mathematics

MDPI

Article

A Semi-Federated Active Learning Framework for Unlabeled Online Network Data

Yuwen Zhou [1,2], Yuhan Hu [2,*], Jing Sun [1], Rui He [1] and Wenjie Kang [3]

[1] College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
[2] Science and Technology on Information Systems Engineering Laboratory, Changsha 410073, China
[3] Hunan Provincial Key Laboratory of Network Investigational Technology, Hunan Police Academy, Changsha 410125, China
* Correspondence: huyuhan16@nudt.edu.cn

Abstract: Federated Learning (FL) is a newly emerged federated optimization technique for distributed data in a federated network. The participants in FL that train the model locally are classified into client nodes. The server node assumes the responsibility to aggregate local models from client nodes without data moving. In this regard, FL is an ideal solution to protect data privacy at each node of the network. However, the raw data generated on each node are unlabeled, making it impossible for FL to apply these data directly to train a model. The large volume of data annotating work prevents FL from being widely applied in the real world, especially for online scenarios, where the data are generated continuously. Meanwhile, the data generated on different nodes tend to be differently distributed. It has been proved theoretically and experimentally that non-independent and identically distributed (non-IID) data harm the performance of FL. In this article, we design a semi-federated active learning (semi-FAL) framework to tackle the annotation and non-IID problems jointly. More specifically, the server node can provide (i) a pre-trained model to help each client node annotate the local data uniformly and (ii) an estimation of the global gradient to help correct the local gradient. The evaluation results demonstrate our semi-FAL framework can efficiently handle unlabeled online network data and achieves high accuracy and fast convergence.

Keywords: network data; federated learning; unlabeled data; heterogeneous data

MSC: 68T09

Citation: Zhou, Y.; Hu, Y.; Sun, J.; He, R.; Kang, W. A Semi-Federated Active Learning Framework for Unlabeled Online Network Data. *Mathematics* **2023**, *11*, 1972. https://doi.org/10.3390/math11081972

Academic Editors: Adrian Sergiu Darabant, Diana-Laura Borza and Catalin Stoean

Received: 22 March 2023
Revised: 12 April 2023
Accepted: 20 April 2023
Published: 21 April 2023

1. Introduction

Along with the explosion of data in devices and network terminals, an ever-increasing number of AI applications and services relying on these devices/terminals are emerging. Nevertheless, subject to laws on data privacy protection, the traditional centralized or decentralized training paradigm of the AI model is no longer feasible in many scenarios [1]. The phenomenon that devices/terminals are unwilling to share their private data which hinders the centralized training is called "data island". To this end, Federated Learning (FL) [2], a novel AI model training and inference framework, is promoted and introduced in many network edge intelligence applications, e.g., network anomaly detection [3] and internet traffic classification [4]. As an effective solution to deal with the "data island" problem and protect data privacy, FL aggregates various network nodes and uses their local parameters or gradient information. Therefore, it trains

a global model together without data moving and sharing. It is practical with respect to protecting data privacy.

In an FL application, the task is defined before learning begins, making FL a typical task-driven learning paradigm. Thus, as a task-driven approach [5], supervised learning is widely used to train a model with explicit functions in FL. For example, models for anomaly detection and attack classification in cybersecurity [6] are all trained via supervised learning.

Note that model training through supervised learning usually needs a large amount of labeled data, whereas data generated in most network nodes lack labels. Consequently, FL cannot be directly applied for secure model training with network data.

In previous work, efforts have been made to address the typical challenges of FL, i.e., communication efficiency, heterogeneous data, limited computation, incentive mechanism, etc. Unfortunately, almost all of the previous works have assumed that the local data of each node are perfectly ready to be used for training. Network data are unprocessed and unlabeled, while model training is completed via supervised learning with labeled data in most scenarios. Thus, it is impractical to execute model training directly with local unlabeled data.

Due to the particular characteristics of network data, we face several challenges when applying FL to network data. First, it is hard to label all the data (namely data annotation) in network applications. For those online network applications, e.g., network anomaly detection [3], data are constantly being generated, making accurate data annotation extremely costly. Thus, it is critical and challenging to minimize the cost of data annotation while maximizing the model benefit. Second, due to the independence of data annotation on each node, the annotated labels could be inconsistent, i.e., different labels may appear for the same data class. For example, in labeling the types of network attacks, the denial of service can be marked as "DoS" in a node, while "DDoS" is used as the label in another node. This issue could bring trouble to FL model training since the standard and uniform label is required for single-task model training. Nevertheless, besides the challenges of annotating data, non-IID data are another crucial challenge in FL, especially in scenarios where local data are unlabeled. As one of the basic technologies in the field of cyberspace security, internet traffic classification is also affected by non-IID data [4]. In addition, experimental studies in [7] show that even the existing state-of-the-art FL algorithms could not be optimal in all scenarios of non-IID data.

In summary, although FL learning could solve the "data island" problem and protect the privacy of data, it suffers from the gradient variance intensified by non-IID data. In addition, the pre-trained model and the global gradient estimation require the server to prepare the task-related data in advance. However, in reality, the local data are constantly generated and unable to be fully labeled, which deviates from the assumption of much state-of-art research. The above challenges have inspired us to design a new FL framework that could jointly address data annotation and non-IID issues. In this way, the new framework is expected to be used for network data. More specifically, we aim to answer the following significant questions in this work: (i) Is it possible to reduce the annotation workload by screening out the most crucial instances in current model training? (ii) Combined with data annotation, is there any way to address the issue caused by non-IID data during the federated optimization?

In this work, we pursue fast convergence of model training and high accuracy of model inference with unlabeled non-IID network data. The paper makes the following major contributions:

- To reduce the cost of data annotation, we introduce the idea of active learning [8] that a pre-trained model is used to test current unlabeled data and the instances with the wrong test result are selected to annotate manually. These manually annotated instances are used to train and update the pre-trained model.
- To eliminate the negative impact of non-IID data, we consider designing a gradient correction mechanism in which an unbiased estimation of the global gradient is used to correct the local gradient so that the gradient variance caused by non-IID data can be eliminated.
- Combining the advantages of active learning and FL, we design an accelerated **semi**-**f**ederated **a**ctive **l**earning (**semi-FAL**) optimization framework to handle the unlabeled and non-IID issues of local data using existing public historical data. The experiment result shows the higher accuracy, faster convergence and robustness of the proposed

framework semi-FAL compared with the other two typical federated learning frameworks.

The rest of the article is organized as follows. The following section reviews related literature. Next, the architecture design of semi-federated active learning is presented. Following that, operation details of semi-FAL are given. Then, the proposed architectures and mechanisms are evaluated in a case study. The discussion section compares our method with others. We also discuss future work in this section. The final section concludes the article.

2. Related Work

This section reviews existing solutions for training with unlabeled data in FL. Then, several representative methods are introduced to reduce the negative impact of heterogeneous data.

2.1. Solutions of Unlabeled Data Challenge

Data annotation is a practical challenge for the implementations of FL since the cost of annotation is generally high. Some studies have been conducted to find solutions for training with unlabeled data [9–11]. Federated Active Learning (F-AL) [9] is proposed to reduce the annotation cost through active learning and sample strategies. In F-AL, instances would be scored by an auxiliary model trained via FL and the instances with the highest scores would be annotated. Unsupervised learning is introduced in [10] and the federated of unsupervised learning method (FedUL) is proposed. In FedUL, the unlabeled client data are transformed into surrogate labeled data and the client model is modified to form a surrogate supervised FL task so that existing FL methods can be used. Dong et al. expected to make efficient use of distributed unlabeled medical data via a robust federated contrastive learning framework [11].

2.2. Solutions of Statistical Challenge

The statistical heterogeneity caused by non-IID data is a crucial factor, which impacts the practical application of FL. Currently, the performance of most FL algorithms can be better guaranteed with the IID data [2], while the convergence would be slowed down in non-IID settings [12]. A lot of studies have been conducted to tackle this issue [13–25]. These solutions are proposed from three aspects, i.e., local data, server setting and update rule. Zhao et al. expected to improve the local non-IID data by sharing a small subset dataset globally in [13]. Jeong et al. applied distillation and augmentation to improve the data distribution structure [14]. In addition, local batch normalization is also used to alleviate the feature shift caused by non-IID data in [16]. Instead of making an effort on local data, Xie et al. propose a multi-center FL framework where clients are divided into several groups according to the distribution of their local data [18]. In this multi-center framework, each center trains a global model. Since clients in the same group have similar distribution data, the global model trained in each center would not be heavily impacted by non-IID data. This multi-center FL framework applies a kind of clustering idea. Based on the clustering idea, more methods have been proposed, such as federated attentive message passing [21], the experience-driven control framework, FAVOR [22]. Moreover, some novel update rules have been designed to address the non-IID issue, such as SCAFFOLD [17], FedProto [19], FedPD [25], ASO-Fed [24]. In SCAFFOLD, control variates are used to reduce the impact of non-IID data. Instead of aggregating model parameters or gradients, FedProto transmits and aggregates prototypes. FedPD is designed from the primal-dual optimization perspective. An asynchronous update strategy is applied in ASO-Fed to tackle the heterogeneous issue.

All the above solutions are proposed under the assumption that local data have been annotated with the uniform rule, i.e., local data could be used to train the local and global models directly. However, data are often unlabeled after being generated in practice, especially in online scenarios. Motivated by the requirement to handle the challenges of data annotation and heterogeneous data together, we design an accelerated federated

optimization framework, semi-FAL, for unlabeled data in the online network. In our semi-FAL, a server is expected to supply the pre-trained model and the global gradient estimation for clients. For this target, a task-driven federated network building architecture is designed to find a node with sufficient computing and data resources as the server from the global perspective. Focus on the local of each node in the federated network, the data-driven collaborative annotation and computation architecture is designed to address the data annotation and non-IID data issues. Then, we propose two-phase model training operations under the two designed architectures, respectively. In phase I, we strategically match the data of network nodes with the task, choosing an optimal node with task-related historical data and plenty of computing resources as the server and other nodes possessing task-related data to form a client set. Accordingly, a basic network of semi-FAL consists of a server node and several client nodes. In phase II, within the basic network formed in phase I, the server would provide a pre-trained model to help select crucial instances and clients would annotate these instances with the unified standard. In addition, an unbiased estimation of the global gradient would be computed and delivered by the server to clients to reduce the gradient variance caused by non-IID data.

3. Framework Design of Semi-Federated Active Learning

3.1. Framework Overview and Design Requirements

In the past decade, the scale of the network (e.g., IoT) has increased dramatically, resulting in massive network data. As the essential part of understanding, managing and operating modern wide-area, data-center and cellular networks [26], most of these data would be unlabeled and non-IID. To realize the effective use of these data, we design a novel FL framework to apply these unlabeled and non-IID data to train a model with fast convergence and high accuracy, as illustrated in Figure 1. Specifically, the framework could be divided into two phases: federated network building and collaborative learning. These two phases have their focus. The former focuses on selecting suitable nodes from the global network to construct the federated network. At the same time, the significant points of the latter are the operations on each node to realize the collaborative data annotation and model training with unlabeled non-IID data. In the design, some basic requirements need to be followed.

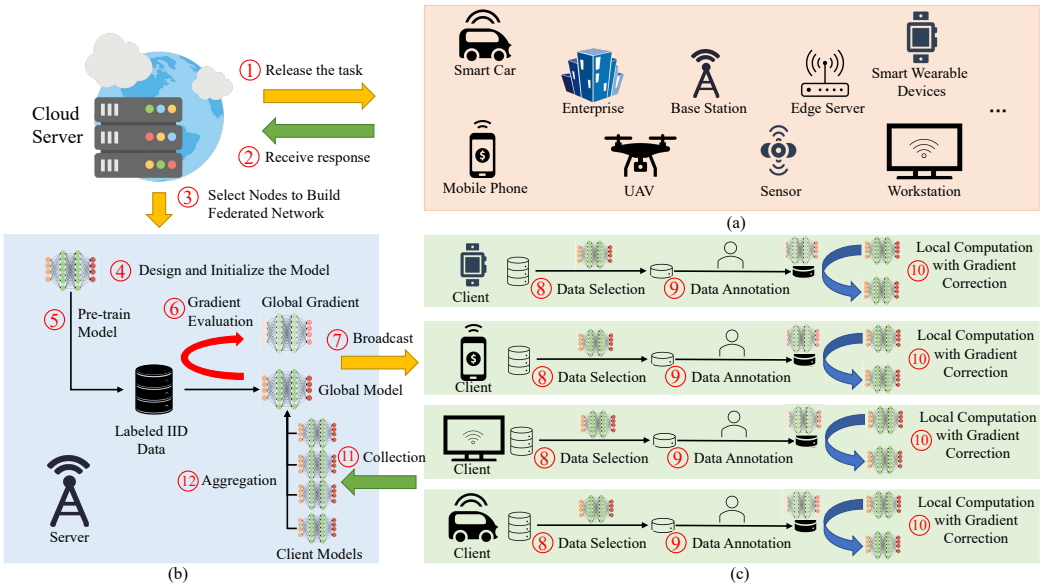


Figure 1. Semi-Federated Active Learning Framework: (**a**) task-driven servers and client selection; (**b**) server operation; (**c**) client operation.

- **Data Privacy Protection.** As the most critical point in data development and utilization, data privacy protection is the most important in our framework. To eliminate the

risk of privacy leakage, the raw data of each network node would be only processed and used locally. Moreover, the model and the gradient transmitted between the server and the client would be encrypted.

- **Robustness for different data and models.** The models and data used to complete the task are different in different scenarios. Thus, the framework we designed needs to be able to deal with different datasets and models in various scenarios.

3.2. Global Perspective: Task-Driven Federated Network Building

Instead of randomly selecting a group of network nodes to build a federated network, we adopt a task-driven server and client selection strategy. Generally speaking, by releasing the task in the global network, we could receive the response from the target group where various network nodes would be contained, such as mobile phones, smart cars, enterprises, etc., as illustrated in Figure 1a. As the base of our semi-FAL, we would like to select a node having sufficiently idle computing resources and task-related labeled data to be the server of the federated network. Nodes with good infrastructures, such as enterprises, base stations and edge servers, usually have more computing resources than those primarily used to provide applications. In addition, data are produced constantly. For historical data, it is made to be a public dataset, deleted or stored in some nodes. For example, historical data that are related to the business would be accessible and found in an enterprise node. Thus, an optimal node with plenty of computing resources and historical data could be found among the target group as the server.

Note that the data stored in the server are expected to be labeled and IID so that the model pre-trained by the server could distinguish each class of data and the global gradient estimated by the server could be approximately unbiased. The IID data could be constructed via data augmentation [27], such as flipping, translation, rotation, etc. In some extreme scenarios, it may be hard to find a node with all data classes to construct an IID dataset. Multiple nodes could be selected as a server group to pre-train a model federatively. In addition, the performance of the pre-trained model would be limited since the data used are just historical data; that is, the pre-training model can only be used as a coarse-grained model to correctly identify the part of unlabeled data. In the following section, we disclose more details on how to use the data in the server to help the data annotation locally and to improve the performance of the final model. In this article, we name the task-driven server and client selection mechanism phase I operation toward the semi-FAL.

3.3. Local Perspective: Data-Driven Collaborative Annotation and Computation

As mentioned before, the server of our federated network would not just play a role in delivering, collecting and aggregating models but also supply some necessary computation with the data in the server to address the challenges of local unlabeled non-IID data. To reduce the cost of data annotation and improve the performance of the model effectively, we further design the collaborative annotation and computation architecture for FL. Specifically, a data-driven collaborative annotation and computation architecture is proposed to realize the unified annotation of local data and reduce the impact of non-IID data on model training. Note that data annotation and model training are interleaved so that this architecture would still be effective in online network settings.

The architecture and detailed design of a data annotation and training system in the server and the client are shown in Figure 1b and Figure 1c, respectively. The design includes two major components: the server and the client.

The Server: As the core of the federated network, the server would continue to undertake the same basic tasks as the server in general FL: global model design and initialization, global model broadcast, local model collection and local model aggregation. However, unlike the server in general FL, the server in our design possesses an IID dataset with labeled data. Thus, the initial model could be trained with this dataset before broadcasting the global model to the client. In addition, an unbiased estimation of the gradient for the

current global model would be computed with the dataset. Then, this gradient would be delivered with the global model to the client together in each round. This global gradient would play a significant role in reducing the gradient variance caused by non-IID data.

The Client: The client nodes are usually the terminal devices that are closest to the users. The data of the client are often raw and produced constantly. To make use of the data, it is necessary to annotate them first. However, manual annotation is costly; that is, it would be unreasonable to annotate all the new data generated in each round manually. In addition, although the cost of automatic annotation via a trained model is low, the effect of annotation would be inferior. Thus, we draw on the idea of active learning. In each round of training, the global model is first used to test the local data and then the instances with wrong test results would be screened out to be annotated manually. After local data annotation, local training is ready to be executed.

With the support of the above data annotation and model training mechanism, FL could be executed with unlabeled and non-IID data. More details on the computing of semi-FAL are disclosed below. Specifically, we name the data-driven collaboration annotation and computation mechanism phase II operation toward the semi-FAL in this article.

4. Operations of Semi-Federated Active Learning

In this section, we first introduce relevant entities involved in the designed architectures and then present details about the aforementioned two key phases in achieving the semi-FAL.

4.1. Involved Entities

Cloud Server: A cloud server is selected to complete the overall coordination of the framework. As illustrated in Figure 1, the cloud server is mainly used to choose suitable nodes from the global network to build the federated network so that the training task can be completed. Additionally, the construction of the federated network would directly affect the total effect of the task with the network data.

Computation-intensive Nodes: The computation-intensive nodes mainly denote network nodes with a well constructed computing environment, such as base stations, edge servers and enterprises. In addition to sufficient computing resources, some historical data would be stored in these nodes. The above conditions fit our requirements for the server in our semi-FAL. Thus, these nodes are the primary candidates for the server in our federated network. The filtering of these nodes could be done through the feedback of nodes after releasing the task.

Terminal devices: Terminal devices are the main force of data generation and usually play the role of the client in a federated network. They are closest to users and the real environment of various applications. The model trained via FL or our semi-FAL would finally be deployed in terminal devices to supply the intelligent services. Thus, these nodes often have the most relevant data for target model training. However, data processing and model training are both energy-intensive processes and the terminal device, especially mobile devices, tends to have limited battery storage. The quality of user experience (QoE) provided by the terminal device is determined by the service response and battery life of the device. Thus, to ensure a good QoE, the computations performed locally are preferably lightweight and fast so as not to occupy and consume too many computing and battery resources. A meaningful way to reduce the cost of model training with unlabeled data is that only the critical instances are selected to be annotated and used to train the local model.

4.2. Phase I: Establishment of Federated Network

Nodes Sets: To build a federated network for the current task, the first step is to identify all nodes that are willing and able to participate in the task. According to the actual conditions of these nodes and the requirements for the server in our semi-FAL, they could be divided into two sets: the server set dominated by computation-intensive nodes and the client set dominated by terminal devices. More factors, such as the connectivity with the

other nodes, the cost to set this node as the server, etc., need to be considered for the server node. For the client node, whether it could provide the manual annotation of the data also needs to be considered.

Federated Network: A general federated network consists of a server and several clients. Given the above two sets of nodes for building the federated network, an optimal server node is expected to be selected to connect to as many client nodes as possible so that more network data can be used federatively to optimize the global model. Thus, the optimal server node would be selected from the server set through careful consideration of task-related data reserves, idle computing resources, communication resources and connectivity in the network.

Further Considerations: We also consider some practical constraints in building the federated network. First, historical data are rare in some emerging fields, so it is hard to find a node with plenty of computing and data resources as a server. At this point, we could find a node in the client set via some incentives to serve as a server. Second, the scale of the client is so large that a server is not enough to sustain the network. Our framework still works when multiple servers are involved in building the federated network.

4.3. Phase II: Collaborative Data Annotation and Model Training

As we make use of the historical data for the model pre-training, the discrimination accuracy of the model to fresh unlabeled data cannot be high. Therefore, in phase II, we leverage the local data to federatively optimize the global model via iterating data annotation, local training and global aggregation operations, thus further improving the inference accuracy of the model.

Data Annotation (Minimize the Annotation Cost): In each client, to reduce the manual annotation cost as much as possible, we need to find the critical instances for local training in each round. As illustrated in Figure 2a, the global model accepted by the client would be used to annotate local unlabeled data automatically at first. This process is equal to the model test; that is, input the unlabeled instance into the model and then output the label of the instance. After outputting the label of all local data, the owner of these data would determine whether these labels are correct. In this process, the data of the same label would be presented to the owner in the form of a batch so that the mislabeled data can be easily detected. These mislabeled instances are the critical instances in this round of training. Therefore, these instances would be selected and annotated manually as shown in Lines 5–6 of Algorithm 1. Furthermore, this process could be replaced by some intelligent methods, such as setting the probability thresholds of model output for a different label. Note that this would lead to the worse non-IID issue where only the key instances of each client are used to execute the local training. Thus, we design a gradient correction mechanism to reduce the negative impact of non-IID data.

Model Training (Reduce the Impact of Non-IID Data): The essence of model training is to continuously optimize model parameters to adapt to training instances. Thus, different distributed data would correspond to different optimization directions. In other words, non-IID data would cause gradient variance, which would make the model deviate from the global optimal. The essence of reducing the negative impact of non-IID data is to eliminate the gradient variance. Therefore, we design a novel gradient descent strategy as illustrated in Figure 2b. After completing the data annotation, for an arbitrary node j, a local gradient estimation g_j^* would be computed with the key instances and the global model. In each epoch of local training, the gradient descent could be formulated as the following:

$$w_{j,r+1} \leftarrow w_{j,r} - \eta_t(\nabla f(x_r, w_{j,r}) - g_j^* + \widetilde{g})$$

where $w_{j,r+1}$ denotes the local model in epoch $r+1$, η_t is the learning rate, f_j denotes the local loss function, x_r denotes the data instance and $\widetilde{g}$ is the global gradient estimation. Intuitively, we use the global gradient estimation $\widetilde{g}$ calculated in the server as the main body and the difference between the real, local gradient and the local gradient estimation $(\nabla f(x_r, w_{j,r}) - g_j^*)$ as the increment to update the local model as illustrated in Figure 2c.

The track in yellow is on behalf of the training process of local model $w_{j,r}$ toward $w_{j,r+1}$ in the general scenario. The grey line close to it denotes the local gradient estimation g_j^*. The red arrows between the two tracks are key updates for the local model. The dotted line in grey represents the global gradient estimation $\widetilde{g}$. The dotted line in blue is the direction of using the global gradient. The semi-FAL tries to add the difference (red arrows) to the global gradient to relieve the gradient variance and preserve the update feature of each client. Therefore, the local model $w_{j,r}$ is conducted to be $w_{j,r+1}$. In this way, the model in each client node would be optimized in a uniform direction so that no gradient variance would be generated. The update rule of global gradient $\widetilde{g}$ and other calculation details are given in Algorithm 1.

Algorithm 1 Semi-FAL: Semi-Federated Active Learning with Unlabeled Data

Input learning rate η_t for clients, learning rate α_t for server, model $\widetilde{w}$ and global gradient $\widetilde{g}$
for each iteration $t = 1, 2, \ldots, T$ **do**
  $S_t \leftarrow$ (sample a set of devices randomly)
  for each device $j \in S_t$ **in parallel do**
    $Y_j = \widetilde{w}(X_j)$
    $x_j^* \leftarrow$ (select the fault instances manually via Y_j)
    Picks x_j^* and computes the local gradient g_j^*
    $g_j^* = \nabla f(x_j^*, \widetilde{w})$
    $w_{j,1} = \widetilde{w}$
    for each local round $r = 1, 2, \ldots, R$ **do**
      $w_{j,r+1} \leftarrow w_{j,r} - \eta_t(\nabla f(x_r, w_{j,r}) - g_j^* + \widetilde{g})$
    end for
    $\Delta w_j = w_{j,R} - \widetilde{w}$
  end for
  server aggregation:
  $\widetilde{w} \leftarrow \widetilde{w} + \alpha_t \frac{1}{|S_t|} \sum_{j \in S_t} \Delta w_j$
  $\widetilde{g} = \widetilde{g} + \frac{1}{|S_t|} \sum_{j \in S_t} g_j^*$
end for

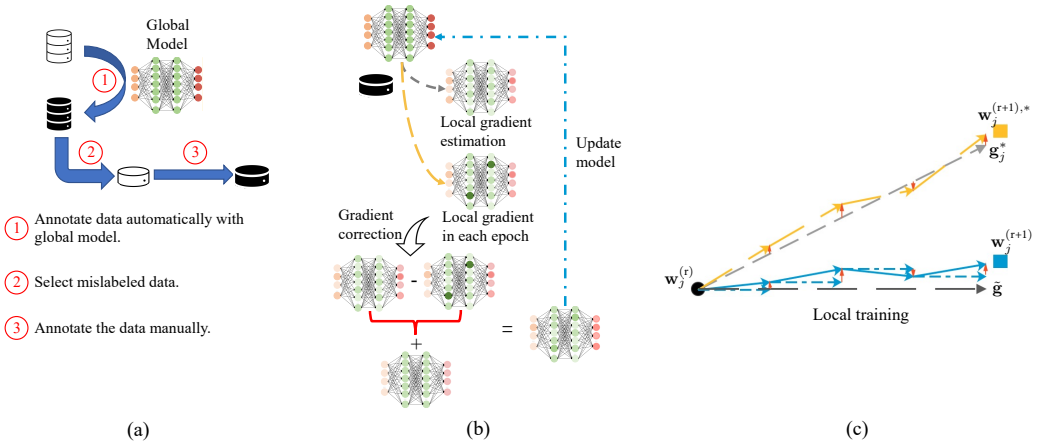


Figure 2. Detailed design of the collaborative data annotation and model training architecture in each client. (**a**) The annotation process of semi-FAL. The mislabeled data would be picked and labeled manually after being labeled by the model. (**b**) The model training process of semi-FAL. The semi-FAL uses gradient correction to direct the local gradient. (**c**) The update direction of local model.

Further Considerations: We further consider some practical considerations during the data annotation and the model training operations. Even if the semi-FAL could accelerate the process that experts label the new data, it still requires the expert to browse all labeled

results generated via the model. Moreover, the framework needs to exchange information between the server node and client nodes. This might arouse the risk of being attacked. Security assurance and communication efficiency are essential to the implementation of FL in reality. To address these issues, we propose some ideas.

First, to ensure that each round of training could be completed quickly, the maximum amount of data to be annotated in each client should be determined by the local computing power. Second, as the model and the global gradient are delivered to the client, the cost of communication in our framework would be higher. Thus, the compression of the global model and the global gradient should be executed via lossless compression techniques, such as Sparse Ternary Compression [28] and Sparse Dithering [29], before broadcast. Third, due to the data of the server being historical data, as new data are generated, the global gradients calculated using the data of the server would be biased. Therefore, we could consider computing an unbiased global gradient estimation in a client node with IID data.

5. A Case Study

5.1. Experimental Setting

Scenario: In this case study, we consider completing online image classification tasks with a federated network consisting of 1 server and 100 clients. A small IID dataset is pre-deployed in the server to simulate the historical data and this dataset is labeled. For the client, we control the increase of the local data in each round and these data are unlabeled. To make the experiment more realistic, the data added to each client node in each round would match the distribution of the local historical data.

Datasets: We choose two classical image classification datasets, MNIST (http://yann.lecun.com/exdb/mnist/, accessed on 26 August 2022) and Fashion-MNIST (https://github.com/zalandoresearch/fashion-mnist, accessed on 26 August 2022), to evaluate the performance of our semi-FAL. MNIST is a handwritten digit dataset containing 60,000 training instances and 10,000 test instances. The Fashion-MNIST dataset consists of 60,000 28×28 greyscale images in 10 classes, with 6000 images per class. To investigate the robustness of our proposed framework for non-IID data, two data setting schemes are given: (1) IID setting: an equal amount of data is allocated to each client randomly. (2) Non-IID setting: to simulate non-IID data, the dataset is always divided and distributed to clients manually. In our design, the data are sorted via labels and then two classes of data are assigned to each client. Note that, to simulate the sequential data in an online network, we control the local data of each client increasing constantly.

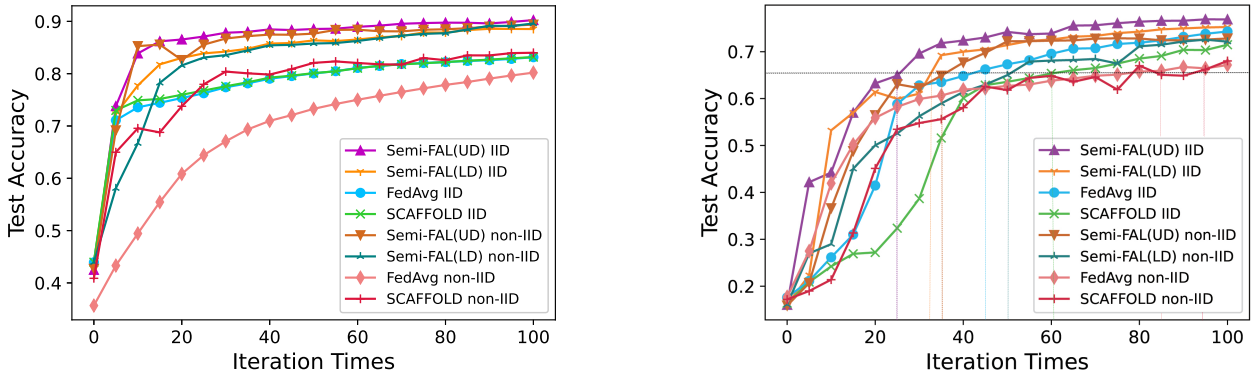
Model: We design two models, **l**ogistic **r**egression (LR) for MNIST and **c**onvolutional **n**eural **n**etwork (CNN) for Fashion-MNIST, in our experiments. Specifically, the CNN is designed with two 5×5 convolution layers (the first with 32 channels, the second with 64, each followed with 2×2 max pooling), a fully connected layer with 512 units and ReLu activation and a final softmax output layer (1,663,370 total parameters).

Benchmarks: We compare the performance of our semi-FAL with two typical FL frameworks, FedAvg [2] and SCAFFOLD (Stochastic Controlled Averaging algorithm) [17]. To ensure the fairness of the comparison, the pre-trained model is set as the initial model in each setting. Specifically, the model test accuracy from the following four settings is compared:

- Semi-FAL(UD): The model is trained with **u**nlabeled **d**ata (UD) and local training is executed with the key instances.
- Semi-FAL(LD): The model is trained with **l**abeled **d**ata (LD) and local training is executed with all local data.
- FedAvg: The model is trained with all local labeled data and local models are directly aggregated via weighted average without gradient correction.
- SCAFFOLD: The model is trained with all locally labeled data through SCAFFOLD. It uses a control variance to correct the 'client-drift' in its local updates.

5.2. Results and Analysis

Semi-FAL yields higher accuracy with benchmarks. Figure 3 summarizes the performance of semi-FAL compared with benchmarks. Figure 3a shows the result of applying different FL models using MNIST, LR. The four models using semi-FAL, no matter whether it is non-IID or IID and unlabeled data or labeled data, could obtain the highest accuracy after 100 iterations. Other contrast methods behave much worse than semi-FAL. The basic model FedAvg using non-IID data achieves the lowest accuracy. This means simply taking the average of the gradient generated from each client node might be inefficient when data are non-IID; more iterations should occur to increase the accuracy. Figure 3b is the result of using Fashion-MNIST, CNN also through 100 iterations. Under this scenario, all of the models behave relevantly unstably. Semi-FAL still obtains the best performance. Figure 3a,b both show that under the same model and dataset, the performance of the model trained via semi-FAL with unlabeled non-IID data is close to that of the model trained with unlabeled IID data. This strongly proves the effectiveness of the gradient correction mechanism in overcoming non-IID data issues.



(**a**) *MNIST*, LR (**b**) *Fashion-MNIST*, CNN

Figure 3. The result of the case study: test accuracy vs. iteration rounds for the MNIST, LR and FEMNIST, CNN in IID and non-IID data settings.

Semi-FAL achieves faster convergence. In the MNIST, LR scenario, the semi-FAL(UD) using IID data could obtain high accuracy over 0.8 within 20 iterations but FedAvg and SCAFFOLD are unable to obtain the proximate performance even beyond 100 iterations. In the Fashion-MNIST, CNN scenario, the dotted lines show other methods need many more iterations than semi-FAL if they want to achieve accuracy at the same level. In addition, as is shown in the graph, the semi-FAL even performs better in unlabeled data than in labeled data scenarios. That is because semi-FAL could achieve faster convergence and high test accuracy with unlabeled data than with labeled data. The difference between these two settings is that a critical instance selection process would be executed before local training with unlabeled data and only the critical instances would be annotated and used to train the local model, while all labeled data would be used to train the model directly in the setting of labeled data. This scheme can select the most helpful instances for current global model training. In fact, this scheme can also be used in labeled data settings to accelerate the convergence of the model.

The performance of semi-FAL is robust to the model and data. We apply two different models, LR and CNN, and two different datasets, MNIST and Fashion-MNIST, to execute the empirical studies. The results in Figure 3 demonstrate that semi-FAL could achieve the best performance among all benchmarks under different model and data conditions. Note that, in all scenarios of our case study, the local data of each node are added constantly to simulate the online situation. Our semi-FAL is also robust to the online network data. However, it is shown that when applying MNIST, LR semi-FAL could achieve much higher accuracy than applying Fashion-MNIST, CNN after 100 iterations. This is common for all frameworks; the possible reason might be that the efficiency of

the CNN is impacted by federated learning and needs more iterations to obtain higher accuracy. The results inspire us to do more research on applying semi-FAL to different models and datasets.

6. Discussion

In the experiments, we compared semi-FAL with two typical FL frameworks. The results show the efficiency of semi-FAL. There are seldom other methods that struggle to combine active learning and FL. Ref. [30] assumes the participating users are willing to share requested data between neighbor users. This puts emphasis on using active learning to select critical data that helps achieve balanced data distribution. However, if the requested data are sensitive, it would violate the principle of FL. Ref. [31] applies federated active learning on medical images. However, the goal of the work is to accelerate the training phase of federated learning but it ignores the impact of non-IID data. Furthermore, it only uses the extant active learning method and federated learning method, while semi-FAL designs a new method to solve the gradient variance. Ref. [32] also attaches importance to the application of federated active learning rather than the improvement of the method. Although ref. [33] touches upon the phenomenon of non-IID data, the authors do not solve it explicitly. Our semi-FAL is a universal framework that could be treated as the complement of these methods.

Future work. There are some possible constraints as discussed in Sections 4.2 and 4.3; we would try to solve them in the next step. In addition, We use typical datasets and models in the experiments while there are lots of new investigations that use deep learning methods to process complex images, e.g., [34]. We could do further research concerning applying semi-FAL on various different models and datasets to check its consistent efficiency. Through our investigation, we found that federal learning is applied to different scenarios: intrusion detection [33], network traffic prediction [4,35], etc. Our semi-FAL shows its superiority on the benchmarks, but its performance in the real environment is still unknown. The gap between reality and experiment is considerable. The predictable challenges derive from the inherent characteristics of FL and we should make further improvements in communication efficiency. In the future, we would use the proposed framework to solve practical problems in reality and test its robustness.

7. Conclusions

In this article, we propose the semi-federated active learning framework to realize accelerated federated optimization for online network data. It is an FL framework designed for training with unlabeled network data. The global model annotates the unlabeled local data automatically. The mislabeled data are viewed as critical instances and they are used to train the local model. The server would estimate the global gradient and use it to correct the local gradient to reduce the negative impact of non-IID data. Results from a case study demonstrate that semi-FAL is effective in dealing with the data annotation and non-IID data issues to realize the fast convergence and high accuracy of model training.

Author Contributions: Software, Y.H.; Writing—original draft, Y.Z.; Writing—review & editing, J.S. and R.H.; Project administration, W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Excellent Youth funding of the Hunan Provincial Education Department [Grant No. 22B0938] and Hunan Province Legal Youth Research Project [Grant No. 22HNFX-D-004].

Data Availability Statement: The data supporting our reported results is open dataset. MNIST could be found in http://yann.lecun.com/exdb/mnist/, accessed on 21 March 2023, and Fashion-MNIST could be found in https://github.com/zalandoresearch/fashion-mnist, accessed on 21 March 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, Q.; Liu, Y.; Cheng, Y.; Kang, Y.; Chen, T.; Yu, H. Federated learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2019**, *13*, 1–207.
2. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Ft. Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
3. Zhao, Y.; Wang, L.; Chen, J.; Teng, J. Network anomaly detection based on federated learning. *J. Beijing Univ. Chem. Technol. Nat. Sci.* **2021**, *48*, 92–99.
4. Mun, H.; Lee, Y. Internet traffic classification with federated learning. *Electronics* **2020**, *10*, 27. [CrossRef]
5. Sarker, I.H. Machine learning: Algorithms, real-world applications and research directions. *SN Comput. Sci.* **2021**, *2*, 1–21.
6. Alazab, M.; Swarna Priya, R.M.; Parimala, M.; Reddy, P.; Gadekallu, T.R.; Pham, Q.V. Federated learning for cybersecurity: Concepts, challenges and future directions. *IEEE Trans. Ind. Inform.* **2021**, *18*, 3501–3509. [CrossRef]
7. Li, Q.; Diao, Y.; Chen, Q.; He, B. Federated learning on non-iid data silos: An experimental study. *arXiv* **2021**, arXiv:2102.02079.
8. Settles, B. *Active Learning Literature Survey*; University of Wisconsin: Madison, WI, USA, 2009.
9. Ahn, J.H.; Kim, K.; Koh, J.; Li, Q. Federated Active Learning (F-AL): An Efficient Annotation Strategy for Federated Learning. *arXiv* **2022**, arXiv:2202.00195.
10. Lu, N.; Wang, Z.; Li, X.; Niu, G.; Dou, Q.; Sugiyama, M. Federated Learning from Only Unlabeled Data with Class-Conditional-Sharing Clients. *arXiv* **2022**, arXiv:2204.03304.
11. Dong, N.; Voiculescu, I. Federated contrastive learning for decentralized unlabeled medical images. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Strasbourg, France, 27 September–1 October 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 378–387.
12. Li, X.; Huang, K.; Yang, W.; Wang, S.; Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv* **2019**, arXiv:1907.02189.
13. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-iid data. *arXiv* **2018**, arXiv:1806.00582.
14. Jeong, E.; Oh, S.; Kim, H.; Park, J.; Bennis, M.; Kim, S.L. Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data. *arXiv* **2018**, arXiv:1811.11479v1.
15. Briggs, C.; Fan, Z.; Andras, P. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–9.
16. Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; Dou, Q. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv* **2021**, arXiv:2102.07623.
17. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. Scaffold: Stochastic controlled averaging for federated learning. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 5132–5143.
18. Xie, M.; Long, G.; Shen, T.; Zhou, T.; Wang, X.; Jiang, J.; Zhang, C. Multi-center federated learning. *arXiv* **2021**, arXiv:2108.08647.
19. Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; Zhang, C. Fedproto: Federated prototype learning over heterogeneous devices. *arXiv* **2021**, arXiv:2105.00243.
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. Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; Zhang, Y. Personalized cross-silo federated learning on non-iid data. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 7865–7873.
22. Wang, H.; Kaplan, Z.; Niu, D.; Li, B. Optimizing federated learning on non-iid data with reinforcement learning. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Virtual, 6–9 July 2020; pp. 1698–1707.
23. Shoham, N.; Avidor, T.; Keren, A.; Israel, N.; Benditkis, D.; Mor-Yosef, L.; Zeitak, I. Overcoming forgetting in federated learning on non-iid data. *arXiv* **2019**, arXiv:1910.07796.
24. Chen, Y.; Ning, Y.; Slawski, M.; Rangwala, H. Asynchronous online federated learning for edge devices with non-iid data. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), IEEE, Virtual, 10–13 December 2020; pp. 15–24.
25. Zhang, X.; Hong, M.; Dhople, S.; Yin, W.; Liu, Y. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *arXiv* **2020**, arXiv:2005.11418.
26. Warraich, E.; Shahbaz, M. Constructing the face of network data. In Proceedings of the SIGCOMM'21 Poster and Demo Sessions, Virtual, 23–27 August 2021; pp. 21–23.
27. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]
28. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef]
29. Albasyoni, A.; Safaryan, M.; Condat, L.; Richtárik, P. Optimal Gradient Compression for Distributed and Federated Learning. *arXiv* **2020**, arXiv:2010.03246.
30. Shullary, M.H.; Abdellatif, A.A.; Massoudn, Y. Energy-Efficient Active Federated Learning on Non-IID Data. In Proceedings of the 2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS), Online, 7–10 August 2022; pp. 1–4.
31. Deng, Z.; Yang, Y.; Suzuki, K.; Jin, Z. FedAL: An Federated Active Learning Framework for Efficient Labeling in Skin Lesion Analysis. In Proceedings of the 2022 IEEE International Conference on Systems, Man and Cybernetics (SMC), Prague, Czech Republic, 9–12 October 2022; pp. 1554–1559.
32. Ahmed, U.; Lin, J.C.W.; Srivastava, G. Semisupervised Federated Learning for Temporal News Hyperpatism Detection. *IEEE Trans. Comput. Soc. Syst.* **2023**, 1–12. [CrossRef]

33. Naeem, F.; Ali, M.; Kaddoum, G. Federated-Learning-Empowered Semi-Supervised Active Learning Framework for Intrusion Detection in ZSM. *IEEE Commun. Mag.* **2023**, *61*, 88–94. [CrossRef]
34. Elhanashi, A.; Lowe, D., Sr.; Saponara, S.; Moshfeghi, Y. Deep learning techniques to identify and classify COVID-19 abnormalities on chest X-ray images. In Proceedings of the Real-Time Image Processing and Deep Learning 2022, Orlando, FL, USA, 3–7 April 2022; Volume 12102, pp. 15–24.
35. Sanon, S.P.; Reddy, R.; Lipps, C.; Schotten, H.D. Secure Federated Learning: An Evaluation of Homomorphic Encrypted Network Traffic Prediction. In Proceedings of the 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2023; pp. 1–6.