} gradient(X', x') - \\ gradient(X, x) \end{matrix}\right|_2 - \varepsilon)$
end for
return X', x'

Many recent studies have shown that attackers can use benign data of inference to train generative adversarial network models, and then generate new malicious traffic to carry out attack operations, so federated learning also faces serious privacy problems. In the face of this problem, we will operate on the data and la-

bels trained by the client model in the user system to ensure the privacy of user data.

Based on the characteristics of the extraction attack, we assume that at least one layer during model training contains weights and biases, then the output of this layer can be written as $w^T x + b$, where (w, b) are the corresponding weights and biases of the layer.

According to the attack characteristics, the attacker will steal the data to get the privacy data through the stolen parameters which depends on the absolute value of the stolen data, so in this paper we use the L_2 distance metric to constrain the distance between the original data and the transformed data and use the $ReLU(x)$ activation function to ensure that the distance between the original data and the transformed data is not infinitely far. Therefore we use (1) to constrain the original and transformed data.

$$argmin_{X'} ReLU(h - ||X' - X||_2). \tag{1}$$

Furthermore, considering that the reconstructed label is different from the reconstructed data, the reconstructed label does not rely on the data label of the original label, so unlike constraining the maximum distance between the original data and the transformed data, increasing the distance does not guarantee the privacy of the label. Therefore we first find the true label i in x, and then minimize x', so that $x'_{\min}$ is as close to x'_i as possible, so that the attacker is highly likely to reconstruct any index except i. The process is as in (2).

$$argmin_{x'} |x'_{\min} - x'_i|, \tag{2}$$

where X is the original data, x is the original label, X' is the virtual data, x' is the virtual label, h is the upper limit of the distance between the original data and the virtual data, and $x'_{\min}$ is the virtual label minimum.

In summary, from (1) and (2) we summarize the process of transforming the user's data and labels as follows:

$$argmin_{X',x'} ReLU(h - ||X' - X||_2) + |x'_{\min} - x'|. \tag{3}$$

And due to the updating method of neural network, the solution of gradient in the back propagation process will produce the phenomenon of *sigmoid* derivative and parameter multiplication. The maximum value of the *sigmoid* derivative is 0.25, and the weights are generally between 0 and 1 initially, so the product is less than 1, so that there will be more than one value less than 1 multiplied together, which will lead to the gradient close to the input layer tends to be 0. In this case, the data and label conversion will fall into a local optimum, and the obtained "virtual" gradient may tend to disappear, which will affect the input layer. In this case, the data and label transformation will fall into a local optimal situation, and the obtained "virtual" gradient may tend to disappear, which will affect the convergence of the global model. The derivative of $ReLU(x)$ activation function is always equal to 1 in

the positive range, so we use $ReLU(x)$ activation function to solve the problem of vanishing gradient. We also set a constant value C during the training process to determine the "virtual" gradient tends to 0 when the conversion is terminated early.

Based on the output of the inclusion of the weight and bias layers, we can introduce the following equation:

$$\nabla w = X \nabla b, \tag{4}$$

$$\nabla w' = X' \nabla b'. \tag{5}$$

And after getting the transformed data and label (X', x'), we use method to protect the user data according to the following equation:

$$||X' - X||_2 \geq \frac{2(||\nabla w' - \nabla w||_2 - ||\nabla b' - \nabla b||_2)}{2M + ||\nabla b' - \nabla b||_2}, \tag{6}$$

where M is an upper bound on the number of paradigms that limit the gradient with respect to the deviation b.

In order to ensure that the data contains as little information about the original data as possible and to augment the data with GAN, we chose to constrain the "virtual" gradients obtained using the transformed data and the labels, in order to keep the distance between the original gradient and the "virtual" gradient. The purpose is to keep the original gradient and the "virtual" gradient always constrained by the L_2 distance. Then, to convert the above constrained problem into a linear optimization problem, we obtain the following expression:

$$argmin_{X',x'} ReLU \left(\left|\left| \begin{matrix} gradient(X', x') - \\ gradient(X, x) \end{matrix} \right|\right|_2 - \varepsilon \right). \tag{7}$$

$gradient(X', x')$ and $gradient(X, x)$ represent the "virtual" gradient and the original gradient, respectively, and ε represents the distance constraint between the "virtual" gradient and the original gradient. Finally, we combine (3) and (7) to transform the original data and labels, in order to make the transformed data not similar to the original data to achieve the privacy protection of client user data and to ensure that the "virtual" gradient conforms to the model training process in federated learning, the transformation process is as follows:

$$\begin{aligned} &argmin_{X',x'} ReLU(h - ||X' - X||_2) + |x'_{\min} - x'| \\ &+ ReLU(||gradient(X', x') - gradient(X, x)||_2 - \varepsilon). \end{aligned} \tag{8}$$

The data and label conversion process is shown in Algorithm 1.

The client gets the transformed "virtual" data X' and the optimal training strategy, and after going through the local DNN network generator, it gets the generated data X^L. Then the "virtual" data X' and the generated data X^L are inputted into the CNN network using the CNN network's discriminator classification. The generator and the discriminator play a game where the generator continuously optimizes

the generated data in an attempt to get the discriminator to recognize the generated data as real, while the discriminator continuously optimizes itself to recognize the generated data. The generator and the discriminator are balanced to obtain a local generator model G^L and a local discriminator model D^L. The local generator model parameters are updated according to (9).

$$Adam\left(\left(\frac{1}{m}\sum_{i=1}^{m}\Delta_\omega L^i + n\right), \omega_i, \alpha_g\right) \rightarrow \omega_i^L, \tag{9}$$

where ω_i^L is the local generator model parameter and α_g is the generator learning rate. The local discriminator model parameters are updated according to (10).

$$Adam\left(\Delta\theta\frac{1}{m}\sum_{i=1}^{m} -D(G^i)\right), \theta_i, \alpha_d) \rightarrow \theta_i^L, \tag{10}$$

where G^i is the current round generator model, θ_i^L is the local discriminator model parameters, and α_d is the discriminator learning rate.

3.3 Personalized Federated Learning Based on Hypernetworks and Attention Mechanism Ensembles for Internet of Things

Different client data often come from different data sources or have different distributions, which also indicates that the uniform model pursued by federated learning may not be suitable for all clients. Therefore, we use Transformer to capture the global dependencies of client data through the self-attention mechanism and extract the personalized features among different clients. In real scenarios, the emergence of new users also tests the model's generalization ability, and we use the hypernetwork to generate personalized self-attention weights for the client's Transformer, which not only enables the model to better adapt to the local data features, but also enhances the model's generalization ability to new users.

The hypernetwork processes the input embedding vectors through its internal fully connected layer to generate the corresponding projection matrix (W^i), which includes the projection parameters of Q, K, and V used for the self-attention layer, expressed as (11):

$$W^i = \left[W_Q^i, W_K^i, W_V^i\right]. \tag{11}$$

This framework uses the attention mechanism for personalized features. The input image is first segmented into sequences at the time of processing with and then a transformation operation is performed on the sequences to convert the sequences into a matrix H. Where Q, K, and V in the attention mechanism are denoted as follows:

$$Q = HW_Q, K = HW_K, V = HW_V. \tag{12}$$

$W = [W_Q, W_K, W_V]$ then represents the projection matrix of Q, K, and V in the hypernetwork.

The generated projection matrix is used to update the self-attention layer of each client. Specifically, the original query, key and value matrices (H_Q, H_K, H_V) are generated by multiplying them with the projection matrix W^i to generate personalized queries, keys and values, and these personalized outputs are subsequently used for the self-attention computation as described in (13):

$$Attention(Q, K, V) = softmax\left(\frac{QK^t}{d^{\frac{1}{2}}}\right)V, \tag{13}$$

where d is the number of columns of Q, K, and V.

Assuming that there are a total of N clients in this federated learning framework, each client is used for its own local dataset D_i and so that user m_i represents client i with m_i samples from different data distributions, and then the whole network dataset is then denoted as D, and M represents the total dataset of all m_i. In this paper, we define $P(\mu;\cdot)$ as representing a personalization model that is optimized according to the personalization feature parameter μ and the optimization objective is:

$$argmin_\mu \sum_{i=1}^{N} \frac{m_i}{M} \mathcal{L}_i\left(\mu^i\right), \tag{14}$$

where $\mathcal{L}_i(\mu^i)$ is the loss function, in this experiment we choose the cross-entropy loss function. And during the training process, the personalized feature parameter $\mu^i = \{W^i, \eta^i\}$ will be divided into the attention parameter in Transformer and the ordinary parameter in the local client, which are W^i, η^i respectively. And the common parameters are aggregated in the federated learning architecture through (15).

$$(\eta^i)^t = \sum_{i=1}^{N} \frac{m_i}{M} \eta^i \tag{15}$$

t then denotes the number of training rounds for the client, and η^i is a common parameter.

After joining the hypernetwork, we define z_i as the input and then define the hypernetwork as $HN(\varphi_i, z_i)$. After joining the hyper network, we define z_i as the input and then define the hyper network as $HN(\varphi_i, z_i)$. After processing the personalized features by the hyper network, the personalized model is obtained as $P[(\mu^i)^t;\cdot] = P[(HN(\varphi, z_i))^t, (\eta^i)^t;\cdot]$. Therefore in this study (14) will be deformed to minimize the loss function as in (16):

$$argmin_{\varphi,z,\eta} \sum_{i=1}^{N} \frac{m_i}{M} \mathcal{L}_i\left(HN(\varphi, z_i), \eta^i\right). \tag{16}$$

The client accepts the global model parameters to get the personalized model $P(\mu;\cdot) = P(W^i, \eta^i;\cdot)$, and after t rounds of training the client gets $P[(\mu^i)^t;\cdot] = P[(W^i)^t, (\eta^i)^t;\cdot]$, where $(W^i)^t$ stays in the client locally to achieve the personalization of the client's model, while $(\eta^i)^t$ is uploaded to the server for aggregation.

Algorithm 2 Pseudo-code for Acquisition of personalized parameters under hypernetworks

Input: training round: t, Client-side embedding vectors: z_i, client: N;

Initialization parameters;
for $i = 1$ to N **do**
  Getting Personalized Model Parameters $\mu = \{W, \eta\}$;
  $W = [W_Q, W_K, W_V]$
  $(\eta^i)^t = \sum_{i=1}^{N} \frac{m_i}{M} \eta^i$
  Build hypernetwork $HN(\varphi, z_i)$
  Projecting the client's attention parameter yields $W^i = \left[W_Q^i, W_K^i, W_V^i\right]$;
  Hypernetwork processing of personalized features to obtain personalized models
  $P[(\mu^i)^t; \cdot] = P[(HN(\varphi, z_i))^t, (\eta^i)^t; \cdot]$
  $argmin_{\varphi,z,\eta} \sum_{i=1}^{N} \frac{m_i}{M} \mathcal{L}_i(HN(\varphi, z_i), \eta^i)$
  Calculate ΔW^i
  Upload ΔW^i and η^i to server;
  $\varphi^t = \varphi^{t-1} - \beta \nabla_\varphi \mathcal{L}_i^{t-1}$;
  $z_i^t = z_i^{t-1} - \beta \nabla_{z_i} \mathcal{L}_i^{t-1}$;
end for
return φ, z_i, η^i

During the communication rounds of federated learning, the client sends the updated model parameters (including personalized parameters from the self-attention layer and shared parameters from other layers) back to the central server. The server aggregates these parameters to update the global model. The hypernetwork parameter (φ) on the server is also updated based on the gradient information collected from the client. As follows:

$$\nabla_\varphi \mathcal{L}_i = \sum_{i=1}^{N} \frac{m_i}{M} \nabla \varphi W_t^i \Delta W^i, \tag{17}$$

$$\nabla_{z_i} \mathcal{L}_i = \sum_{1=1}^{N} \frac{m_i}{M} \nabla_{z_i} W_t^i \Delta W^i, \tag{18}$$

where ∇_φ and ∇_{z_i} are gradient operators.

The above process is iterated over multiple communication rounds until the model performance reaches the desired goal or the stopping condition is satisfied. In each iteration, the hypernetwork generates a new projection matrix to progressively improve the personalized self-attention layer of the model, t rounds as follows,

(19) for hypernetwork parameter updating and (20) for client embedding.

$$\varphi^t = \varphi^{t-1} - \beta \nabla_{\varphi} \mathcal{L}_i^{t-1}, \tag{19}$$

$$z_i^t = z_i^{t-1} - \beta \nabla_{z_i} \mathcal{L}_i^{t-1}. \tag{20}$$

β is the learning rate for global updates, $\mathcal{L}_i^{t-1}$ is the cross-entropy loss function, and z_i^{t-1} is the client embedding. As shown in Algorithm 2.

The generator model parameter of the server global model and the discriminator model parameter of the server global model are represented by the formulas:

$$\omega^{t+1} = \frac{\sum(\lambda_i * \omega_i^L)}{\sum \lambda_i}, \tag{21}$$

$$\theta^{t+1} = \frac{\sum(\lambda_i * \theta_i^L)}{\sum \lambda_i}, \tag{22}$$

where ω^{t+1} is the server generator model parameter after aggregating the local generator model parameters, λ_i is the weight of the i^{th} client, and ω_i^L is the local generator model parameter of the i^{th} client; θ^{t+1} is the server discriminator model parameter after aggregating the local discriminator model parameters, and θ_i^L is the local discriminator model parameter of the i^{th} client.

4 EXPERIMENTATION AND ANALYSIS

In this section, we present our experimental setup and conducted experiments for image categorization tasks with different data sources to make extensive experimental arguments. In detail, we use two datasets for comparison, CIFAR-10, CIFAR-100, MNIST and MNIST-M. Both of these datasets are used for image classification tasks, and both of them can be used as baseline datasets in machine learning research, but they still have some differences. CIFAR-10 and CIFAR-100 are color image datasets that contain many different object classes, while MNIST and MNIST-M are grayscale image datasets focusing on handwritten digit recognition. MNIST and MNIST-M have the same label distribution but different data feature distributions. While CIFAR-10 and CIFAR-100 differ in the number of categories they contain, each category in their respective datasets has an equal number of images uniformly distributed.

4.1 Datasets and Preprocessing

In the experiments, the IID data were set up so that the data distributions of different clients were similar and independent. The IID data were constructed by randomly selecting data in the training set of the CIFAR-10 dataset using a no-playback approach. But for the federated learning framework, the independent and identically distributed data situation is the ideal data distribution situation,

which will have high training accuracy, but in real life, there are data heterogeneity situations that affect the data distribution. For example, in a federated learning client, some clients have a large amount of data in category A and a small amount of data in category B. However, another part of the client may have a large amount of data in category B and a small amount of data in category C. The accuracy of the federated learning model under such conditions will be affected. And there is a great difference between the client data, so it is a great challenge for personalized federated learning.

In this experiment, we use 100 clients to complete the training of federated learning. In the IID data setup, we randomly assign the training set to 100 clients. However, for the Non-IID data setup, we randomly select 70 % of the categories in the dataset for allocation, i.e., 7 categories in the CFAR-10 dataset and 70 categories in CIFAR-100. The remaining categories are then selected from the remaining 30 % of the dataset, which is sufficient to ensure that the training samples are different, to ensure the reliability of the experimental results, and to help identify the limitations of the model in dealing with a small number of categories. In our experiments, we also distinguished between Pathological Distribution (Labeled Unbalanced Distribution) and Dirichlet Distribution for the data.

UNIT (%)	CIFAR-10				CIFAR-100			
Settings	Pathological		Dirichlet		Pathological		Dirichlet	
Client	50	100	50	100	50	100	50	100
FedAvg	51.37	46.72	56.77	57.95	15.78	14.29	18.40	21.44
FedProx	53.05	58.14	58.65	56.27	19.24	21.31	19.28	20.72
FedPer	83.11	81.01	77.17	74.08	49.04	41.43	22.45	19.59
pFedMe	86.02	85.29	75.86	74.60	49.45	45.73	31.32	25.54
FedBN	88.00	86.94	74.33	75.11	49.86	48.62	28.89	28.51
pFedHN	87.84	87.24	71.84	68.63	59.73	53.00	33.65	29.67
pFedGP	88.54	87.26	–	–	63.18	60.95	–	–
FedRoD	88.33	88.83	74.78	73.30	56.15	55.64	27.71	29.25
ours	89.87	88.27	80.64	80.08	67.79	63.94	46.34	43.38

Table 1. Comparison of the accuracy of different federated learning algorithms

As for the MNIST and MNIST-M datasets, we classify the datasets through two perspectives: quantity-based label imbalance and distribution-based label imbalance. Quantity-based labeling imbalance focuses on the fact that each client has only a predetermined number of labeled samples. In contrast, distribution-based label imbalance uses the Dirichlet distribution to assign labels to each client, and we still use $\alpha = 0.2$ from the above experiments.

In the field of machine learning, accuracy is a crucial metric for evaluating the performance of a classifier. It is defined as the ratio of correctly classified samples to the total number of samples, making it an intuitive and easily understandable measure of effectiveness. The formula for calculating the accuracy rate is as fol-

lows:

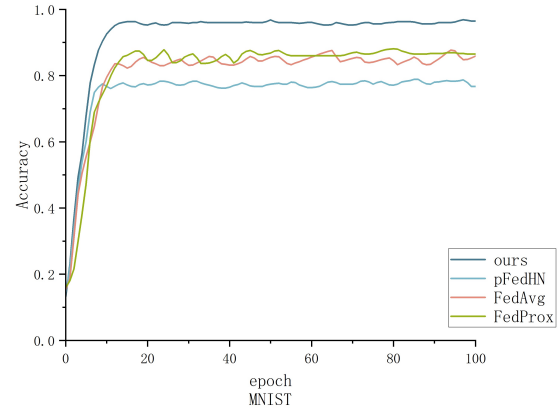
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{23}$$

$$RTA = \frac{Accuracy_{model}}{Accuracy_{baseline}} \times 100\,\%. \tag{24}$$

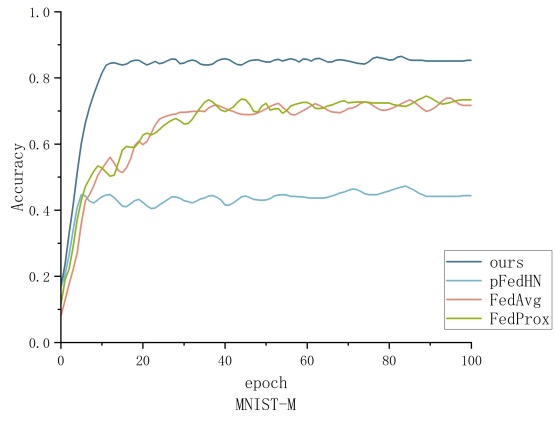
True Positive (TP) refers to the count of samples that are correctly identified as positive by the classifier. True Negative (TN) represents the number of samples that are accurately classified as negative. False Positive (FP) occurs when a classifier incorrectly labels negative samples as positive, while False Negative (FN) refers to positive samples that are mistakenly classified as negative. Mean Relative Test Accuracy (MRTA) refers to the average of the relative test accuracy obtained in each experiment over multiple experiments to obtain a more stable and comprehensive evaluation result.

4.2 Performance Analysis of Different Federated Learning Algorithms

In this paper, in order to verify that our method has better performance against heterogeneous data, we compare the method we use with some of the remaining personalized federated learning algorithms. According to Table 1, we can know that the accuracy of the method we use is higher than the other methods in most conditions, and has better performance when facing heterogeneous data. Since the FedAvg algorithm for heterogeneous data simple averaging does not bring good results, while FedBN and FedProx are based on FedAvg using batch normalization and introducing approximation terms to solve the problems caused by heterogeneity, which is not the optimal solution to solve the problem in complex heterogeneous scenarios. The performance of FedPre largely depends on the data distribution, so when facing the situation of complex data distribution, the performance of this model will be lower than our model performance. pFedMe and FedRod algorithms, due to their algorithmic characteristics, have a great shortcoming in the face of complex real-world scenarios and the situations that new users are constantly joining. The pFedHN algorithm, which is also applicable to the hypernetwork structure, will have the problem of decreasing model accuracy when the personalization requirement is high due to the lack of research on the attention mechanism. In summary, for the CIFAR-10 dataset, the FedRoD algorithm has about 0.5 % higher accuracy than the method we used, except for the Pathological distribution, when the number of clients is 100, and the method we used is higher than the rest of the algorithms in the rest of the conditions. The average accuracy improvement is 10.3 % for the CIFAR-10 dataset and 19.4 % for the CIFAR-100 dataset.



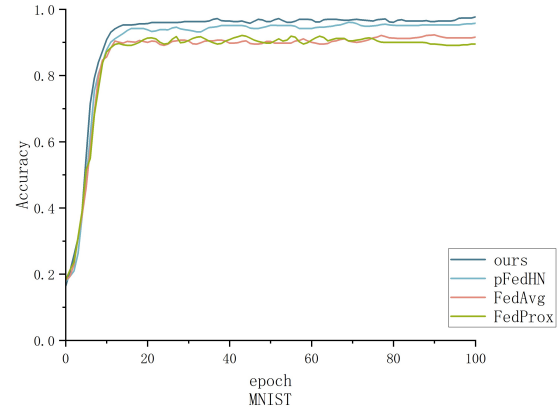
a)



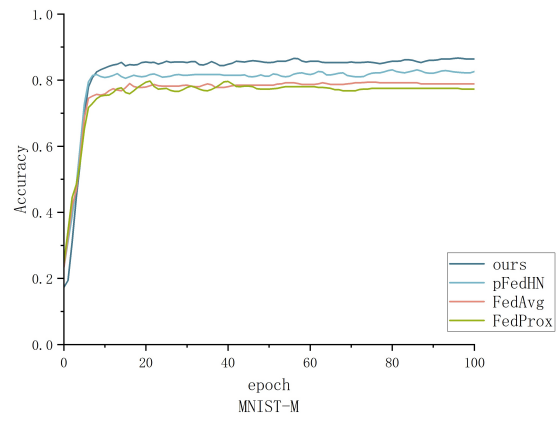
b)

Figure 4. Comparison of the accuracy of MNIST and MNIST-M datasets with different federated learning algorithms under quantity-based labeling imbalance

In the above experiments we verified that our method has better performance when facing differences in the number of categories, but the reality is more complex. Therefore, we further validate the model performance when the data features are distributed differently. We compare FedAvg, FedProx, pFedHN, and our method using two unbalanced methods for MNIST, as shown in Figures 4 and 5. Among them, Figure 4 compares the accuracies of the four methods under the number-based label imbalance setting, and we can see from the figure that our method has a better performance in terms of accuracy as well as convergence. We can see that although FedAvg and FedProx have better accuracy than pFedHN, the FedAvg and FedProx methods fluctuate more due to the data heterogeneity problem.



a)



b)

Figure 5. Comparison of the accuracy of MNIST and MNIST-M datasets with different federated learning algorithms under distribution-based label imbalance

On the other hand, the pFedHN method with the addition of hypernetwork has more stable play. Therefore, we use the hypernetwork to focus on the personalized features of client data to achieve the personalization of client models, fully solve the problem of model accuracy degradation caused by Non-IID data, and thus use the hypernetwork to improve the convergence speed and stability of the model. Figure 5 compares the accuracies of the four methods in the distribution-based label imbalance setting, where our method still has better accuracy than FedAvg, FedProx, and pFedHN. In the distribution-based label imbalance setting, although pFedHN also has good performance, the method is unstable and the accuracy fluctuates a lot compared to the quantity-based label imbalance setting. Therefore, it can be shown that

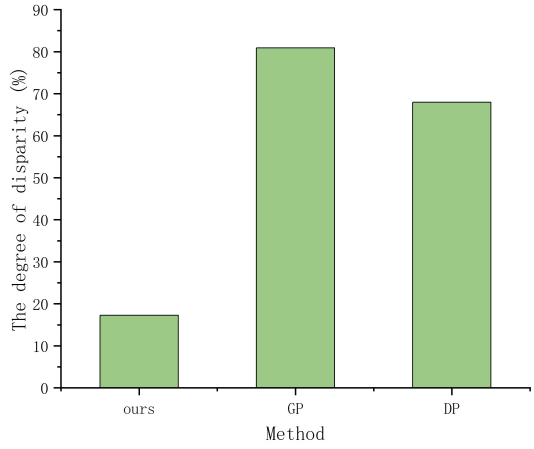


Figure 6. Performance comparison of different privacy protection technologies

our method has better performance when facing multiple real-data heterogeneous scenarios.

4.3 Client-Side Privacy Protection Analysis

UNIT (%)	CIFAR-10				CIFAR-100			
Settings	Pathological		Dirichlet		Pathological		Dirichlet	
Client	50	100	50	100	50	100	50	100
Local model	84.35	82.19	69.55	66.72	55.68	49.32	27.73	23.31
FedAvg*	52.78	48.52	62.45	60.03	34.48	30.75	38.51	35.01
FedProx*	49.79	46.01	61.98	60.58	30.76	29.83	37.25	33.48
ours	89.87	88.27	80.64	80.08	67.79	63.94	46.34	43.38

Table 2. Transformer performance comparison (* indicates that the algorithms use Transformer instead of the original neural network architecture)

In order to verify the effectiveness of our privacy-preserving mechanism based on data and label transformations for the client, we train an adversarial network model using the user data and labels extracted by the "attacker" to obtain the extracted data. We defend against such attacks by constraining the data and labels used for model training at the source. We compare the Gradients Perturbation and Differential Privacy methods commonly used in research. In the experiments, the Gradients Perturbation method introduces noise to achieve perturbation, which may directly affect the generalization ability of the model, while Differential Privacy protects individual privacy by adding randomness to the dataset. However, it may also result in the statistical characterization of the data becoming less accurate.

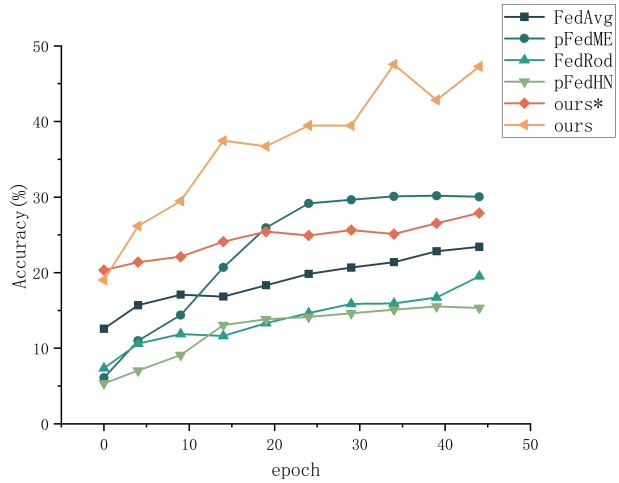


Figure 7. Comparison of generalizability against new user accessions in the CIFAR-10 dataset (* indicates that the Hypernetwork structure is not used)

Therefore, in Figure 6, we show that our proposed method of using transformed data for local model training has better privacy-preserving performance compared to the gradient perturbation and differential privacy methods, as the transformed data is as far away from the original data as possible, and therefore better resistant to the attacker's extraction attacks. This suggests that the transformed data we use can contain less user information, reduce the risk of reconstructing the user's private data after it is extracted by a malicious attacker, and better protect the user's privacy.

4.4 Generalization on New Clients

Considering that the model may perform well against the training data and that in reality there will be a constant stream of new users joining the federated learning network, we conducted an experimental test of the model's generalization ability. The test of model generalization ability can help us understand whether our proposed model is truly learning data features from experimental data, and not simply memorizing the training data to recognize data features. In this subsection, we operate on the CIFAR-10 dataset using the Dirichlet distribution and then select 20 % of the clients as new clients for the experiment.

The pFedMe algorithm necessitates the tuning of hyperparameters for various task scenarios, while FedRod requires a complex training strategy to effectively handle prediction tasks. Additionally, training the hypernetwork in the pFedHN method demands greater computational resources and scenario-specific adjustments to the hypernetwork's structure and parameters, it has poor performance against

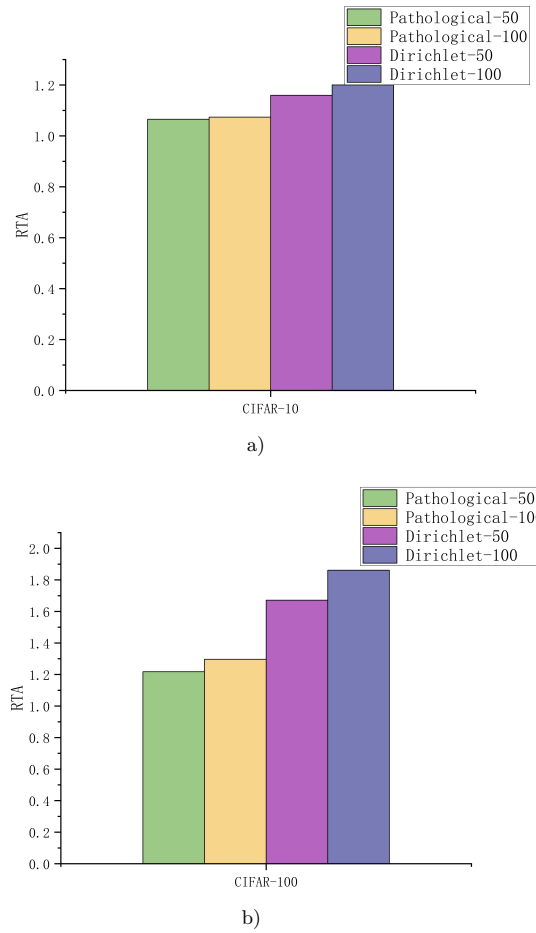


Figure 8. Comparison of RTA of federated learning algorithms using different Transformer architectures in the CIFAR-10 and CIFAR-100 dataset

new user joining. As shown in Figure 7, in the face of new user joining, the weights do not increase linearly with new user joining because we use the hypernetwork to generate personalized weights for each client. The experimental results show that our model has better generalization ability, which indicates that the model we use is more resistant to small changes or noise in the data, which means that the model still maintains a better performance in the face of complex or imperfect data. And it also shows that our model has better generalization ability to new data, when faced with data growth or rapid changes, our model is better able to adapt to the addition of new data without the need to retrain the model.

UNIT (%)	CIFAR-10								
α	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$
FedAvg	49.22	56.77	58.35	59.73	62.84	62.98	63.28	64.75	66.17
FedProx	60.23	58.65	55.36	54.97	52.19	53.08	51.36	50.08	51.32
pFedMe	76.86	75.86	73.29	70.97	72.84	73.26	71.55	71.26	73.22
pFedHN	75.98	71.84	69.75	70.69	68.53	69.15	70.16	67.32	66.87
FedRoD	74.16	74.78	73.98	72.62	72.69	71.37	71.28	70.96	71.18
ours	82.14	80.64	77.68	78.76	77.98	76.54	76.17	75.88	73.29

Table 3. Effect of comparing α on accuracy in the CIFAR-10 dataset

4.5 Ablation Experiment

4.5.1 Performance Comparison of Different Transformer Network Architectures

In order to verify the negative impact of the FedAvg algorithm on the accuracy of training models on heterogeneous data, we compared several classical algorithms with our algorithm. For fairness, we changed the neural network in the original algorithm to the same transformer structure as in this experiment. As can be seen from the data in Table 2, the method we used has a better performance in the face of heterogeneous data and the accuracy of the locally trained model is higher than that of the federated learning model after using the FedAvg algorithm, since the FedAvg algorithm negatively affects the heterogeneous data. We take the CIFAR-10 and CIFAR-100 dataset as an example, as shown in Figure 8, it can be seen that the relative test accuracies of our model and the local model are both greater than 1, so it can be concluded that the accuracy of the model processed by our framework is higher than the accuracy of the local model after training.

UNIT (%)	CIFAR-10				CIFAR-100			
Settings	Pathological		Dirichlet		Pathological		Dirichlet	
Client	50	100	50	100	50	100	50	100
Local model	84.35	82.19	69.55	66.72	55.68	49.32	27.73	23.31
pFedHN	87.84	87.24	71.84	68.63	59.73	53.00	33.65	29.67
ours*	86.95	84.55	74.96	72.03	62.01	56.51	36.28	32.19
ours	89.87	88.27	80.64	80.08	67.79	63.94	46.34	43.38

Table 4. Performance comparison with and without Hypernetwork (* indicates that the Hypernetwork structure is not used)

The average relative test accuracy (MRTA) of this experiment is 1.12 in the CIFAR-10 dataset, and 1.51 in the dataset CIFAR-100. These results can prove that the model accuracy of our experimental approach is higher than that of the local model in the face of heterogeneous data, and therefore, our method has a very good performance in the federated learning environment with respect to different data

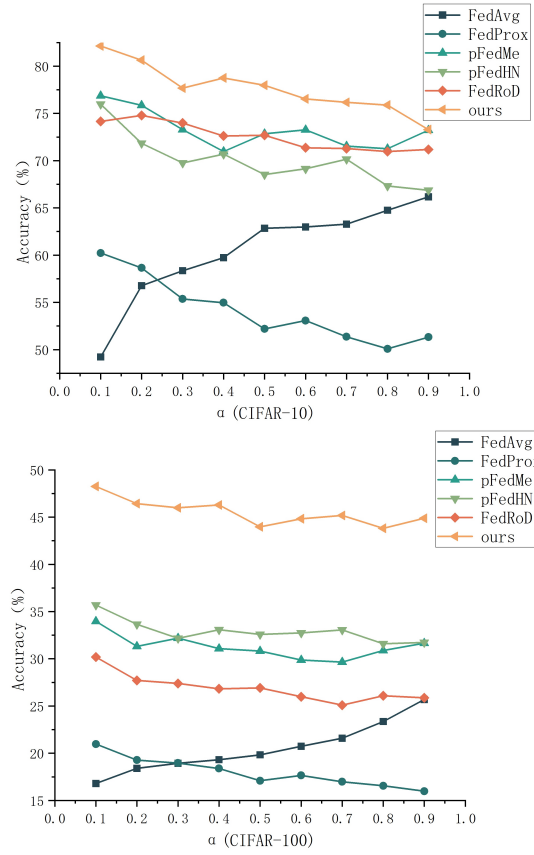
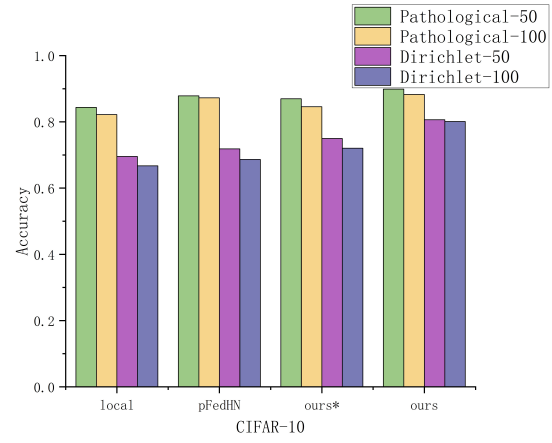


Figure 9. Effect of α coefficient on accuracy under CIFAR-10 and CIFAR-100 dataset

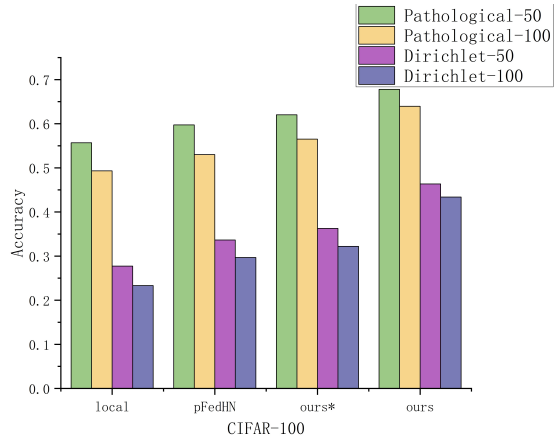
features. Therefore, our method has good performance for different data features in a federated learning environment, can solve the data heterogeneity problem well, and can build a personalized federated learning framework that conforms to the data distribution of the whole network.

4.5.2 The Effect of the Magnitude of the α Coefficient in the Dirichlet Distribution

The Dirichlet distribution is a multivariate probability distribution that is usually used in scenarios such as Bayesian statistics and topic modeling in machine learning (e.g., LDA), in which the α coefficient plays an important role. In order to study the impact brought by the data distribution situation on the model accuracy and to verify the performance of our algorithm in different situations, we hereby determine whether our algorithm performs better in the face of different scenarios by examining



a)



b)

Figure 10. Hypernetwork performance analysis in the CIFAR-10 and CIFAR-100 dataset (* indicates that the Hypernetwork structure is not used)

the impact of the α coefficient on the model accuracy in the Dirichlet distribution. When the value of α is small, the distribution will be more concentrated, meaning that there will be fewer categories with higher probabilities, while others will have probabilities close to zero. In this case, the distribution will be sparser, indicating a stronger preference for certain categories. When the value of α is large, the distribution will be more even, indicating less variation in preferences for individual categories.

Whereas data heterogeneity is a key problem to be solved in personalized federated learning, in order to verify that our method outperforms other algorithms,

we conducted comparison experiments for the value of α, choosing 50 clients in the CIFAR-10 and CIFAR-100 datasets, respectively. Table 3 and Figure 9 show the effect of α on accuracy. We can see that some of the remaining federated learning algorithms are unable to adequately capture the user's personalized information when α becomes smaller, i.e., when the data distribution is more different, and are unable to address the negative impact of data heterogeneity, but our method still has a better performance.

4.5.3 Hypernetwork Performance Analysis

We also made experiments to verify whether the hypernetwork has improved the model performance or not, and the results of the experiments are shown in Table 4 and Figure 10. We compared the accuracy of the local model without hypernetwork, pFedHN with hypernetwork, and the model accuracy of the method in our article with and without the hypernetwork structure. By comparing the magnitude of the accuracy of the models in our experimental approach using and not using the hypernetwork structure, we can see that the hypernetwork has a better performance in the face of personalized features, and it can better encode the client's personalized information into the client's input as a way to improve the personalized performance of the model. And it improves by 2–8 % and 5–11 % in the CIFAR-10 and CIFAR-100 datasets, respectively. Comparing again to the pFedHN method that uses the hypernetwork structure, our model still has better performance.

5 CONCLUSIONS

In this paper, we propose an innovative personalized federated learning strategy, pFedHT, which has good applicability in the federated learning domain, not only for IID data, but also for Non-IID data, and at the same time can satisfy the needs of organizations for personalized services. This strategy enhances the diagnostic and predictive capabilities of local models by sharing personalized feature information across organizations. We employ data transformation to ensure the privacy and security of clients' data, while utilizing a personalized federated learning framework to cope with the heterogeneity of data. This approach not only protects users' privacy, but also enables the model to better adapt to distributed data sources from different organizations. This is particularly important for the distributed system, as different organizations may have user populations with different characteristics and need to ensure that the trained model is applicable to all clients.

However, according to our research findings, the adoption of hypernetwork technique, although it has significant advantages in data privacy protection and data heterogeneity problems, may bring high resource consumption during its operation, especially during the training phase when it may encounter the slow speed problem. One of our primary future research goals will be to implement advanced optimization techniques to minimize the number of parameters, making the model more lightweight and suitable for deployment on resource-constrained IoT devices. This

will ensure that our method is not only accurate and secure, but also efficient and scalable across various systems with different hardware capabilities. Additionally, we aim to explore broader applications of personalization. While this study focuses on federated learning in IoT environments, the personalization mechanism we developed can be applied to other domains, such as healthcare, finance, and smart cities, where heterogeneous data and privacy concerns are similarly critical. Future research will include the design of personalized federated learning frameworks tailored to these fields, taking into account their specific data types, privacy requirements, and system constraints.

Declaration

Conflict of interest: The authors declare that they have no conflict of interest.

Ethics approval and consent to participate: This study does not involve medical ethics approval and consent to participate.

Consent for publication: All authors of this study have agreed to be published in your journal.

Competing interests: No competing interests in this study.

Funding: The study is supported by the National Natural Science Foundation of China (Grant No. 62402338).

Authors'contributions: All authors contributed to the study conception and design. Huiqi Zhao and Lu Liu designed and analyzed the study and led the experimental process. Lu Liu and Fang Fan participated in the collection and organization of data, writing and revision of the first draft. Huiqi Zhao and Zhihan Lyu carried results validation and analysis. Huiqi Zhao and Sibo Qiao obtained funding for the study. All authors discussed and approved the final version of the paper together and agreed to take responsibility for the content presented.

Acknowledgements: This study acknowledges the support of Shandong University of Science and Technology, Tiangong University and Uppsala University.

Availability of data and material: The datasets used or analysed during the current study are available from the corresponding author on reasonable request.

REFERENCES

[1] Sattler, F.—Wiedemann, S.—Müller, K. R.—Samek, W.: Robust and Communication-Efficient Federated Learning from Non-i.i.d. Data. IEEE Transactions on Neural Networks and Learning Systems, Vol. 31, 2019, No. 9, pp. 3400–3413, doi: 10.1109/TNNLS.2019.2944481.

[2] Zhang, L.—Shi, Y.—Chang, Y. C.—Lin, C. T.: Hierarchical Fuzzy Neural Networks with Privacy Preservation for Heterogeneous Big Data. IEEE Transactions on Fuzzy Systems, Vol. 29, 2020, No. 1, pp. 46–58, doi: 10.1109/TFUZZ.2020.3021713.

[3] Shi, Y.—Zhang, L.—Cao, Z.—Tanveer, M.—Lin, C. T.: Distributed Semisupervised Fuzzy Regression with Interpolation Consistency Regularization. IEEE Transactions on Fuzzy Systems, Vol. 30, 2021, No. 8, pp. 3125–3137, doi: 10.1109/TFUZZ.2021.3104339.

[4] Yu, L.—Huang, J.: Cyclic Federated Learning Method Based on Distribution Information Sharing and Knowledge Distillation for Medical Data. Electronics, Vol. 11, 2022, No. 23, Art. No. 4039, doi: 10.3390/electronics11234039.

[5] Qu, L.—Zhou, Y.—Liang, P. P.—Xia, Y.—Wang, F.—Adeli, E.—Fei-Fei, L.—Rubin, D.: Rethinking Architecture Design for Tackling Data Heterogeneity in Federated Learning. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10051–10061, doi: 10.1109/CVPR52688.2022.00982.

[6] Mothukuri, V.—Parizi, R. M.—Pouriyeh, S.—Huang, Y.—Dehghantanha, A.—Srivastava, G.: A Survey on Security and Privacy of Federated Learning. Future Generation Computer Systems, Vol. 115, 2021, pp. 619–640, doi: 10.1016/j.future.2020.10.007.

[7] Melis, L.—Song, C.—De Cristofaro, E.—Shmatikov, V.: Exploiting Unintended Feature Leakage in Collaborative Learning. CoRR, 2018, doi: 10.48550/arXiv.1805.04049.

[8] Truex, S.—Liu, L.—Gursoy, M. E.—Yu, L.—Wei, W.: Demystifying Membership Inference Attacks in Machine Learning as a Service. IEEE Transactions on Services Computing, Vol. 14, 2021, No. 6, pp. 2073–2089, doi: 10.1109/TSC.2019.2897554.

[9] Koloskova, A.—Stich, S. U.—Jaggi, M.: Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication. CoRR, 2019, doi: 10.48550/arXiv.1902.00340.

[10] Zhao, Y.—Li, M.—Lai, L.—Suda, N.—Civin, D.—Chandra, V.: Federated Learning with Non-IID Data. CoRR, 2018, doi: 10.48550/arXiv.1806.00582.

[11] Wu, Q.—Chen, X.—Zhou, Z.—Zhang, J.: FedHome: Cloud-Edge Based Personalized Federated Learning for In-Home Health Monitoring. CoRR, 2020, doi: 10.48550/arXiv.2012.07450.

[12] Li, L.—Duan, M.—Liu, D.—Zhang, Y.—Ren, A.—Chen, X.—Tan, Y.—Wang, C.: FedSAE: A Novel Self-Adaptive Federated Learning Framework in Heterogeneous Systems. 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–10, doi: 10.1109/IJCNN52387.2021.9533876.

[13] Chai, Z.—Ali, A.—Zawad, S.—Truex, S.—Anwar, A.—Baracaldo, N.—Zhou, Y.—Ludwig, H.—Yan, F.—Cheng, Y.: TiFL: A Tier-Based Federated Learning System. CoRR, 2020, doi: 10.48550/arXiv.2001.09249.

[14] Daliang Li, J. W.: FedMD: Heterogenous Federated Learning via Model Distillation. CoRR, 2019, doi: 10.48550/arXiv.1910.03581.

[15] Arivazhagan, M. G.—Aggarwal, V.—Singh, A. K.—Choudhary, S.:

Federated Learning with Personalization Layers. CoRR, 2019, doi: 10.48550/arXiv.1912.00818.

[16] McMahan, H. B.—Moore, E.—Ramage, D.—Hampson, S.—Agüera y Arcas, B.: Communication-Efficient Learning of Deep Networks from Decentralized Data. CoRR, 2016, doi: 10.48550/arXiv.1602.05629.

[17] Li, T. A.—Sahu, K.—Zaheer, M.—Sanjabi, M.—Talwalkar, A.—Smith, V.: Federated Optimization in Heterogeneous Networks. CoRR, 2018, doi: 10.48550/arXiv.1812.06127.

[18] Chen, H. Y.—Chao, W. L.: On Bridging Generic and Personalized Federated Learning for Image Classification. CoRR, 2021, doi: 10.48550/arXiv.2107.00778.

[19] Melis, L.—Song, C.—De Cristofaro, E.—Shmatikov, V.: Exploiting Unintended Feature Leakage in Collaborative Learning. CoRR, 2018, doi: 10.48550/arXiv.1805.04049.

[20] Geiping, J.—Bauermeister, H.—Dröge, H.—Moeller, M.: Inverting Gradients – How Easy Is It to Break Privacy in Federated Learning? CoRR, 2020, doi: 10.48550/arXiv.2003.14053.

[21] Sun, J.—Li, A.—Wang, B.—Yang, H.—Li, H.—Chen, Y.: Soteria: Provable Defense Against Privacy Leakage in Federated Learning from Representation Perspective. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9307–9315, doi: 10.1109/CVPR46437.2021.00919.

[22] Boenisch, F.—Dziedzic, A.—Schuster, R.—Shamsabadi, A. S.—Shumailov, I.—Papernot, N.: When the Curious Abandon Honesty: Federated Learning Is Not Private. CoRR, 2021, doi: 10.48550/arXiv.2112.02918.

[23] Ha, D.—Dai, A.—Le, Q. V.: HyperNetworks. CoRR, 2016, doi: 10.48550/arXiv.1609.09106.

[24] Shamsian, A.—Navon, A.—Fetaya, E.—Chechik, G.: Personalized Federated Learning Using Hypernetworks. CoRR, 2021, doi: 10.48550/arXiv.2103.04628.

[25] Ma, X.—Zhang, J.—Guo, S.—Xu, W.: Layer-Wised Model Aggregation for Personalized Federated Learning. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10082–10091, doi: 10.1109/CVPR52688.2022.00985.

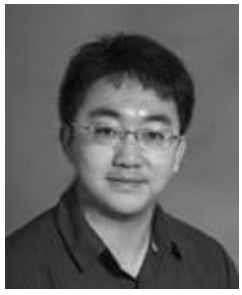
[26] Sun, G.—Mendieta, M.—Luo, J.—Wu, S.—Chen, C.: FedPerfix: Towards Partial Model Personalization of Vision Transformers in Federated Learning. CoRR, 2023, doi: 10.48550/arXiv.2308.09160.

Lu Liu is affiliated with Shandong University of Science and Technology, M.Sc. graduated from Shandong University of Science and Technology in 2025. His main research interests are federated learning, artificial intelligence, industrial internet, and cyber security.

Huiqi Zhao Ph.D., Associate Professor, Master's supervisor, is in urgent need of talents in the key support areas of Shandong Province. Research interests: industrial control security, Internet of Things security, data privacy protection, evolutionary computing, etc. He has presided over and participated in a number of scientific research projects such as major scientific and technological innovation projects of Shandong Provincial Key R & D Program, Shandong Provincial Natural Science Foundation, and Shandong Provincial Civil Air Defense Scientific Research Project, he published more than 20 SCI papers, and authorized more than 10 invention patents. He serves as the director of Shandong Data Innovation and Open Laboratory, the director of Tai'an Industrial Information Security Engineering Laboratory, a member of the Functional Safety and Information Security Committee of the China Association for Standardization, a member of the Chinese Association of Automation, a member of the Network Information Service Committee of the Chinese Society of Automation, a member of the China Computer Federation, and a member of the Network Security Committee of the Shandong Computer Federation.

Fang Fan is a teacher from the Intelligent Equipment Institute, Shandong University of Science and Technology, graduated from Shandong University of Science and Technology with a Ph.D. Her main research interests are artificial intelligence, intelligent optimization, industrial internet, and cyber security. She has long been engaged in the research and application of intelligent computing, network security and related fields.

Sibo Qiao is Associate Professor in the Tiangong University, Tianjin, China. He received his Master's and Ph.D. degrees from the China University of Petroleum, Qingdao, China, in 2020 and 2023, respectively. His research interests include digital twins, federated learning, deep learning, edge computing and image processing.



Zhihan Lyu is Associate Professor in Uppsala University, Sweden. He received his Ph.D. degree from Paris 7 University and the Ocean University of China, Qingdao, China, in 2012. His research interests include deep learning, virtual reality, computer vision, 3D visualization and graphics, Big Data, GIS, cloud computing, edge computing.