Gossip Distillation: Decentralized Deep Learning Transmitting neither Training Data nor Models

Taisuke Moriwaki
Tokyo Institute of Technology

Kazuyuki Shudo
Kyoto University

***Abstract*—While deep learning requires a large amount of training data to obtain a highly accurate model, it is not always possible to collect the data in one place for privacy and other reasons. Furthermore, eliminating centralized servers and allowing all nodes to communicate in a decentralized way, improves fault tolerance and eliminates the unfairness of servers getting learned models first.**

In existing methods, a node transmits a learned model to other nodes. Our proposal, Gossip Distillation is to apply Knowledge Distillation to communication between nodes. A node transmits inference results on common data, not the model itself. The method reduces the amount of communication and makes it possible to combine multiple different models for each node. For CINIC-10, only 5.18 MiB of the inference results are transmitted between nodes though existing methods requires 49.03 MiB of ResNet-18 as the main model to be transmitted. In addition, the proposed method allows different sub models to be trained in parallel with the main model. Achieved accuracies are comparable with existing centralized methods.

***Index Terms*—Decentralized deep learning, gossip, Knowledge Distillation**

I. Introduction

Training deep neural networks typically requires large and diverse data sets. It hinders learning with data that are distributed to multiple organizations and cannot be shared with others. Federated Learning [1] have enabled deep learning without sharing data among learning nodes.

But Federated Learning comes with the constraint that users must trust a centralized server and the organization operating it. To address this issue, decentralized distributed learning has become increasingly important in modern society, with proposals such as Oguni et al. [10], [11].

In such decentralized learning, a node transmits a model to other nodes as well as Federated Learning. Our proposal, Gossip Distillation is to apply Knowledge Distillation to such communication between nodes. A node transmits inference results on common data, not the model itself. It reduces the size of data transmitted between nodes by several times as shown in Table IV and Figure 1. For CINIC-10, only 5.18 MiB of the inference results are transmitted though existing methods requires 49.03 MiB of ResNet-18 as the main model to be transmitted. In addition, the proposed method allows different sub models, DenseNet-121 and MobileNet V3 to be traind in parallel with the main model. The trained models achieve comparable accuracies with existing centralized methods.

This paper is organized as follows. Related work is shown in Section II. Section III presents our proposed method. Section IV shows experimental setups and results, and Section V summarizes our contribution.

II. Related work

A. Federated Learning

Federated Learning uses a central server and numerous connected devices, including mobile devices and IoT devices. In contrast to traditional centralized learning, where data is sent to the central server for training, Federated Learning sends model gradient information instead. The node devices perform training using this information, and then the model gradients are sent back to the server to integrate them into a new model. This process is repeated iteratively. As a result, data does not have to be shared with others, making it possible to maintain privacy.

B. Knowledge Distillation

Knowledge Distillation is a method for achieving high accuracy even with lightweight and simple models. This is to use the results of training a model with a large and complex structure for training a lightweight model.

This improves accuracy because more detailed data can be used for training than the usual training data given by 0–1.

C. Decentralized Learning via Adaptive Distillation (DLAD)

Decentralized Learning via Adaptive Distillation (DLAD) [6] is a centralized method that incorporates the concept of Knowledge Distillation into Federated Learning. While Federated Learning sends the gradient of the learned model to the server, DLAD infers the distillation dataset in addition to the learned model on the private dataset. The inference results are sent to the central server, which adaptively integrates them according to the contents of the private dataset, and the server trains a new model that can output the integrated results.

Contrary to its name, DLAD is a centralized method involving the central server. In contrast, our proposed method is decentralized.

D. Distributed machine learning with gossip

Gossip [7] is a method to multicast messages over a network and is often used in a distributed system. Gossip learning [8] is a decentralized distributed learning method using gossip. It is for classification problems with linear models, not for deep learning. This method is a privacy-conscious learning method that does not move data from one node to another

Proc. IEEE Virtual Conference on Communications
(IEEE VCC 2023), November 2023

and uses gossip as a communication method. Takahashi et al. [9] improved gossip learning by manipulating the selection probability according to the order of the nodes. Oguni et al. applied gossip-based communication to deep learning to achieve decentralized deep learning [10], [11]. They delat with the issue of heterogeneity of learning nodes [10] and resolved congestion by changing the communication frequency [11]. Hu et al. [12] tried to combine gossip into Federated Learning. Specifically, the weights of models owned by multiple clients are combined using the gossip technique. Therefore, gossip has been considered as a promising technique for decentralized learning.

III. Gossip Distillation

In this chapter, we describe the proposed method, Gossip Distillation.

A. Problem Formulation

Let $x \in \mathcal{X}$ be an input sample (e.g. image or sound), and $y \in \mathcal{Y}$ be the label for the input sample. In this research, we will focus on N-class classification problems where $\mathcal{Y} = \{1, 2, \ldots, N\}$.

To formulate the problem in classification learning, we consider the existence of multiple clients.

Assuming that there are N nodes $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_N$, each possessing a labeled dataset $\mathcal{D}_i = (\mathcal{X}_i, \mathcal{Y}_i)$ for $i \in 1, 2, \ldots, N$ where $\mathcal{X}_i = \{x_j^{(i)}\}$ and $\mathcal{Y}_i = \{y_j^{(i)}\}$, also $y_j^{(i)}$ is the label annotated to $x_j^{(i)}$.

The set $\mathcal{D}_i$ can only be viewed by node $\mathcal{N}_i$, and cannot be viewed by any other nodes $\{\mathcal{N}_j | j \neq i\}$ in the network.

Hereafter, we reffer D_i to as Private Data (PD).

Each node has its own model $\mathcal{M}_i : \mathcal{X} \to \mathcal{Y}_i$ learned using PD.

Since it can be fully assumed that real-world data is Non-IID, PD is also Non-IID, Therefore, even if the same image is inferred using $\mathcal{M}_i$ and $\mathcal{M}_j$ $(i \neq j)$, it is not necessarily guaranteed to yield the same result.

Our goal is to infer the output results of both $\mathcal{M}_i$ and $\mathcal{M}_j$ correctly and identically by utilizing each other's classification ability.

B. Distillation

To transfer the classification ability of a node's $\mathcal{M}_i$ to other models, we use the idea of Knowledge Distillation [5] in this study. In addition to PD, there exists a label-free dataset that can be viewed by any node, called the Distillation Dataset (DD). To create DD, a separate set of data, the Common Data $\mathcal{X}c$ (CD), is prepared apart from PD.

By using DD, the classification ability that is only possessed by the model owned by the node trained on PD can be propagated to other models. However, it becomes difficult to simultaneously send and receive knowledge from multiple nodes, so the gossip (pull) method is used to communicate one-on-one repeatedly.

Specifically, when a node $\mathcal{N}_{tgt}$, which owns the model $\mathcal{M}_{tgt}$, infers $\mathcal{X}_c$, it creates a list $\mathcal{L}_{tgt}$, and the number of times data has been transmitted so far, $\mathcal{C}_{tgt}$. The receiving node, $\mathcal{N}_{get}$, which has its own $\mathcal{X}_c$, receives $\mathcal{L}_{tgt}$ and $\mathcal{C}_{tgt}$, and creates its own list $\mathcal{L}_{get}$ and count $\mathcal{C}_{get}$. The lists are then integrated by a weighted average as follows:

$$\mathcal{L}mix^{(i)} = \frac{\mathcal{L}_{tgt}^{(i)} \times \mathcal{C}_{tgt} + \mathcal{L}_{get}^{(i)} \times \mathcal{C}_{get}}{\mathcal{C}_{tgt} + \mathcal{C}_{get}} \; i \in \{1, 2, \ldots, n\} \quad (1)$$

where n is the total number of nodes in the CD.

Then, we create a label list $\mathcal{Y}c$ using the maximum element of the list generated by Equation 1, i.e., $\mathcal{Y}c^{(i)} = \max \mathcal{L}mix^{(i)}$. After ward we create DD, $\mathcal{D}dist = (\mathcal{X}c, \mathcal{Y}c)$ with images as $\mathcal{X}c$ and labels as $\mathcal{Y}c$. DD is repeatedly regenerated every time another inference list is incorporated through communication, and then the model $\mathcal{M}_i$ is trained again using the regenerated DD. Ultimately, it is expected that each node's $\mathcal{M}_i$ will acquire the classification ability of all data in PD.

Algorithm 1 Gossip Distillation (local phase)

Input: Initial Weight W_0, Each model M_i, Each Private Data PD_i, Common Data $\mathcal{X}_c$, number of Nodes N
Output: Each Inference List IL_i, Trained Weight W_i

```
function LocalPhase(W_0, PD_i, DRS)
    for all each client number do
        W_i ← Update(W_0, PD_i)
        IL_i ← Inference(W_i, X_c)
    end for
end function
```

C. Proposed method

1) Local phase: First, training the model $\mathcal{M}_i$ on the PD_i owned by each node.

Then, after inferring the Distillation Dataset on $\mathcal{M}_i$, saving the inference results for each class. Specifically, we followed the algorithm 1.

2) Gossip phase: In this section, we update the model M_i by exchanging the IL_i generated in section III-C1 between nodes, creating a DD, and training it. We repeat this process for a number of rounds that has been set in advance.

1) Specify any node other than oneself.
2) Receive the inference list of that client and the number of times the loop is executed for the specified client.
3) Create DD according to equation 1.
4) Learn the model owned by oneself using DD.
5) Update the inference list for transmission by inferring the image of DD using the learned model.

Specifically, as shown in Algorithm2.

IV. Experiments

In this chapter, we will conduct experiments using the proposed Gossip Distillation method. We will compare the proposed method with DLAD (Section II-C) [6], which uses Knowledge Distillation in centralized distributed learning, as a previous study.

Algorithm 2 Gossip Distillation (gossip phase)

Input: Each Trained Weight W_i, Round count R, Target Node N_{tgt}, User Node N_{usr}
Each Node has set $N = (\mathcal{L}, \mathcal{C})$
function GOSSIPPHASE(R, N_{usr})
  for all Rounds $r = 1, 2, \ldots, R$ **do**
    $tgt \leftarrow$ Randomly selected node id
    download $N_{tgt} = (\mathcal{L}_{tgt}, \mathcal{C}_{tgt})$
    function MAKE DISTILLATION DATASET(N_{tgt}, N_{usr})
      $image$...Images for Distillation Dataset
      $label$...make as follows
      1. Integrate like Equation 1 by using (N_{tgt}, N_{usr})
      2. The highest-valued index of each element is the label.
      return $DD_i = (image, label)$
    end function
    $W_i \leftarrow Update(W_i, DD_i)$
    $IL_i \leftarrow Inference(W_i, DRS)$
  end for
end function

TABLE I
DATA VARIANCE USED IN THE EXPERIMENT.

	U_{5n+1}	U_{5n+2}	U_{5n+3}	U_{5n+4}	U_{5n+5}
IID	$0-9$	$0-9$	$0-9$	$0-9$	$0-9$
Non-IID #1	$0, 1$	$2, 3$	$4, 5$	$6, 7$	$8, 9$
Non-IID #2	$0-4, 5$	$0-4, 6$	$0-4, 7$	$0-4, 8$	$0-4, 9$
Non-IID #3	$0, 1, 2, 3$	$0, 4, 5, 6$	$1, 4, 7, 8$	$2, 5, 7, 9$	$3, 6, 8, 9$
Non-IID #4			i (U_i)		

TABLE II
NEURAL NETWORK MODELS USED IN EXPERIMENTS. (SIZE IS CAPACITY OF PARAMETERS IN THE MODEL)

Network	Number of parameters	Size (MiB)
ResNet-18	$11,689,512$	49.03
DenseNet-121	$7,978,856$	33.47
MobileNet V3 (small)	$2,542,856$	10.66

A. Datasets

In this section, we describe the datasets used in the experiment. We use MNIST, CIFAR-10, and CINIC-10 [13] in accordance with DLAD.CINIC-10 is a huge dataset consisting of 270,000 images extracted from CIFAR-10 and ImageNet, making it a challenging dataset.

In all datasets, we use 20% of the training dataset as private data and the remaining 80% as distillation data, simulating real-world situations where there are more unlabeled data than labeled data.

Regarding the dataset used for private data, we randomly select it but fix the random seed to ensure that it has the same distribution during the experimental stage. In addition, we split the PD into n clients, where each node $\mathcal{N}_i$ has a dataset $\mathcal{D}_i = (\mathcal{X}_j, \mathcal{Y}_j)_{j=1}^{n/i}$.

In this experiment, we split 20% of the data, which is 60,000 for MNIST and 50,000 for CIFAR-10, without duplicates. Although DLAD allows duplicates, we do not permit them in this experiment. As a result, the size of the PD owned by each client is about 2% of the training dataset.

For CINIC-10, 90,000 samples are prepared for both the training dataset and the validation dataset. Therefore, we can use the validation dataset as distillation data, but we chose to split the training dataset into 90,000 parts, similar to the ratio of MNIST and CIFAR, to create the aforementioned situation.

We tested using five different data distributions, as shown in Table I. In addition to the same distribution as DLAD, we added one distribution where the data distribution varies among clients.

- **IID** All clients have data with probability $p_i = [0.1, 0.1, 0.1, \ldots, 0.1]$.
- **Non-IID** $\#1$ Each client has data for 2 out of 10 classes. Specifically, $p_{5k+1} = [0.5, 0.5, 0, 0, \ldots, 0], p_{5k+2} = [0, 0, 0.5, 0.5, 0, \ldots, 0], \ldots$ such that each client has different classes.
- **Non-IID** $\#2$ All clients have data for classes 0–4, but only one client has data for classes 5–9. Specifically, $p_{5k+1} = [0.1, 0.1, \ldots, 0.1, 0.5, 0, \ldots, 0], p_{5k+2} = [0.1, 0.1, \ldots, 0.1, 0, 0.5, 0, \ldots, 0], \ldots, p_{5k+5} = [0.1, 0.1, \ldots, 0.1, 0, \ldots, 0, 0.5]$ such that each client has different classes.
- **Non-IID** $\#3$ Each client has data for 4 classes, and each class is owned by two different client groups i.e.,

$$p_k = \begin{cases} 0.25 & \text{(class included in private client)} \\ 0 & \text{(otherwise)} \end{cases}$$

  The classes owned by each client node are shown in Table I.
- **Non-IID** $\#4$ Each client has data for only one class, assuming a stricter situation than previous studies. Specifically, $p_{10k+1} = [1, 0, \ldots, 0], p_{10k+2} = [0, 1, 0, \ldots, 0], \ldots, p_{10k+10} = [0, \ldots, 0, 1]$ such that each client owns a different class.

The theoretical values of the accuracy of the models held by each node trained using only PD are expected to be (IID, Non-IID #1, Non-IID #2, Non-IID #3, Non-IID #4) $= (1, 0.2, 0.6, 0.4, 0.1)$.

B. Models

We use the Deep Residual Network (ResNet) [14] as the main client model for this experiment. For experiments using different models, we utilize the Densely-connected Convolutional Networks (DenseNet) [15] and MobileNet V3 (small) [16]. The former is used for comparison with DLAD, while the latter is used to investigate the impact of using a lightweight and simple model in conjunction with ResNet-18. The sizes of the three models used in this experiment are shown in Table II.

In each model, pre-trained weights were employed, followed by 200 epochs with a learning rate of 5×10^{-6} using SGD Optimizer, and a mini-batch of size 40 for the local round. In addition, only one epoch of learning is performed at each step in the gossip phase. The learning rate and mini-batch size

TABLE III
DETAILS OF THE MACHINE USED IN EXPERIMENTS.

	Machine specification
OS	Ubuntu 20.04.2 LTS
CPU	Intel Xeon Platinum 8368 Processor (38 core, 2.4 GHz) × 2
GPU	NVIDIA Tesla A100-PCIE-40GB

TABLE IV
SIZE OF DISTILLATION DATASET TRANSMITTED BETWEEN NODES AT EACH STEP IN THE GOSSIP PHASE (TO BE COMPARED WITH TABLE II).

Dataset	Size (MiB)
MNIST	3.46
CIFAR-10	2.88
CINIC-10	5.18

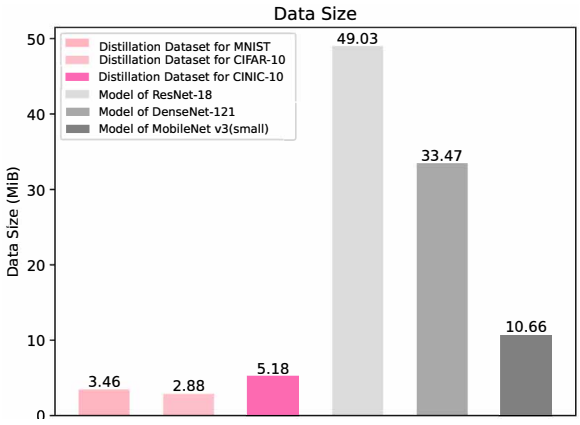

Fig. 1. Size of data transmitted between nodes at each step in the gossip phase.

were set to the same conditions as those used for training the local model.

C. Experiment setup

The specification of the machine used in all the following experiments is shown in Table III. A computer network and communication over it are simulated on the machine, but deep learning actually took place on the machine.

D. Experimemt details

In this experiment, we will first observe the accuracy under the same conditions as DLAD and verify the differences. Next, as explained above, we will conduct experiments combining ResNet-18, DenseNet-121, and MobileNet V3 (small), respectively, to compare and verify the accuracy. Consequently, we will also measure the size of IL_i transmitted in the gossip phase. This will allow us to compare the size of the IL_i transmitted with methods such as Federated Learning, which transmits the weights of the model.

E. Results

The proposed method reduced the size of data transmitted between nodes. Table IV shows the sizes of Distillation Datasets, that is transmitted between nodes at each step in the gossip phase. They are much smaller than the sizes of models transmitted in the existing techniques shown in Table VI. For CINIC-10, only 5.18 MiB of inference results are transmitted between nodes in the proposed method though the existing methods require 49.03 MiB of ResNet-18 as the main model to be transmitted. Figure 1 summarizes the sizes of Distillation Datasets and the sizes of models.

The proposed method achieved comparable accuracy with existing centralized methods. Table V and Figure 2 show the classification accuracy when training ResNet-18 on each dataset and method. The values represent the median accuracy on 10 nodes. The higher values in DLAD and Gossip Distillation are bolded. First, the overall accuracy of each model in the gossip phase is higher than the accuracy of each model in the local phase in all experiments, indicating that distillation makes sense. Comparing the accuracy in the gossip phase with that of DLAD and the case where all data are in one place (labeled), we found that the accuracy in the gossip phase is almost equal to, and sometimes exceeds, that of DLAD, and can even compete well with labeled.

In the local phase, the accuracy was low, due to the small number of data owned by the PDs, but the accuracy of each model with respect to the PDs was high regardless of IID or non-IID (>99% in MNIST and >75% in CINIC-10). This fact may be the reason for the significant improvement in accuracy for the test data in the gossip phase, soon after the start of the information transfer round. In addition, a comparison of the accuracy of the Non-IID4 pattern in the same data set shows that the results are almost the same. From this result, it can be inferred that all knowledge is spread over the rounds, regardless of the PD owned by each user. Figure 3 plots the accuracy at the end of the local phase and the accuracy in the gossip phase for the IID, Non-IID #1∼3 of CIFAR-10 in Table V. The graphs show that the accuracy increases in the early rounds, indicating the significance of Gossip Distillation.

The proposed method enabled combination of multiple different models. Table VI shows show achieved accuracy in case combining multiple different models. In this case, ResNet-18 and other neural networks were trained with 5 nodes each. Table VI shows the median accuracy of each of the five nodes. Other datasets also show similar or slightly higher accuracy than ResNet-18 alone. The results using MobileNet V3 (small) also show the same level of accuracy as those obtained using ResNet-18 alone. The results with MobileNet also show that Gossip Distillation is also accurate, although the accuracy is lower than that of ResNet. The results of the experiment using the same conditions, except that 10 models were replaced by 10 MobileNets, showed an accuracy of about 0.47, indicating that the same level of accuracy can be achieved by combining multiple models. This indicates that Gossip Distillation can be performed by combining different models.

V. CONCLUSION

In this study, we applied Knowledge Discovery to communication in decentralized deep learning. The proposed method reduced the size of data transmitted between nodes by several

TABLE V
ACCURACY. (MODELS ARE ALL RESNET-18, $n = 10$)

Dataset		MNIST					CIFAR-10					CINIC-10				
Distribution	IID	NIID1	NIID2	NIID3	NIID4	IID	NIID1	NIID2	NIID3	NIID4	IID	NIID1	NIID2	NIID3	NIID4
DLAD [6]	**0.9821**	**0.9820**	**0.9828**	**0.9840**		**0.7314**	0.6657	0.6847	0.7027		**0.6323**	**0.6266**	0.5666	0.5934	
labeled [6]	0.9836	0.9868	0.9845	0.9857		0.7115	0.8127	0.7576	0.7755		0.6256	0.6880	0.6183	0.6574	
Gossip Distillation (local phase)	0.7089	0.1861	0.4698	0.3504		0.3870	0.1669	0.2693	0.2798		0.3596	0.1592	0.2536	0.2609	
Gossip Distillation (gossip phase)	0.9644	0.9647	0.9669	0.9680	0.9715	0.7187	**0.7159**	**0.7181**	**0.7226**	0.7280	0.5935	0.6096	**0.5973**	**0.6064**	0.6157

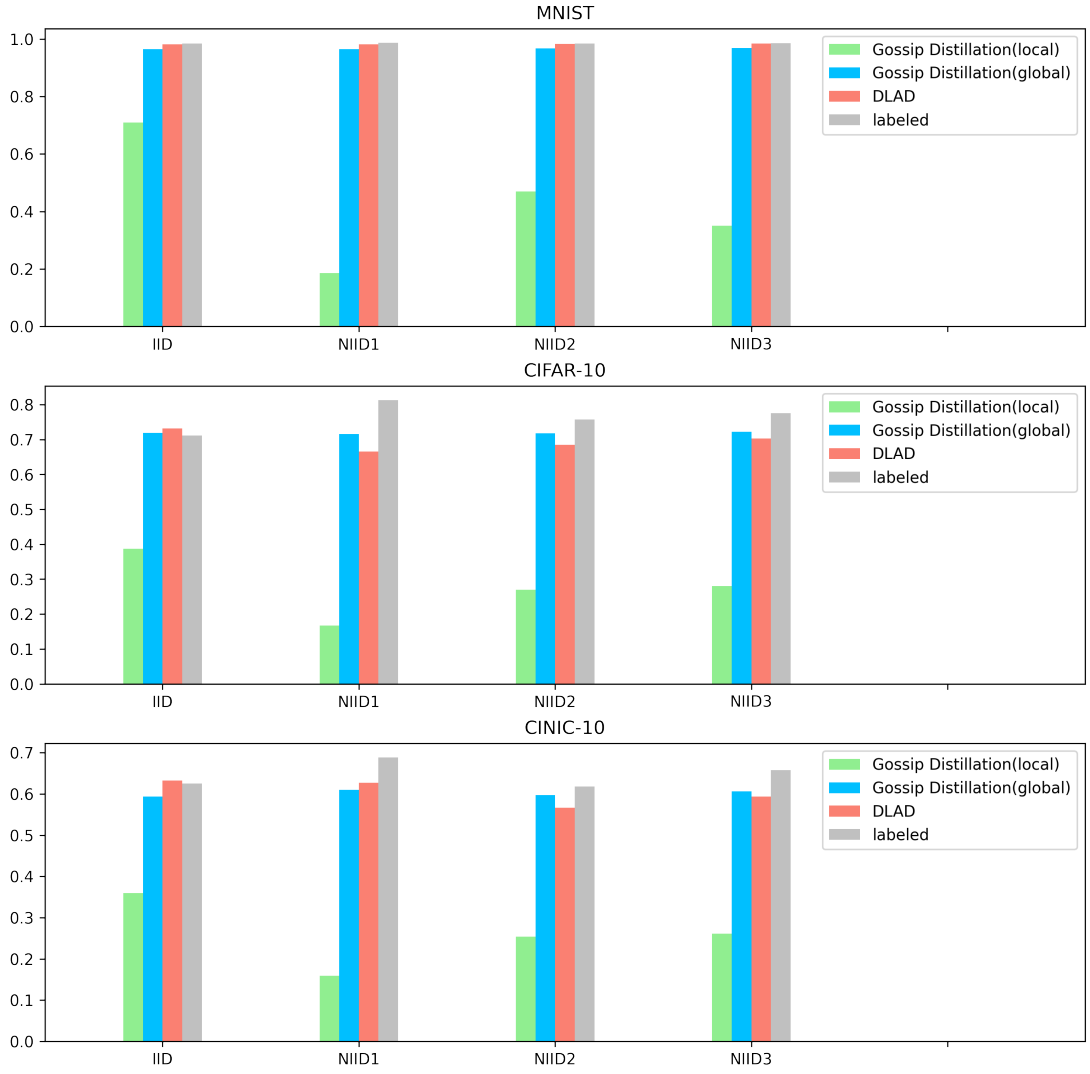

Fig. 2. Accuracy achieved at the end of local and gossip phases, compared with DLAD and the case where all data are in one place (labeled).

TABLE VI
ACCURACY FOR THE CASE WHERE MULTIPLE MODELS COMBINED ($n = 10$)

Main Model		ResNet-18			DenseNet-121		MobileNet V3(small)		
Sub Model			DenseNet-121						
Dataset		MNIST	CIFAR-10	CINIC-10	MNIST	CIFAR-10	CINIC-10		
Non-IID # 1	DLAD [6]	– / –	0.6657 / 0.6642	– / –	– / –	– / –	– / –		
Res/other	Gossip Distillation	0.9709 / 0.9781	**0.7279** / **0.7403**	0.6096 / 0.6180	0.9718 / 0.9540	0.7265 / 0.5550	0.6077 / 0.4796		

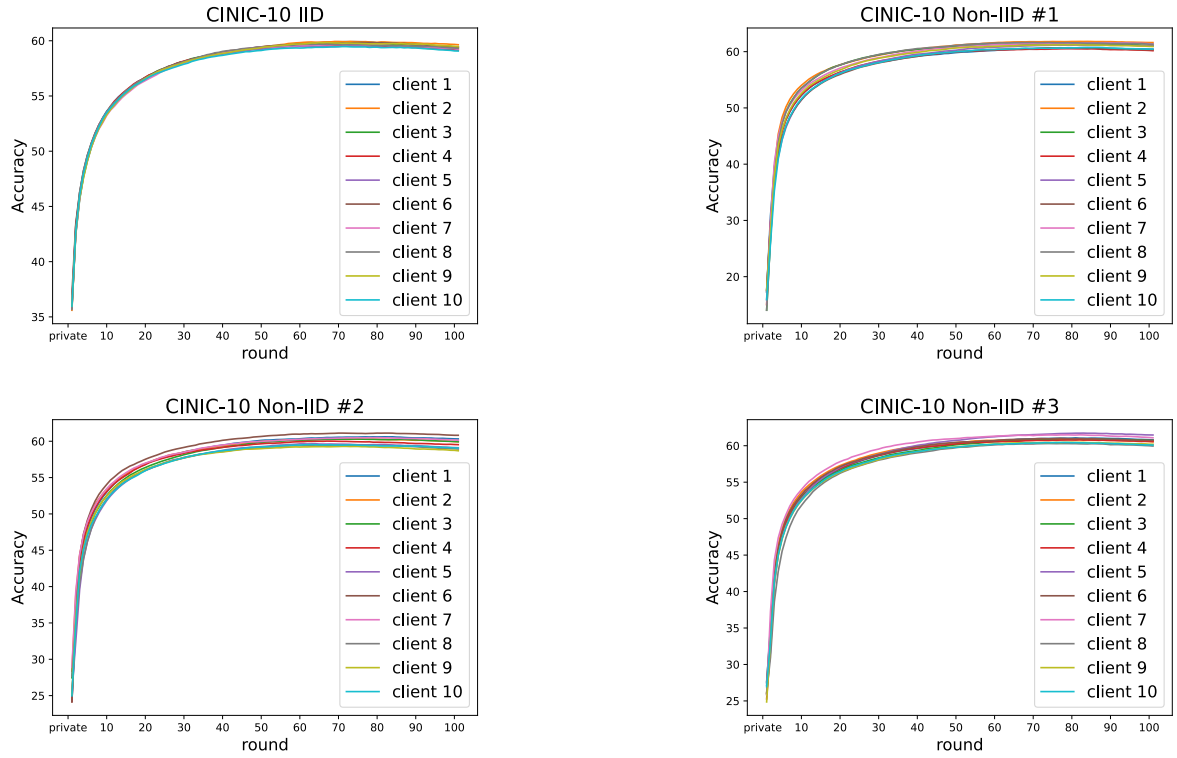

Fig. 3. Accuracy for each round in the gossip phase.

times. It also enabled combination of various models depending the performance of the learning devices. Experimental results showed that the proposed method can achieve accuracy comparable to that of existing centralized methods.

Future work includes experiments on more realistic networks and large number of devices. Enabling trustless distributed learning is a possible direction because it is difficult to establish trust with such a large number of devices.work

ACKNOWLEDGMENTS

We would like to express our gratitude to Dr. Ryo Yonetani, for discussion.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B.A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proc. AISTATS*, Fort Lauderdale, Florida, USA, April 2017.

[2] Rauniyar, A., "Federated Learning for Medical Applications: A Taxonomy, Current Trends, Challenges, and Future Research Directions," *arXiv e-prints, 2022*, doi:10.48550/arXiv.2208.03392.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li and H. Vincent Poor,"Federated Learning for Internet of Things: A Comprehensive Survey," *in IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622-1658, thirdquarter 2021, doi: 10.1109/COMST.2021.3075439.

[4] Y. Qu et al., "Decentralized Privacy Using Blockchain-Enabled Federated Learning in Fog Computing," *in IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5171–5183, June 2020, doi: 10.1109/JIOT.2020.2977383.

[5] Hinton, G., Vinyals, O., and Dean, J., "Distilling the Knowledge in a Neural Network," *in NIPS Deep Learning and Representation Learning Workshop*, 2015.

[6] J. Ma, R. Yonetani and Z. Iqbal, "Adaptive Distillation for Decentralized Learning from Heterogeneous Clients," *in 2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, 2021 pp. 7486-7492.

[7] S. Maarten van and T. Andrew S., "Distributed Systems Third edition," 2020.

[8] O. Róbert , H. István and J. Márk, "Gossip Learning with Linear Models on Fully Distributed Data," *Concurrency and Computation Practice and Experience*, 2013, 25, 10.1002/cpe.2858.

[9] Y. Takahashi and K. Shudo, "Unbiased machine learning methods for data on P2P networks," *DEIM Forum 2017*, 2017.

[10] H. Oguni and K. Shudo, "Addressing the Heterogeneity of A Wide Area Network for DNNs," Proc. *IEEE CCNC 2021*, Las Vegas, NV, USA, 2021, pp. 1-6, doi: 10.1109/CCNC49032.2021.9369585.

[11] H. Oguni and K. Shudo, "Communication Scheduling for Gossip SGD in a Wide Area Network," *IEEE Access*, Vol.9, pp.77873–77881, doi: 10.1109/ACCESS.2021.3083639.

[12] Hu, C., Jiang, J., and Wang, Z., "Decentralized Federated Learning: A Segmented Gossip Approach", *arXiv e-prints*, 2019. doi:10.48550/arXiv.1908.07782.

[13] Darlow, L. N., Crowley, E. J., Antoniou, A., and Storkey, A. J., "CINIC-10 is not ImageNet or CIFAR-10", *arXiv e-prints*, 2018. doi:10.48550/arXiv.1810.03505.

[14] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016 pp. 770-778.

[15] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.

[16] A. Howard et al., "Searching for MobileNetV3," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 1314-1324, doi: 10.1109/ICCV.2019.00140.