applied sciences

MDPI

Article

Few-Shot Federated Learning: A Federated Learning Model for Small-Sample Scenarios

Junfeng Tian [1,2], Xinyao Chen [1,2,*] and Shuo Wang [1]

[1] School of Cyber Security and Computer, Hebei University, Baoding 071000, China; tjf@hbu.edu.cn (J.T.); wangshuo3127@gmail.com (S.W.)
[2] Key Laboratory on High Trusted Information System in Hebei Province, Hebei University, Baoding 071000, China
* Correspondence: 20218019032@stumail.hbu.edu.cn

Abstract: Traditional federated learning relies heavily on mature datasets, which typically consist of large volumes of uniformly distributed data. While acquiring extensive datasets is relatively straightforward in academic research, it becomes prohibitively expensive in practical applications, especially in emerging or specialized medical fields characterized by data scarcity. This poses a significant challenge. To address this issue, our study introduces a federated learning model that integrates few-shot learning techniques and is complemented by personalized knowledge distillation to further enhance the model's classification accuracy. This innovative approach significantly reduces the dependence on large-scale datasets, enabling efficient model training under limited data conditions. Our experimental evaluations conducted on small-scale datasets, including Omniglot, FC100, and mini-ImageNet, indicate that our model surpasses existing state-of-the-art federated learning models in terms of accuracy, achieving a substantial improvement. Specifically, on the FC100 dataset, the classification accuracy of the conventional federated learning algorithm FedAvg was merely 19.6%, whereas the method proposed in this study achieved a classification accuracy of 41%, representing an improvement of more than double. This advancement not only highlights our model's superiority in alleviating the challenges of limited data availability, but also expands the applicability of federated learning to a broader range of applications.

Keywords: federated learning; few-shot learning; knowledge distillation; small data sample scenarios

Citation: Tian, J.; Chen, X.; Wang, S. Few-Shot Federated Learning: A Federated Learning Model for Small-Sample Scenarios. *Appl. Sci.* **2024**, *14*, 3919. https://doi.org/10.3390/app14093919

Academic Editor: Antonio Fernández-Caballero

Received: 30 March 2024
Revised: 26 April 2024
Accepted: 30 April 2024
Published: 4 May 2024

1. Introduction

In the current realm of machine learning and deep learning, federated learning has emerged as a highly focused research direction, offering the potential to construct global models in decentralized data environments. Federated learning allows multiple participants to train models in a distributed manner without centralizing data storage in a single server, thus preserving data privacy and security. However, while current federated learning frameworks excel with large-scale, high-quality datasets, their performance significantly diminishes with small-scale, few-shot datasets [1].

In practical applications [2–4], training processes often encounter situations where participants may possess a limited amount of data or are exploring an emerging field with scarce available data. This underscores the imperative need for further investigations on how to ensure adequate robustness in joint learning under scenarios with limited data samples, thereby facilitating efficient model training without compromising the availability of the global model. Addressing this issue is critical for the extensive deployment of federated learning.

In response to the challenges faced by traditional federated learning in training global models under few-shot data scenarios, this chapter introduces a novel federated learning model: the Few-Shot Federated Learning model (hereinafter referred to as FsFL). This

model incorporates few-shot learning [5] to overcome the limitations of traditional federated learning in few-shot scenarios. By integrating few-shot learning techniques into the federated learning model training process, the model aims to enable learning and generalization from only a small number of samples. Integrating few-shot learning techniques into the federated learning system reduces the dependency of traditional federated learning model training on large-scale datasets, achieving the objective of model training in few-shot data scenarios. This allows for effective model training progression even with extremely limited data. Simulation experiments have demonstrated the enhanced robustness of the federated learning system in few-shot data scenarios, expanding the applicability of federated learning.

Our contributions:

1. Overcoming the Challenges of Training Models with Extremely Limited Data: In practical applications, acquiring a large volume of labeled data can be prohibitively expensive. A few-shot federated learning model has been designed that enables the normal progression of model training even in scenarios with limited data availability.
2. Enhancing Privacy Protection: In an era where privacy and security issues are of paramount concern, ensuring that user privacy is not compromised during model training is crucial. Therefore, federated learning has been adopted as the foundational framework. This framework avoids the privacy leaks that could occur through the sharing of client's original data.
3. Facilitating Personalized Learning on the Client Side: Considering real-world scenarios, especially those with extremely limited data, where data distributions are often non-independent and identically distributed (non-IID), and participants' classification tasks can vary greatly, personalized knowledge distillation has been introduced. This enables clients to perform partial knowledge distillation to the server, thereby further improving the training efficiency of the model.

2. Related work

2.1. Few-Shot Learning

Few-shot learning and federated learning are paradigms in machine learning where the goal is to acquire "knowledge" from a very limited number of samples and use this "knowledge" to achieve a certain objective. Few-shot learning can be further divided into few-shot classification [6,7], few-shot regression [8], and few-shot detection [9,10]. The methods involved in few-shot classification include metric learning methods, model fine-tuning methods, and meta-learning methods. The core idea of metric learning methods is categorizing samples based on the computation of distances or similarities between them, enabling the identification of unknown samples by their distance or similarity to known categories. Key approaches include the following:

1. Siamese Networks: Koch et al. [11] first introduced the concept of Siamese networks in their paper, employing two identical networks to learn the similarity between input pairs. These networks share the same weights and architecture and are trained jointly to minimize or maximize some distance metric, thereby determining whether two inputs belong to the same category. Thomas et al. [12] proposed a method based on Siamese networks for building text classifiers, embedding texts and labels into a common vector space, and using a similarity function to calculate the similarity between two items. Niamh et al. [13] innovated on the Siamese network architecture by introducing the Stop Loss function to prevent representation collapse, simplifying the training process and enhancing model robustness.
2. Matching Networks: Vinyals et al. [14] introduced matching networks, incorporating attention mechanisms and memory modules to enable the model to learn a matching function for a small-sample task directly from the support set. The design of matching networks allows the model to consider all samples in the support set at each step, learning how to make effective predictions from a few samples in an end-to-end manner. Cao et al. [15] proposed a Bi-directional Matching Network architecture,

incorporating a semantic alignment model and combining appearance flow, relation flow, and mutual information flow for sample alignment and comparison. This method addresses the challenge of image classification in few-shot settings by leveraging the deep semantic relationships between images, significantly improving the classification performance. Zhang et al. [16] presented SGMNet, a meta-learning framework based on scene graph matching for few-shot remote sensing scene classification, introducing a Graph Construction Module (GCM) and a Graph Matching Module (GMM) to effectively utilize the co-occurrence and spatial correlation features of remote sensing images, enhancing the classification performance.

3. Prototypical Networks: Snell et al. [5] introduced Prototypical Networks, which classify by computing prototypes for each category—the averages of all sample features within a class—and then classifying query samples based on their distance to these prototypes. The essence of Prototypical Networks is to represent each category in the feature space such that the features of samples within the same category are as close together as possible, while those of different categories are far apart. This approach has shown excellent performance in few-shot learning tasks, especially in one-shot and few-shot learning scenarios, effectively improving model accuracy and the capacity for generalization. Zhou et al. [17] proposed a new architecture based on Prototypical Networks, LDP-Net, employing a dual-branch network structure to predict the category of an input image using a global branch method. The image is then randomly cropped and the cropped image is used as a new input for prediction. After prediction, knowledge distillation is used to reinforce the consistency between the overall and partial predictions of the image, thereby enhancing the model's generalization performance. Qin et al. [18] proposed a robust network supervised learning method based on Prototypical Networks to address noise and domain discrepancies in network data. This method introduces a small number of real-world samples as "truth" and uses contrastive learning to minimize the distance between network data and this "truth".

The aforementioned studies primarily focus on methods such as the optimization of network structures to enhance the efficiency of local models in scenarios involving small data samples. In contrast, this paper enhances training effectiveness by integrating few-shot learning within a federated learning framework through multi-party co-training.

2.2. Federtated Learning

Federated learning, a specialized form of distributed machine learning, aims to address the challenge of data silos. Initially proposed and designed by Google, the FedAvg algorithm [19] distinguishes itself from traditional centralized machine learning algorithms by distributing the model from the server to multiple clients. Clients locally train the model using their private data and then upload the training results to the server. The server aggregates the uploaded learning parameters or gradients, iterating this process until the global model converges. Numerous classical works in this field [19–21], with FedAvg widely applied across industries, demonstrate exceptional performance in scenarios with large and high-quality datasets. However, these methods exhibit significant performance degradation when faced with small-scale, small-sample datasets, especially those requiring data to be independently and identically distributed (iid). Unfortunately, in practical applications, datasets often deviate from ideal conditions in terms of size, distribution, and quality, leading to a substantial decrease in the accuracy of the final trained global model.

3. Preliminary

3.1. Knowledge Distillation

The concept of knowledge distillation was first introduced by Hinton et al. in [22], comprising a system of teacher and student models [23–26]. The so-called teacher model refers to a large, structurally complex, and redundantly trained network that acts as the

"teacher" during training, instructing a simpler, under-trained "student" model to achieve a performance as close as possible to that of the teacher model.

This process is realized through the use of soft and hard labels along with corresponding loss functions. Specifically, it begins by calculating the soft output of the teacher model, which is the raw prediction values, or logits, from the last fully connected layer of the neural network—values that have not been activated by the softmax function. Using the probability distribution after applying the softmax function would limit the knowledge transferred to the student model and might even have a negative effect by amplifying errors. These prediction values represent the teacher model's confidence level for each class.

$$q_i = \frac{exp\left(\frac{z_i^T}{T}\right)}{\sum_j exp\left(\frac{z_j^T}{T}\right)} \tag{1}$$

Here, z_i^T denotes the logit output of the teacher model for the ith class, and T is the temperature parameter that controls the smoothness of the output probability distribution. It is evident that a smaller T value, which brings the function closer to the softmax function, makes q_i more similar to the probability distribution after activation. Conversely, a larger T value results in a smoother probability distribution curve from the softmax output, allowing the student model to acquire knowledge with richer information entropy.

After obtaining the logits from the teacher model, the student model's logits are calculated in a similar manner. Here, z_i^S represents the logit output of the student model for the ith class.

$$p_i = \frac{exp\left(\frac{z_i^S}{T}\right)}{\sum_j exp\left(\frac{z_j^S}{T}\right)} \tag{2}$$

Subsequently, a loss function is employed to enable the student model to absorb and compress the knowledge from the teacher model.

$$L_{soft} = KL(Q_i, P_i) \times T^2 \tag{3}$$

Here, Q_i and P_i represent the probability distributions of the teacher and student models, respectively. The difference between these probability distributions is calculated using the Kullback–Leibler divergence, with the temperature parameter used to adjust it, ensuring that the gradients do not become too small as the temperature value increases.

Throughout the knowledge distillation process, in addition to computing the soft target loss, it is also necessary to calculate the hard target loss. The hard target loss, also known as hard label loss, enables the student model to learn not only the decision boundaries and related knowledge from the teacher model through soft labels, but also to make accurate predictions from the true labels. Typically, the cross-entropy loss function is used to calculate the hard label loss between the true labels y and the probability distribution P_i of the student model:

$$L_{hard} = CrossEntropy(y, P_i) \tag{4}$$

Therefore, by introducing a parameter α to control the weights, a total loss function for knowledge distillation can be obtained:

$$L_{total} = \alpha L_{hard} + (1 - \alpha) L_{soft} \tag{5}$$

3.2. Few-Shot Learning

Few-shot learning aims to enable machine learning models to learn and generalize from a minimal number of samples. The development of few-shot learning draws inspiration from human learning patterns, where humans exhibit proficiency in recognizing new

objectives with minimal data. Few-shot learning methods are generally classified into two categories: metric-based and gradient-based methods.

Metric-based methods leverage the similarity between samples for few-shot learning. Prototypical networks [5], for instance, predict the classification of query samples based on the Euclidean distance between query samples and samples in the support set. Gradient-based methods, on the other hand, update the model parameters during training rounds to enhance generalization in small-sample scenarios. For instance, in [8], the model parameters are adjusted based on the support set to achieve rapid generalization.

In the typical scenario of few-shot learning, the dataset is divided into four parts. Three hyperparameters, namely N-way, K-shot, and Q-query, are utilized to further delineate the few-shot learning task. Here, N governs the number of classes in a single task, K denotes the quantity of data per class among the N classes, and Q determines the size of the data used in a single task. Figure 1 is an illustrative diagram of a five-way, one-shot, one-query few-shot learning scenario; in the illustrated diagram, an image is input as a query, and through the process of few-shot learning, it is possible to identify a label within the support set that closely corresponds to the input, namely, Bulldog.

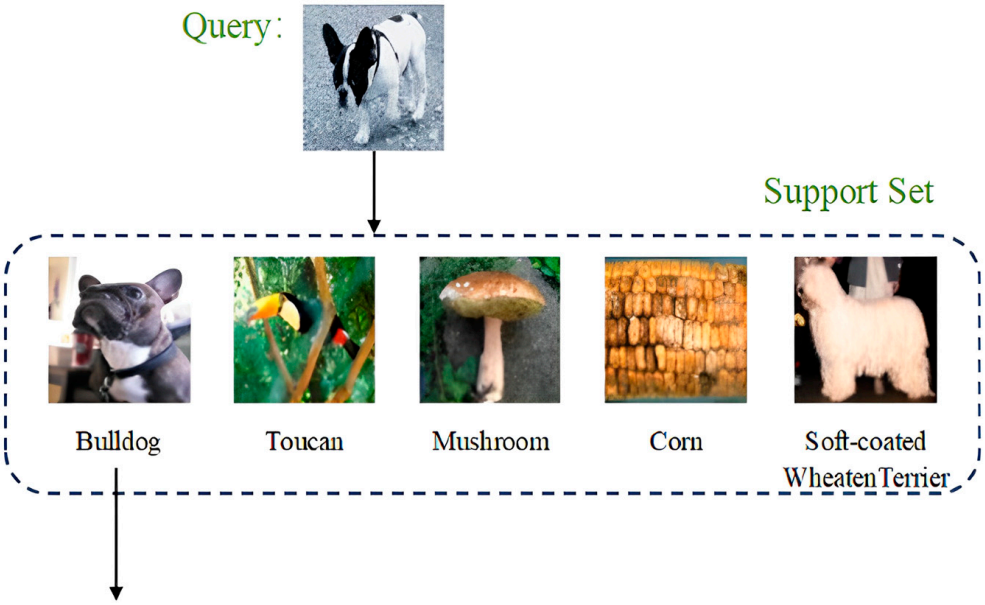

Figure 1. Schematic diagram of a small sample.

4. Model System

4.1. Model Learning Process

Figure 2 illustrates the overall workflow of the model, which is as follows:

1. Initialization. The global model is initialized on the server side and distributed to all clients participating in the training.
2. Local Model Training by Clients. Each client trains the received global model on its local dataset. Specifically, clients calculate the prototypes for each class in their local dataset and based on these prototypes, complete the training of the model.
3. Client Parameters Upload. After the local model training is completed in the current round, each client uploads the model update parameters to the server.
4. Server-Side Aggregation. Upon receiving the model update parameters from the clients, the server averages and aggregates these parameters to obtain the global model parameters. After aggregation, the global model is evaluated for performance, specifically testing its classification accuracy on a test set.
5. Iterative Training. The server distributes the updated global model to all clients, who then commence the next round of model training.

6. Personalized Knowledge Distillation. When clients receive a global model that is not in an initialized state, they use their local model as the student model and the received global model as the teacher model. Through a personalized knowledge distillation algorithm, the student model is guided to learn knowledge relevant to its local classification task from the teacher model.

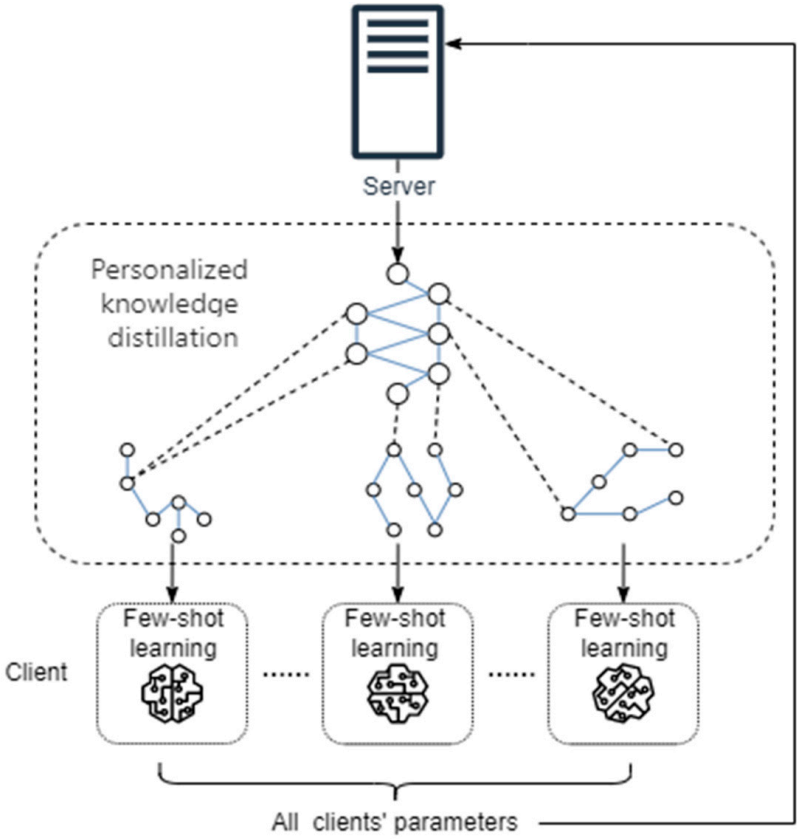

Figure 2. Schematic of the FsFL model.

Algorithm 1 shows the specific workflow.

Algorithm 1 Few-shot Federated Learning

Input: Client Collection{}
Output: Global model with completed training

Server side sends initial global model to client side
for Client{}... **do**
    Clients use local data for model few-shot learning
    Upload model training parameters to the server
end for
for Client{}... **do**
    Average aggregation of model parameters uploaded by clients
end for
while Global model does not converge
    for Client{} **do**
        $w_g \leftarrow PersonalizedKD(M_c, M_g)$
        $w_c^{t+1} = (w_c^t + w_{g)}/2$
    end for
Repeat the process until the model converges

4.2. FsFL

This section describes the training process of the FsFL model in detail. Initially, in each round, clients randomly sample their local dataset to obtain a meta-training task set $D_{mt} = \{D_{support}, D_{query}\}$ and conduct preliminary training on the meta-learner for this dataset.

$$\begin{aligned} w'_{t+1} &= w'_t - \alpha \nabla L\left(w'_t; D_{support}\right) \\ &= w'_t - \alpha \frac{1}{|D_{support}|} \sum_{x_i \in D_{support}} \nabla L(w'_t; x_i) \end{aligned} \tag{6}$$

The cross-entropy loss between the global model parameters of the current round t and the data samples in the support set is calculated to determine the direction of model updates for the next round, with the learning rate controlling the pace of the model updates. Subsequently, the model's learning effectiveness is evaluated on D_{query}, and the model is adjusted based on its classification performance on $D_{support}$.

Specifically, the training process on the client side proceeds as follows: Firstly, feature extraction f_θ is performed on the samples in the $D_{support}$ set using a neural network to obtain feature vectors. Then, for each category c, the mean feature vector p_c of that category is calculated.

$$p_c = \frac{1}{|S_c|} \sum_{(x_i, y_i) \in S_c} f_\theta(x_i) \tag{7}$$

The image data x in D_{query} is classified based on the calculation of the L2 distance d for x.

$$p(y = c|x) = \frac{\exp(-d(f_\theta(x), p_c))}{\sum_{c'} \exp(-d(f_\theta(x), p_{c'}))} \tag{8}$$

The model parameters θ are optimized to achieve the minimum loss on D_{query}.

$$\min_\theta L(f_\theta) = \sum_{(x_i, y_i) \in D_{query}} L(f_\theta(x_i), y_i) \tag{9}$$

If C represents the total number of categories, substituting Equation (8) results in the expanded expression below.

$$\begin{aligned} L(f_\theta(x_i), y_i) &= -y_i \log \frac{\exp\left(-d\left(f_\theta(x_i),\, p_{y_i}\right)\right)}{\sum_{c=1}^{C} \exp(-d(f_\theta(x_i),\, p_c))} \\ &= -y_i\left(-d\left(f_\theta(x_i), p_{y_i}\right)\right) + \log \sum_{c=1}^{C} \exp(-d(f_\theta(x_i), p_c)) \end{aligned} \tag{10}$$

In the System Model section, the server-side aggregation of w_c is performed as follows:

$$w_g = \sum_{c=1}^{K} \frac{w_c}{K} \tag{11}$$

The server aggregates the parameters w_c received from the K clients by summing them and then taking the average, resulting in the global model parameters w_g.

4.3. Personalized Knowledge Distillation Based on Student Model Classification Tasks

In the text on Personalized Knowledge Distillation, a scenario is considered where each client possesses different data samples with varying data distributions. Consequently, the final parameters obtained from model training, tailored to each client's specific classification tasks, also differ. If clients directly extract parameters from the global model provided by the server, they will inevitably acquire parameters related to other clients' classification tasks, which are irrelevant to their own. This directly impacts the clients' ability to conduct subsequent rounds of local training. To address this issue, when the server sends a non-initial global model to the clients, personalized knowledge distillation is employed on the client model. This allows clients to selectively extract parameters relevant to their local classification tasks, thereby avoiding the interference of unnecessary parameters in model

training. Initially, the classification task of the client model focuses on a specific set of categories C_m, while the server focuses on a broader set of categories C_b. The distillation target is for the client to distill knowledge from the server on the categories C_m. Here, the output probabilities of the client model and the server model for a sample $C_x (C_x \in C_m)$ are defined as $P_c(C_x)$ and $P_s(C_x)$, respectively. The objective is to minimize the prediction probability difference between $P_c(C_x)$ and $P_s(C_x)$ as much as possible to achieve the effect of personalized distillation. The loss function is defined as follows [10,23–28]:

$$L_{KD} = -\sum_{C_x} P_c(C_x) \log P_s(C_x) \tag{12}$$

where the output probability $P_c(C_x)$ for the client model is given by the following formula:

$$P_c(C_x) = \frac{\exp\left(\frac{z(C_x)}{T}\right)}{\sum_{k \in C_m} \exp\left(\frac{z(C_k)}{T}\right)} \tag{13}$$

and the output probability $P_s(C_x)$ for the server model is defined as follows:

$$P_s(C_x) = \frac{\exp\left(\frac{z'(C_x)}{T}\right)}{\sum_{k \in C_m} \exp\left(\frac{z'(C_k)}{T}\right)} \tag{14}$$

Furthermore, the model's prediction probabilities are softened by utilizing the temperature parameter T. Initially, the difference between the maximum logit output by the server model and the logit of the true category, known as the confidence discrepancy Δz_x, is calculated.

$$\Delta z_x = \max\left(C_x^t\right) - C_x^t * y_x \tag{15}$$

C_x^t represents the logit values for all categories by the server model on sample x, and y_x is the true label of the sample. In this section, the exponential function is chosen as the confidence discrepancy function, and the smoothing factor S is set to 2 to ensure sufficient sensitivity to the confidence discrepancy Δz_x:

$$f(\Delta z_x) = e^{\frac{\Delta z_x}{S}} \tag{16}$$

Moreover, a hyperparameter w_i is introduced into the system to facilitate the extraction of task-specific objectives.

$$w_i = I(C_i) = \begin{cases} 0.9, (C_x \in C_m) \cap (C_x \in C_b) \\ 0.1, C_x \notin C_m \end{cases} \tag{17}$$

Consequently, this leads to the derivation of the final adaptive temperature parameter T_i and the ultimate loss function L'_{KD}:

$$T_i = 1 + w_i * (T_{max} - 1) * \frac{f(\Delta z_x) - 1}{f(\Delta z_x) + 1} \tag{18}$$

The final loss function L'_{KD} is obtained by substituting T_i into Equations (13) and (14), respectively.

$$L'_{KD} = -\sum_{C_x} w_i P_c(C_x) \log P_s(C_x) \tag{19}$$

The design of this temperature parameter facilitates the confidence in sample classification based on the server-side model and the relevance to the client's local tasks. Ultimately, this achieves the effect of enabling clients to selectively extract parameters from the server-side model for knowledge distillation. The workflow is shown in Algorithm 2.

Algorithm 2 PersonalizedKD

Input: Server's model, Client's model
Output: Updated Client's model
for Client{} **do**
    The client receives the global model from the server
    The values of the hyperparameters w_i are determined according to Equation (17)
    For data related to the local classification task in the server, the confidence discrepancy is calculated using Equation (15)
    The corresponding temperature parameters are obtained according to Equation (18)
    The probabilities of the client-side model p_c, and the server-side model p_s, are calculated separately.
    The knowledge distillation loss is calculated using Equation (19)
    The client model parameters are updated through backpropagation
end for

5. Experiment

5.1. Datasets

This section of the experiment uses three datasets commonly utilized in the field of few-shot learning; these three datasets were selected because each category within them contains a small amount of data, and they are prone to forming non-IID data distribution patterns, making them highly suitable for evaluating the model proposed in this paper:

Omniglot [6], a dataset comprising images of handwritten characters from 1623 different character classes derived from alphabets of various natural languages around the world. Each class contains 20 distinct samples, with each original data sample being a single-channel image of 105 × 105 pixels in size.

FC100 [29], an image dataset based on the classic Cifar-100 dataset, consisting of 100 classes, each with 600 samples of 32 × 32 resolution three-channel data. The dataset is further divided into train (60 classes), validation (20 classes), and test (20 classes) subsets.

MiniImageNet [14], extracted directly from the ImageNet dataset, also comprises 100 classes, each with 600 samples of 84 × 84 pixel size. Like the others, it is divided into train (64 classes), validation (16 classes), and test (20 classes) subsets.

These datasets are integral to the research of few-shot learning, providing diverse challenges due to their varied class compositions and image resolutions. The structured division into training, validation, and test subsets facilitates a comprehensive evaluation of the learning algorithms under investigation, allowing for a balanced assessment of their generalization capabilities across different data distributions.

5.2. Performance Evaluation

Classification accuracy is used as the evaluation metric in the model comparison experiments, with the accuracy formula presented as follows:

$$Acc = \frac{1}{|D_{test}|} \sum_{i=1}^{|D_{test}|} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{20}$$

Here, $|D_{test}|$ denotes the number of categories within the test dataset. TP (True Positives) refers to the count of positive instances correctly predicted as positive; TN (True Negatives) indicates the count of negative instances correctly predicted as negative; FP (False Positives) represents the number of negative instances incorrectly predicted as positive; and FN (False Negatives) denotes the number of positive instances incorrectly predicted as negative.

In Section 5.3 of the experiments, we selected FedAvg as the representative algorithm for classical federated learning and FedFSL [30] as an advanced algorithm for small-sample federated learning to conduct comparative experiments. These model comparison exper-

iments used classification accuracy as the evaluation metric, with the specific accuracy formula presented in Equation (20).

5.3. Result & Analysis

From a straightforward observation, it is evident that FedAvg exhibits significantly lower accuracy on the Omniglot dataset compared to the method FsFL introduced in this paper (Figure 3). Furthermore, under a one-shot scenario, the FedAvg model completely loses its usability. This observation underscores the feasibility of integrating few-shot learning techniques into federated learning systems, enhancing their capability to operate effectively in few-shot scenarios. On the other hand, it also confirms the previously mentioned point that traditional federated learning algorithms urgently need improvement to perform well in few-shot learning contexts. In addition, we also compared our method with advanced FedFSL, and the comparison results showed that the accuracy of the proposed method is better than FedFSL in both one-shot and five-shot scenarios.

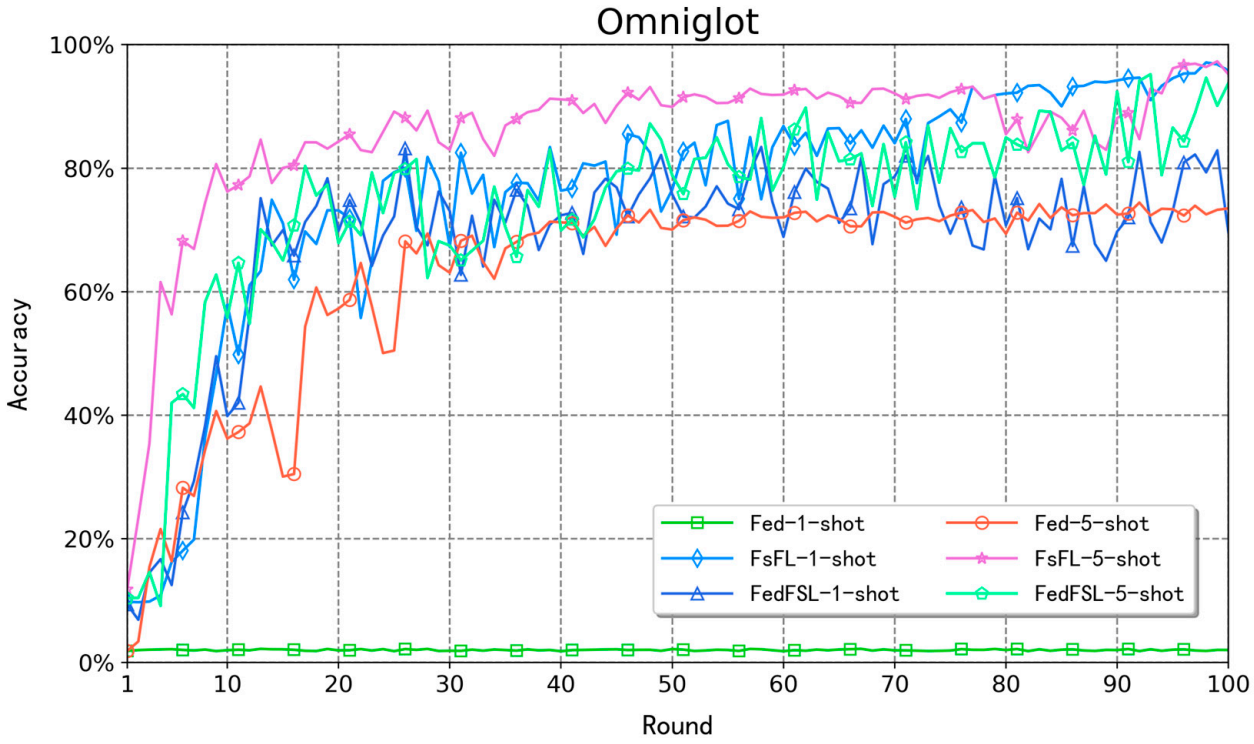

Figure 3. Performance on the Omniglot dataset.

Given that the FC100 (Figure 4) dataset encompasses a wide variety of objects and scenes from the natural world, the content of the images is more diverse and complex. As the complexity of the dataset increases, it generally leads to a decrease in the overall model accuracy. However, it is still observable that the FsFL method proposed in this paper maintains the usability of the model to a certain extent, and that the training effects of the model remain superior to those of traditional federated learning methods and FedFSL.

The experiments conducted on the previous two datasets reveal that even when employing the FsFL method, the control groups (one-shot and five-shot) show minimal differences in the accuracy of the model. This discrepancy could likely be attributed to the inherent limitations of the chosen training models. Consequently, a more complex dataset, MiniImageNet (Figure 5), was selected for further experimentation. The results demonstrated that, across four different scenarios, the model accuracy curves exhibited the anticipated shape, achieving the envisioned effects.

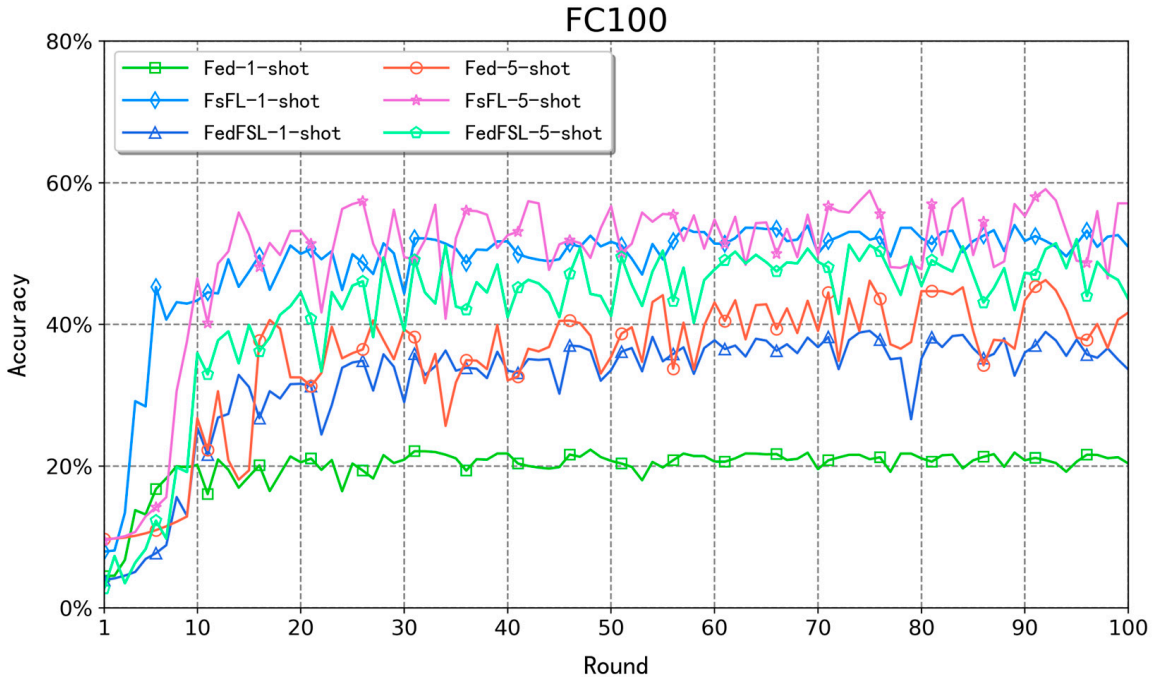

Figure 4. Performance on the FC100 dataset.

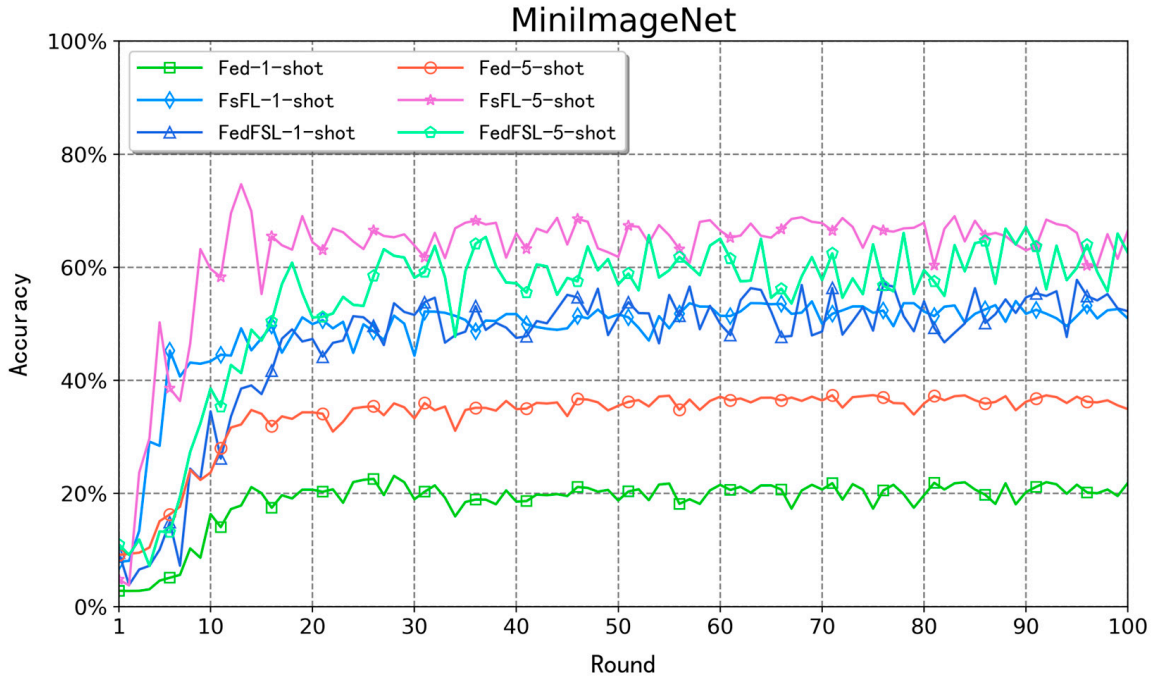

Figure 5. Performance on the MiniImageNet dataset.

The overall experimental results are as follows in Table 1:

Table 1. Comparison of accuracy across three datasets.

Dataset	Omniglot		FC100		MiniImageNet	
Setting	**One-Shot**	**Five-Shot**	**One-Shot**	**Five-Shot**	**One-Shot**	**Five-Shot**
FsFL	95.8 ± 1.54	96.4 ± 2.03	41.00 ± 2.10	57.51 ± 1.48	49.4 ± 1.21	68.2 ± 1.57
Fedavg	2	71.2 ± 1.99	19.6 ± 1.00	40.9 ± 1.03	18.7 ± 0.86	44.1 ± 0.97
FedFSL	78.1 ± 1.45	89.7 ± 2.06	38.60 ± 2.00	50.90 ± 1.08	53.52 ± 1.1	61.56 ± 1.66

We evaluated the model's performance by calculating the F1 scores on three datasets under the five-shot condition. The results are presented in Table 2 for a further assessment of the model's performance.

Table 2. F1-score.

Dataset	Omniglot	FC100	MiniImageNet
FsFL	0.907	0.603	0.712
Fedavg	0.685	0.434	0.394
FedFSL	0.875	0.593	0.632

Based on the analysis of the experimental results:

FsFL: The method presented in this paper maintains a high accuracy rate after reaching convergence on three image datasets; even under the extreme conditions of one-shot learning, it still preserves a considerable accuracy rate. It can be intuitively observed that our method far surpasses the baseline methods. This phenomenon indicates that introducing few-shot learning techniques into federated learning systems can improve the system's model training effects in small-sample scenarios.

FedAvg: In contrast, the performance of the classic federated learning algorithm under these conditions is barely satisfactory, which further proves that traditional federated learning struggles to find effective applications in small-sample scenarios. Additionally, the effectiveness of our approach on the FC100 and MiniImageNet datasets is also related to the neural network used for extracting image features.

FedFSL: Only on the MiniImageNet dataset, the method proposed in this paper performs better in the one-shot scenario, but the F1 score is lower than the method proposed in this paper on all three datasets, further demonstrating the effectiveness of the method proposed in this paper.

6. Conclusions

In this paper, the challenges of applying federated learning in few-shot scenarios are explored, and a new model, FsFL, is introduced. This model builds on federated learning by integrating few-shot learning techniques. A personalized knowledge distillation approach has been devised that enables client models to partially distill knowledge from server-side models, thereby enhancing the robustness of federated learning. This allows it to complete model training and ultimately converge in scenarios with extremely limited data, showing considerable classification accuracy in few-shot scenarios. This model was tested on three image datasets, and the results indicate that in one-shot scenarios, the performance improvement exceeded 100%. In five-shot scenarios, there was also a significant enhancement, for instance, a more than 70% improvement on the MiniImageNet dataset, and over 30% on the other two datasets. At the same time, the model accuracy and F1 score obtained by this method are also better than advanced FedFSL schemes. These results highlight the superiority of our proposed method in scenarios with small data samples; at the same time, it also reflects the value of the model in practical scenarios.

Author Contributions: Conceptualization, X.C.; Methodology X.C.; Software X.C.; Validation X.C.; Formal Analysis X.C.; writing—original draft preparation, X.C.; Writing—Review and Editing, J.T.; Funding Acquisition, J.T.; Supervision, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Hebei Province, grant number F2021201049, Natural Science Fund Committee of Hebei Province (F2021201058) and Funds of the Central Government for Local Science and Technology Development (236Z0701G).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All of datasets used in this experiment are publicly available datasets at the following URLs: MiniImageNet. https://www.kaggle.com/datasets/arjunashok33/miniimagenet/download?datasetVersionNumber=1 (accessed on 29 April 2024); FC100. https://github.com/learnables/learn2learn/blob/master/learn2learn/vision/datasets/fc100.py (accessed on 29 April 2024); Omniglot. https://www.kaggle.com/datasets/watesoyan/omniglot (accessed on 29 April 2024).

Conflicts of Interest: The authors declare that this study was conducted without any commercial or financial relationships that could be considered potential conflicts of interest.

References

1. Zhang, J.; Shi, Y. A Personalized Federated Learning Method Based on Clustering and Knowledge Distillation. *Electronics* **2024**, *13*, 857. [CrossRef]
2. Li, Y.; Wen, G. Research and Practice of Financial Credit Risk Management Based on Federated Learning. *Eng. Lett.* **2023**, *31*, 271.
3. Yang, D.; Xu, Z.; Li, W.; Myronenko, A.; Roth, H.R.; Harmon, S.; Xu, S.; Turkbey, B.; Turkbey, E.; Wang, X.; et al. Federated semi-supervised learning for COVID region segmentation in chest CT using multi-national data from China, Italy, Japan. *Med. Image Anal.* **2021**, *70*, 101992. [CrossRef] [PubMed]
4. Zhuang, W.; Gan, X.; Wen, Y.; Zhang, S. Optimizing performance of federated person re-identification: Benchmarking and analysis. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 1–18. [CrossRef]
5. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
6. Lake, B.M.; Salakhutdinov, R.; Tenenbaum, J.B. Human-level concept learning through probabilistic program induction. *Science* **2015**, *350*, 1332–1338. [CrossRef] [PubMed]
7. Tian, Y.; Wang, Y.; Krishnan, D.; Tenenbaum, J.B.; Isola, P. Rethinking few-shot image classification: A good embedding is all you need? In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 266–282.
8. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
9. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645. [CrossRef] [PubMed]
10. Fan, Q.; Zhuo, W.; Tang, C.-K.; Tai, Y.-W. Few-shot object detection with attention-RPN and multi-relation detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4013–4022.
11. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015.
12. Müller, T.; Pérez-Torró, G.; Franco-Salvador, M. Few-shot learning with siamese networks and label tuning. *arXiv* **2022**, arXiv:2203.14655.
13. Belton, N.; Hagos, M.T.; Lawlor, A.; Curran, K.M. Fewsome: One-class few shot anomaly detection with siamese networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 2977–2986.
14. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29.
15. Cao, C.; Zhang, Y. Learning to compare relation: Semantic alignment for few-shot learning. *IEEE Trans. Image Process.* **2022**, *31*, 1462–1474. [CrossRef] [PubMed]
16. Zhang, B.; Feng, S.; Li, X.; Ye, Y.; Ye, R.; Luo, C.; Jiang, H. SGMNet: Scene graph matching network for few-shot remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5628915. [CrossRef]
17. Zhou, F.; Wang, P.; Zhang, L.; Wei, W.; Zhang, Y. Revisiting prototypical network for cross domain few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 20061–20070.
18. Qin, Y.; Chen, X.; Chen, C.; Shen, Y.; Ren, B.; Gu, Y.; Yang, J.; Shen, C. Fopro: Few-shot guided robust webly-supervised prototypical learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 2101–2109.
19. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
20. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
21. Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; Poor, H.V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7611–7623.
22. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.

23. Ye, H.-J.; Ming, L.; Zhan, D.-C.; Chao, W.-L. Few-shot learning with a strong teacher. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *46*, 1425–1440. [CrossRef] [PubMed]
24. Parnami, A.; Lee, M. Learning from few examples: A summary of approaches to few-shot learning. *arXiv* **2022**, arXiv:2203.04291.
25. Kang, D.; Cho, M. Integrative few-shot learning for classification and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9979–9990.
26. Yang, G.; Tae, H. Federated Distillation Methodology for Label-Based Group Structures. *Appl. Sci.* **2023**, *14*, 277. [CrossRef]
27. Sun, Q.; Liu, Y.; Chua, T.-S.; Schiele, B. Meta-transfer learning for few-shot learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 403–412.
28. Tan, Z.; Wang, S.; Ding, K.; Li, J.; Liu, H. Transductive linear probing: A novel framework for few-shot node classification. In Proceedings of the Learning on Graphs Conference, Virtual, 9–12 December 2022; pp. 4:1–4:21.
29. Oreshkin, B.; Rodríguez López, P.; Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; Volume 31.
30. Fan, C.; Huang, J. Federated few-shot learning with adversarial learning. In Proceedings of the 2021 19th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Philadelphia, PA, USA, 18–21 October 2021; pp. 1–8.