Robust Federated Learning With Contrastive Learning and Meta-Learning

Huan Zhang[1], Yuxiang Chen[1,2], Kuanching Li[1,2*], Yuhui Li[3], Sisi Zhou[1], Wei Liang[1,2], Aneta Poniszewska-Maranda[4]

[1] School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan (China)
[2] Sanya Research Institute, Hunan University of Science and Technology, Sanya (China)
[3] Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong SAR (China)
[4] Institute of Information Technology, Lodz University of Technology, Łódź (Poland)

* Corresponding author: aliric@hnust.edu.cn

Received 18 December 2024 | Accepted 9 April 2025 | Published 26 September 2025

Abstract

Federated learning is regarded as an effective approach to addressing data privacy issues in the era of artificial intelligence. Still, it faces the challenges of unbalanced data distribution and client vulnerability to attacks. Current research solves these challenges but ignores the situation where abnormal updates account for a large proportion, which may cause the aggregated model to contain excessive abnormal information to deviate from the normal update direction, thereby reducing model performance. Some are not suitable for non-Independent and Identically Distribution (non-IID) situations, which may lead to the lack of information on small category data under non-IID and, thus, inaccurate prediction. In this work, we propose a robust federated learning architecture, called FedCM, which integrates contrastive learning and meta-learning to mitigate the impact of poisoned client data on global model updates. The approach improves features by leveraging extracted data characteristics combined with the previous round of local models through contrastive learning to improve accuracy. Additionally, a meta-learning method based on Gaussian noise model parameters is employed to fine-tune the local model using a global model, addressing the challenges posed by non-independent and identically distributed data, thereby enhancing the model's robustness. Experimental validation is conducted on real datasets, including CIFAR10, CIFAR100, and SVHN. The experimental results show that FedCM achieves the highest average model accuracy across all proportions of attacked clients. In the case of a non-IID distribution with a parameter of 0.5 on CIFAR10, under attack client proportions of 0.2, 0.5, and 0.8, FedCM improves the average accuracy compared to the baseline methods by 8.2%, 7.9%, and 4.6%, respectively. Across different proportions of attacked clients, FedCM achieves at least 4.6%, 5.2%, and 0.45% improvements in average accuracy on the CIFAR10, CIFAR100, and SVHN datasets, respectively. FedCM converges faster in all training groups, especially showing a clear advantage on the SVHN dataset, where the number of training rounds required for convergence is reduced by approximately 34.78% compared to other methods.

Keywords

Contrastive Learning, Federated Learning, Meta-Learning, Non-Independent and Identically Distribution (Non-IID).

DOI: 10.9781/ijimai.2025.09.004

I. Introduction

In the era of artificial intelligence, where various types of data (such as images, audio, and text) are growing exponentially, and demands for data privacy protection are becoming increasingly stringent, federated learning decentralizes model training to the client side. It eliminates the need to share private data by transmitting local model updates, which are then aggregated on a server to form a global model. Federated learning effectively addresses the data requirements of artificial intelligence and is widely applied in finance [1], Internet of Things [2], healthcare [3] and other fields [4], [5]. For example, in COVID-19 testing, federated learning is used, and different medical institutions use private chest X-ray images to train models, which can avoid privacy regulations, privacy leaks, and other issues [3]. However, two major factors affecting model performance are the unbalanced data distribution and clients' vulnerability to attacks, in federated learning.

Whether it is the unbalanced data distribution or the local data anomalies caused by client attacks, both can lead to deviations in local model training, thereby reducing the accuracy of the global model.

Please cite this article as:
H. Zhang, Y. Chen, K. Li, Y. Li, S. Zhou, W. Liang, A. Poniszewska-Maranda. Robust Federated Learning with Contrastive Learning and Meta-Learning, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 6, pp. 38-51, 2026, http://doi.org/10.9781/ijimai.2025.09.004

Therefore, it is essential to consider the impact of non-Independent and Identically Distribution (non-IID) on the accuracy of the global model, while also distinguishing between anomalies caused by attacks and biases resulting from unbalanced data distribution.

To address the accuracy issues caused by non-IID data, existing studies have employed methods such as shared mapping spaces [6], global feature dictionaries [7], data distribution information [8], and the addition of loss terms related to the global model during local training [9], [10] to enhance model accuracy under non-IID conditions. However, these methods do not consider the presence of data anomalies and overlook that abnormal data can affect the update direction of the global model.

Existing methods primarily focus on robust aggregation techniques to address the decline in model accuracy caused by data anomalies. These methods aim to reduce the impact of model updates from abnormal data on the global model by eliminating or adjusting the weights of abnormal updates. However, removing abnormal updates can lead to the loss of certain data information [11]–[14], making it impossible to predict some categories of data and compromising the stability of the model. While methods that adjust the weights of abnormal updates [15], [16] do not lose data information, they can suffer from reduced accuracy when faced with many contaminated clients. Both approaches lack the utilization of real data information and may not be suitable for non-IID situations.

To address the decline in model accuracy caused by data anomalies under non-IID conditions, this paper proposes a robust method called FedCM. This approach effectively trains local models on benign clients to obtain noise-resistant models. Doing so reduces the negative impact of data anomalies on global updates and improves the accuracy of the global model. FedCM consists of the following components: 1) Contrastive Learning Based on Improved Data Features: This method gains improved local features by weighting local model features from the previous round, less affected by noise. By mitigating the detrimental effects of noisy features, this approach brings similar data points closer together and enhances model accuracy. 2) Meta-Learning Based on Gaussian Noise Model Parameters: This approach employs contaminated data with randomly added Gaussian noise alongside the original data to train noise-resistant local models through meta-learning. The global model is then used to correct the update direction of the local model, which improves its generalization capability, alleviates non-IID issues, and enhances the model's robustness to data anomalies , which is applicable to multiple practical application scenarios.For example, in the field of medical image analysis, hospitals can use FedCM to jointly train disease detection models while adapting to the data distribution of different institutions; in the financial fraud detection scenario, FedCM can enhance the model's adaptability to different bank fraud patterns and improve detection accuracy; in IoT anomaly detection, FedCM optimizes feature extraction and model generalization capabilities to enable smart devices to more accurately identify anomalies. These features make FedCM have broad application potential in real-world environments with data privacy and non-IID data.

The main contributions of this work are as follows:

- A robust federated learning method, FedCM, is proposed, which consists of contrastive learning based on improved data features and meta-learning based on Gaussian noise model parameters, which can effectively improve the model accuracy.
- A contrastive learning method based on improved data features is proposed, which performs contrastive learning using the weighted data features of the previous and current rounds of models to reduce the impact of noise features and thus improve the model's accuracy.
- Meta-learning is performed based on Gaussian noise model parameters to obtain noise-resistant model parameters, thereby improving the robustness of the model to data anomalies under non-IID.
- On the CIFAR10, CIFAR100, and SVHN datasets, the proposed method improves the model accuracy by at least 8.2%, 6.1%, and 0.58%, respectively. Additionally, on the SVHN dataset, the number of training rounds required for convergence is reduced by approximately 34.78% compared to other methods.

The remainder of this work is organized as follows: Section II reviews and analyzes the existing achievements related to the unbalanced data distribution in federated learning. Meanwhile, Section III briefly introduces the foundational knowledge and algorithmic theories relevant to our scheme. Section IV provides a detailed description of the proposed solution, and Section V presents a thorough analysis of the experimental results. Finally, Section VI concludes the paper with a summary and directions for future research.

II. Related Work

In federated learning, non-IID data is a significant factor affecting model accuracy. There has been relevant research on the issue of statistical heterogeneity of data. For example, FedProx [9] addresses the non-IID problem by introducing a regularization term to train local models, thereby narrowing the gap between local model updates. However, the uncertainty and dynamics of data anomalies make selecting the optimal regularization parameter a challenge. FedAlign [10] trains models using alignment loss to ensure consistent model updates and alleviate the non-IID issue. However, FedAlign performs poorly in the presence of data anomalies.

The Virtual Homogeneity Learning (VHL) method [6] maps the data of each client to a virtual homogeneous space, then utilizes the representation of the obtained unified feature distribution to train the model and employs a weighted aggregation mechanism from the virtual homogeneous space to obtain a global model, thereby improving model accuracy. However, anomalous data can lead to abnormal information in the feature distribution, misleading the model update direction and decreasing accuracy. FedCA [7] uses a feature dictionary to assist local training and enhance feature consistency to solve the non-IID problem. However, anomalous data can introduce abnormal features into the feature dictionary, misleading the model update direction and reducing accuracy. The Classifier Calibration with Virtual Representations (CCVR) algorithm [25] uses virtual representations generated by a Gaussian mixture model to calibrate the classifier and introduces a weighting mechanism to balance the impact of models trained on different clients on the global model, thus improving model accuracy under non-IID conditions. However, anomalous data may lead to weight imbalance, affecting the global model.

MOON [17] mainly uses the current local model, the global model and the previous round of local model for contrastive learning to solve the non-IID problem. However, the existence of data anomalies may cause the update direction of the global model to deviate, and MOON's local training depends on the data features extracted by the global model, which leads to a decrease in model accuracy. FedProc [18] mainly uses class prototypes as global knowledge to propagate to all clients. Each client uses the global class prototype for contrastive learning to make the local training direction consistent with the global one. FedPCL [19] uses local and global prototypes for contrastive learning to solve the non-IID problem; however, if data anomalies occur, the feature mean of the data may deviate from the true category center, resulting in errors in the global class prototype, which may cause fluctuations in the client training direction and affect model convergence.

In response to the label noise problem, Jinchui Zhang et al. [22] proposed a noise-aware local model training mechanism, which mainly uses a label correction network to convert noisy labels into soft labels, and optimizes the model through meta-learning to reduce the impact of noise on the performance of the local model. FedLN [21] mainly uses embedding-based discovery of noisy labels to estimate noise, uses the nearest neighbor label correction to correct labels, uses adaptive knowledge distillation to guide local training to reduce the impact of incorrect labels, and uses noise-aware weighted averaging on the server to reduce the impact of noisy clients. The above two methods are used to solve the label data noise problem. In response to the data noise problem, FedNS [23] uses gradient norm analysis to identify noisy clients in the initial round, and uses noise-aware aggregation methods to optimize model aggregation, thereby reducing the impact of noisy clients. It is suitable for scenarios with obvious noise data. FedNS mainly operates at the client level, not directly at the data level. If the proportion of client noise data is small, FedNS may not be able to distinguish well. The abnormal data generated by the client attack is random and unknowable, so FedNS may sometimes not recognize significant gradient changes, which affects FedNS's detection ability.

On the client side, Mean Augmented Federated Learning (MAFL) [20] mixes local updates to approximate Mixup data augmentation, improving model generalization and effectively addressing the non-IID issue. However, data anomalies can produce feature mean shifts and amplify abnormal knowledge, reducing model accuracy. Astraea [8] uses the global data distribution for data augmentation and creates intermediaries to regulate local model training on clients, employing adaptive weights to balance the contribution of each local model, thus effectively solving the data heterogeneity problem and improving model accuracy. However, suppose a client has severe anomalous data. In that case, it may prevent the self-balancing mechanism from effectively calibrating the data distribution, causing the global model to aggregate the anomalous information from that local model, amplifying the impact of the anomaly and reducing accuracy. Additionally, some methods utilize knowledge distillation [26]–[28] and personalization [29], [30] to obtain globally shared knowledge to address the non-IID problem. Still, anomalous data may cause global knowledge to contain abnormal information, thus lowering model accuracy.

Regarding model aggregation methods, FedAvg [31] is the mainstream aggregation method. However, it cannot effectively address the reduction in model accuracy caused by anomalous data. Existing methods primarily minimize the impact of anomalous data on the global model by eliminating anomalous updates and adjusting the weights of these updates, achieving robust aggregation.

Methods for updating models by eliminating anomalous updates include trimmed mean, median, Krum method, and norm bound. The trimmed mean method [11] calculates the mean of each dimension of local updates after removing the maximum and minimum values to obtain the aggregated global model. This calculation is simple and can effectively resist potential anomalous updates, but it may lead to the loss of information corresponding to small or large categories in the data. The median method [12] updates the global model by calculating the median of model updates, reducing the impact of large deviations on the global model, effectively alleviating the problem of data heterogeneity, and decreasing the interference of malicious updates on model aggregation. However, when the data distribution is highly non-IID, model updates trained on overly concentrated data from certain categories or insufficiently trained small data categories may deviate significantly. Krum algorithm [24] calculates the Euclidean distance between local updates and selects a local update most similar to n-m-2 adjacent updates to aggregate into a global model, thus eliminating relatively deviated local updates and making the model more robust, where m is the expected number of malicious clients. However, this method struggles to effectively distinguish between noisy and benign updates under non-IID conditions, reducing model accuracy. The Multi-Krum algorithm is an extension of Krum that is suitable for non-IID scenarios. The norm bound algorithm [13] sets a threshold, treating local updates with norms above this threshold as malicious. Only benign updates are aggregated during aggregation, effectively reducing malicious updates' impact on global performance. However, if many malicious local updates occur in a single communication round, the aggregated global model will only retain the knowledge of a few categories, significantly reducing model accuracy and slowing convergence.

Methods that adjust the weights of anomalous updates include RFA [15], which utilizes an approximate geometric median operation as an aggregation method, effectively reducing the impact of anomalous data on the global model under non-IID conditions. Residual [16] uses a median estimator to calculate the residual of each local model parameter and dynamically adjusts the aggregation weight based on the residual. The larger the residual, the more significant the gap between the local update and the global model, which increases the likelihood that the corresponding client contains anomalous or malicious data, thus requiring a smaller aggregation weight. This adaptive method is more robust and can withstand uncertain and dynamic attacks better.

Although anomaly detection [32]–[34] can be employed on the client side to improve model accuracy by eliminating local anomalous data, under non-IID conditions, small category data on the client may be misidentified as anomalous data, leading to the loss of information for small category samples and making it impossible to predict such data accurately. Table I summarizes the above methods from the perspective of adapting to non-IID, adapting to data anomalies(abbreviated as AD), and adapting to the coexistence of non-IID and data anomalies. Among them, the methods based on robust aggregation are also divided according to whether there is a problem of information loss in small sample data.

TABLE I. Summary of Each Method

	Approaches	non-IID	AD	non-IID+AD	RSSI
Client-level	Fedprox [9], FedAlign [10], MOON [17], FedProc [18], FedPCL [19], MAFL [20]	✓	label noise		
	FedLN [21], [22]	✓			
Aggregation-based	RFA [15], Residual [16], FedNS [23]	✓	✓	✓	✓
	trimmed mean [11], Median [12], Krum [24], Norm [13]	✓	✓	✓	
Feature-based	VHL [6], FedCA [7], CCVR [25], Astraea [8]	✓			

[1] Note: ✓ indicates the scenario that the method is suitable for.
[2] Note: RSSI indicates that small sample data information is retained.

III. Preliminaries

In this section, we first formulate the question of the robustness of federated learning to data anomalies under non-IID conditions and then introduce the foundation of FedCM.

A. Problem Formulation

In federated learning, a classification task is implemented. Under non-IID, assume that there are n clients in total, and the sample data x and the true label y on client i are both distributed under distribution D_i, and D_i are different for each client. Let the dataset of client i be S_i, the model be f, and the model parameter be θ. Then, the model's prediction value for the sample data is $p = f(x; \theta)$. The main goal of training θ in traditional federated learning FedAvg (the default aggregation method in this article) is:

$$\min_{\theta} L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|S_i|}\sum_{x\in S_i} L(f(x;\theta), y) \tag{1}$$

Considering that the client's local private data may be attacked and thus generate abnormal data, the local model trained using abnormal data may cause the global model to be contaminated, thereby decreasing the accuracy, as shown in Fig. 1.

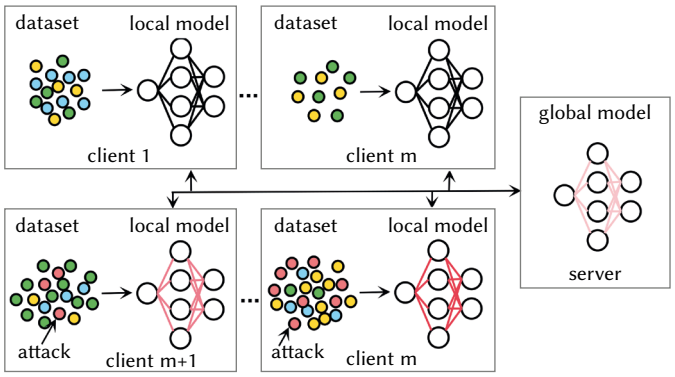


Fig. 1. In federated learning with non-IID, the client is attacked and thus generates abnormal data.

In Fig. 1, the red dots represent abnormal data generated by the attack, and the dots of other colors represent different categories of data. The red straight line indicates that the abnormal data affects the model. The color depth suggests the degree of noise in the model, *i.e.*, the degree to which the abnormal data affects the model. The darker the color, the more significant the impact.

Existing methods mainly reduce the impact of abnormal models on the global model by adjusting the weight of abnormal local models during aggregation or eliminating abnormal updates on the server side. Let the benign client subset be S_n, the attacked client subset be S_a, the robust global aggregation function be g, and the local model parameter received from client i ($i \in S_n \cup S_a$) in round t be θ_i^t. Then the aggregation weight assigned to client i is $w_i^t = g(\theta^t, I)$, and the parameters of the updated global model are:

$$\theta^{t+1} = \sum_{i=1}^{n} g(\theta^t, i) \times \theta_i^t \tag{2}$$

This reduces the impact of local updates obtained from abnormal data training on model accuracy.

Considering that the robust aggregation method on the server side ignores that the proportion of attacked clients is relatively large and may not be suitable for non-IID, the FedCM proposed in this article aims to make the model more robust to abnormal data and have better accuracy by adjusting the local model training process under non-IID.

B. Contrastive Learning

Moco [35] mainly uses momentum encoder and queue storage to construct a large number of negative samples and maintain the stability of feature representation. The query feature $q = f_q(x_q)$ is extracted using the main encoder $f_q(\cdot)$, and the key feature $k = f_k(x_k)$ is extracted using the momentum encoder $f_k(\cdot)$. The contrast loss InfoNCE is:

$$L_q = -\log\frac{\exp(q \cdot k^+/\tau)}{\sum_{i=0}^{K}\exp(q \cdot k_i/\tau)} \tag{3}$$

Where, x_q is the query, x_k is the key, k^+ is the corresponding positive sample (i.e., different perspectives of the same image), k_i is the negative sample taken from the queue, and τ is the temperature hyperparameter used to control the distribution of similarity. If the data used is image data, q and k^+ can be constructed by random changes such as cropping and color change.

The core idea of SimCLR [36] is to make the enhanced same images (positive samples) close in the feature space, while different images (negative samples) are far away.

Assuming a dataset D of size N, the contrast loss NT-Xent is:

$$L_i = -\log\frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N}\mathbf{1}_{[k\neq i]}\exp(\mathrm{sim}(z_i, z_k)/\tau)} \tag{4}$$

Where z_i and z_j are different enhanced views of the same image, τ is a temperature parameter used to control the softness/hardness of contrastive learning. Negative samples come from other samples in the mini-batch.

Contrastive learning (SupCon) [37] mainly improves the feature representation of the model by bringing the features of samples of the same type closer and the features of samples of different types farther apart. In a dataset, there are M categories $C_k(k = 1, 2, .., M)$ and a total number of samples N, then the contrast loss is:

$$L_{SupCon} = \sum_{i\in I}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(sim(z_i, z_p)/\tau)}{\sum_{a\in A(i)}\exp(sim(z_i, z_a)/\tau)} \tag{5}$$

Where I represents all samples, $P(i)$ represents samples of the same category as sample i, $A(i)$ represents all samples except sample i, $sim(z_i, z_j)$ represents the cosine similarity between feature z_i of sample i and feature z_j of sample j, and τ is the temperature parameter.

Since MoCo and SimCLR construct positive and negative samples through data augmentation, and SupCon directly uses category labels, it has better generalization ability and is suitable for scenarios where data are not independent and identically distributed, so SupCon is used for improvement.

C. Meta Learning

Meta-learning mainly trains models on a set of tasks and extracts experience from them to quickly adapt to new tasks and improve the model's performance in new tasks. A task distribution $p(\Gamma)$, task $\Gamma_i \sim p(\Gamma)$ is trained on a dataset D_i, and meta-learning obtains the model parameters θ by minimizing the following loss, making the model more adaptable to new tasks:

$$\min_{\theta}\sum_{\Gamma_i\sim p(\Gamma)} L_{\Gamma_i}(\theta_i, D_i) \tag{6}$$

MAML [40] mainly trains the initial parameters of the model so that it can achieve good generalization performance on the new task with only a small amount of gradient updates. In the meta-training phase, a batch of tasks is randomly selected from the task distribution p(Γ). For each task, the gradient $\nabla_\theta \mathcal{L}_{\Gamma_i}(f_\theta)$ of the model under the current parameters θ is first calculated, and then the adapted parameters are obtained by gradient descent $\theta'_i = \theta - \alpha\nabla_\theta \mathcal{L}_{\Gamma_i}(f_\theta)$. Then, based on the

performance of the adapted parameters on the task (i.e. $\mathcal{L}_{T_i}(f_{\theta_i'})$), the initial parameters are updated, and the update formula is:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta_i'}) \tag{7}$$

Where α is the step size of the gradient update within the task, and β is the step size of meta-learning.

IV. Proposed Method

A. Overview

We propose a robust federated learning framework, FedCM, to solve the problem of model performance degradation caused by data anomalies under non-IID conditions. The overall framework of FedCM, as shown in Fig. 2, mainly includes contrastive learning based on improved data features and meta-learning based on Gaussian noise model parameters.

Contrastive learning based on improved data features is used to improve model accuracy–using the data features of the local model of this round and the previous round, calculating their cosine similarity, weighting to obtain the improved features, and performing contrastive learning training on the enhanced features to get a local model with higher accuracy.

Meta-learning based on Gaussian noise model parameters is used to improve the robustness of the model to abnormal data– adding noise to the local data to obtain noise data, using the global model to adjust the model update direction, and using the noise data and the original data to get a noise-resistant local model through meta-learning training.

In the model training process of federated learning, first, the client receives the latest global model from the server as a local model. Then, the client uses FedCM to perform model training locally on the local dataset to obtain the local model. Finally, the client uploads the trained local model to the server for model aggregation to get the global model.

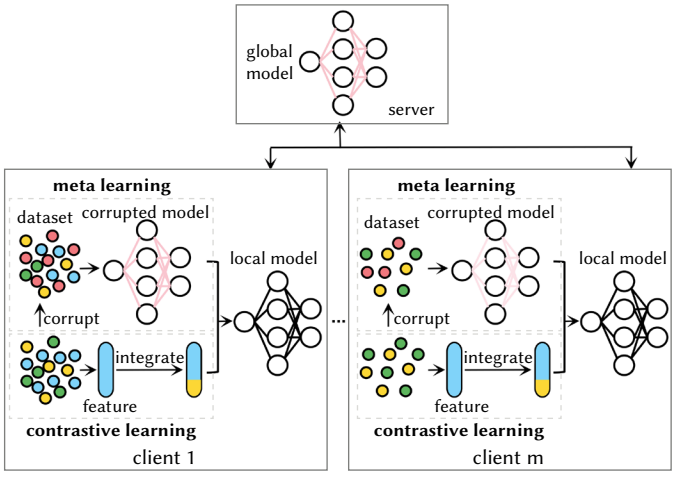


Fig. 2. Overall framework of FedCM.

B. Contrastive Learning Based on Improved Data Features

To further improve the model's accuracy, contrastive learning is employed to effectively enhance the similarity of feature representations for similar data while reducing the similarity of feature representations for data from different classes, thereby strengthening the model's accuracy.

On client i, for each sample $(x, y) \in S_i$, the model's backbone network f_{backbone} is used to obtain the feature $h = f_{\text{backbone}}(x; \theta_i)$, thus the contrast loss is obtained as:

$$L_{cl}(\theta_i) = \sum_{j \in S_i} \frac{-1}{|P(j)|} \sum_{k \in P(j)} \log \frac{\exp(sim(h_j, h_k)/\tau)}{\sum_{a \in A(j)} \exp(sim(h_j, h_a)/\tau)} \tag{8}$$

where $P(j)$ represents the samples of the same category as sample j, and $A(j)$ represents all samples except sample j. The temperature hyperparameter τ is used to adjust the "smoothness" of the distribution.

Considering that the global model is greatly affected by data anomalies, in contrast, the local model in the previous round is less affected by noise interference, the similarity $score = sim(h_{pre}, h_{cur}) = \frac{h_{pre} \cdot h_{cur}}{\|h_{pre}\| \times \|h_{cur}\|}$ between the feature h_{pre} obtained by the previous round model and the feature h_{cur} obtained by the current round model is used to calculate the proportional coefficient used to adjust feature:

$$\alpha = \begin{cases} 1 & , \text{if score} < 0 \\ 1 - \text{score}, & \text{if } 0 \leqslant \text{score} < \mu \\ \text{score} & , \text{if score} \geqslant \mu \end{cases} \tag{9}$$

In Equation(9), the contrastive similarity threshold is μ ($\mu = 0.5$), when $score < 0$, it means that the feature similarity is very low, which rarely occurs, and it means that the direction of the feature after the local model is updated seriously deviates from the direction of the feature of the previous round of model, so only the features of the local model are considered at this time; when $0 \leq score < \mu$, although the similarity is low, the current local model is mainly considered, so a larger proportional weight should be assigned to the local model; when $score \geq \mu$, the similarity is higher, but because the current local model is more affected by noise, it still needs to be corrected by the previous round of local model. Thus, the corrected features are obtained as follows:

$$\hat{h} = \alpha \times h_{\text{cur}} + (1 - \alpha) \times h_{\text{pre}} \tag{10}$$

Using the corrected features to get the contrast loss as:

$$L_{cl} = \sum_{j} \in S_i \frac{-1}{|P(j)|} \sum_{k \in P(j)} \log \frac{\exp(sim(\hat{h}_j, \hat{h}_k)/\tau)}{\sum_{a \in A(j)} \exp(sim(\hat{h}_j, \hat{h}_a)/\tau)} \tag{11}$$

Next, using this loss and the original cross-entropy loss, the basic loss is obtained as:

$$L = L_{cl} + L_{CE} \tag{12}$$

where $L_{CE} = \frac{1}{|S_i|} \sum_{(x,y) \in S_i} - \sum_{c=1}^{C} y_c \log(f_c(x; \theta_i))$ is the cross entropy loss. Gradient descent is used to update local model parameters. This improves the similarity of feature representations of similar data to a certain extent, making them more clustered, thereby improving the model's accuracy while maintaining a certain degree of robustness.

C. Meta-Learning Based on Gaussian Noise Model Parameters

In federated learning, abnormal clients are highly likely to cause deviations in the update direction of the global model, thereby reducing the model's accuracy. To enhance the noise resistance of federated learning, this research introduces meta-learning to learn noise features and build a model with stronger noise resistance. The core idea is to integrate noise features into the original model to obtain noise-resistant model parameters, thereby reducing the impact of noisy data on the model. The specific approach is as follows.

First, use the local model to learn the features of the noise data. Due to the randomness and unknown nature of the attack, this paper uses random Gaussian noise to add noise to the original data to simulate data poisoning attacks. The small batch in the local dataset D_i of client i is $B_i = \{(x, y)\}$, where x is the sample data and y is the true label. The Gaussian distribution is set to $\mathcal{N}(m, std)$, where m is the mean of the Gaussian distribution, which is set to 0 to avoid excessive deviation of

the data, causing the model to be too biased towards the noise direction and resulting in performance degradation. *std* is the standard deviation of the Gaussian distribution, which is set to a random number. After adding random Gaussian noise to the sample data x, the noisy small batch obtained is $\hat{B}_i = \{(\hat{x}, y) | \hat{x} = x + \mathcal{N}(m, std)\}$.

We train the original model $f(\theta_i)$ using the traditional cross entropy loss on a noisy mini-batch $\hat{B}_i$ and update the model parameters using stochastic gradient descent:

$$L_{\text{noise}} = \frac{1}{|\hat{B}_i|} \sum_{(\hat{x},y)\in\hat{B}_i} L_{CE}\left(f(\hat{x};\theta_i), y\right) \tag{13}$$

$$\hat{\theta}_i \leftarrow \theta_i - \eta \nabla_{\theta_i} L_{noise}(\theta_i) \tag{14}$$

where η is the learning rate and $\hat{\theta}_i$ is model parameter with noisy. Thus, the noisy local model $f(\hat{\theta}_i)$ is obtained. Then, a meta-update of the local model will be performed. The noisy local model will guide model training, thereby obtaining noise-resistant model parameters. Use the noisy local model $f(\hat{\theta}_i)$ to obtain a correct prediction y for the original data x, thereby optimizing the model parameters.

$$L_{\text{meta}} = \frac{1}{|B_i|} \sum_{(x,y)\in B_i} L_{CE}\left(f(x;\hat{\theta}_i), y\right) \tag{15}$$

Since the data is not independent and identically distributed, and adding noise may cause a certain degree of deviation in the model update's direction, correcting the noisy model parameters is necessary. This article uses the global model to fine-tune the noisy local model to improve the non-IID problem. The loss of the global model with the noisy local model is:

$$L_{\text{kd}} = \frac{1}{|B_i|} \sum_{(x,y)\in B_i} \text{KL}\left(f(x;\theta) || f(x;\theta_i)\right) \tag{16}$$

where $\text{KL}(p||q)$ is the KL divergence between probability distributions p and q, and $f(\theta)$ is the global model. Using the above loss, the greater the deviation between the noisy local model and the global model, the more penalty is added so that the deviation becomes smaller after the update; the non-IID problem is alleviated to a certain extent. Therefore, the total loss of the meta-learning part is:

$$L_m = L_{\text{meta}} + L_{kd} \tag{17}$$

As only the original model parameters are updated when the model parameters are updated, the influence of noise on the original model can be reduced.

D. Overall Algorithm

The FedCM method is mainly used to solve the problem of decreased model accuracy caused by data anomalies under non-IID. As shown in Algorithm 1, FedCM is mainly implemented through contrastive learning based on improved data features and meta-learning based on Gaussian noise model parameters. In contrastive learning based on enhanced data features, the similarity is calculated using the local model features of the previous round and the current round to obtain improved data features (lines 16-19 in Algorithm 1), and then the enhanced data features are used for contrastive learning (lines 20-21 in Algorithm 1) to improve the accuracy of the model; in meta-learning based on Gaussian noise model parameters, Gaussian noise is added to the original data, and then the noisy data and the original data are used for meta-learning to train the local model (lines 23-29 in Algorithm 1) to improve the robustness of the model to abnormal data.

In Algorithm 1, lines 1 to 11 show the model aggregation on the server side; lines 12 to 31 show the process of training the local model using FedCM. Among them, about the acquisition of S_a: in practical federated learning scenarios, malicious clients can be identified and detected using statistical information derived from robust aggregation methods. For instance, the Euclidean distance computed by the Krum algorithm [24], the median estimated by the Median algorithm [12], and the extreme values removed by the trimmed mean method [11] serve as indicators of potential abnormal clients. Clients that are consistently flagged and excluded across multiple training rounds can be classified as compromised, forming an estimated set of attacked clients S_a.

V. Experimental Results

In this section, we evaluate the effectiveness of the proposed method FedCM on multiple different datasets and different data pollution levels, compare it with various baseline methods, and analyze the impact of different components and different hyperparameters on the overall performance.

A. Experimental Settings

Data. In the experiment, three benchmark datasets for image classification were used: CIFAR10 [41], CIFAR100 [41], and SVHN [42] datasets. The CIFAR10 and CIFAR100 datasets contain 60,000 32x32 pixel images with 10 and 100 categories, respectively, of which 5,000 images are training sets and 1,000 images are test sets; the SVHN dataset includes 73,257 training set images and 26,032 test set images, with image pixels of 32x32 and a total of 10 categories. The Dirichlet distribution $D_r(N, \beta)$ is used to control the balance of data distribution. The degree of non-IID of the data is controlled by adjusting its parameter β (set to 0.1, 0.5, 1, and 10 respectively), where N is the total number of clients, the default value is 20, the higher the β value, the smaller the degree of non-IID and the closer to IID, and the default value is 0.5. And distribute the data divided by Dirichlet distribution to different clients.

Implementation details. The number of communication rounds is set to 100, and half of the clients are randomly selected to participate in training in each round of communication. In each local training, only one round of training is performed. Since the ResNet converges faster in the federated learning environment with different data distributions in image classification tasks, this paper uses ResNet as the basic network [43]. ResNet18 is used as the backbone network [44], and the SGD optimizer is used, with a learning rate, momentum, and weight decay of 0.01, 0.9, and 1e-5, respectively. The batch size of local training is 64. The temperature of the contrastive learning part is 0.07. Data augmentation, such as horizontal flipping, random cropping, and color jittering, is used during local training. Among the total clients, the proportion of attacked clients defaults to 0.2, *i.e.*, among 20 clients, there are four attacked clients. In each round of communication, clients are randomly selected to participate, so the situation of participating attacked clients is different. The default percentage of contaminated samples is 0.8 (i.e., 80 out of 100 samples are corrupted). The method proposed in [45] adds random noise to the client's original data. The degree and mode of damage of the random noise are random to achieve the uncertainty of abnormal data caused by the attack, which is closer to the actual situation. The intensity of the attack can be changed by changing the proportion of attacked clients or contaminated samples.

Baselines. A total of 9 baselines are compared: (1) FedAvg algorithm is not robust and has no defense; (2) Median calculates the median of local updates to obtain the aggregated global model instead of using the mean; (3) Trimmed-mean removes the maximum and minimum values of each dimension and then takes the average as the aggregated model parameter value; (4) Multi-krum algorithm uses the Euclidean distance to calculate the distance between updates and selects several local updates that are most similar to adjacent updates

Algorithm 1. FedCM algorithm

Input: initial model θ_0, model parameters $\theta_{i,t}$ of the i-th client in the t-th round, the dataset of client i is S_i, and E is number of epoch of local training , the subset of clients attacked is S_a.

On the server side:

for iteration t **do**
  Select the client S_t for this round of training
  for each client $i \in S_t$ **do**
    if client $i \in S_a$ **then**
      $\theta_{i,t} \leftarrow L_{CE}((x', y); \theta_{i,t})$ ▷ update model parameters by cross-entropy loss
    else
      $\theta_{i,t} \leftarrow$ ***ClientUpdateOfFedCM***$(\theta_{i,t})$
    end if
  end for
  $\theta_{t+1} = \sum_{i \in S_t} \theta_{i,t} \times |S_i| / \sum_{i \in S_t} |S_i|$ ▷ update the global model
end for

On the client side: **ClientUpdateOfFedCM**$(\theta_{i,t})$

for local epoch k for 1 to E **do**
  $B \leftarrow$ The local dataset is divided into batches of size $|B|$
  for each sample $(x, y) \in B$ do
    // Contrastive learning based on improved data features
    h_{pre}, h_{cur} ▷ get local features of the previous round and this round
    $score = dot\ (h_{pre} / |\ 0h_{pre}|\ 0, h_{cur} / |0h_{cur}|\ 0)$ ▷ calculate cosine similarity
    $\alpha \leftarrow$ Eq. 9 ▷ calculate feature ratio
    $h = \alpha \times h_{cur} + (1 - \alpha) \times h_{pre}$ ▷ get improved features
    $L_{cl} \leftarrow$ Eq.11 calculate contrast loss
    $L = L_{cl} + L_{CE}$ ▷ calculate loss
    //Meta-learning based on Gaussian noise model parameters:
    $x' \leftarrow x + g$ ▷ get noisy data
    $L_{noise}((x', y); \theta_{i,t}) \leftarrow$ Eq. 13 ▷ calculate the loss of the noisy model
    $\theta'_{i,t} \leftarrow \theta_{i,t} - \eta \nabla L_{noise}$ ▷ get noisy model parameters
    $L_{meta}((x, y); \theta'_{i,t})) \leftarrow$ Eq.15 ▷ calculate meta-loss
    $L_{kd}((x, y); \theta_t, \theta'_{i,t})) \leftarrow$ Eq.16 ▷ calculate the KL divergence loss of the global and local model
    $L_m = L_{meta} + L_{kd}$ ▷ total loss of the meta-learning part
    $\theta_{i,t} \leftarrow \theta_{i,t} - \eta \nabla (L_m + L)$ ▷ update model parameters
  end for
end for

for aggregation;(5) Norm algorithm sets a threshold and aggregates local updates whose norm below the threshold; (6) Residual uses the median estimator to calculate the residual and uses the residual to calculate the aggregation weight of each local update for aggregation; (7) RFA uses the geometric median operation for aggregation; (8) FedProx uses the Euclidean distance between the global and the local model parameters as a regularization term on the client side and adds it to the training objective; (9) Moon mainly uses improved contrast loss to optimize local training. For all baselines, the same experimental settings as the method in this paper are used for evaluation. For the Residual algorithm, the confidence interval was set to 2.0, and the threshold was set to 2.0; for RFA, the smoothing parameter was set to 1e-6, and the maximum number of Weizfeld iterations was set to 100.

Evaluation. All methods are evaluated using the same model and experimental settings (e.g., number of clients, proportion of attacked clients, proportion of sample contamination, β of Dirichlet distribution, number of local model training rounds, total number of communications), and accuracy is used as the evaluation metric:

$$\text{Accuracy} = \frac{n_{\text{correct}}}{n_{\text{total}}} \tag{18}$$

Where n_{total} represents the total number of predicted samples on the test set, and n_{correct} represents the number of samples with the correct predicted category. The average accuracy of the last ten rounds on the test set is used as the final accuracy.

B. Accuracy Comparison Experiment

Under the default experimental settings (attacked client ratio ar = 0.2, contaminated sample ratio NR = 80%, non-IID ($\beta = 0.5$)), the accuracy of each method is compared and analyzed to evaluate the performance of the proposed method.

As shown in Table II, according to ANOVA, the FedCM method is significantly better than most baseline methods on cifar10 and cifar100, which shows that the FedCM method has better performance.

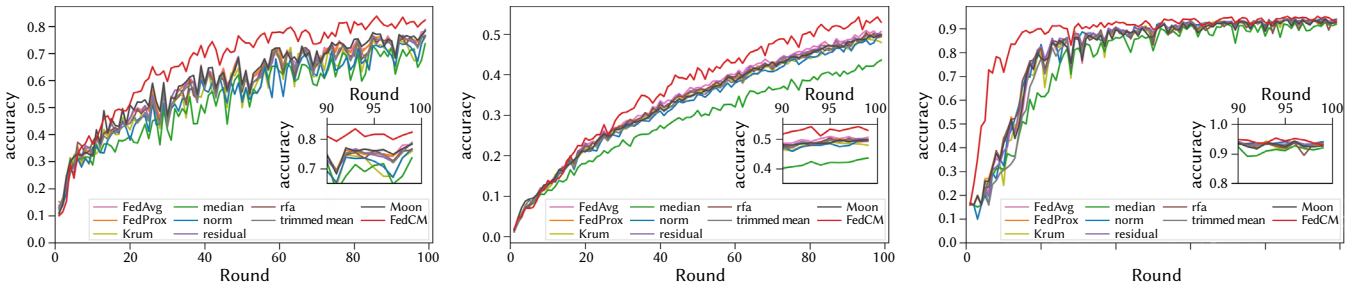
In Fig. 3, the last ten rounds are magnified for display.

As shown in Fig. 3(a) and (b), on the CIFAR10 and CIFAR100 datasets, the model accuracy of the FedCM method is higher than that of all the comparison methods, increasing by 8.2% and 6.1% respectively (the specific values can be seen in Table II, the data when $\beta = 0.5$), which shows that the FedCM method has better accuracy under non-IID and data anomalies.

TABLE II. COMPARISON OF THE ACCURACY OF EACH METHOD UNDER DIFFERENT NON-IID LEVELS

	Approaches	FedAvg	FedProx	Krum	Median	Norm	Residual	RFA	Trimmed Mean	MOON	FedCM
CIFAR10	β = 0.25	68.78	69.98	66.75	57.21	65.95	68.99	68.11	67.09	68.92	**77.55**
	β = 0.5	75.39	75.21	71.69*	68.88*	72.13*	74.81	74.49*	74.88	75.84	**81.61**
	β = 1.0	82.11	82.12	81.20	78.83	81.09	81.88	82.30	81.6	81.45	**84.56**
	β = 10	85.24	85.10	84.83	84.22	84.93	85.22	85.23	85.61	85.03	**87.93**
	β = 100	85.32	85.58	85.35	84.68	85.12	85.86	85.77	85.77	85.54	**88.17**
CIFAR100	β = 0.25	50.71	50.53	49.19	36.09	48.12	49.79	50.50	48.65	50.51	**53.02**
	β = 0.5	50.11**	49.32**	48.12**	42.17**	48.02**	49.41**	49.03**	49.09**	49.12**	**53.20**
	β = 1.0	51.72	51.47	50.44	47.17	51.03	51.59	51.61	51.62	51.55	**54.12**
	β = 10	53.8	53.45	53.05	50.00	53.22	53.87	54.22	53.67	53.54	**56.87**
	β = 100	53.65	53.61	53.19	50.95	53.24	54.11	53.86	53.76	53.75	**56.26**
SVHN	β = 0.25	89.68	90.55	88.83	84.29	87.41	90.4	89.08	89.04	89.29	**93.08**
	β = 0.5	93.05	93.04	92.96	91.14*	93.75	93.59	92.83	93.26	92.89	**94.30**
	β = 1.0	94.79	94.80	94.32	93.88	94.44	94.83	93.97	94.67	94.71	**95.81**
	β = 10	95.63	95.60	95.54	95.53	95.42	95.63	95.48	95.58	95.50	**96.39**
	β = 100	95.65	95.80	95.62	95.57	95.53	95.74	95.60	95.72	95.82	**96.53**

[1] Note: The highest accuracy is marked in bold, and the second highest accuracy is underlined.
[2] Note: "*" indicates the p-values of the ANOVA between FedCM and baseline p < 1e-5, "**" indicates p < 1e-6.



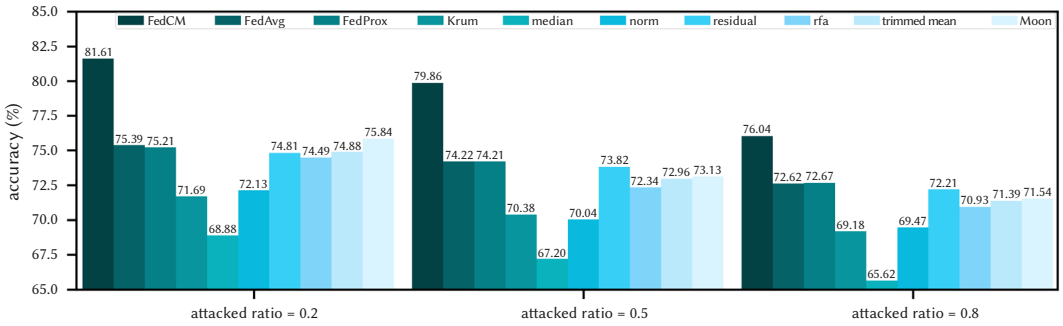
(a) CIFAR10 (b) CIFAR100 (c) SVHN

Fig. 3. Performance comparison of various methods under different datasets.

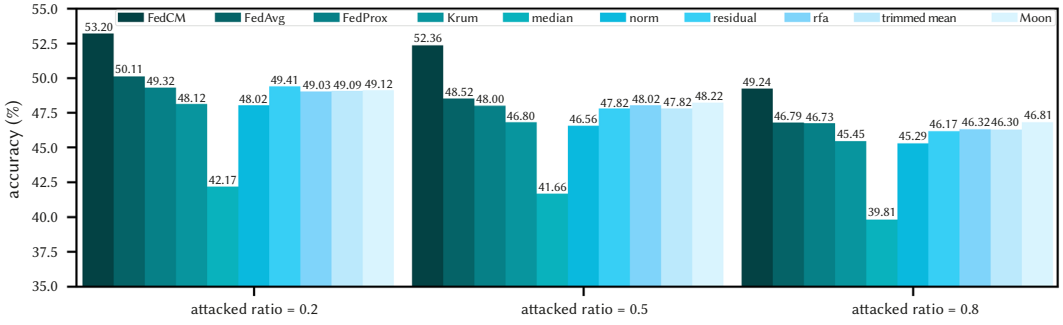
As shown in Fig. 3(c), on the SVHN dataset, although the accuracy of FedCM and the highest accuracy of the comparison method are similar, it can be seen that the accuracy of the FedCM method improves faster than other methods in the first ten rounds and is close to convergence in the 15th round. In contrast, other methods are close to convergence in the 23rd round. The number of training rounds required for convergence was reduced by 34.78%. FedCM demonstrates a clear advantage in convergence efficiency. Due to the use of contrastive learning in the FedCM method, the previous round's local model is employed to enhance feature representations, reducing interference from noisy information and minimizing the model's susceptibility to noise. As a result, the model's categories predictions are more accurate, leading to minor fluctuations in FedCM during the first twenty rounds. This further indicates that contrastive learning based on improved data features is effective and maintains stability.

In summary, FedCM shows good accuracy and improves convergence speed under non-IID conditions with data anomalies.

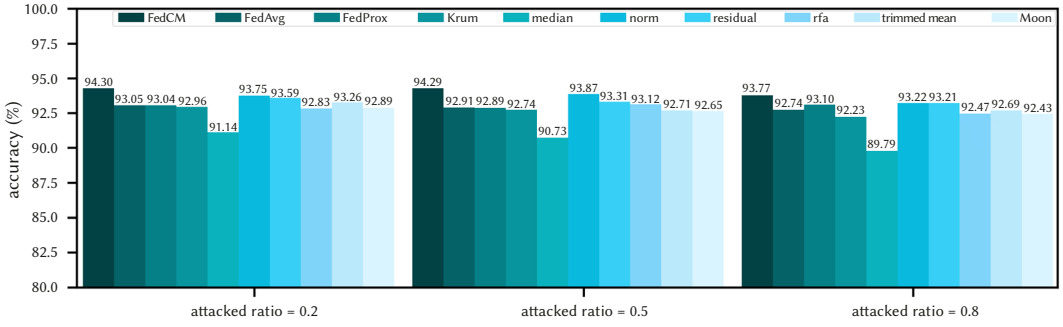
Accuracy under non-IID degree. As shown in Table II, FedCM has the best model accuracy under different non-IID degrees (β = 0.25, 0.5, 1, 10), indicating that FedCM still maintains good accuracy for data with high non-IID degree. When $\beta = 0.5$ changes to 0.25, the degree of decrease in FedCM's accuracy is lower than that of the comparison method, indicating that FedCM is less sensitive to the degree of non-IID. Since the meta-learning based on Gaussian noise model parameters in FedCM uses the global model to adjust the update direction of the anti-noise model, it can reduce the deviation of the local model from the global model when the local model is updated under non-IID so that a local model that is more in line with the update direction of the global model can be trained to improve generalization. As the degree of non-IID decreases, the accuracy of the FedCM method also increases, indicating that FedCM is suitable for both non-IID and IID. In the CIFAR10, CIFAR100, and SVHN datasets, FedCM improves by at least 2.7%, 4.5%, and 0.58%, respectively, compared with the comparison methods, indicating that FedCM has better model accuracy at different degrees of non-IID. As β increases, the degree of non-IID decreases, and the data distributions of each client become more similar. As shown in Table II, the accuracy of the model using the method in this paper basically increases as the beta value increases. This is because when the data distribution is more similar, the contrastive learning part aggregates the characteristics of similar data better, and the meta-learning learns more noise knowledge that fits the attacked data, thereby better correcting the model update direction and increasing the model accuracy.



(a) CIFAR10



(b) CIFAR100



(c) SVHN

Fig. 4. Performance comparison of various methods under different attacked ratio.

C. Robustness Analysis

Experiments are conducted under different experimental parameter settings to evaluate the robustness of the proposed method. It mainly includes two parameters: (1) the proportion of attacked clients AR and (2) the proportion of sample contamination NR.

Robustness to the proportion of attacked clients. As shown in Fig. 4, FedCM has the best average model accuracy under different proportions of attacked clients (AR = 0.2, 0.5, 0.8) compared with the comparison methods. In the non-IID case ($\beta = 0.5$), on CIFAR10, when the proportion of attacked clients is 0.2, 0.5, and 0.8, FedCM improves the average accuracy of the comparison methods by 8.2%, 7.9%, and 4.6%, respectively. And under different proportions of attacked clients, on the CIFAR10, CIFAR100, and SVHN datasets, FedCM improves the average accuracy of the comparison methods by at least 4.6%, 5.2%, and 0.45%, respectively. These indicate that FedCM is more robust regarding the proportion of attacked clients and has better model accuracy. As the number of attacked clients increases, the information of malicious updates in the model increases, while the direction of model updates shifts. Some methods aggregated by extracting partial updates may cause this situation in which the number of attacked clients participating in certain rounds is significant. Thus, most aggregated model updates are noisy, resulting in model shifts. However, the meta-learning based on Gaussian noise model parameters in FedCM extracts noise-resistant model parameter information, thereby correcting the direction of the model update and improving model performance, further illustrating the effectiveness of the meta-learning based on Gaussian noise model parameters.

Robustness to the proportion of contaminated samples. As shown in Table III, FedCM has higher accuracy than the comparison methods on different data sets under different proportions of contaminated samples (NR=25%, 0.5, 0.8, 1). For example, under CIFAR100, FedCM improves the average accuracy of the Median method by at least 22.2%; under CIFAR10, CIFAR100, and SVHN data sets, the FedCM method improves the accuracy of the RFA method by at least 8.4%, 5.2%, and 0.98%, respectively. As the proportion of contaminated data increases, the feature information of the contaminated data increases, the model is more affected by the contaminated data, and the model is prone to deviation. Due to the randomness of the contaminated data and the non-IID data distribution ($\beta = 0.5$), the model accuracy of each method will fluctuate to a certain extent as the amount of contaminated sample data increases. Although the FedCM method has some fluctuations, its model accuracy is still higher than that of the comparison method. Since meta-learning based

TABLE III. PERFORMANCE COMPARISON OF VARIOUS METHODS UNDER DIFFERENT SAMPLE CONTAMINATION RATIOS

Approaches		FedAvg	FedProx	Krum	Median	Norm	Residual	RFA	Trimmed Mean	MOON	FedCM
CIFAR10	NR=25%	75.32	76.5	72.64	69.25	73.21	75.20	74.75	74.69	75.10	**81.07**
	NR=50%	75.34	75.09	71.99	68.63	72.38	75.70	73.53	74.85	75.79	**80.75**
	NR=80%	75.39	75.21	71.69	68.88	72.13	74.81	74.49	74.88	75.84	**81.61**
	NR=100%	74.67	75.62	70.87	68.51	72.24	74.91	74.24	74.46	76.24	**81.13**
CIFAR100	NR=25%	50.71	50.26	48.69	43.27	48.73	50.34	50.25	49.26	49.68	**52.90**
	NR=50%	49.77	49.55	48.77	42.56	48.84	49.43	49.61	49.12	49.15	**52.86**
	NR=80%	50.11	49.32	48.12	42.17	48.02	49.41	49.03	49.09	49.12	**53.20**
	NR=100%	49.69	49.02	48.04	42.26	47.91	48.92	49.58	48.19	48.85	**53.18**
SVHN	NR=25%	93.30	93.35	92.86	90.93	93.94	93.57	93.34	93.18	92.72	**94.26**
	NR=50%	93.11	93.13	93.00	90.76	93.92	93.72	93.15	93.22	92.49	**94.67**
	NR=80%	93.05	93.04	92.96	91.14	93.75	93.59	92.83	93.26	92.89	**94.30**
	NR=100%	93.23	93.28	92.75	90.80	93.74	93.45	92.93	93.01	92.60	**94.19**

[1] Note: The highest accuracy is marked in bold, and the second highest accuracy is underlined.

on Gaussian noise model parameters in the FedCM method introduces noise for learning, the model can predict the contaminated data well, and using the global model for correction reduces the possibility and degree of model deviation caused by the introduced noise so that the model update direction is more accurate, showing that FedCM is robust in the proportion of contaminated samples.

TABLE IV. PERFORMANCE COMPARISON OF FEDCM WITH AND WITHOUT ABNORMAL UPDATES REMOVED ON THE SERVER SIDE UNDER THE DEFAULT EXPERIMENTAL SETTINGS

Scenario	CIFAR10	CIFAR100	SVHN
with-a-r	79.94	51.55	93.31
without-a-r	81.61	53.20	94.30

[1] Note: "with-a-r" means that the model of the attacked client is excluded when the model is aggregated on the server side; "without-a-r" means that the model of the attacked client is not excluded when the model is aggregated on the server side.

In addition, as shown in Table IV: On the cifar10, cifar100, and svhn datasets, compared with the case where abnormal updates were removed on the server side, the accuracy of the model using FedCM increased by 2.09%, 3.2%, and 1.06% respectively when abnormal updates were not removed on the server side, indicating that the accuracy of the model that processes the attacked clients is better than that of the model that directly excludes them. Because in the attacked clients, there is some data that has not been attacked, which can provide effective data information to help the model improve its accuracy. Especially in the case of heterogeneous data, if some categories only exist in the attacked clients, directly removing these clients will cause the model to be unable to learn the information of this category, affecting the overall performance. Therefore, it further illustrates the rationality of the method in this paper to use meta-learning to learn noise knowledge.

The FedCM method has good model accuracy and robustness under different proportions of attacked clients and contaminated samples.

D. Ablation Experiments

Under the default experimental settings, ablation experiments are conducted by splitting and combining components, mainly comparing the following four variants: **(1) No components (none)**: Only FedAVG is used for aggregation. **(2) Only meta-learning**: Only meta-learning based on Gaussian noise model parameters is used on the client. **(3) Only contrastive learning**: Only contrastive learning based on improved data features is used on the client. **(4) All components (full)**: The entire FedCM is used on the client, and FedAvg aggregation is used on the server.

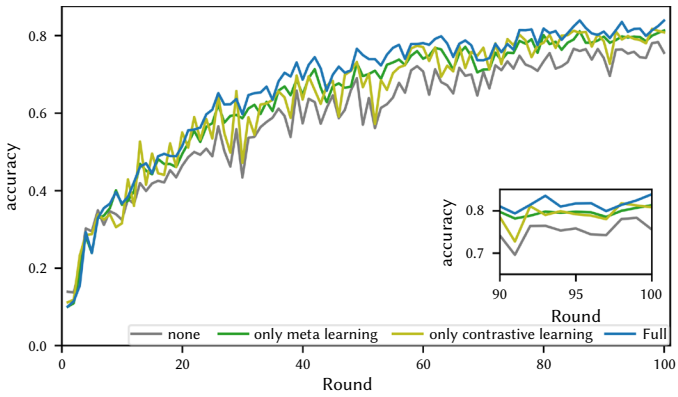
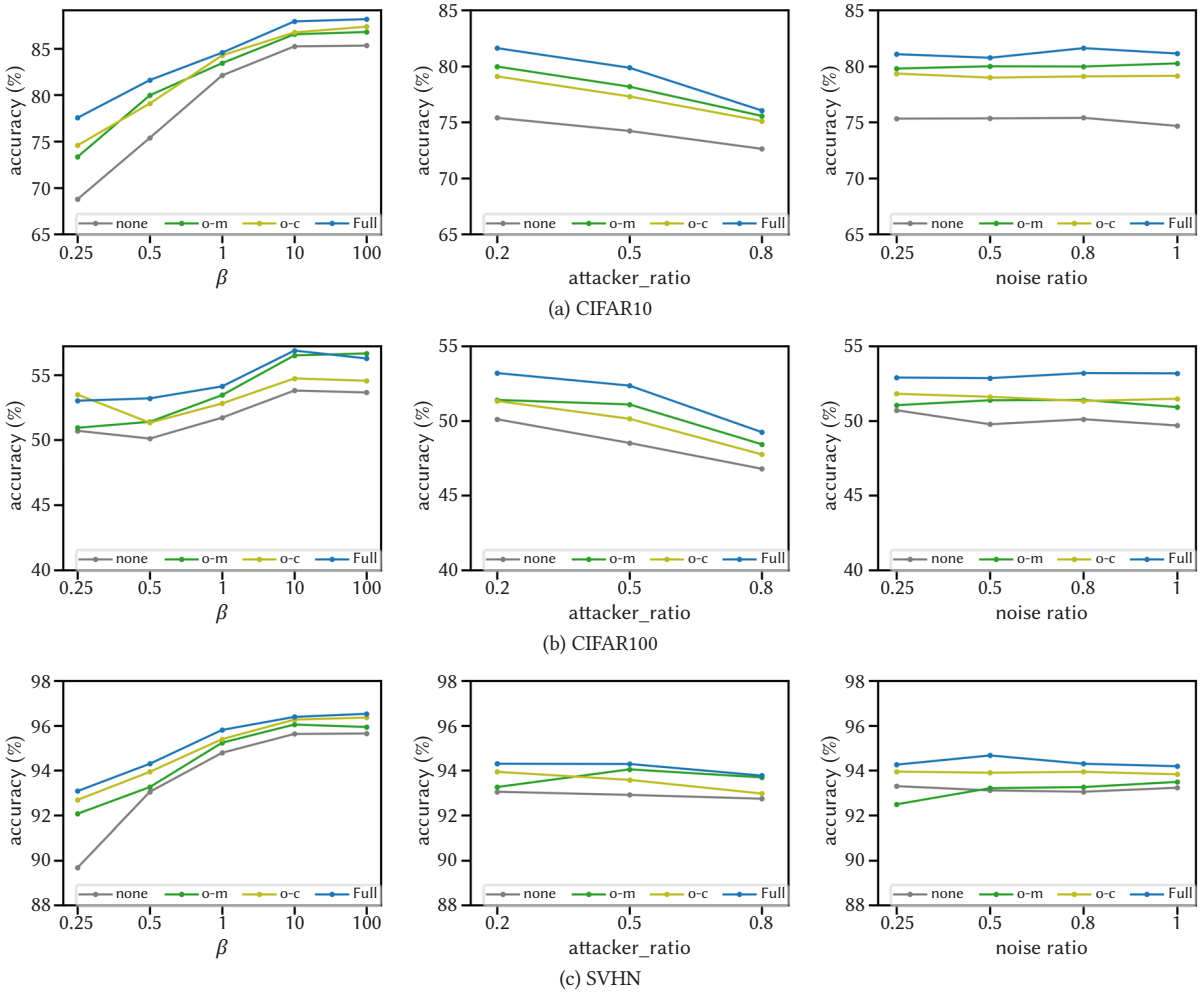


Fig. 5. Performance comparison of ablation experiments of FedCM on CIFAR10.

As shown in Fig. 5, the accuracies of the above four cases are 75.39, 79.59, 79.23, and 81.61, respectively. When only meta-learning is performed, compared with no components, the overall fluctuation is minor, and the accuracy is improved, indicating that meta-learning based on Gaussian noise model parameters maintains the stability of model performance and reduces the model's sensitivity to abnormal data. The model accuracy is improved when only contrastive learning is performed, compared with no components. Still, the fluctuation is significant, indicating that contrastive learning based on improved data features can improve the model's accuracy but is affected by noise data. When all components are present, the overall performance is improved compared with meta-learning alone, indicating that contrastive learning based on improved features can improve component performance; compared with only contrastive learning, the model accuracy is improved, and the fluctuation is slight, indicating that meta-learning can effectively improve the robustness of the



(a) CIFAR10

(b) CIFAR100

(c) SVHN

Fig. 6. Performance comparison of ablation experiments of FedCM.

model to data anomalies. Combining meta-learning and contrastive learning can improve the model's accuracy on abnormal data. Since directly using local features for contrastive learning will make abnormal information more prominent, contrastive learning based on improved data features uses the previous round of anti-noise model to improve features and slow down the extraction of noise features; it will smooth the performance improvement and make the model more stable. In short, when non-IID and there are data anomalies, the two components of FedCM proposed in this paper improve the model's accuracy to a certain extent and make the model more stable. In Fig. 6, none, o-m, o-c, Full indicate the above four cases in turn. As shown in Fig. 6, under most different beta, ar, and nr, the accuracy of the FedCM model is improved compared to using only the meta-learning and only the contrast learning part, indicating that the combination of the meta-learning part and the contrast learning part is effective in most of the cases.

E. Hyperparameter Sensitivity Analysis

Experiments were conducted under different hyperparameter settings to evaluate the sensitivity of the proposed method to hyperparameters. There are mainly three hyperparameters: (1) contrastive similarity threshold μ, (2) temperature of the contrast learning part τ , and (3) learning rate lr.

As shown in Fig. 7(a) and (b), as the contrastive similarity threshold or the temperature of the contrastive learning part increases, the accuracy of the model will fluctuate, but the change is small, which shows that the contrastive similarity threshold μ and the temperature hyperparameter of the contrast learning part τ have relatively little effect on the model performance. As shown in Fig. 7(c), when the learning rate is too large ($lr = 0.1$), it will lead to unstable model convergence and a decrease in model accuracy, which shows that the learning rate has a greater impact on model performance.

F. Scalability Analysis

Experiments were conducted under different client numbers (nc = 20, 50, 100, and training rounds were 100, 200, and 300, respectively) to evaluate the scalability of the proposed method.

As shown in Table V, as the number of clients increases, although FedCM decreases, its overall accuracy is higher than other methods. This is because as the number of clients increases, the client data becomes sparser, and the noise knowledge learned by meta-learning is closer to the noise knowledge of the data distribution. In addition, the contrastive learning part uses category information to perform loss calculation, which enhances the extraction of category features, thereby having better model performance. Therefore, FedCM has good scalability.

G. Time Complexity Analysis

The following is an analysis of the time complexity of the method in this paper. The time complexity of the model aggregation on the server side is $O(N + d)$, where N is the number of clients and d is the number of model parameters. When using FedCM for local model training, the time complexity of the contrastive learning part is $O(m \times b \times f)$, and the time complexity of the meta-learning part is $(m \times (f + d))$, so

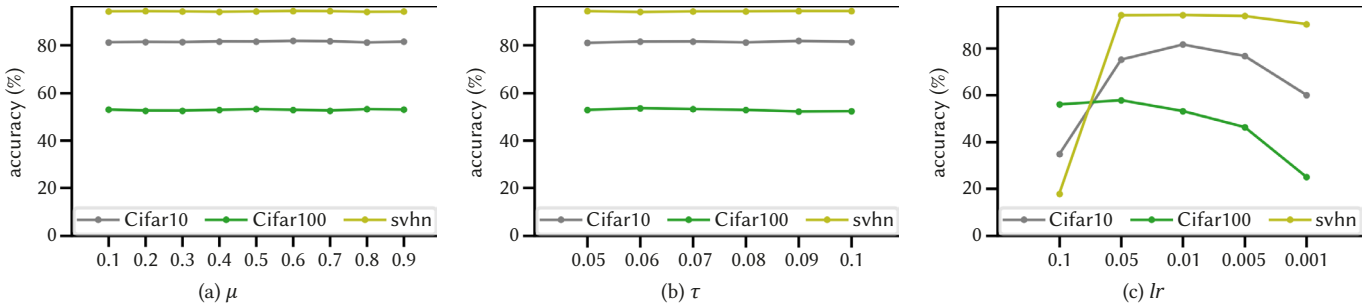


Fig. 7. Comparison of model accuracy of FedCM under different contrastive similarity threshold μ, temperature of the contrast learning part τ, and learning rate lr.

TABLE V. Performance Comparison of the Methods Under Different Client Numbers

Approaches		FedAvg	FedProx	Krum	Median	Norm	Residual	RFA	Trimmed Mean	MOON	FedCM
CIFAR10	NC=20	75.39	75.21	71.69	68.88	72.13	74.81	74.49	74.88	75.84	**81.61**
	NC=50	69.60	70.42	69.51	66.69	62.72	71.88	73.18	71.06	71.67	**80.47**
	NR=100	69.54	69.37	70.42	63.46	47.91	71.38	70.50	72.17	70.86	**78.64**
CIFAR100	NC=20	50.11	49.32	48.12	42.17	48.02	49.41	49.03	49.09	49.12	**53.20**
	NC=50	50.03	49.62	49.65	43.53	42.41	48.96	49.95	49.04	50.67	**56.32**
	NR=100	40.36	40.04	39.52	30.62	27.94	37.26	39.88	39.13	41.52	**48.70**
SVHN	NC=20	93.05	93.04	92.96	91.14	93.75	93.59	92.83	93.26	92.89	**94.30**
	NC=50	94.46	94.3	94.47	94.30	92.80	94.55	94.73	94.70	94.20	**95.78**
	NR=100	93.61	93.46	93.64	92.48	85.89	93.78	93.93	93.83	92.80	**94.95**

[1] Note: The highest accuracy is marked in bold, and the second highest accuracy is underlined.

the total time complexity of using FedCM for local model training is $O(E\times m\times(b\times f+f+d))$, where m is the local dataset size, f is the feature dimension, E is the number of local training rounds, b is the batch size, usually, $d >> f$ and $d >> b$, abbreviated as $O(E\times m\times d)$. Therefore, the total time complexity of each round of training is $O(N\times d)+O(E\times m\times d)$, where $E\times m>>N$, usually, abbreviated as $O(E\times m\times d)$. It can be seen that the computational overhead of the FedCM algorithm is mainly determined by the size of the local training dataset m and the number of model parameters d.

VI. Concluding Remarks and Future Work

Data anomalies under non-IID will lead to degraded model performance. Existing methods focus on robust aggregation on the server side, ignoring the proportion of malicious updates, and some methods have non-IID problems. This paper proposes a robust federated learning framework, FedCM, which has two parts: contrastive learning based on improved data features and meta-learning based on Gaussian noise model parameters. Experiments have shown that FedCM has higher model accuracy and faster convergence under different non-IID conditions and is robust regarding the proportion of attacked clients and the proportion of contaminated samples. Future research can explore how to achieve effective, robust model updates when the client situation is unknown and how to expand the proposed method to achieve practical applications in more abnormal situations and personalized federated learning.

Acknowledgment

We would like to thank all the teachers and classmates for their help in the process of writing the paper, as well as the reviewers for their in-depth review and revision suggestions. Finally, we would like to thank the project fund for its support. This work was supported in part by the Joint Key Project of National Natural Science Foundation of China under Grant U2468205, in part by the National Natural Science Foundation of China under Grant 62202156 and Grant 62472168; in part by the Hunan Provincial Key Research and Development Program under Grant 2023GK2001 and Grant 2024AQ2028; in part by the Hunan Provincial Natural Science Foundation of China under Grant 2024JJ6220; in part by the Research Foundation of Education Bureau of Hunan Province under Grant 23B0487.

References

[1] A. Abadi, B. Doyle, F. Gini, K. Guinamard, S. K. Murakonda, J. Liddell, *et al.*, "Starlit: Privacy-preserving federated learning to enhance financial fraud detection," *arXiv preprint arXiv:2401.10765*, 2024.

[2] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, G. Srivastava, "Federated-learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2021, doi: https://doi.org/10.1109/JIOT.2021.3077803.

[3] S. K. Das, N. R. Moparthi, S. Namasudra, R. González Crespo, D. Taniar, "A smart healthcare system using consumer electronics and federated learning to automatically diagnose diabetic foot ulcers," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 9, no. 2, pp. 5–17, 2025, doi: https://doi.org/10.9781/ijimai.2024.10.04.

[4] Z. Xu, R. Zhang, W. Liang, K.-C. Li, K. Gu, X. Li, *et al.*, "A privacy-preserving data aggregation protocol for internet of vehicles with federated learning," *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2024, doi: https://doi.org/10.1109/TIV.2024.3411313.

[5] W. Liang, J. Xiao, Y. Chen, C. Yang, K. Xie, K.-C. Li, B. Di Martino, "Tmhd: Twin-bridge scheduling of multi-heterogeneous dependent tasks for edge computing," *Future Generation Computer Systems*, vol. 158, pp. 60–72, 2024.

[6] Z. Tang, Y. Zhang, S. Shi, X. He, B. Han, X. Chu, "Virtual homogeneity learning: defending against data heterogeneity in federated learning,"

Proceedings of the 39th International Conference on Machine Learning, pp. 21111–21132, 2022.

[7] F. Zhang, K. Kuang, L. Chen, Z. You, T. Shen, J. Xiao, *et al.*, "Federated unsupervised representation learning," *Frontiers of Information Technology & Electronic Engineering*, vol. 24, no. 8, pp. 1181–1193, 2023, doi: https://doi.org/10.1631/FITEE.2200268.

[8] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, *et al.*, "Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications," *2019 IEEE 37th international conference on computer design (ICCD)*, pp. 246–254, 2019, doi: https://doi.org/10.1109/ICCD46524.2019.00038.

[9] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, Talwalkar, V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020, doi: https://doi.org/10.48550/arXiv.1812.06127.

[10] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, C. Chen, "Local learning matters: rethinking data heterogeneity in federated learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8397–8406, 2022.

[11] D. Yin, Y. Chen, R. Kannan, P. Bartlett, "Byzantine- robust distributed learning: towards optimal statistical rates," *International conference on machine learning*, pp. 5650–5659, 2018.

[12] C. Xie, O. Koyejo, I. Gupta, "Generalized byzantine- tolerant sgd," *arXiv preprint arXiv:1802.10116*, 2018.

[13] Z. Sun, P. Kairouz, A. T. Suresh, H. B. McMahan, "Can you really backdoor federated learning?," *arXiv preprint arXiv:1911.07963*, 2019.

[14] S. Zhou, K. Li, Y. Chen, C. Yang, W. Liang, A. Y. Zomaya, "Trustbcfl: mitigating data bias in iot through blockchain-enabled federated learning," *IEEE Internet of Things Journal*, vol. 11, no. 15, pp. 25648–25662, 2024, doi: https://doi.org/10.1109/JIOT.2024.3379363.

[15] K. Pillutla, S. M. Kakade, Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022, doi: https://doi.org/10.1109/TSP.2022.3153135.

[16] S. Fu, C. Xie, B. Li, Q. Chen, "Attack-resistant federated learning with residual-based reweighting," *arXiv preprint arXiv:1912.11464*, 2019.

[17] Q. Li, B. He, D. Song, "Model-contrastive federated learning," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 2021.

[18] X. Mu, Y. Shen, K. Cheng, X. Geng, J. Fu, T. Zhang, *et al.*, "Fedproc: Prototypical contrastive federated learning on non-IID data," *Future Generation Computer Systems*, vol. 143, pp. 93–104, 2023.

[19] Y. Tan, G. Long, J. Ma, L. Liu, T. Zhou, J. Jiang, "Federated learning from pre-trained models: A contrastive learning approach," *Advances in neural information processing systems*, vol. 35, pp. 19332–19344, 2022.

[20] T. Yoon, S. Shin, S. J. Hwang, E. Yang, "Fedmix: Approximation of mixup under mean augmented federated learning," *arXiv preprint arXiv:2107.00233*, 2021.

[21] V. Tsouvalas, A. Saeed, T. Ozcelebi, N. Meratnia, "Labeling chaos to learning harmony: Federated learning with noisy labels," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 2, pp. 1–26, 2024.

[22] J. Zhang, D. Lv, Q. Dai, F. Xin, F. Dong, "Noise- aware local model training mechanism for federated learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 4, pp. 1–22, 2023.

[23] H. Li, M. Funk, N. M. Gürel, A. Saeed, "Collaboratively learning federated models from noisy decentralized data," *2024 IEEE International Conference on Big Data (BigData)*, pp. 7879–7888, 2024, doi: https://doi.org/10.1109/BigData62323.2024.10825502.

[24] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, "Machine learning with adversaries: byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, pp. 119–129, 2017.

[25] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, J. Feng, "No fear of heterogeneity: classifier calibration for federated learning with non-IID data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5972– 5984, 2021.

[26] Z. Zhu, J. Hong, J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," *International conference on machine learning*, pp. 12878– 12889, 2021.

[27] D. Li, J. Wang, "Fedmd: heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[28] H. Wang, Y. Li, W. Xu, R. Li, Y. Zhan, Z. Zeng, "Dafkd: domain-aware federated knowledge distillation," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20412–20421, 2023.

[29] Y. Wu, S. Zhang, W. Yu, Y. Liu, Q. Gu, D. Zhou, *et al.*, "Personalized federated learning under mixture of distributions," *International Conference on Machine Learning*, pp. 37860–37879, 2023.

[30] J. Lu, H. Liu, R. Jia, J. Wang, L. Sun, S. Wan, "Toward personalized federated learning via group collaboration in iiot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 8, pp. 8923–8932, 2022, doi: https://doi.org/10.1109/TII.2022.3223234.

[31] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artificial intelligence and statistics*, vol. 54, pp. 1273–1282, 2017.

[32] Y. Lei, S. L. Wang, C. Su, T. F. Ng, "Oes-fed: a federated learning framework in vehicular network based on noise data filtering," *PeerJ Computer Science*, vol. 8, p. e1101, 2022, doi: https://doi.org/10.7717/peerj-cs.1101.

[33] T. Tuor, S. Wang, B. J. Ko, C. Liu, K. K. Leung, "Overcoming noisy and irrelevant data in federated learning," *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5020–5027, 2021, doi: https://doi.org/10.1109/ICPR48806.2021.9412599.

[34] S. Duan, C. Liu, Z. Cao, X. Jin, P. Han, "Fed-dr- filter: Using global data representation to reduce the impact of noisy labels on the performance of federated learning," *Future Generation Computer Systems*, vol. 137, pp. 336–348, 2022, doi: https://doi.org/10.1016/j.future.2022.07.013.

[35] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, "Momentum contrast for unsupervised visual representation learning," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

[36] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, "A simple framework for contrastive learning of visual representations," *International conference on machine learning*, pp. 1597–1607, 2020.

[37] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, *et al.*, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18661–18673, 2020.

[38] S. Ravi, H. Larochelle, "Optimization as a model for few-shot learning," *International conference on learning representations*, 2017.

[39] Y. Li, W. Liang, K. Xie, D. Zhang, S. Xie, K. Li, "Lightnestle: quick and accurate neural sequential tensor completion via meta learning," *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pp. 1–10, 2023, doi: https://doi.org/10.1109/INFOCOM53939.2023.10228967.

[40] C. Finn, P. Abbeel, S. Levine, "Model-agnostic meta- learning for fast adaptation of deep networks," *International conference on machine learning*, pp. 1126– 1135, 2017.

[41] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[42] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, *et al.*, "Reading digits in natural images with unsupervised feature learning," *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 2, p. 4, 2011.

[43] B. K. Alotaibi, F. A. Khan, Y. Qawqzeh, G. Jeon, D. Camacho, "Performance and communication cost of deep neural networks in federated learning environments: An empirical study," 2024, doi: https://doi.org/10.9781/ijimai.2024.12.001.

[44] S. Han, S. Park, F. Wu, S. Kim, C. Wu, X. Xie, *et al.*, "Fedx: Unsupervised federated learning with cross knowledge distillation," *European Conference on Computer Vision*, pp. 691–707, 2022, doi: https://doi.org/10.1007/978- 3-031-20056-4_40.

[45] D. Hendrycks, T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *arXiv preprint arXiv:1903.12261*, 2019.

Huan Zhang

Huan Zhang received bachelor's degree from Changsha University in 2019. She is currently pursuing a master's degree at Hunan University of Science and Technology. Her current research interests include artificial intelligence, network security, and anomaly detection.

Yuxiang Chen

Yuxiang Chen received the Ph.D. degree from Hunan University, Changsha, China, in 2021. He is currently an Assistant Professor with Hunan University of Science and Technology, Xiangtan, China. His research interests include network monitoring, network security, big data, and AI.

Kuanching Li

Kuanching Li is a Professor at the School of Computer Science and Engineering, Hunan University of Science and Technology. Dr. Li has co-authored over 150 conference and journal papers, holds several patents, and serves as an associate and guest editor for various scientific journals. He has also held chair positions at several prestigious international conferences. His research interests include cloud and edge computing, big data, and blockchain technologies. Dr. Li is a Fellow of the IET.

Yuhui Li

Yuhui Li is a first-year PhD student at The Hong Kong Polytechnic University. He received an M.Eng. degree from Hunan University in 2024. Prior to that, he received a B.Eng. degree from Shantou University in 2021. He has published several high-quality peer-reviewed papers in top journals and conferences, including IEEE TC, IEEE INFOCOM, IEEE TDSC, IEEE TITS, IEEE ICME, IEEE TCBB, and IEEE SCC/SSE. His research interests include network measurements, service computing, network security, and deep learning.

Sisi Zhou

Sisi Zhou received her Bachelor's and Master's degrees in 2009 and 2012, respectively, and currently pursuing a Ph.D. degree at Hunan University of Science and Technology. Her research interests include information security and privacy protection, with a focus on blockchain technology and federated learning.

Wei Liang

Wei Liang (Senior Member, IEEE) received the Ph.D. degree from Hunan University, Changsha, China, in 2013. He is a Postdoctoral Scholar with Lehigh University, Bethlehem, PA, USA, from 2014 to 2016. He is currently a Professor with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests include intelligent transportation, security of IoV, blockchain, embeded system and hardware IP protection, and security management in wireless sensor networks.

Aneta Poniszewska-Maranda

Aneta Poniszewska-Maranda received M.Sc. degree in computer science from Lodz University of Technology in 1998; PhD degree in computer science in 2003 from Universite of Artois in France; DSc degree in computer science from Czestochowa University of Technology, Poland in 2014.Her research interests include: software engineering, information systems security, analysis and design of information systems, multi-agent-based systems, cloud computing, internet of things, mobile security, blockchain, data analysis, machine learning, data processing, distributed systems, optimization. She has published more than 160 research papers in journals, conference proceedings and books. She is a reviewer in more than 40 research international journals, member of Editorial Board and Reviewer Boards of research journals and Chair, Vice-Chair and PC member of many international scientific conferences from all over the world. She is also the member of ACM, IEEE, AIS and INSTICC research organizations.