Hindawi
Security and Communication Networks
Volume 2022, Article ID 7424094, 9 pages
https://doi.org/10.1155/2022/7424094

Research Article

Federated Learning Framework Based on Data Value Evaluation in Industrial IoT

Chao Ma, Haiyang Yu, Zheng Li, and Zhen Yang

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

Correspondence should be addressed to Haiyang Yu; yuhaiyang@bjut.edu.cn

Received 4 August 2022; Revised 27 October 2022; Accepted 11 November 2022; Published 8 December 2022

Academic Editor: Marimuthu Karuppiah

With the continuous maturity and development of the big data technology system, deep learning has been widely used in the field of the Industrial Internet of Things. However, the traditional training model with centralized data is prone to the leakage of private information in the industry, such as facial information. In recent years, federated learning solves the problem of privacy leakage caused by data sharing by not sharing data and only contributing to local models. Federated learning does not share data, which also makes it impossible to evaluate the contribution of each client to the federated task. We propose a federated learning framework based on data value evaluation. In this method, under the premise of effectively completing the training task of federated learning and ensuring the privacy of client data, a data value evaluator is designed in the central server, and the model uploaded by the client is evaluated to obtain the corresponding selection probability as the model aggregation weight. Experimental results show that the proposed method improves the accuracy of the global model obtained by existing federated aggregation.

1. Introduction

1.1. Introduction. In recent years, the integration of artificial intelligence and the Industrial Internet of Things has promoted the development of the intelligence industry [1]. Due to its powerful big data modeling, classification, and identification capabilities [2], deep learning has been applied to solve data-driven problems in the Industrial Internet of Things [3, 4]. The IoT networks consist of smart devices to exploit sensors to accumulate information or utilize implanted cameras to take photos and record videos [5]. With the development of the Industrial Internet of Things and the popularization of smart devices, a large amount of high-value data has been generated in the field of the Industrial Internet of Things, which is very conducive to training more accurate models for actual production [6]. In the previous deep learning, various data providers aggregated their data for training. Nagarajan et al. [7] proposed a new FoG-assisted cloud network architecture based on the Internet of Things. This architecture accumulates patients' real-time medical data through multiple medical Internet of Things sensor networks and uses the depth learning algorithm deployed on the FoG-based medical platform to analyze these data. However, most of the data in the industrial Internet of Things field contains private information, such as fingerprint information. If these big data are shared with other trainers, it will cause heavy losses to users and the company. In addition, many countries in the world are committed to protecting the privacy of countries, enterprises, and citizens and have promulgated relevant laws to limit them [1]. As a result, the data owner can only use a small amount of local data to train the model, which is not conducive to the improvement of the generalization of the model, resulting in the overfitting of the model. Therefore, deep learning urgently needs to solve the problem of data silos caused by data privacy.

Federated learning (FL) was first proposed by the Google team in 2016 and applied to the Google input method Gboard system to realize the candidate word prediction of the input method [8]. Federated learning only requires clients to share aggregated gradients or model parameters, not client data. It preserves the privacy of clients' local data

and avoids model overfitting by training on data from multiple sources. Therefore, federated learning has received extensive attention from academia and industry.

The goal of federated learning is to share model parameters that are trained only with local data between clients, which not only gives full play to the advantages of big data but also avoids data privacy leakage. At the same time, client model training can be easily performed in parallel. However, in most current federated learning [1, 9, 10], all clients can obtain the same federated learning model in each round of communication, even if they contribute to a global model based on local data differences. This may be unfair because although different clients contribute differently to the global model, they all end up with the same "reward;" that is, each client has the same global model and the profit given by the federation task issuer. Su et al. [11] proposed IDES, which provides a centralized reputation management scheme to detect malicious nodes in the vehicular network. In practice, this difference in contributions can be due to several reasons, the most obvious being that different clients have different data values [12]. Obviously, if the customer's data set is correct and is standard sample data, the model trained with these data usually has higher accuracy. However, if the data samples of a client have a large number of label errors or the sample data contains a lot of noise, the accuracy of the model trained by the client using such samples may be poor. Furthermore, since existing federated learning frameworks cannot evaluate contributions, this environment may be vulnerable to free-rider attacks. A free-rider client can just upload some worthless model parameters to the central server but still end up with the same global model and corresponding rewards [13]. In federated learning, clients will consume the resources of their devices, including computation, communication, and power for training models or transferring data. Without adequate compensation, clients may be reluctant to participate in federated tasks and share models they trained after spending a lot of resources. Furthermore, federated learning frameworks still face various security risks [14]. Song et al. [15] showed that the central server can deduce some sensitive information about the client through the model gradient uploaded by the client [16]. In addition, a curious central server can recover some private information of client data through generative adversarial networks. Due to these risks, clients are even less likely to actively participate in federated learning tasks. Therefore, giving them adequate rewards reasonably is a means of incentivizing clients to participate in federated missions. The reward mechanism of federated learning can influence the client's decision. In federated learning, clients use different sample data to train models individually, which will affect the accuracy of the final global federated model. Therefore, a challenge for federated learning systems is how to evaluate each client's contribution to the global model and then assign rewards to clients.

There are two main perspectives for motivating clients to participate in federated tasks. On the one hand, game theory in the field of economics is introduced into federated learning, and the client and the central server decide their strategies according to their respective resource consumption and benefits in federated learning. On the other hand, researchers in the computer field strive to achieve an assessment of the value of each client's data [17].

We propose a federated learning framework based on client-side data value evaluation (FLDVE). This method uses a reinforcement learninglike approach, designing an estimator and a small private standard validation dataset on the federated server side, using the accuracy of the global model on the validation set as a reinforcement signal to the estimator. FLDVE can both protect the privacy and measure the contribution of each client to the federated global model more accurately. The main contributions of this study are as follows:

(i) We propose a method for evaluating the contribution of federated clients, which utilizes the idea of reinforcement learning to evaluate the contribution of federated clients to the federated global model through an evaluator in the federated server.

(ii) Based on the contribution of each federated client to the global model obtained by the evaluator, our method adopts a federated weighted aggregation method based on contribution evaluation, so that clients with high-value data can get a larger aggregation weight. In this way, the role of higher-contribution clients can be fully utilized.

(iii) Our experiments on different datasets have verified the effectiveness of our proposed method. The results show that our method can better measure the contribution of each client in federated learning and the test accuracy of the global model is higher than that of traditional federated learning.

The remainder of the paper is structured as follows: the related work is presented in Section 2. We give the system design in Section 3. Section 4 provides the experimental setup and the performance evaluation, respectively. Finally, we conclude the work as a whole in Section 5.

1.2. Symbol Description. In this section, we explain the abbreviations that appear in the paper.

IoT: the industrial internet of things.

FL: federated learning.

FLDVE: a federated learning framework based on client-side data value evaluation.

FLI: federated learning incentive.

DVRL: a data value evaluation framework using reinforcement learning.

SGD: stochastic gradient descent.

MNIST: modified national institute of standard and technology.

CNN: convolutional neural network.

CIFAR-10: a publicly available dataset funded by the Canadian institute for advanced research.

ReLU: rectified linear unit.

2. Related Work

The Internet of Things is an emerging technology in all aspects of the world today. With the increase in security issues, data transmission should be more secure to avoid unauthorized access to confidential information. Various methods based on cryptography have also been studied [18]. Nagarajan et al. [19] proposed an intelligent city dynamic food supply chain based on the Internet of Things, which can ensure food quality, provide vehicle routes, and track pollution sources. Nagarajan et al. [20] proposed an encryption technology to provide effective security using data from embedded devices or medical storage databases.

Federated learning uses another idea to achieve privacy protection, it only requires clients to share aggregated gradients or model parameters, not client data. In federated learning, how to evaluate the client's contribution to the federated global model is a problem that needs to be solved. Shapley value is a common indicator for fair and quantitative assessment of user marginal contribution. Sim et al. [21] introduced Shapley to federated learning, and based on this, they designed an incentive scheme that can calculate marginal contribution. It gives a different federation model for each client as a reward. Although the incentive mechanism of deep learning has studied how to evaluate the value of each client's training data, it requires high additional computational overhead and cannot be directly applied to complex federated tasks.

In federated learning, to attract clients with high-value data, Song et al. [22] proposed a Shapley value-based contribution index to evaluate the contribution of different clients in federated learning. By using a combination of different training submodels, the contribution index of different clients is calculated. Therefore, it requires a lot of computational and time overhead, and also cannot be applied to practical scenarios. To overcome this problem, the authors approximately reconstruct the model on different combinations of clients by federated learning of intermediate models at each iteration to avoid additional training. The auction mechanism in economics also applies to federated learning. Due to the large amount and quality gap between the training data of different clients, this gap can lead to large performance degradation of federated learning. Reference [23] proposed a new multidimensional federated learning incentive framework considering the multidimensional dynamic edge resources in federated learning. The authors use game theory to derive the optimal policy that each client can maximize its benefit and use expected utility to guide the central server to select the client that maximizes the server's benefit to train a global model. In federated learning, the training and commercialization of models take time to complete. Therefore, there is a delay before the central server can compensate the client. Traditional federated learning does not study this mismatch between compensation and client contributions. To create a fair federated incentive environment to attract clients with more high-value training data to participate in federated tasks, Yu et al. [24] proposed Federated Learning Incentive (FLI). It dynamically allocates a preset budget among clients in the federation, maximizing collective utility and minimizing disparity between clients.

Reinforcement learning [25, 26] is a research hotspot in the field of machine learning, mainly used to achieve decision optimization. The basic idea of reinforcement learning is that when the subject performs an action, the environment will switch to a new state, and the subject will constantly adjust its strategy according to the rewards received from the environmental feedback to achieve optimal decision-making, which is mainly used to solve decision-making problems. First, the agent perceives the current state, and selects the action to perform from the action space A; the environment feeds back the corresponding reward according to the operation done by the agent and transfers it to the new state. The agent adjusts its strategy through the rewards it receives and makes new decisions for new states. The goal of reinforcement learning is to find an optimal policy so that the agent can obtain the maximum long-term cumulative return. Yoon et al. [27] recently proposed a data value evaluation framework using reinforcement learning (DVRL), which expresses the evaluation of data samples as a meta-learning framework, where the model can give selection probabilities for different samples and decide whether to select the sample to join the train. In this way, through a series of reinforcement processes, high-value samples are mostly selected when training the model, while low-value samples are discarded.

Traditional federated learning cannot evaluate the value of customers' private data, and therefore cannot consider each customer's contribution when aggregating models. Models trained using samples from clients with low-quality data will reduce the accuracy of the aggregated model. Our method proposed uses the idea of reinforcement learning to evaluate the contribution of the federation client and takes the contribution into account in the federation aggregation to solve the above problems.

3. Proposed Method

3.1. Overall Structure. A typical architecture of federated learning in IoT is shown in Figure 1. The federation architecture consists of two roles, the client and the central server. The client is the data owner of the Internet of Things, and the server is responsible for collecting and aggregating the model uploaded by the client. We define $C = \{1, \ldots, n\}$ as the client set C, and the private dataset on the client i is represented as D_i. The representation of data samples in D_i is (x_j, y_j), where x_j is the input vector of sample j in D_i, y_j is the corresponding sample label. Each client C_i uses its local dataset D_i to train a local model with ω_i parameters and sends the local model parameters to the central server. After the central server collects all the model parameters uploaded by the client, they are weighted and aggregated to obtain a new global model. Finally, the new global model is issued to all clients, and the clients perform the next round of iterative updates on this basis until for convenience, let $f(x_j, y_j; \omega)$ or $f_j(\omega)$ represents the loss function of a sample j. In this framework, data owners act as clients to cooperate in training machine learning models. Specifically, the training

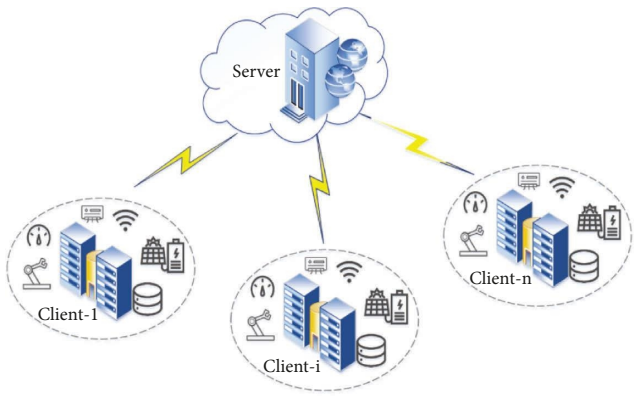

Figure 1: Typical architecture of federated learning in IoT.

process of joint learning can be summarized as initialization, local model training, and global aggregation.

Traditional federated learning cannot evaluate the private data value of clients, so the contribution of each client cannot be taken into account when the model is aggregated. The local model trained by the client using the local dataset reflects the value of the local dataset. If the contribution of each local model to the federated task can be assessed, it will indirectly reflect the value of the client's local dataset to the federated task. To achieve this goal, Federated Learning Framework Based on Data Value Evaluation (FLDVE) is proposed to evaluate the contribution of local models to federated tasks as the data value of the clients.

3.2. FLDVE. As shown in Figure 2, FLDVE has two roles of client and server. The responsibility of the client is the same as in traditional federated learning. A client receives the model sent by the server, then trains the model using its local dataset and uploads the parameters of the trained model to the server. Different from traditional federated learning, we set up an evaluator in the server to evaluate each client's contribution to the federated global model and also set up a validation dataset in the server. The validation set is a manually designed dataset without label noise. The evaluator is a deep neural network that learns the contribution of each client-uploaded model to the global model while obtaining a reward signal from the validation set. In FLDVE, the server collects the local model parameters uploaded by each client and then takes the model parameters as the input of the estimator, and the output of the estimator is the selection probability of each local model. When the server aggregates all the local models, the selection probability is used as the aggregation weight of the corresponding local model to update the global model. In this FLDVE setting, the action of the agent (evaluator) is its model selection, and the environment, that encompasses the federation model training and evaluation, correspondingly gives a reward for each action, based on the state of current round models. The server calculates the reward signal using the test accuracy of the aggregated global model on the validation set and then applies the reward signal to the evaluator to make the evaluator reach a new state. In general, both the federated task and the estimator are directed toward improving the test accuracy of the global model on the validation set.

In this paper, we assume that the federated learning system is conducted in a secure and trusted environment. In other words, all clients in this scenario are honest but curious, especially since the central server is trusted. The standard validation set D_v on the central server side is chosen according to the needs of the task model owner and is private to the central server. In the experiment, since the encryption/decryption has nothing to do with the scheme proposed in this paper, the encryption/decryption process for model transmission is omitted.

Reinforcement learning mainly consists of three parts: the agent, the environment, and the reward function, and it focuses on how the agent chooses the optimal action to maximize the cumulative reward given by the environment. The strategy of reinforcement learning determines the action of the agent; that is, the agent will output the action to be performed for a given input according to the strategy. The strategy is generally denoted as π. The combination of the s output by the environment and the action output by the agent is a trajectory, that is,

$$\tau = \{s_1, a_1, s_2, a_2, \ldots, s_t, a_t\}. \tag{1}$$

For a given agent's parameter ϕ, we can calculate the probability that a certain trajectory τ occurs as follows:

$$\begin{aligned} p_\phi(\tau) &= p(s_1)p_\phi(a_1|s_1)p(s_2|s_1,a_1)p_\phi(a_2|s_2)p(s_3|s_2,a_2)\cdots \\ &= p(s_1)\prod_{t=1}^{T} p_\phi(a_t|s_t)p(s_{t+1}|s_t,a_t). \end{aligned} \tag{2}$$

The cumulative reward for this trajectory is

$$R(\tau) \coloneqq \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t). \tag{3}$$

The goal of policy π_ϕ to maximize the expectation is

$$V^{\pi_\phi}(\mu) = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\phi}[R(\tau)]}. \tag{4}$$

Therefore, the policy gradient updating formula in the form of REINFORCE is

$$\nabla V^{\pi_\phi}(\mu) = \mathbb{E}_{\tau \sim \Pr_\mu^{\pi_\phi}}\left[R(\tau)\sum_{t=0}^{\infty}\nabla \log \pi_\phi(a_t|s_t)\right]. \tag{5}$$

The complete process of FLDVE is as follows:

Initialization. The federated server first determines the architecture of the federated model and randomly initializes the parameters of the global model. Then, the federated server sends the initialized model parameter ω^0 to each client.

Step 2: Local model training. In the t round of training, each client uses its local dataset to train based on the received global model ω_i^t, and updates to get a new local model ω_i^t. Then, each client sends the updated local model parameters ω_i^{t+1} to the central server. The goal of

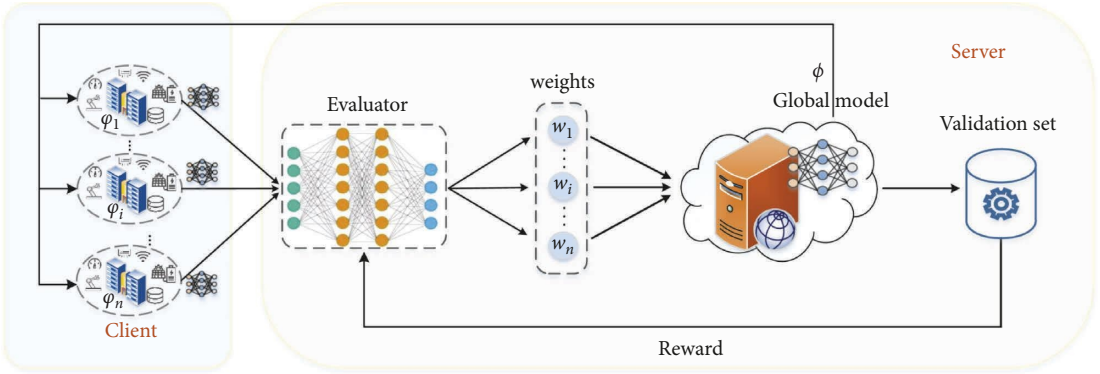

Figure 2: The structure diagram of FLDVE method.

client i in t-th round is to minimize the empirical loss $F(\omega_i^t)$ based on the local dataset, i.e.,

$$\omega_i^t = \underset{\omega_i^t}{\operatorname{argmin}} F\left(\omega_i^t\right),$$
$$F\left(\omega_i^t\right) = \frac{1}{|D_i|}\sum_{j\in D_i} f_j\left(\omega_i^t\right), \tag{6}$$

where $|D_i|$ denotes the number of samples in dataset D_i. The update process for each client can be implemented by performing stochastic gradient descent (SGD)

$$\omega_i^t = \omega_i^t - \eta\nabla F\left(\omega_i^t\right), \tag{7}$$

where $\eta\nabla F(\omega_i^t)$ represents the gradient of loss function, η is the learning rate.

Step 3: The central server collects the local models uploaded by all clients θ_i^t, which is introduced into the value evaluator to obtain the selection probability of each local model. The specific implementation is as follows.

In FLDVE, the central server side estimator is described as a proxy, and the local model parameter θ_i^t uploaded by the client will be used as the input of the central server estimator. The selection probability W^t corresponding to the local output of the evaluator. Then, when the models of the central server are aggregated, the selection probabilities of the estimators are used as weights for the corresponding local models to update the global model. After model aggregation, the reward signal R is calculated based on the accuracy of the global model on D_v.

In this paper, the reward function is defined to be directly related to the test accuracy of the aggregated global model on the server-side validation set D_v. The loss of the aggregated model is evaluated on a validation set and compared to the previous loss δ to determine the reward. The reward function $R(S^t)$ is calculated as follows:

$$R\left(S^t\right) = \frac{1}{|D_v|}\sum_{k=1}^{|D_v|} L_v\left(f_{\theta G}\left(x_k\right), y_k\right) - \delta, \tag{8}$$

where S^t is action space, $|D_v|$ is the size of the validation set D_v, L_v^t is the loss function of global model on validation set D_v, and δ is previous loss L_v^{t-1}. Contrary to FedAvg, the global model parameters are updated as follows:

$$\theta_G^{t+1} \leftarrow \theta_G^t - \frac{\alpha_\theta}{\sum_{i=1}^N W_i^t}\sum_{i=1}^N W_i^t \Delta\theta_i^t, \tag{9}$$

where θ_G^t is global model parameters for round $t+1$, α_θ is learning rate and $\alpha_\theta > 0$.

With the log-derivative trick, we have the following equation:

$$\nabla_\phi p\left(W^t|\phi\right) = p\left(W^t|\phi\right)\nabla_\phi \log\, p\left(W^t|\phi\right), \tag{10}$$

where

$$\nabla_\phi \log\, p\left(W^t|\phi\right) = \sum_{i=1}^n \left(W_i^t \nabla_\phi \log\, \omega_i^t + \left(1 - W_i^t\right)\nabla_\phi \log\left(1 - \omega_i^t\right)\right). \tag{11}$$

Then, the evaluator's model parameters ϕ can be optimized by gradient ascent method with learning rate α:

$$\phi^{t+1} \leftarrow \phi^t + \alpha_\phi \sum_{i=1}^n R\left(S^t\right)\nabla_\phi \log\, p\left(S^t|\phi\right)|_{\phi^t}. \tag{12}$$

Step 4: Global aggregation. In each round, the parameter server aggregates the local model parameters from the client and replaces the global model with the averaged model. Models trained with high-value data will obtain higher aggregated weights. Then, the aggregated global model parameters θ_G^{t+1} are sent back to each client. The global model parameters are updated as follows:

$$\theta_G^{t+1} \leftarrow \theta_G^t - \frac{\alpha_\theta}{\sum_{i=1}^N W_i^t}\sum_{i=1}^N W_i^t \Delta\theta_i^t, \tag{13}$$

where θ_G^t is global model parameters for round $t+1$, α_θ is learning rate and $\alpha_\theta > 0$.

Step 5: Then, the central server distributes the aggregated global model ϕ_{t+1} to each client.

The above steps will be repeated until the desired accuracy or required number of rounds is achieved.

4. Experiments

In this section, we evaluate the effectiveness of our scheme on different joint tasks based on MNIST and CIFAR10. Because these two data sets reflect the characteristics of IoT device data. We choose the accuracy of the model on the test set as the metric to demonstrate the effectiveness of FLDVE. The overall accuracy is calculated by summing the number of correctly predicted values and dividing by the total number of predicted values [28].

4.1. Experimental Setup. The training samples of the federated learning client may contain corrupted samples due to reasons such as the cheap labeling process of the training data. Automatically identifying datasets with corrupted samples is very beneficial for improving the accuracy of federated task models. Ideally, client models with a higher proportion of noisy labeled samples would be assigned lower weights when federated aggregation. In the following experiments, we add different proportions of label noise to the data samples of different clients to simulate the data label errors caused by mislabeling in real scenes.

In the experiments, a total of $c = 5$ clients participated in federated tasks, and each client participated in all FL rounds. Clients use their private data to train local models and then upload the model parameters to the central server, which is responsible for aggregating the collected submodels into a global model. We use the image classification task of the MNIST dataset as the federated task. Considering the host performance, we mainly use several small proxy datasets to test the algorithm's accuracy. In the experiments, CNN is chosen as the unified federated model, and the CNN has one fully connected layer (256 units and ReLU function), one softmax output layer, and two 5×5 convolutional layers. To fully study the two-gang learning, we selected 12,000 samples from MNIST, of which 10,000 samples were randomly divided into 5 groups equally and assigned to each client, and the remaining 2,000 samples were stored in a central server as a validation set. 2000 samples (except these 12000 samples) were randomly selected from the unused samples as the test set. To verify the ability of FLDVE to identify dataset anomalies in federated learning, a training set is generated using corrupted data. Label noise is added to some client-side datasets. For MNIST, we made the following sets of settings: (i) set-1: one of the client's data are added with 50% random label noise; (ii) set-2: add random label noise of 10%, 20%, 30%, 40%, 50% to 5 clients; (iii) set-3: one of the client's data are added with 90% of the specified label 9 noise.

4.2. Performance Evaluation. We implement a convolutional neural network as a federated model. Specifically, in this group of experiments, the CNN model consists of two 3×. It is composed of 3 volumes of integration layers, each of which has 16 and 32 ReLU activated channels and is normalized. Each convolutional layer is followed by a 2×2 max pooling. The last max-pooling layer is followed by two fully connected layers consisting of 64 units with ReLU activations and another 10 units with softmax activations. The batch size, the number of epochs for local training E, and the number of FL rounds are set to $B = 50$, $E = 30$, and 30, respectively. The optimizer for each client used SGD with a learning rate of 0.25.

Similar to the previous MNIST setting, we selected 3000 samples from CIFAR10, of which 2500 samples were equally divided into 5 groups and distributed to each client, and the remaining 500 samples were stored in the central server as the validation set. 500 samples were randomly selected from the unused samples as the test set. To verify the ability of FLDVE to identify data anomalies in FL, a CIFAR10 set was generated using corrupted data. Label noise is added to some of the clients' data sets.

For CIFAR10, we made the following sets of settings: (i) set-1: one of the client's data are added with 50% random label noise; (ii) set-2: add random label noise of 10%, 20%, 30%, 40%, 50% to 5 clients; (iii) set-3: one of the client's data are added with 90% of the specified label 9 noise.

FedAvg is the aggregation method of federated averaging in traditional federated learning. In DVRL, the accuracy of the predictive model is improved by removing low-value samples from the training dataset, and its implementation in federated learning is to remove the local models of the parties with low-value datasets when aggregating, we also call it DVRL.

The experimental results (Table 1, Figures 3 and 4) show that when noise labels are added to the client-side dataset, the accuracy of the traditional federated learning global model on the test dataset decreases compared to the noise-free case. The model accuracy of FLDVE drops less than traditional federated learning. Traditional federated learning cannot distinguish high/low-contribution clients, while the FLDVE method with an evaluator can evaluate the contribution of each client to the global model and give different weights when the models are aggregated. After assigning the corresponding aggregation weights to different clients through FLDVE, the accuracy of the global model is significantly improved. We clearly see that the accuracy of the global model decreases in the aggregation method of the DVRL method. The reason may be that after the local model of a client is removed, its positive contribution to the global model is also removed. In the field of Industrial IoT, there are usually few joint clients, and deleting some clients directly will have a greater impact on the global model.

Client datasets may contain low-value training samples, such as mislabeled or noisy samples. Models trained on datasets with low data quality often perform poorly, and methods to automatically assess data quality will be very helpful in distinguishing datasets with clean or noisy labels. FLDVE evaluates clients' contributions to the global model by using an estimator and a clean validation dataset on the federated server, and low-contribution models will be assigned lower aggregation weights when aggregating, which reduces their negative effect on the global model.

Table 1: The accuracy of the global model on the validation set.

Set		Methods		
		FedAvg	DVRL	FLDVE
MNIST	Clean	0.9960	—	—
	Set-1	0.990	0.984	**0.992**
	Set-2	0.982	0.978	**0.990**
	Set-3	0.974	0.970	**0.986**
CIFAR-10	Clean	0.7929	—	—
	Set-1	0.7589	0.7329	**0.7747**
	Set-2	0.6829	0.6722	**0.7010**
	Set-3	0.6778	0.6429	**0.6913**

The numbers in the table refer to the accuracy of the model in the test set. The bold values mark the maximum accuracy in a certain set.

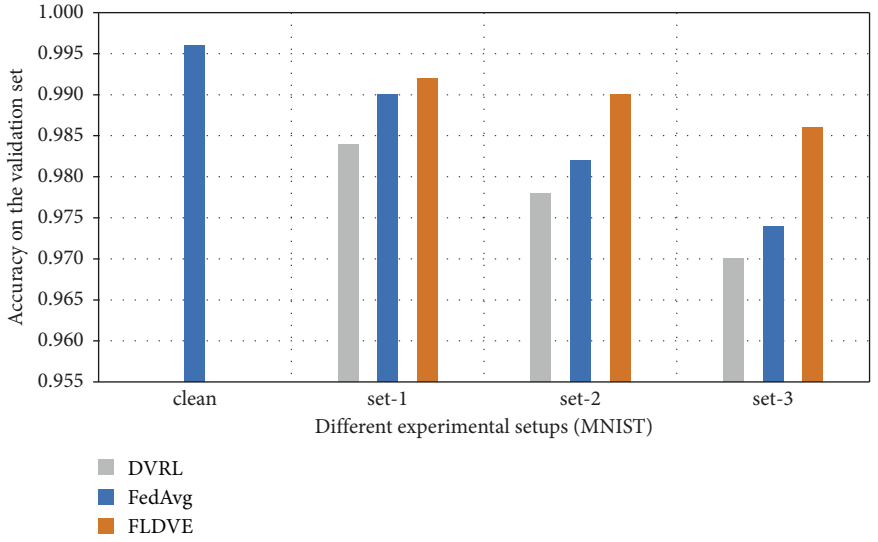

Figure 3: Accuracy of MNIST training set on verification set.

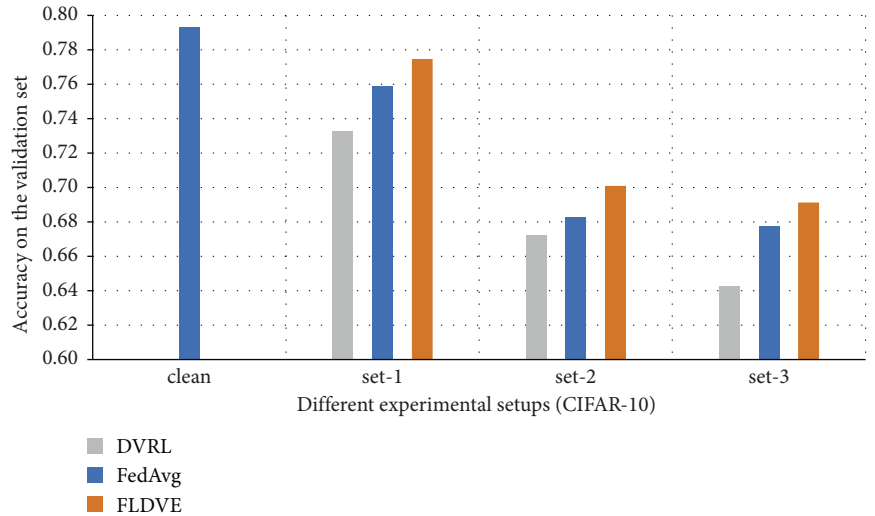

Figure 4: Accuracy of CIFAR10 training set on verification set.

For the client, when the proportion of correctly labeled samples in its data set is relatively large, the calculated model gradient is relatively stable, and it is easier to eliminate the influence of abnormal samples on the model training direction. Therefore, the value of the training dataset is considered to be positively correlated with the customer's

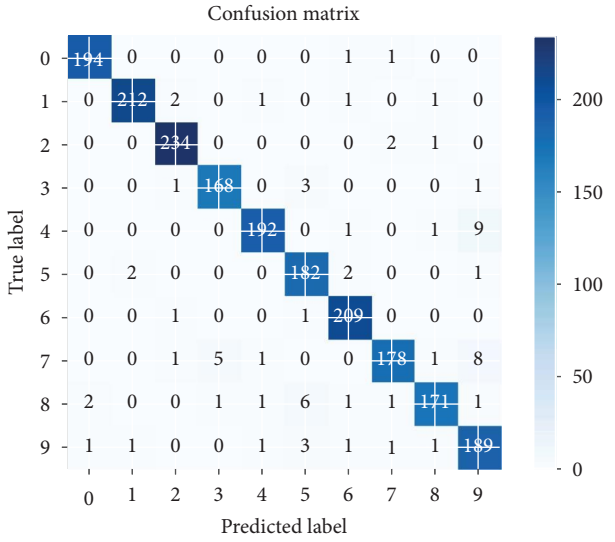

Figure 5: Confusion matrix of FLDVE on the test set at set-2. (MNIST).

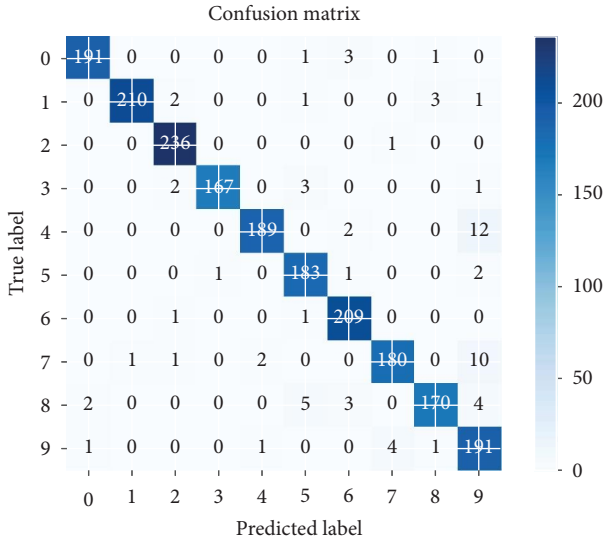

Figure 6: Confusion matrix of FLDVE on the test set at set-3. (MNIST).

contribution. The larger the dataset quality gap between clients, the more obvious the advantage of FLDVE, which is also validated on the CIFAR10 dataset (Figure 4). Experimental results show that FLDVE can evaluate the client's contribution, which in turn improves the accuracy of the federated global model. Confusion matrices (Figures 5 and 6) also show the effectiveness of the experiments on the test set.

5. Conclusion

Accurately assessing each federation client's contribution to the federation's mission is critical. In this paper, we propose FLDVE, a client-side data value-based federated learning framework that uses a reinforcement learninglike approach to evaluate client-uploaded models. When models are aggregated on the server side, client-side data values are used as weights for the local model, since it is obvious that data values can represent their contribution to the global model. For the client, when its dataset is clean, the parameters calculated by the model are relatively stable. Therefore, the number of clean data samples is positively related to the client's contribution to the federated task. The experimental results strongly support the ability of FLDVE to accurately assess the contribution of each client to the federated task and enable the global model with better accuracy. At the same time, this method can also be used to reduce the impact of poisoned clients on the global model. When the proposed FLDVE in this paper is applied to the IoT field, it can not only protect the data privacy of federated clients but also improve the accuracy of federated models by evaluating the value of client data. Future research may consider how to use the client contributions obtained by FLDVE to distribute benefits efficiently to all clients.

Data Availability

The simulation experiment data used to support the findings of this study are available from the corresponding author upon request. The MINST data used to support the findings of this study have been deposited in the MNIST database of handwritten digits (http://yann.lecun.com/exdb/mnist/). The CIFAR-10 data used to support the findings of this study have been deposited in the CIFAR-10 dataset (https://www.cs.toronto.edu/~kriz/cifar.html).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant no. 2020YFB2009501), the National Natural Science Foundation of China (Grant no. 62271456), the R&D Program of Beijing Municipal Education Commission (Grant no. KM202210005026), Major Research Plan of National Natural Science Foundation of China (Grant no. 92167102), and Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions (Grant no. CIT&TCD20190308). It was also supported by the Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education.

References

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: concept and applications," *ACM Trans. Intell. Syst. Technol.*vol. 10, no. 2, pp. 1–19, 2019.

[2] B. Hu, Z.-H. Guan, N. Xiong, and H.-C. Chao, "Intelligent impulsive synchronization of nonlinear interconnected neural networks for image protection," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3775–3787, 2018.

[3] R. Coulter, Q.-L. Han, L. Pan, J. Zhang, and Y. Xiang, "Data-driven cyber security in perspective—intelligent traffic analysis," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3081–3093, July 2020.

[4] G. Lin, S. Wen, Q.-L. Han, J. Zhang, and Y. Xiang, "Software vulnerability detection using deep neural networks: a survey," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1825–1848, 2020.

[5] S. Tu, M. Waqas, S. U. Rehman, T. Mir, Z. Halim, and I. Ahmad, "Social phenomena and fog computing networks: a novel perspective for future networks," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 32–44, 2022.

[6] C. Zhou, A. Fu, S. Yu, H. Wang, and Y. Zhang, "Privacy-preserving Federated Learning in Fog Computing," *IEEE Internet Things J.*vol. 7, no. 11, pp. 10782–10793, 2020.

[7] S. M. Nagarajan, G. G. Deverajan, P. Chatterjee, W. Alnumay, and U. Ghosh, "Effective task scheduling algorithm with deep learning for internet of health things (IoHT) in sustainable smart cities," *Sustainable Cities and Society*, vol. 71, p. 102945, Article ID 102945, 2021.

[8] H. B. McMahan, E. Moore, D. Ramage, and B. A. Arcas, "Federated Learning of Deep Networks Using Model Averaging," 2016, https://arxiv.org/abs/1602.05629.

[9] P. Kairouz, H. B. McMahan, B. Avent et al., "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2020.

[10] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: challenges, methods, and future directions," 2019, https://doi.org/10.48550/arXiv.1908.07873.

[11] S. Su, Z. Tian, S. Liang, S. Li, S. Du, and N. Guizani, "A reputation management scheme for efficient malicious vehicle identification over 5G networks," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 46–52, June 2020.

[12] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1486–1500, 2020.

[13] H. Huang, B. Zhang, Y. Sun, C. Ma, and J. Qu, "Delta-DAGMM: a free rider attack detection model in horizontal federated learning," *Security and Communication Networks*, vol. 2022, Article ID 8928790, 13 pages, 2022.

[14] M. Waqas, S. Tu, Z. Halim, S. U. Rehman, G. Abbas, and Z. H. Abbas, "The role of artificial intelligence and machine learning in wireless networks security: principle, practice and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5215–5261, 2022.

[15] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proc. Of ACM CCS*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds., pp. 587–601, Association for Computing Machinery, New York, NY, USA, 2017.

[16] Y. Li, Y. Li, H. Xu, and S. Ren, "An adaptive communication-efficient federated learning to resist gradient-based reconstruction attacks," *Security and Communication Networks*, vol. 2021, Article ID 9919030, 16 pages, 2021.

[17] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, pp. 2597–2604, IEEE, Los Angeles, CA, USA, December 2019.

[18] G. G. Deverajan, V. Muthukumaran, C.-H. Hsu, M. Karuppiah, Y.-C. Chung, and Y.-H. Chen, "public key encryption with equality test for industrial internet of things system in cloud computing," *Transactions on Emerging Telecommunications Technologies*, vol. 33, p. e4202, 2021.

[19] S. M. Nagarajan, G. G. Deverajan, P. Chatterjee, W. Alnumay, and V. Muthukumaran, "Integration of IoT based routing process for food supply chain management in sustainable smart cities," *Sustainable Cities and Society*, vol. 76, no. 2022, Article ID 103448, 2022.

[20] S. M. Nagarajan, G. G. Deverajan, U. Kumaran, M. Thirunavukkarasan, M. D. Alshehri, and S. Alkhalaf, "Secure data transmission in internet of medical things using RES-256 algorithm," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8876–8884, 2022.

[21] R. Sim, Y. Zhang, M. C. Chan, and B. K. H. Low, "Collaborative machine learning with incentive aware model rewards," 2020, https://arxiv.org/abs/2010.12797.

[22] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, pp. 2577–2586, Los Angeles, CA, USA, December 2019.

[23] R. Zeng, S. Zhang, J. Wang, and X. Chu, "FMore: an incentive scheme of multi-dimensional auction for federated learning in MEC," in *Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pp. 278–288, Singapore, November 2020.

[24] H. Yu, Z. Liu, Y. Liu et al., "A Sustainable incentive scheme for federated learning," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 58–69, 2020.

[25] R. Sutton and A. Barto, *Reinforcement Learning: An introduction*, MIT Press, Cambrige, MA, USA, 2017.

[26] S. Tu, M. Waqas, S. U. Rehman et al., "Reinforcement learning assisted impersonation attack detection in device-to-device communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1474–1479, 2021.

[27] J. Yoon, A. S. Arik, and T. Pfister, "Data valuation using reinforcement learning," *International Conference on Machine Learning*, PMLR, , July 2020.

[28] S. K. Punia, M. Kumar, T. Stephan, G. G. Deverajan, and R. Patan, "Performance analysis of machine learning algorithms for big data classification: ML and AI-based algorithms for big data analysis," *International Journal of E-Health and Medical Communications*, vol. 12, no. 4, pp. 60–75, 2021.