Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **10**, Number **4**, November **2025**

Towards Federated Learning with Noisy Labels via Mix-up Prediction and Contrastive Learning

Cuiwei Peng[1], Jun Ke[2], Zhengming Li[2], Huiwu Huang[2*], Jiahui Chen[2]

[1]School of International Education
Guangdong University of Technology, Guangzhou 510006, China
vivianoop3@gmail.com

[2]School of Computer Science and Technology
Guangdong University of Technology, Guangzhou 510006, China
3122004948@mail2.gdut.edu.cn, lizhengming2345@gmail.com,
hedy@gdut.edu.cn, csjhchen@gmail.com

*Corresponding author: Huiwu Huang

Received January 6, 2024, revised August 1, 2024, accepted May 21, 2025.

Abstract. *In the federated learning scenario, there are some noisy label samples in the local data due to inconsistent user labeling angles and other errors. To tackle the issue that the data of the local client is non-independent and identically distributed, and the samples have noise labels, we proposed a robust federated learning model based on hybrid prediction and contrastive learning. It expands the output difference between the original data of different categories, effectively regularizes the local training process, and improves the model's accuracy. Firstly, the original data is augmented. Secondly, the Mix-up method is used to generate the mixed prediction for the original and augmented data, and the sharpening operation is used to enhance the regularization of the mixed prediction. Then, the contrast loss is used to perform contras learning on the original data, augmented data, and different categories of data, reducing the output difference between the original data and augmented data and increasing the accuracy and generalization ability of the model under non-independent and identical distribution data. Finally, the experimental results show that the local model training method can improve the model's accuracy in the case of non-independent and identically distributed data and noisy labels in the samples.*
Keywords: Federated Learning, Noisy Label, Non-independent data, Identically distributed data

1. **Introduction.** As a distributed machine learning technique, federated learning [1] has emerged as a viable approach to solving data islands. However, in federated learning, noise labels exist in local datasets due to inconsistent user annotations and high annotation costs in some domains. In tasks like image classification, when models are locally trained on datasets containing noisy label samples, they tend to fit incorrectly to these noisy labels, leading to misguided updates and decreased model performance. Consequently, employing the trained model for classification tasks results in predictions deviating from the true labels during model inference. Furthermore, in federated learning scenarios, the data samples across different local clients are not identically distributed; their local datasets exhibit non-independent and non-identically distributed characteristics such as class imbalances or quantity disparities, further impacting the accuracy of the global model. Currently, most research efforts on mitigating the impact of noisy label samples on model training accuracy focus on centralized training paradigms. Noisy label

learning aims to train deep neural networks that are robust to noisy labels. Researchers have proposed various approaches to address this issue in centralized learning environments, which can be broadly categorized into five classes [2]: robust network architectures [3][4], robust regularization methods [5][6], robust loss function designs [7][8], loss value correction methods [9][10], and sample selection methods [11].

However, most existing methods cannot be directly applied due to the limited resources of local client devices. For instance, approaches such as [80,81] require complex and computationally intensive procedures to filter out noisy labels, which are challenging to execute on resource-constrained local client devices. Additionally, due to data privacy concerns, prior information regarding the dataset cannot be obtained, and the relatively smaller and non-independent and identically distributed (non-IID) nature of data on local client devices in federated learning complicates direct noise label filtering through sample selection, leading to a reduction in available information and hindering model training. These reasons render centralized training paradigms impractical for federated learning. In federated learning, most existing methods [12, 13, 14] attempt to introduce a clean labeled benchmark dataset on the server to estimate the noise level of local clients or select samples more likely to be correctly labeled for local training. However, implementing these methods poses two problems. Firstly, maintaining such a perfectly labeled, task-relevant auxiliary dataset on the server is impractical and comes with potential risks of data bias. Secondly, simply discarding samples with noisy labels would discard crucial information about the data distribution [10]. Recently, Xu et al. [15] proposed a multi-stage label correction framework to utilize local client data, including noisy labels. However, this approach heavily relies on a small subset of completely clean local clients. It lacks flexibility, making it challenging to implement in dynamically changing federated scenarios in the real world.

1.1. **Motivation.** To investigate the impact of noisy labels on federated learning, this section conducts experiments based on FedAvg at different noise levels, testing the model accuracy under clean data samples (no noisy labels) or symmetric noisy label scenarios using data augmentation methods. The basic experimental setup is as follows: 5 out of 100 clients are selected to participate in federated learning, with a local epoch of 5, a batch size of 60, a learning rate of 0.15, and a nine-layer convolutional neural network model. The experiments utilize the SGD optimizer with a momentum of 0.9 and weight decay of 0.0001.

As shown in Table 1 and Figure 1, it can be observed that in the original data domain with noise, the performance of the global model initially increases, then gradually decreases. This phenomenon persists in many other noise settings, indicating the presence of memory effects in deep networks influenced by negative knowledge acquired from data with noisy labels. Data augmentation can improve the model's robustness to some extent in the presence of noisy label data. In Figure 4-1, FedAvg converges the fastest on the original dataset without noisy labels, while models trained on datasets with noise labels and 30-degree random rotation data augmentation converge slowest. When trained on data with 40% symmetric noise, applying random rotation data augmentation within 30 degrees shows better robustness to noisy labels. On the Fashion-MNIST dataset, the warm-up period t_w is approximately 20 rounds (considering both the original and augmented domains). This suggests that data augmentation methods can combat noisy labels and enhance model robustness.

1.2. **Our Contribution.** We propose a noise-robust federated learning method based on mixed prediction and contrastive learning to enhance the performance of global models in

TABLE 1. Test accuracy on Fashion-MNIST datasets with various augmentation methods against symmetric noisy label

Dataset	**Fashion-**	**MNIST**
Noice	Yes	No
Raw data	0.72	0.92
Random horizontal flip	0.75	0.92
Random rotation [-5°, 5°]	0.74	0.91
Random rotation [-15°, 15°]	0.77	0.91
Random rotation [-30°, 30°]	0.78	0.90

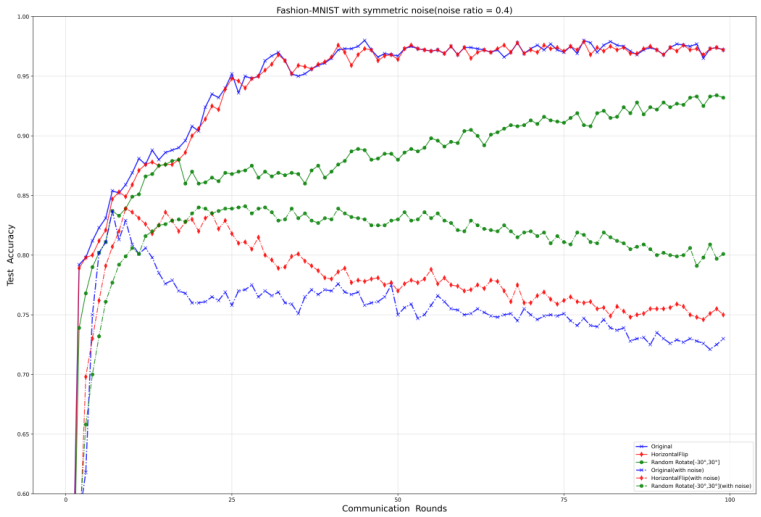

FIGURE 1. The accuracy of the original and application data enhanced Fashion-MNIST was tested under symmetric noise

federated learning scenarios with non-IID data and noisy label samples. Our contributions are as follows:

- We introduce a noise-robust federated learning method based on mixed prediction and contrastive learning, suitable for environments without auxiliary datasets, effectively addressing the presence of noisy label samples in datasets and mitigating the degradation of global model performance.
- We propose a noise-robust federated learning method based on mixed prediction and contrastive learning, capable of contrastive learning between original and augmented data, reducing the output discrepancy between the model on original and augmented data, thus improving model accuracy and generalization under non-IID data distribution.
- We conduct extensive experiments on two benchmark datasets and various noise levels. Compared to state-of-the-art methods, our proposed approach demonstrates superior performance, validating the effectiveness of our method.

1.3. **Organization.** The remaining sections are arranged as follows. In Section 3, we will explain some preliminaries for constructing the proposed model. In Section 4, we will describe the proposed model in detail. In Section 5, we will evaluate the proposed model experimentally. Finally, in Section 6, we will summarize the proposed model.

2. **Related Work.** To address the issue of non-independent and identically distributed (Non-IID) data, Zhao et al. [16] proposed a solution that involves creating a shared data

subset between the central server and local clients to balance the bias caused by Non-IID data. For medical data, the adaptive enhancement method LoadaBoost FedAVG [17] first optimizes local models with high cross-entropy and then sends gradients back to the central server to improve federated learning efficiency. Combining data augmentation methods, contrastive learning [18] leverages intrinsic relationships between data for feature learning, which can be applied to downstream tasks. Contrastive learning has since gained widespread attention. SimCLR [19], a notable method, includes an encoder and a projection head. It performs data augmentation on raw images to simulate input from various perspectives. Contrastive loss is then used to maximize the similarity of different data augmentations of the same target and minimize the similarity between different targets. For datasets with limited labels, it outperforms self-supervised learning on classification tasks on ImageNet. MoCo [20] treats image data as query vectors and key vectors, using an encoder and a momentum encoder to compute contrastive loss for learning feature representations. Grill et al. [21] addressed the need for negative samples in contrastive learning with two asymmetric networks. In their approach, one network learns from the output of another, requiring only positive samples during pre-training, which achieves better results and reduces the batch size impact on the model. MoCo-v2 [22] builds on MoCo by incorporating advantages from SimCLR in data augmentation and projection head design, optimizing performance while reducing computational costs. Zhang et al. proposed a supervised learning framework with contrastive loss constraints in the intermediate layers of the network, using multiple projection heads to connect and train the intermediate layers.

Conversely, MOON [23] is a straightforward and effective method combining federated learning with contrastive learning. By comparing the representations of local models with the global model and optimizing the consistency of these representations, MOON addresses the Non-IID data issue at the model level. Some federated learning optimization methods add regularization terms to reconstruct local loss functions, such as FedProx [24] and SCAFFOLD [25]. FedProx introduces a regularization term in local optimization, using L2-norm distance to directly constrain local updates, preventing extreme deviations of local models and ensuring consistency and stability in model updates. SCAFFOLD introduces control variates to correct local updates and address local client drift during local updates.

For the issue of data with noisy labels, Tuor et al. [14] proposed using a benchmark model trained on an auxiliary dataset as a sample selector. Each client then trains only on the samples selected by this model. Chen et al. [26] maintains an auxiliary dataset with correct labels on the server to measure the quality of local client data. Yang et al. [27] proposed measuring each client's noise ratio based on a clean validation dataset and using this for client selection. These approaches require the server to have a perfectly labeled auxiliary dataset. However, obtaining such a perfectly labeled auxiliary dataset in federated learning is impractical due to labelers' skill, bias, and hardware reliability. Simply discarding noisy samples or reducing the contribution of noisy clients can lead to losing important information about data distribution. Some methods do not require an auxiliary dataset. RobustFL [13] proposes maintaining consistent decision boundaries between clients and the server by exchanging class-level centroids, but this can lead to privacy leaks as centroids can infer original data information. FedCorr [15] introduces a multi-stage label correction framework to handle noisy labels in federated learning, but it relies on completely clean clients, which is unrealistic. Due to the dynamic nature of federated learning, client device connections are not always stable or ideal. However, supervision by a single network can introduce new noise into the training process.

3. Preliminaries.

3.1. Federated Learning.

Federated learning is first proposed by Konecný et al. [28]. The main idea of federated learning is to efficiently perform machine learning among multiple actors or computing nodes based on datasets distributed on multiple devices to optimize the target model while ensuring privacy in big data. Federated learning systems typically have two main components: a central server and various local client nodes. The central server's primary responsibilities include managing the global model, which involves issuing global training tasks, collecting local models from different parties, and aggregating them to update the global model. Each participating node is responsible for training the local model using its local dataset based on the global task.

The general steps of traditional federated learning are as follows:

A user initiates a federated training task, sets up the global model, and initializes its parameters. The central server then sends this information to the other participating nodes. Upon receiving the global model, the other nodes train it using their local datasets and upload the updated models to the central server after training. The central server aggregates the local models from all participating nodes in the current global round using an aggregation algorithm, thus updating the global model. The updated global model is then sent to all participating nodes as the initial global model for the next round. Steps (2), (3), and (4) are repeated until the pre-specified number of rounds is completed. Federated learning has the following characteristics: On one hand, training is conducted directly on the local datasets of the participating nodes rather than sending the datasets to the central server. On the other hand, during the interaction between local clients and the central server, local private data is not exchanged—only model parameters are uploaded and downloaded. These two aspects effectively ensure the privacy and security of user data.

3.2. Federated Learning with Non-IID Issue.

In a federated learning scenario, local data at each node is characterized by non-independent and identically distributed (Non-IID) problem [16]. There are two types of Non-IID data:

- Quantity Bias: In federated learning, the size of local datasets held by different nodes varies. Some nodes have significantly larger datasets than others, with differences potentially reaching tenfold or even hundredfold. This heterogeneity in data size is common in real-world applications.
- Distribution Bias: The distribution of labels differs between nodes, even if the sample distribution under the same label might be similar. For example, different hospitals may have varying distributions of disease categories they treat, depending on their specialization, such as general, gynecological, and psychiatric hospitals.

These scenarios represent common data distribution phenomena. The Non-IID nature of data across nodes results in significant differences between the local models trained by each node. If all local models are used for each round of global updating, it can mitigate the impact of Non-IID data but severely slows down the convergence of the global model, affecting efficiency. Conversely, if only a subset of nodes is selected for global updates, the randomly chosen nodes may not represent the overall data, leading to biases in the global model updates.

3.3. Federated Learning with Label Noise.

Label noise is a common problem in machine learning, especially for deep learning on large datasets [12].

3.4. **Notations and Problem Statement.** Consider a federated learning system comprising n local clients and 1 server. S Representing the set of N local clients, each local client contains a local dataset with a total of n_k data samples $D_k = \{(x_k^i, y_k^i)\}_{i=1}^{n_k}$, and the data is $D = \{D_k\}_{i=1}^{N}$, and is categorized into M classes, with some classes having samples containing noisy labels. During the communication round t between the server and local clients, a subset of local clients S_t is selected to receive the global model provided by the server. Based on the global model, the selected clients perform E epochs of local training. The selected clients download the global model w_t and treat it as their local model w_t^k, training it on their local dataset D_k to minimize the cross-entropy loss (CE Loss). The cross-entropy loss is typically defined as follows:

$$(L)_{CE} = CE\left(f_\theta\left(x\right), y\right).$$

Here, w_t^k is denoted as θ, where f_θ represents a deep neural network. For a data sample x, the model's predicted output is $f_\theta(x)$, with the corresponding label being y, which may potentially be a noisy label. After executing local training, each local client sends back the updated local model w_t^k to the server. Subsequently, the server aggregates the uploaded local models to generate a new global model w_{t+1} for participation in the next round of training.

Then, the problem statement of our work aims to learn a global model that performs well under non-IID data distribution scenarios and noisy label samples with the above notations.

Note that the phenomenon of memory effects in deep networks suggests that during training, deep network models tend to first fit the correctly labeled data, with samples having smaller losses more likely to be correctly labeled, before fitting the data with noisy labels. Hence, there exists a brief warm-up period t_w during model training, during which the model's performance sees an early improvement. Subsequently, the model develops basic learning and recognition capabilities. However, if given labels, some of which may be erroneous noisy labels, are directly used as supervision signals, the model tends to fit the labels to minimize the cross-entropy loss function. This results in the network model gradually memorizing the noisy labels, further leading to a decline in model performance in later stages. Therefore, we also need to avoid the phenomenon of overfitting to noisy labels.

4. **The Proposed Model.**

4.1. **Overview.** The approach primarily focuses on optimizing the local client's local training process by enhancing the confidence of model predictions through data augmentation, mixed prediction, and sharpening operations. This aims to prevent the training model from overfitting to noisy labels. Additionally, by utilizing contrastive learning, the method reduces the output discrepancy between original and augmented instances, expands the output discrepancy between original data and different class data, and mitigates the phenomenon of non-IID data.

Figure 2 illustrates the overall framework of the noise-robust federated learning method based on mixed prediction and contrastive learning.

Initially, the server initializes the global model and sends it to randomly selected local clients according to a preset ratio. Upon receiving the global model, each local client performs data augmentation on its original data, mixes the original and augmented data using the MixUp method, and enhances the confidence of the mixed prediction results through sharpening operations. The sharpened results are then input into the cross-entropy loss function to compute the cross-entropy loss value.

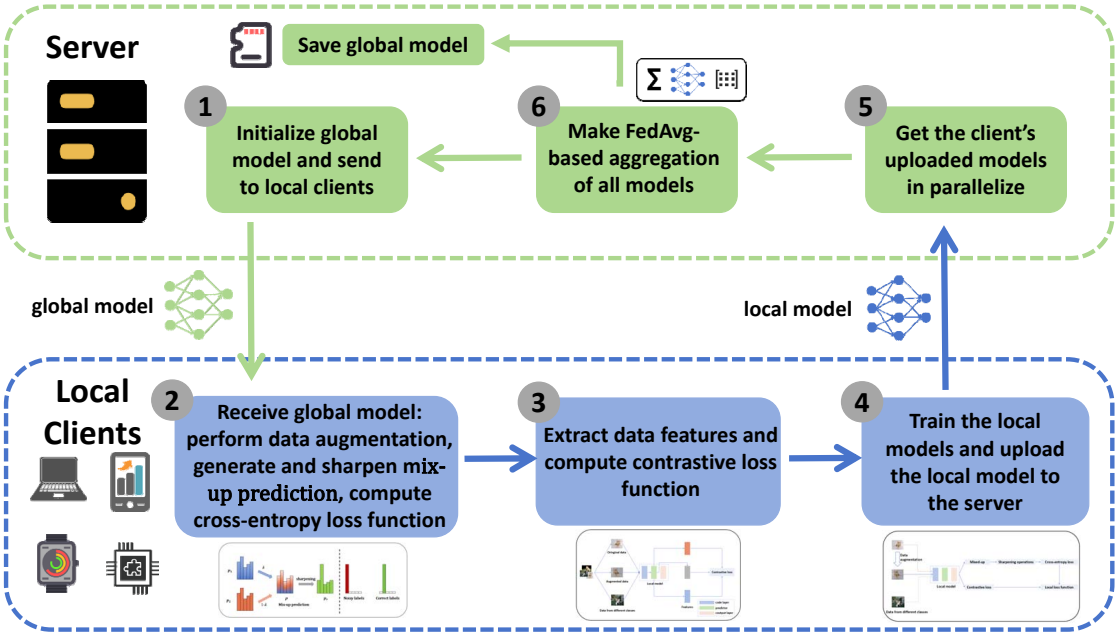

FIGURE 2. Framework of the proposed model

Subsequently, the original, augmented, and other classes' data are input into the network model. The extracted feature representations are then input into the contrastive loss function to compute the contrastive loss value. The local loss function consists of both the classical cross-entropy and contrastive loss functions. The local training process is illustrated in Figure 3.

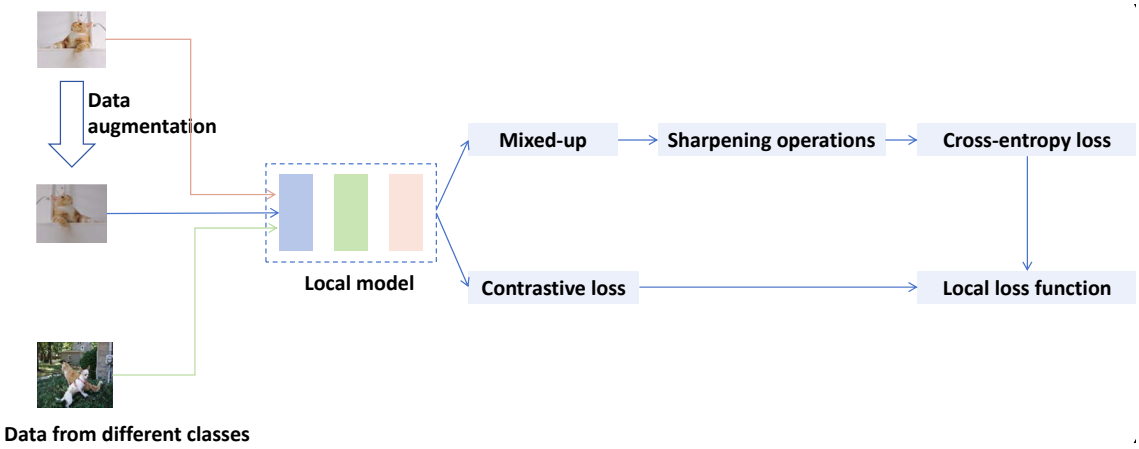

FIGURE 3. Workflow of the local models

The newly generated local models from selected participating local clients are uploaded to the central server. The server receives local models w_t^k uploaded by several local clients. Using the FedAvg approach, the server aggregates these local models w_t^k and generates the updated global model w_{t+1}.

4.2. **Mix-up Prediction.** In each round t, the selected local client k receives the global model $w_(t)$ and updates it as the local model w_t^k. Data augmentation can transform each data sample into two versions: the original datax_i and the augmented data x_aug. The original and augmented data are fed into the local model, producing two corresponding output logits, o_1 and o_2. By applying the Softmax function to o_1 and o_2, we obtain

the corresponding predictions p_1 and p_2. To utilize both predictions simultaneously, we randomly select a λ from a Beta distribution to mix $p_1$and p_2, resulting in a combined prediction p calculated with the following formulation.

$$p = \lambda * p_1 + (1 - \lambda) * p_2.$$

In MixMatch, a sharpening operation generates pseudo-labels for unlabeled data, and further mixed data augmentation is applied to the overall data for semi-supervised learning. The smoothness assumption in many semi-supervised learning methods suggests that the classifier's decision boundary should not pass through high-density regions of the marginal data distribution. Entropy minimization can help satisfy this assumption by adding a loss term to minimize the entropy of the model's predictions.

Inspired by this assumption, but differing from previous work, we apply a sharpening operation to the mixed prediction p to increase In MixMatch, a sharpening operation generates pseudo-labels for unlabeled data, and further mixed data augmentation is applied to the entire dataset for semi-supervised learning. The smoothness assumption in many semi-supervised learning methods indicates that the classifier's decision boundary should not pass through high-density regions of the marginal data distribution. Entropy minimization can help meet this assumption by adding a loss term to minimize the entropy of the model's predictions.

Inspired by this assumption, but differing from previous work, we apply a sharpening operation to the mixed prediction p to enhance the confidence of the model's predictions, resulting in the sharpened prediction p_s:

$$p_{s,i} = \text{Sharpen}(p, T)_i := \frac{p_i^{1/T}}{\sum_{j=1}^{M} p_j^{1/T}},$$

where i represents the i-th class, and T is the sharpening temperature. Next, p_s is used for the cross-entropy loss calculation, which can be expressed as:

$$L_{CE} = CE(p_s, y).$$

As shown in Figure 4, the algorithm proposed in this chapter directly uses the sharpened mixed prediction p_s for the cross-entropy loss calculation. The underlying motivation is to enforce the model to make more confident (lower entropy) predictions for each sample. The original model's output logits tend to fit the given labels, which may include erroneous noisy labels, to reduce the cross-entropy loss. Consequently, the network might form some incorrect recognition patterns. However, sharpened predictions can enhance the model's prediction confidence, compelling the model to make low-entropy predictions. This results in a higher cross-entropy loss compared to the unsharpened predictions, preventing the model from accumulating errors in the self-ensemble logic. Therefore, reducing the cross-entropy loss becomes more challenging, making it harder for the local model to memorize noisy labels. This approach acts as an implicit regularization method to avoid overfitting noisy labels.

4.3. **Contrastive Loss.** Considering the presence of non-independent and identically distributed (non-IID) data and noisy labels in federated learning, we leverage data augmentation and contrastive learning to address the non-IID issue. In our data augmentation approach, the original and augmented data are treated as positive pairs in contrastive learning, while the original data and other data from different classes are treated as negative pairs.

Our contrastive learning method is illustrated in Figure 5. Data samples are processed through the network model to obtain feature vectors. Using cosine similarity, we compute

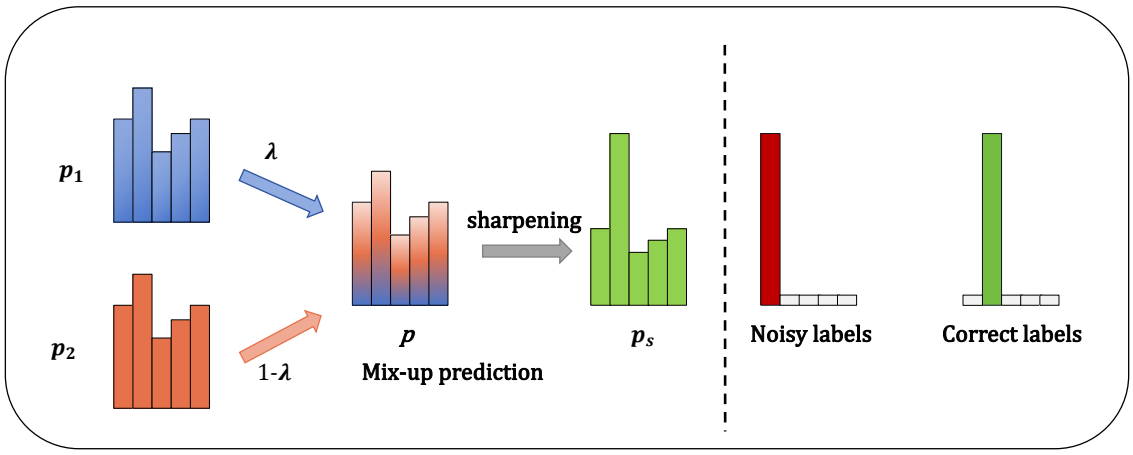

FIGURE 4. An intuitive understanding for Mix-up prediction

the distances between pairs of vectors. For the original data x_i and the augmented data x_{aug}, the loss function is defined as:

$$L_{con} = -\log \frac{\exp(\text{sim}(z(x_i), z(x_{aug}))/\tau)}{\sum_{j=1}^{N} 1[j \neq i] \exp(\text{sim}(z(x_i), z(x_j))/\tau)},$$

where τ is a preset temperature hyperparameter; $z(x_i)$, $z(x_{aug})$, and $z(x_j)$ are the feature vectors obtained by inputting the original data x_i, its augmented data x_{aug}, and different-class data x_j into the local model w_t^k at round t. N is the number of different-class data samples compared during training.

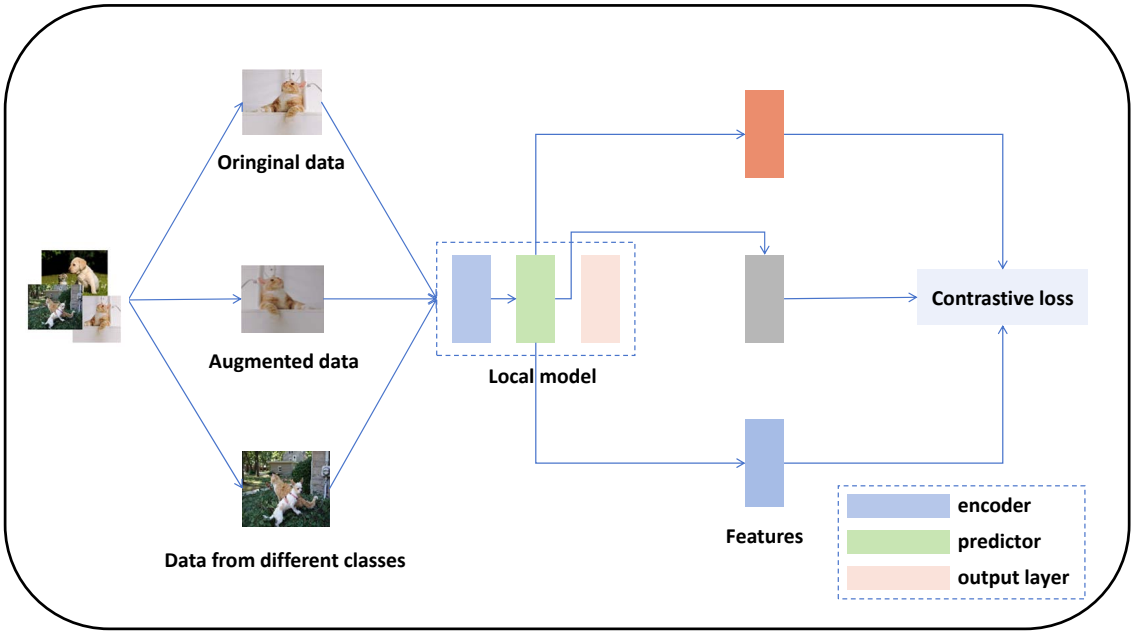

FIGURE 5. An intuitive understanding for contrastive learning

For an input data pair (x, y), the total loss function is defined as:

$$L = \mu L_{con} + L_{ce},$$

where L_{ce} denotes the classic cross-entropy loss function, and μ is a weight factor to adjust the importance of the contrastive loss function.

The contrastive loss function aims to tackle the non-IID data problem by reducing the output differences between the original and augmented data while increasing the output differences between data from different classes. This effectively regularizes the local training process, enhancing the model's convergence accuracy. Note that during the warm-up rounds t_w, the weight factor μ is linearly increased from 0 to its final value to prioritize fitting the primary task in the early learning stages, gradually optimizing the contrastive loss.

4.4. **Model Aggregation.** Our method follows the FedAvg protocol for model aggregation on the server side. The server aggregates the uploaded local model parameters as follows:

$$w_{(t+1)} = \sum_{k \in S_t} \frac{n_k}{n_{total}} w_t^k.$$

Here, $w_{(t+1)}$ represents the global model parameters for the next round $t+1$. w_t^k denotes the local model parameters from client k at round t, n_k is the number of local data samples for the selected local client k, and n_{total} is the total number of data samples across all selected local clients.

5. Experiment Results.

5.1. **Experimental Setting.** The basic experimental setup is as follows. The total number of local clients N is set to 100. Each local client's dataset contains an equal number of samples but from different classes. The number of local iterations (epochs, E) for each client is set to 5, with a local batch size of 60.

Experiments are conducted on the MNIST and Fashion-MNIST benchmark image datasets. Synthetic noisy labels are generated by replacing the original labels in the dataset with typical noise types (symmetric flipping). Experiments are performed with different noise ratio levels ϵ.

When processing data, the dataset is first divided into 10 parts based on the original data classes. An equal amount of noisy label samples is uniformly injected for each class. Each class uses a fixed random seed to inject noise according to the specified noise type and ratio for fair comparison. Subsequently, samples with noisy labels are distributed to clients based on the clients' true label distribution. In the non-IID setting, each local client has a small random subset of samples from different classes.

All programs are written in Python 3.8.0. The methods proposed in this chapter and other baselines are implemented using the PyTorch library (PyTorch 1.7.1). Experiments are conducted on a Linux (Ubuntu) server with an NVIDIA GeForce RTX 3090 Ti GPU (24GB RAM). For MNIST and Fashion-MNIST, a 9-layer convolutional neural network is used. The learning rate is set to 0.15. In each round, 5% of the local clients (out of 100 total clients) are selected to participate in training. The SGD optimizer with momentum 0.9 and weight decay of 10^{-4} is used for fair comparison. Unless otherwise specified, the accuracy reported in the tables is the average accuracy of the global model over the last 10 rounds.

We compared with the following baselines: FedAvg[17], Symmetric Cross Entropy (SCE) [7], FedLSR[29], and Robust Federated Learning (RobustFL) [13]. The main idea of SCE is to introduce a symmetric loss term, RCE, in addition to the cross-entropy loss. It highlights that relying solely on unreliable labels for cross-entropy loss can be flawed and uses model output logits to form RCE as additional supervision, effectively handling

TABLE 2. Coefficient μ selection

Noise Level	0.3	0.4	0.5
MNIST	0.15	0.20	0.25
Fashion-MNIST	0.15	0.20	0.25

label noise. RobustFL aims to gradually form global class feature centroids by collecting local client class feature centroids and performing entropy regularization to enhance model predictions, thereby achieving global supervision.

According to the relevant papers, we generally follow a consistent hyperparameter setup for each baseline. For the SCE algorithm evaluated on MNIST and Fashion-MNIST, α and β are fixed at 0.1 and 1, respectively. Note that for SCE, FedAvg aggregates the weights of the locally trained models on the server.

Hyperparameters: The sharpening temperature T follows the MixMatch setting and is set to 1/2. The warm-up rounds t_w for MNIST and Fashion-MNIST are set to 10 and 20 rounds, respectively. As the noise level increases, the corresponding coefficient μ is adjusted empirically, as shown in Table 2. For data augmentation, random rotations within 30 degrees are applied to MNIST and Fashion-MNIST.

5.2. **Result.** The main results are presented in Table 3, Figure 6, and Figure 7. The experimental results indicate that the method proposed in this chapter shows significant resistance to noisy labels under varying noise levels. Compared to the FedAvg method, which ignores the presence of noisy labels, other methods that address noisy labels (SCE, FedLSR, RobustFL) generally exhibit better performance. For the Fashion-MNIST dataset, when each client only has data from three classes, the proposed method outperforms the baseline algorithms in terms of accuracy. Additionally, Robust Federated Learning collects and transmits class feature centroids from client datasets to form overall class features as global supervision, guiding the local training process. This approach may lead to privacy leakage because these centroids can potentially be used to infer private information from the original data.

Overall, the method proposed in this chapter leverages additional reliable supervision and focuses on optimizing the local training process without transmitting extra sensitive information, thus further protecting data privacy.

TABLE 3. Accuracy on MNIST and Fashion-MNIST datasets with various noise levels

Dataset	Method	Noise Ratio 0.3		Noise Ratio 0.4		Noise Ratio 0.5	
		3 Classes	5 Classes	3 Classes	5 Classes	3 Classes	5 Classes
MNIST	FedAvg	60.68	71.36	52.25	79.96	45.98	50.63
	SCE	69.52	77.57	60.57	87.94	54.15	59.78
	FedLSR	96.02	98.50	92.58	98.72	94.54	98.09
	RobustFL	89.64	90.28	87.88	88.53	79.95	89.48
	Ours	**96.20**	**98.80**	**95.75**	**98.72**	**95.03**	**98.27**
F-MNIST	FedAvg	41.96	54.20	38.75	55.96	30.98	46.63
	SCE	50.46	64.97	47.07	64.13	40.05	55.20
	FedLSR	61.13	77.75	58.96	77.48	63.79	78.72
	RobustFL	63.04	79.57	61.30	77.45	58.58	70.90
	Ours	**66.91**	**81.03**	**71.33**	**83.07**	**69.87**	**82.58**

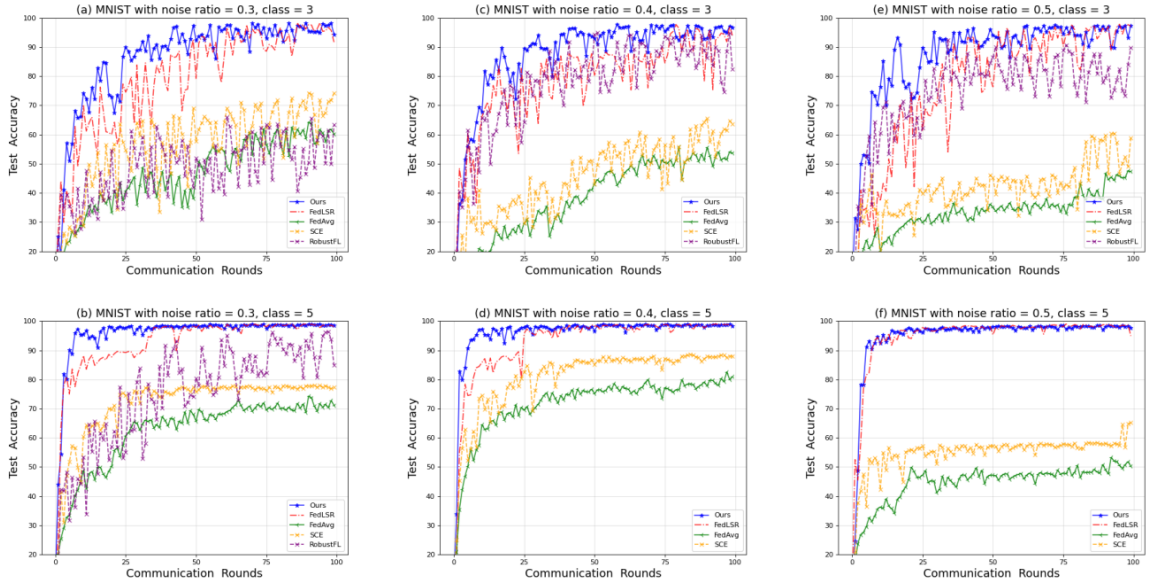

FIGURE 6. The accuracy over communication rounds on MNIST under symmetric noise

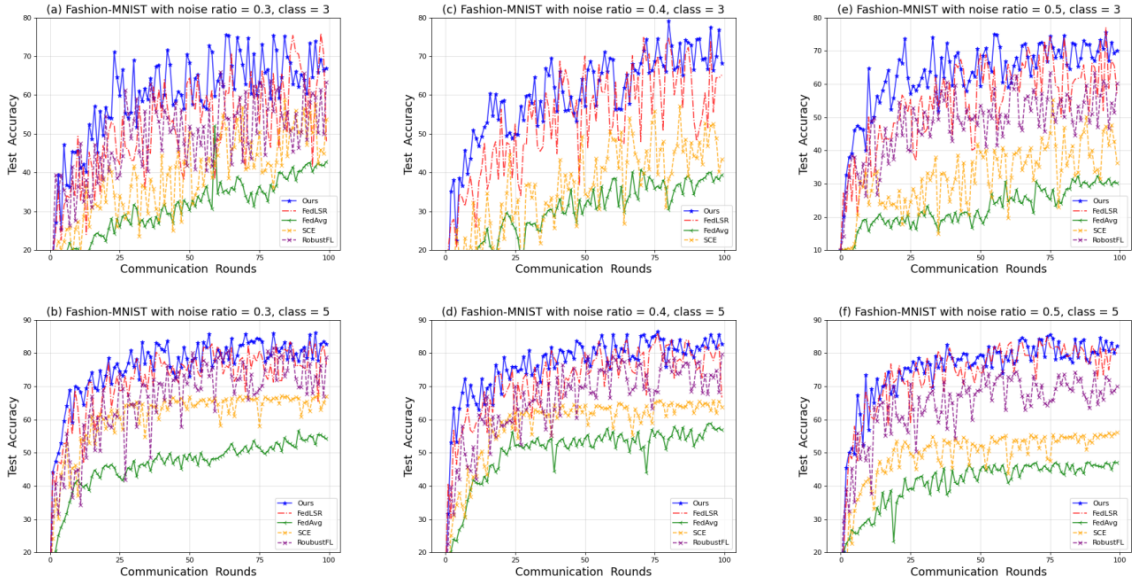

FIGURE 7. Evaluation for different batch size

5.3. **Ablation Study.** The proposed method consists of two main components. Mixed prediction implicitly regularizes training by enhancing model discrimination confidence to avoid overfitting noisy labels, while contrastive learning explicitly regularizes model output consistency at the instance level to address non-iid data problems. The effectiveness of each component was evaluated through ablation experiments conducted on the MNIST and Fashion-MNIST datasets, with a symmetric noise ratio of 0.4 and three classes of data per local client.

Mixed prediction: The efficacy of mixed prediction was evaluated by removing it from the model and assessing its impact. The results are presented in Table 4. A method

was employed where only enhanced data expanded the original dataset without using mixed prediction and trained using cross-entropy loss. Corresponding results indicate that compared to solely sharpening the original prediction p_1, randomly sampling mixed p and then sharpening p significantly improves robustness in extreme scenarios. Mixed prediction proves to be a crucial component ensuring the superior performance of the proposed method.

Contrastive loss: The effectiveness of contrastive loss was evaluated by removing it from the model and assessing its impact. The results are presented in Table 4. The model's performance suffers without using contrastive loss, particularly in imbalanced data distribution.

TABLE 4. Ablation study

Dataset	Method	Accuracy
Fashion-MNIST	FedAvg (Baseline)	38.75%
	Mixed Prediction	57.29%
	Contrastive Loss	61.15%
	Ours	**68.33%**
MNIST	FedAvg (Baseline)	52.25%
	Mixed Prediction	88.68%
	Contrastive Loss	92.17%
	Ours	**95.75%**

6. **Conclusion.** In this work, we explore federated learning algorithms in scenarios where data is non-IID, and samples contain noisy labels, aiming to mitigate the negative impact of noisy labels. This research addresses potential challenges faced when deploying practical federated learning systems, such as federated medical analysis systems. We propose a local self-regularization approach to implicitly prevent models from overfitting noisy labels. We also utilize contrastive learning techniques to explicitly regularize the model output differences between original and augmented instances. Experimental results demonstrate that this method can withstand various noise levels on benchmark datasets with non-IID data and noisy labels, outperforming baseline algorithms. There remains ample future work to explore, including investigating its convergence properties, studying interpretability, and achieving theoretical breakthroughs in federated learning systems.

Acknowledgment. This research was supported by the CCF-Sangfor 'Yuanwang' Research Fund (No. 20240206).

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[2] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8135–8153, 2023.

[3] J. Yao, J. Wang, I. W. Tsang, Y. Zhang, J. Sun, C. Zhang, and R. Zhang, "Deep learning from noisy image labels with quality embedding," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1909–1922, 2018.

[4] K. Lee, S. Yun, K. Lee, H. Lee, B. Li, and J. Shin, "Robust inference via generative classifiers for handling noisy labels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3763–3772.

[5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
[6] X. Xia, T. Liu, B. Han, C. Gong, N. Wang, Z. Ge, and Y. Chang, "Robust early-learning: Hindering the memorization of noisy labels," in *International Conference on Learning Representations*, 2020.
[7] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 322–330.
[8] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, "Normalized loss functions for deep learning with noisy labels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6543–6553.
[9] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1944–1952.
[10] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," in *International Conference on Machine Learning*. PMLR, 2019, pp. 312–321.
[11] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances in neural information processing systems*, vol. 31, pp. 8536–8546, 2018.
[12] Y. Chen, X. Yang, X. Qin, H. Yu, P. Chan, and Z. Shen, "Dealing with label quality disparity in federated learning," *Federated Learning: Privacy and Incentive*, pp. 108–121, 2020.
[13] S. Yang, H. Park, J. Byun, and C. Kim, "Robust federated learning with noisy labels," *IEEE Intelligent Systems*, vol. 37, no. 2, pp. 35–43, 2022.
[14] T. Tuor, S. Wang, B. J. Ko, C. Liu, and K. K. Leung, "Overcoming noisy and irrelevant data in federated learning," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5020–5027.
[15] J. Xu, Z. Chen, T. Q. Quek, and K. F. E. Chong, "Fedcorr: Multi-stage federated learning for label noise correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 184–10 193.
[16] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.
[17] H. Li, Y. Yifeng, F. Zeng, Z. Shifa, D. Hao, and L. Dianbo, "Loadaboost: Loss-based adaboost federated machine learning on medical data," *arXiv preprint arXiv:1811.12629*, 2018.
[18] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 243–22 255, 2020.
[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1597–1607.
[20] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.
[21] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.
[22] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.
[23] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 713–10 722.
[24] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.
[25] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for on-device federated learning," *arXiv preprint arXiv:1910.06378*, vol. 2, no. 6, 2019.
[26] C. Yiqiang, Y. Xiaodong, Q. Xin, Y. Han, C. Biao, and S. Zhiqi, "Focus: Dealing with label quality disparity in federated learning," *arXiv preprint arXiv:2001.11359*, 2020.
[27] M. Yang, H. Qian, X. Wang, Y. Zhou, and H. Zhu, "Client selection for federated learning with label noise," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 2193–2197, 2021.

[28] J. Konecnỳ and H. B. McMahan, "Federated learning: Strategies for improving communication efficiency," 2016.

[29] X. Jiang, S. Sun, Y. Wang, and M. Liu, "Towards federated learning against noisy labels via local self-regularization," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 862–873.