Ferrari: A Personalized Federated Learning Framework for Heterogeneous Edge Clients

Zhiwei Yao, Jianchun Liu, Hongli Xu, Lun Wang, Chen Qian, Yunming Liao

IEEE TRANSACTIONS ON MOBILE COMPUTING (Oct. 2024)

IF: 7.7

Faranak Solat

December 2024

Intelligent networking and computing for Media and Energy Systems Lab

가천대학교 소프트웨어학과
Gachon University

A brief preview on Abstract

Training the local models with pseudo-labeling

↓ Issue

Insufficient labeled data problem

↓ Solve

Federated Semi-Supervised Learning (FSSL)

Focusing heterogenous clients (non-IID scenario)

- The number of model migration
- The quality of pseudo-labels

↓ Impact on

Training Performance (e.g., efficiency and accuracy)

Introduction - Motivation

- By 2025, there will be 75.44 billion Internet of Things (IoT) devices
- These devices will generate a massive amount of data every year
- The modern cloud-centric applications can collect the generated distributed data from the devices
- Transferring this huge amount of data to Parameter Server (PS) is communication costly and has privacy-related issues

It motivates to use applications of Federated Learning (FL)

Introduction - Motivation

- In practical scenarios, due to the high labeling costs and lack of expertise
- There are always insufficient annotated (or labeled) data on the edge clients resulting poor performance on FL

It motivated to use Federated Semi-Supervised Learning (FSSL)

- However, training the large amount of unlabeled non-IID data has high computation and communication cost on clients

It motivated to use Personalized Federated Learning (PFL)

Introduction - Motivation

- However, the clients with labeled data still struggle to obtain component personalized models due to insufficient knowledge of local data distribution

It motivated to focus on the seeking labeling assistance from similar models for better personalization

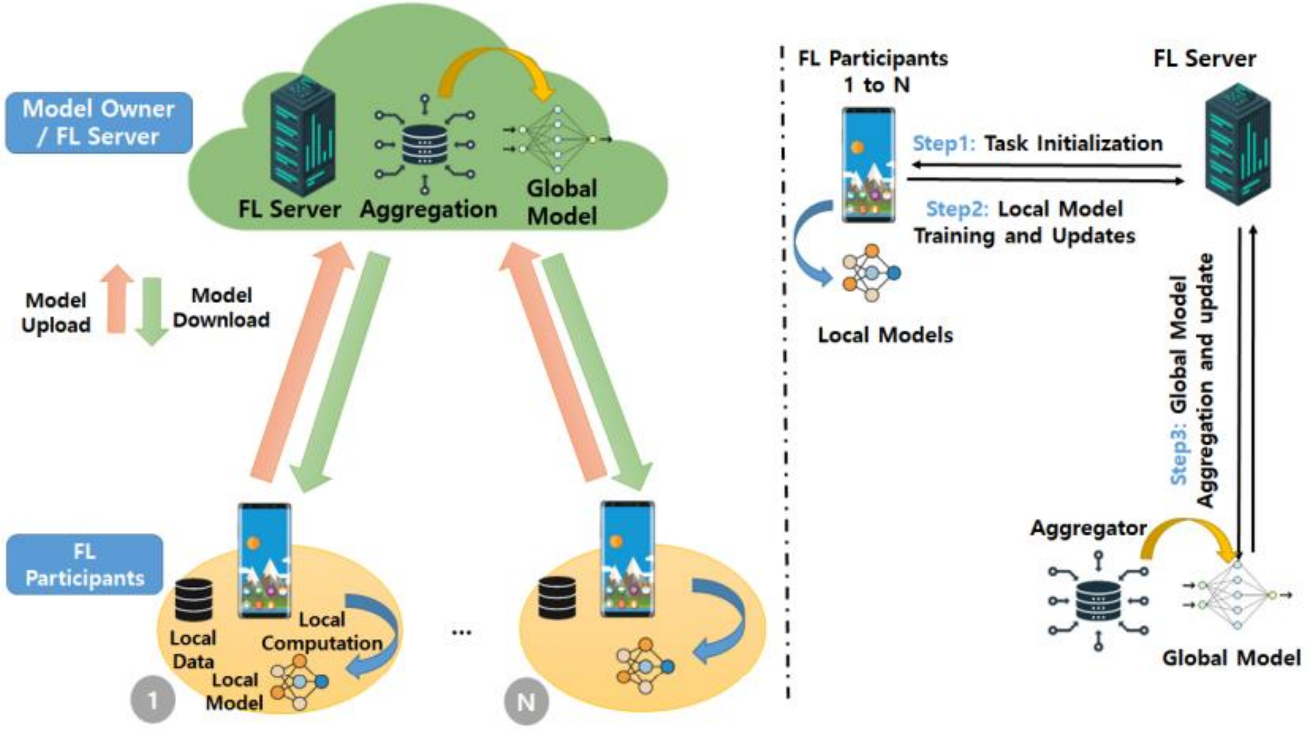
Introduction - Problem Statements: Federated Learning (FL)

- Emerging to address the above privacy challenge is FL proposed by Google in 2016.

- **Federated Learning (FL)**

Including 3 steps:

- Task initialization
- Local model training and update
- Global model aggregation and update



<General FL training process involving N participants>

Introduction – Problem Statements: Personalized Federated Learning (PFL)

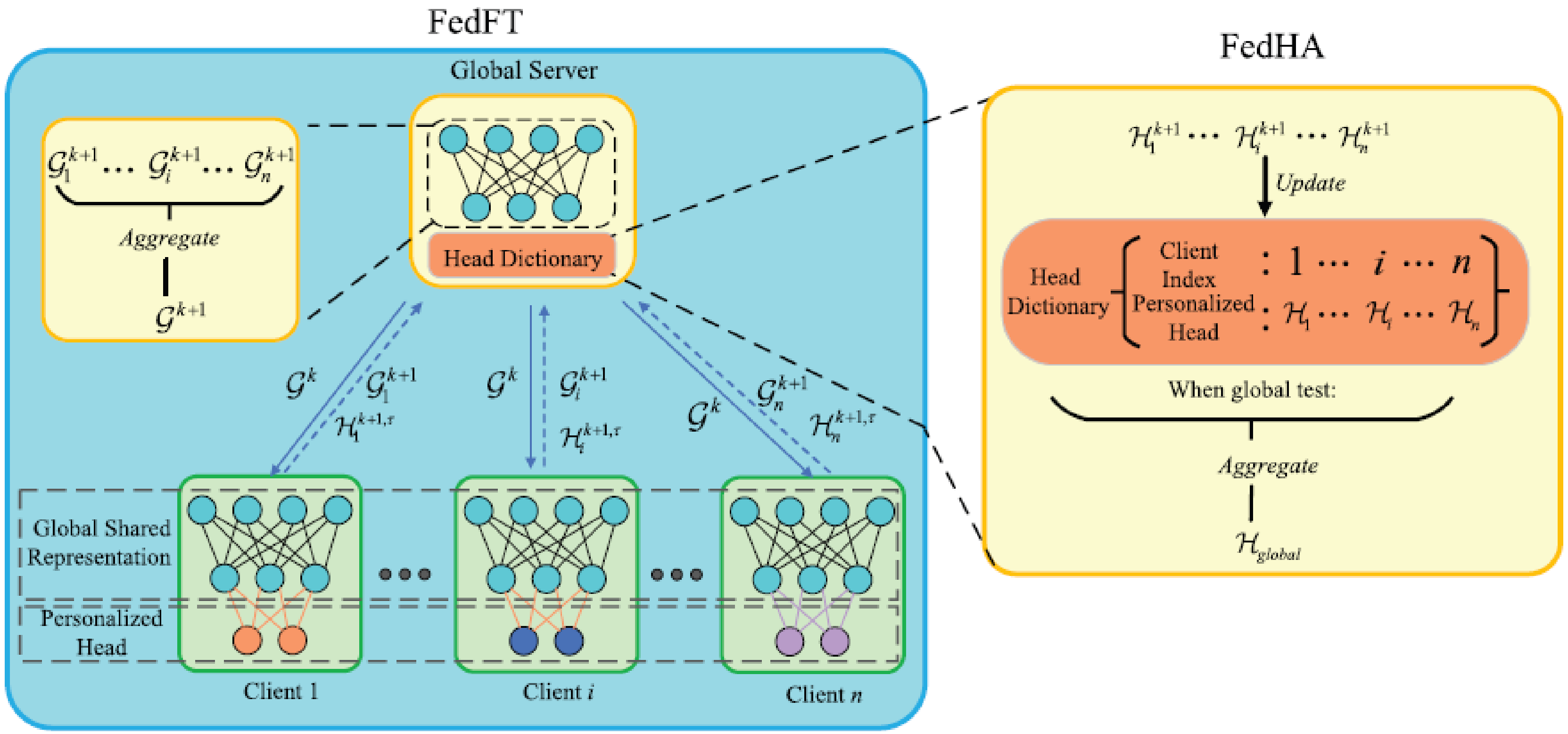
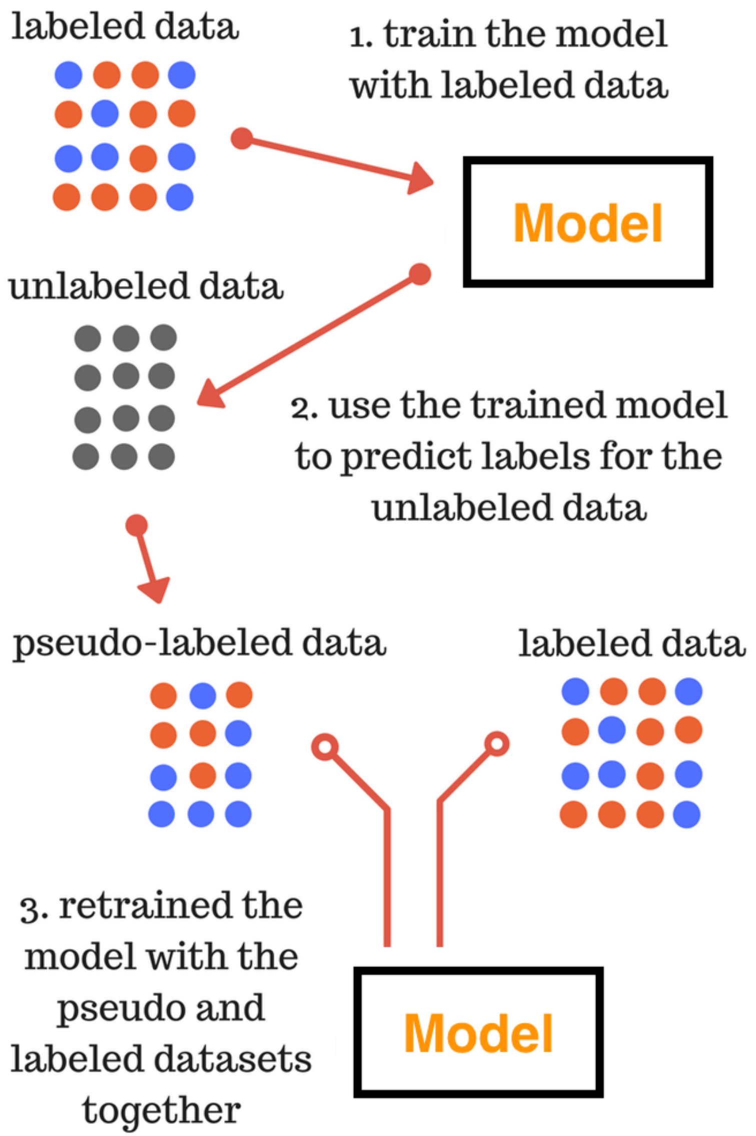


Fig. 1. Overall structure diagram of the FedFTHA method. The left half is mainly the FedFT method: the server and the client jointly train a global shared representation $\mathcal{G}$ and multiple personalized heads in the form of $\mathcal{H}_i$. The right half is the FedHA method: save personalized heads in the head dictionary of the global server, and provide the server with a global head $\mathcal{H}_{\text{global}}$ during global testing.

Introduction – Problem Statements: Semi-Supervised Learning (SSL)

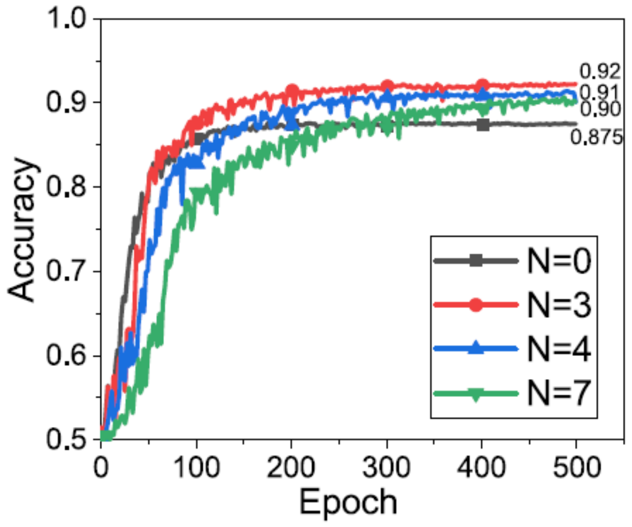
- Semi-Supervised Learning (SSL): To tackle label scarcity by leveraging the unlabeled data
- Two primary categories:
  - Consistency regularization based algorithms
  - Pseudo-labeling based algorithms
- In Pseudo-labeling based algorithms, instead of manually labeling the unlabeled data, we give approximate labels on the basis of the labelled data.



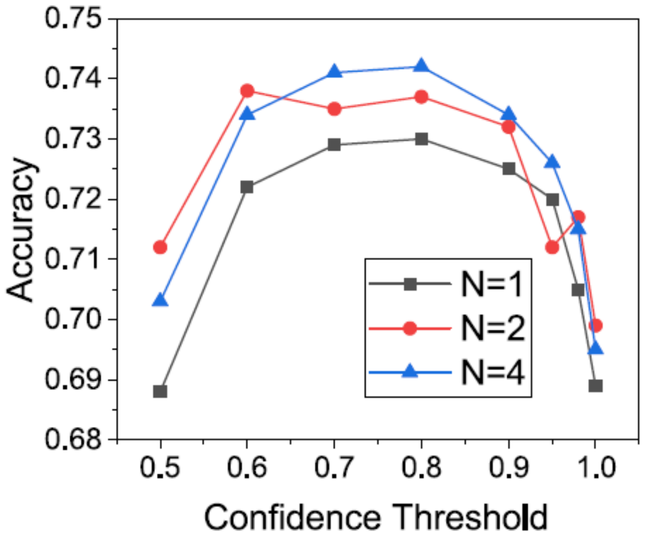
<Pseudo-Labeling SSL Process>

Introduction – Optimization Variables

- The number of model migrations ($\mathrm{N_i}$)
- The confidence threshold ($\mathrm{C_i}$)



(a) Accuracy with different N

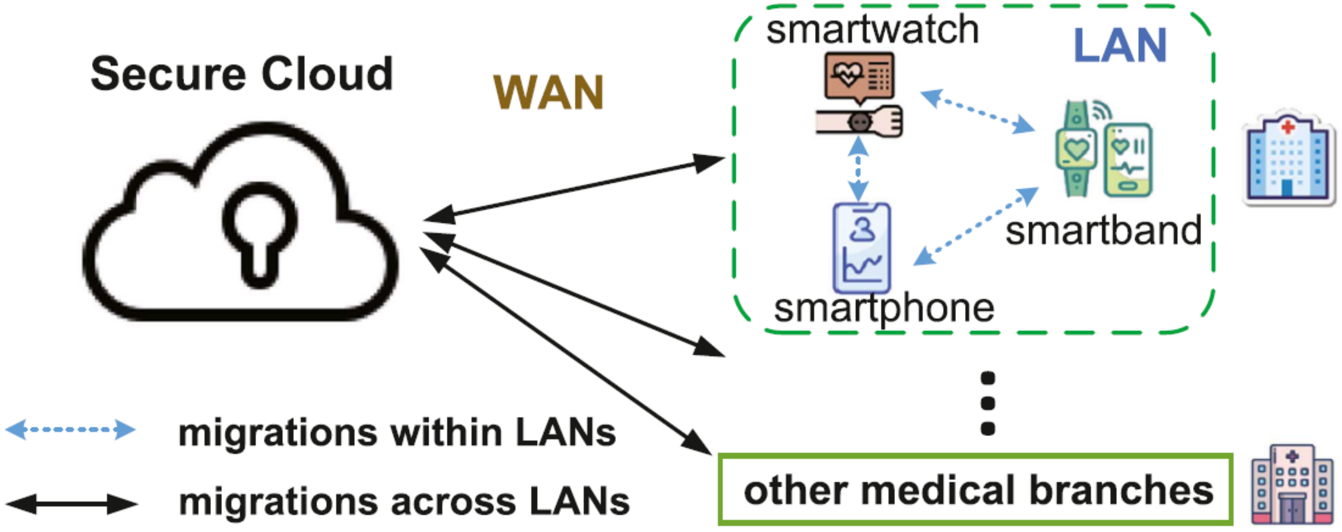


(b) Accuracy with different N and C

<Effect of different numbers of model migrations N and thresholds C on the test accuracy.>

Introduction – Optimization Variables

- The number of model migrations (N_i)
- The confidence threshold (C_i)



<LAN-aware model migrations in the healthcare system.>

TABLE I
COMMUNICATION TIME OF MIGRATING THE THREE MODELS WITHIN/ACROSS LANS

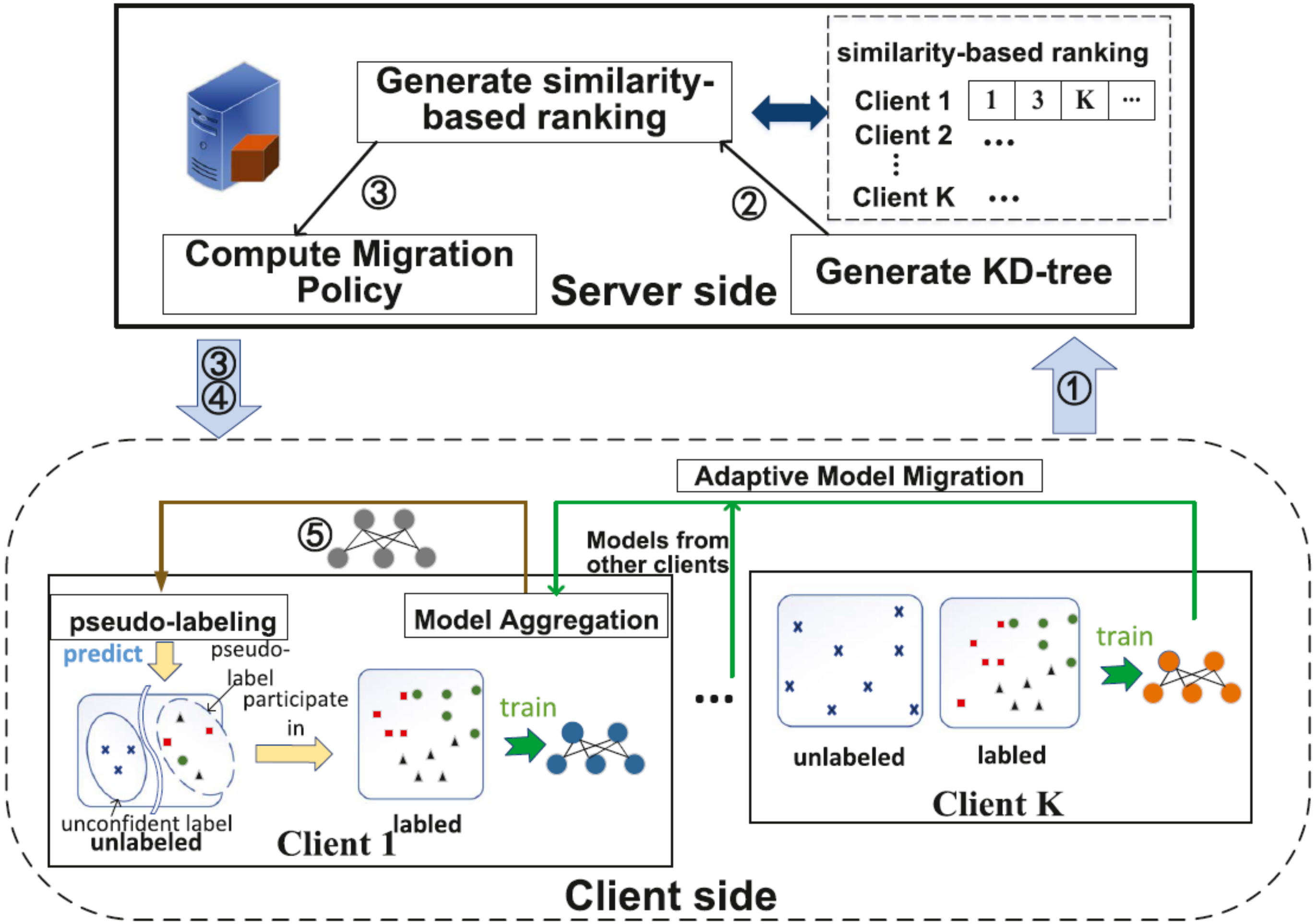
Model	Size (MB)	across LANs (s)	within LANs (s)
AlexNet	14.62	5.31	1.46
CNN	13.32	4.84	1.33
VGG-16	129.76	47.19	12.98

Introduction - Contributions

- To perform **adaptive model migrations** and **utilize the aggregated personalized model** to produce **high quality pseudo-labels** on local unlabeled data for heterogeneous clients.
- Proposing **EPIC**, a **greedy-based algorithm**, to adaptively determine the proper **number of model migrations and confidence threshold** for each client at every epoch.
- Implementing the system model on a physical platform with 30 edge clients (Ferrari provides 1.2~ 5.5× speedup without scarifying model accuracy, compared with existing baselines)

System model

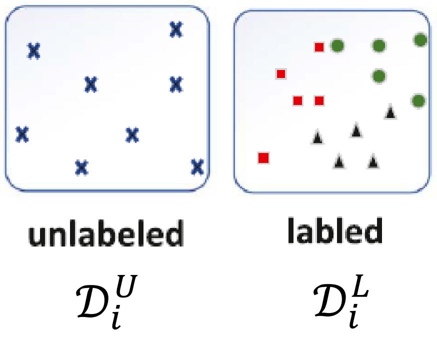
- An efficient personalized FSSL system, called Ferrari
- **F**ederated S**e**mi-Supe**r**vised Lea**r**ning with **A**daptive and Pe**r**sonalized Model M**i**gration (Ferrari)
- K: The number of clients
- w_i: The local model for client i
- d: The dimensions of local model w_i

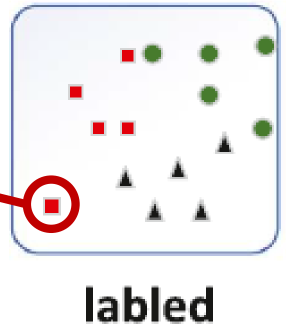


<System Overview>

Problem Formulation

- $\mathcal{D}_i = \mathcal{D}_i^L \cup \mathcal{D}_i^U$
- $\mathcal{D}^L = \cup_{i=1}^K \mathcal{D}_i^L$ & $\mathcal{D}^U = \cup_{i=1}^K \mathcal{D}_i^U$
- $\mathcal{M}_L = \sum_{i=1}^K \mathcal{M}_i$ (Data samples in labeled dataset $\mathcal{D}^L$)
- $\mathcal{D}^L = \{(x_l, y_l)\}_{l=1}^{\mathcal{M}_L}$
- x_l: The features of the l-th data sample
- y_l: The corresponding one-hot label





Sequence Diagram

Client

1) Client i trains the local model w_i^t on its labeled dataset $\mathcal{D}_i^L$

$Q(x_l, w_i^t)$: The predicted class distribution

The supervised loss function: (cross-entropy loss)

$$\mathcal{F}_i^s(w_i^t) = \mathbb{E}_{(x_l, y_l) \sim \mathcal{D}_i^L} f(y_l, Q(\pi_1(x_l), w_i^t)) \qquad (1)$$

π_1: A weak data augmentation, such as **Random Horizontal Flipping** or **Random Cropping**

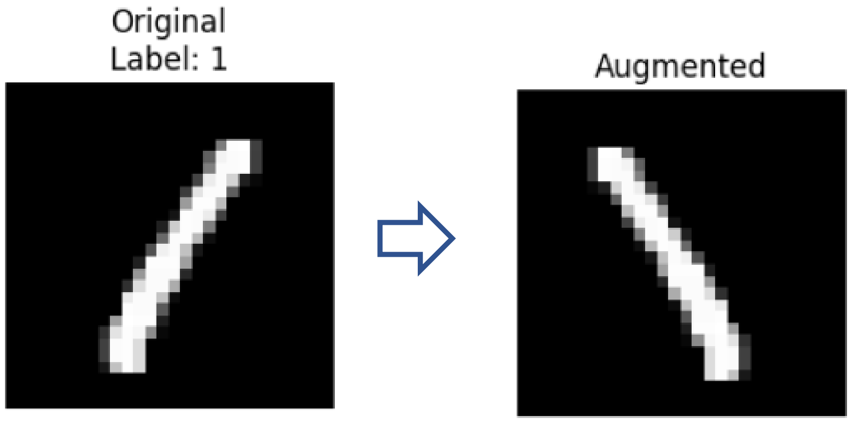
Data Augmentation Methods: Random Horizontal Flipping & Random Cropping

Random Horizontal Flipping

Artificially expand the size and diversity of a dataset

Flip an image horizontally (left-to-right) with a specified probability (often 50%)

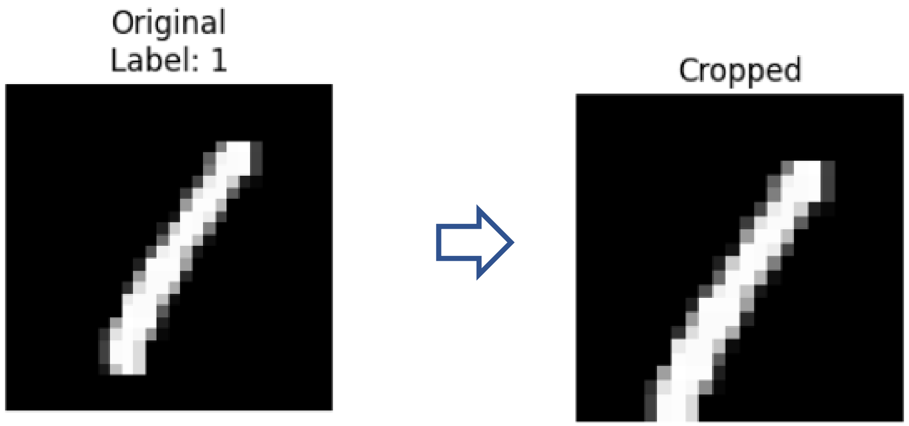
If the image is flipped, the pixels on the left side are swapped with the corresponding pixels on the right side, creating a mirror image along the vertical axis.



Random Cropping

Introduce variability and improve the robustness of models

Randomly selecting a rectangular region from the original image and resizing it (if necessary) to match the desired dimensions.



Sequence Diagram

Client

2) Client i makes labels on its unlabeled dataset $\mathcal{D}_i^U$

$$\hat{y}_j = \operatorname*{argmax}_q p_{j,q} \qquad (2)$$

$\hat{y}_j$ may not be the ground-truth label because of the prediction error.

The consistency loss:

$$\mathcal{F}_i^u(\bar{w}_i^t) = \mathbb{E}_{(x_j, \hat{y}_j) \sim \mathcal{D}_i^U} f(\hat{y}_j, \mathcal{Q}(\pi_2(x_j), \bar{w}_i^t)) \qquad (3)$$

$\overline{w}_i^t$: The aggregated model to use for training the unlabeled dataset

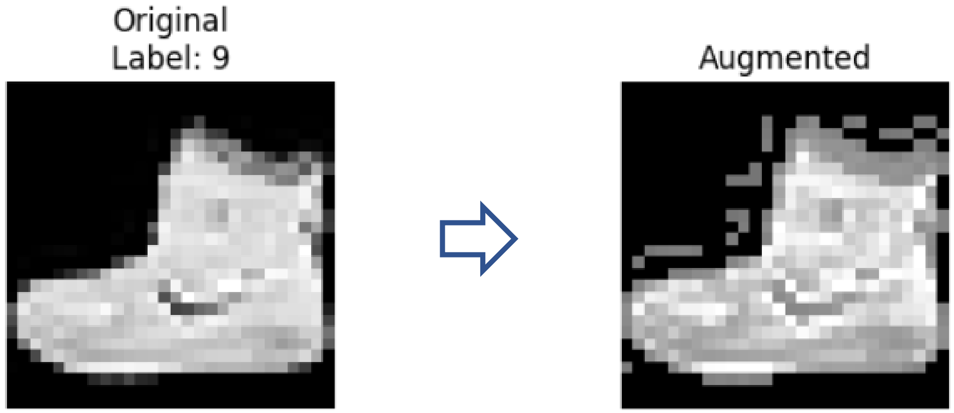
π_2: A strong data augmentation, such as **RandAugment**

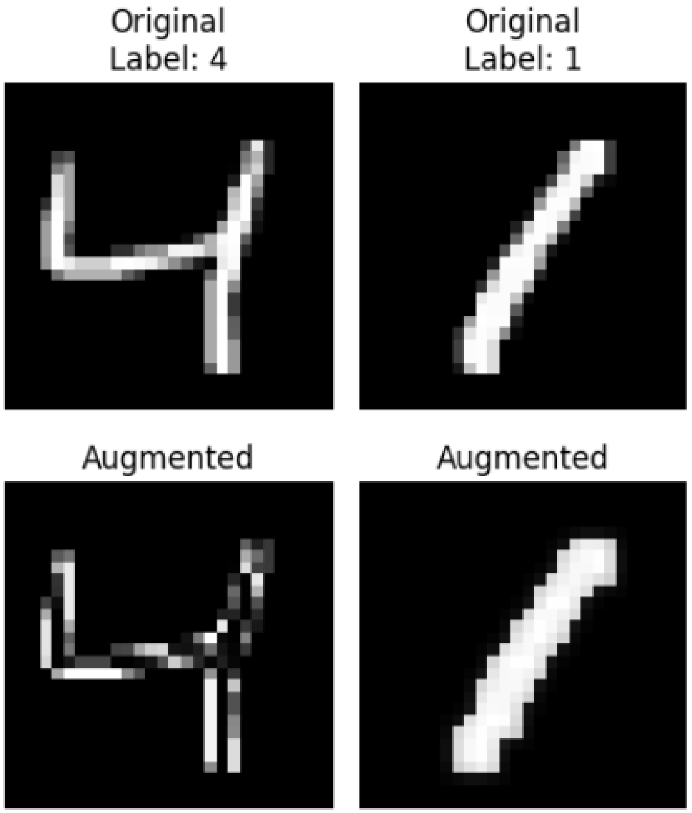
Data Augmentation Method (strong): RandAugment

It automates the process of selecting and applying data augmentation techniques by introducing randomness and simplicity.

Two hyper-parameters:

- N: The number of augmentation transformations to apply.
- M: The magnitude or intensity of these transformations.





Sequence Diagram

Client

PS

3) Local training with aggregated by local models from other clients for $\mathcal{T}_s$ epochs and then uploads the model to PS

4) Collects all local models

5) Computes the similarity-based ranking using Gaussian K-Dimensional (KD)-Tree

6) Generates the migration policy and broadcast among clients

7) Training using the new migration policy and do process until reaching the convergence

Problem Formulation

- FSSL aims to optimize (labeling function):

$$f^* = \min_{\boldsymbol{w} \in \mathbb{R}^d} \mathcal{F}_i(w_i^t) = \min_{\boldsymbol{w} \in \mathbb{R}^d} (\mathcal{F}_i^s(w_i^t) + \mathcal{F}_i^u(\bar{w}_i^t)) \qquad (4)$$

- The optimization Problem:

$$\min_{t \in \{1,2,\ldots,T\}, C_i} \frac{1}{K} \sum_{i=1}^{K} [\mathcal{F}_i(w_i^t)]$$

$$\text{s.t.} \quad \sum_{t=1}^{T} e_{i,t} \leq E^i, \quad \forall i \qquad (5)$$

$$\sum_{i=1}^{K} (N_i b + l) \leq B^t, \quad \forall t \qquad (6)$$

$e_{i,t}$: The computation cost on client i at the t-th epoch

E^i: The computing resource budget for client i

l: The communication cost for each client i to server

b: The communication cost for migrating the model between clients

B^t: The communication resource for all clients budget at the t-th epoch

To solve the optimization problem using a greedy-based algorithm EPIC

Problem Formulation

- Energy-efficient Privacy-preserving Intelligent Communication (EPIC)

Advantages of Greedy EPIC Algorithm:

- **Simplicity**: Computationally less expensive compared to exhaustive search.
- **Scalability**: Works well with large-scale systems where global optimization is computationally infeasible.
- **Practicality**: Easily implementable in real-world systems.

Challenges:

- **Local Optima**: May get stuck in suboptimal solutions.
- **Trade-offs**: Requires careful design of the utility function to balance energy, privacy, and communication efficiency.

A. Alsharif, M. Nabil, S. Tonyali, H. Mohammed, M. Mahmoud, and K. Akkaya, "EPIC: Efficient privacy-preserving scheme with EtoE data integrity and authenticity for AMI networks," *IEEE Internet of Things Journal*, *6*(2), 3309-3321, 2018.

Problem Formulation

- The key idea of EPIC is to **greedily migrate models** according to **the similarity-based ranking** and **heterogeneous resource budget** of clients.
- Search space adjustment:

$$\mathcal{R}_i(N_i, C_i) = \frac{|\frac{1}{K}\sum_{i=1}^{K}[\mathcal{F}_i(w_i^t)] - \mathcal{F}_i(w_i^t)|}{\Delta cost} \quad (10)$$

$\mathcal{R}_i(N_i, C_i)$: The reward of client i with proper N_i and C_i in EPIC:

$\Delta cost$: The communication cost for migrating N_i models

Algorithm

Algorithm 1: Joint Optimization of N_i and C_i by EPIC.

Data: $N_i^{\max}$, the similarity-based ranking ($\mathcal{S}$), δ_i

Result: the number of model migrations N_i, and confidence threshold C_i for each client at every epoch

Initialize $N_i^{\max} = K/2,\ B^t,\ E^i$ **for** *each epoch* $t = \{1, \mathcal{T}_s, 2\mathcal{T}_s, \ldots, T\}$ **do**

  for *each client* $i = \{1, \ldots, K\}$ **do**

    $N_i \leftarrow 1$, $C_i \leftarrow 1.0$ Calculate δ_i by Eq. (11); /* the search space size of the current client i */

    for $\mathcal{N} \in \{N_i' - \delta_i, N_i' + \delta_i\}$ *based on* $\mathcal{S}$ **do**

      /* N_i' is the number of model migrations at last epoch */

      if $\mathcal{F}_i^u(\mathcal{N})$ *satisfies Eq.*(8) **then**

        /* $\mathcal{F}_i^u$ is loss function on the unlabeled data */

        $N_i^{\max}$=$\mathcal{N}$; /* the upper bound of N_i */

    while $N_i \leq N_i^{\max}$ *and* $K \cdot (N_i b + l) \leq B^t$ **do**

      N_i=N_i+1 /* determine N_i under communication constraints */

    Migrate the top N_i models according to $\mathcal{S}$ N_i=max$\{1, N_i - 1\}$ **while** $C_i \in [C_i^{\min}, C_i^{\max}]$ *and* $T \cdot e_{i,t} \leq E^i$ *and* $Acc(N_i, C_i) < Acc(N_i, C_i - 0.1)$ **do**

      $C_i = C_i - 0.1$ /* determine C_i under computing constraints */

    C_i=min$\{1.0, C_i + 0.1\}$

Performance Evaluation

System Implementation:

Parameter Server:

- Intel(R) Core(TM) i9-10900X CPU
- 4 NVIDIA GeForce RTX 2080Ti GPUs
- 128 GB RAM

Clients (30):

- 10 NVIDIA Jetson TX2
- 10 NVIDIA Jetson Xavier
- 10 NVIDIA Jetson AGX

TABLE II
TECHNICAL SPECIFICATIONS OF EDGE CLIENTS

	AI Performance	**GPU Type**
Jetson TX2	1.33 TFLOPS	256-core Pascal
Jetson NX	21 TOPS	384-core Volta
Jetson AGX	22 TOPS	512-core Volta
	CPU Type	**ROM**
Jetson TX2	Denver 2 and ARM 4	8 GB LPDDR4
Jetson NX	6-core Carmel ARM 8	8 GB LPDDR4x
Jetson AGX	8-core Carmel ARM 8	32 GB LPDDR4x

Performance Evaluation

Datasets:

CIFAR-10, SVHN (10 Classes), and Human Action Recognition (HAR)

CIFAR-10 (AlexNet):

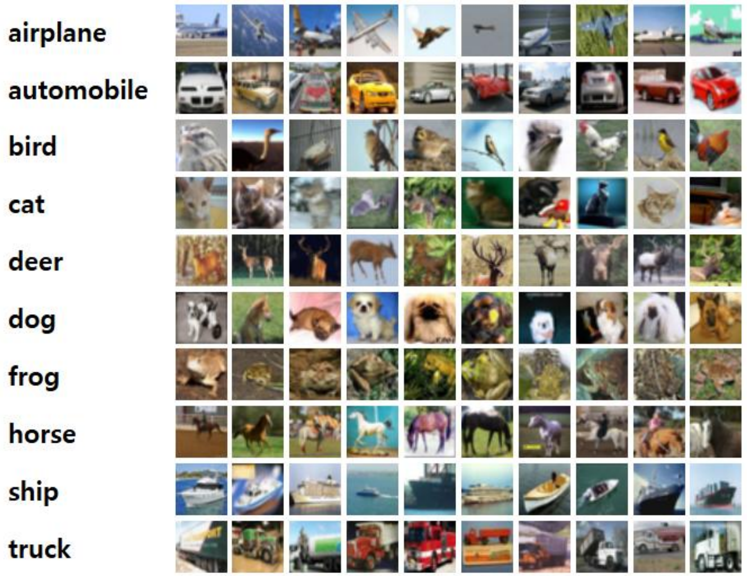
- Train dataset: 10,000 Labeled + 40,000 Unlabeled
- Test dataset: 10,000

SVHN (CNN):

- Train dataset: 50,225 (ratio 0.8 unlabeled)
- Test dataset: 26,032

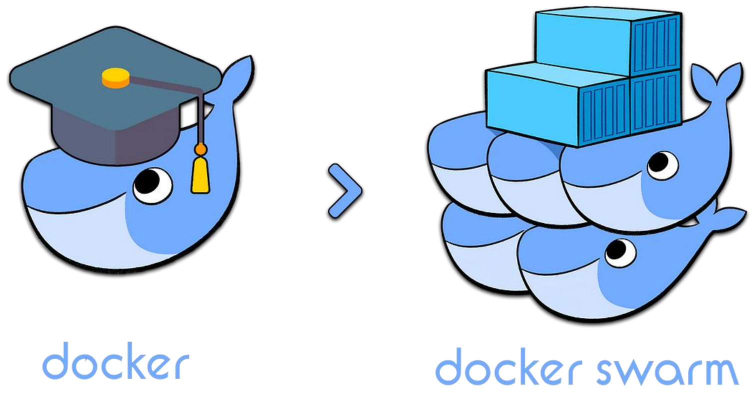
HAR (CNN):

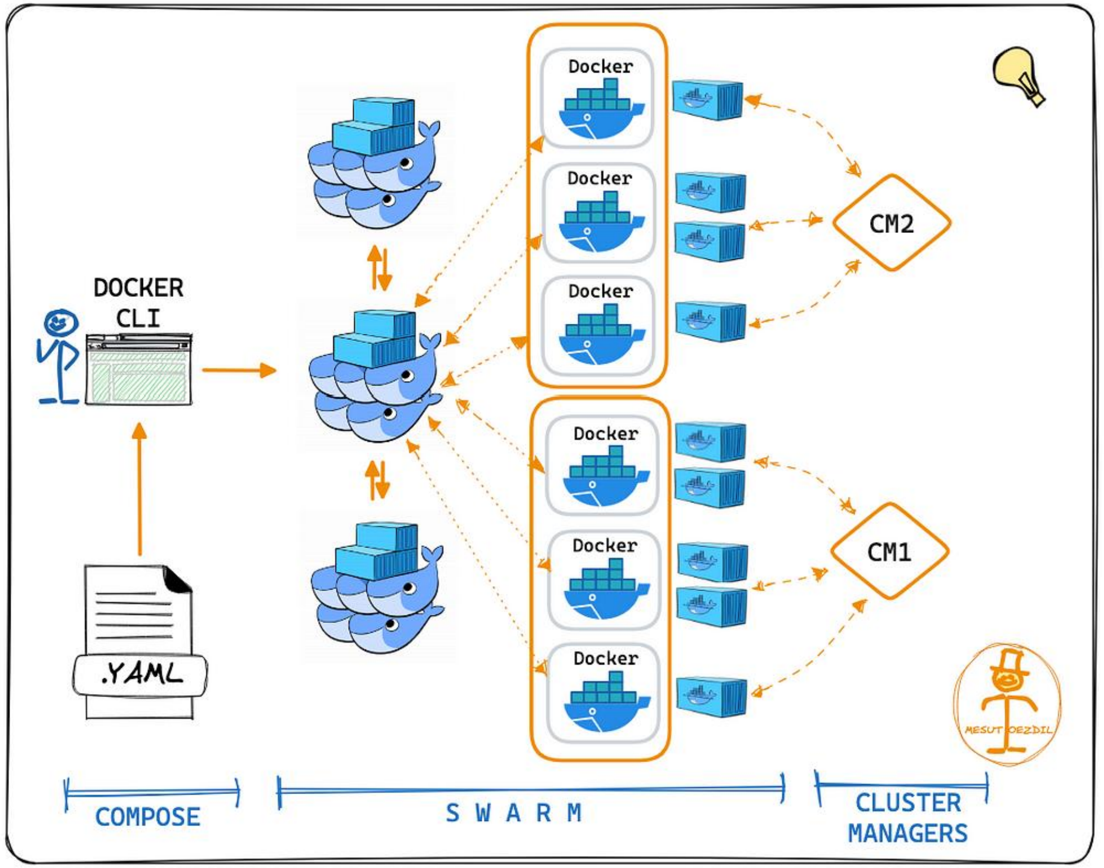
- 6 activities: Sitting, Standing, Laying Down, Walking, Walking Downstairs, Walking Upstairs

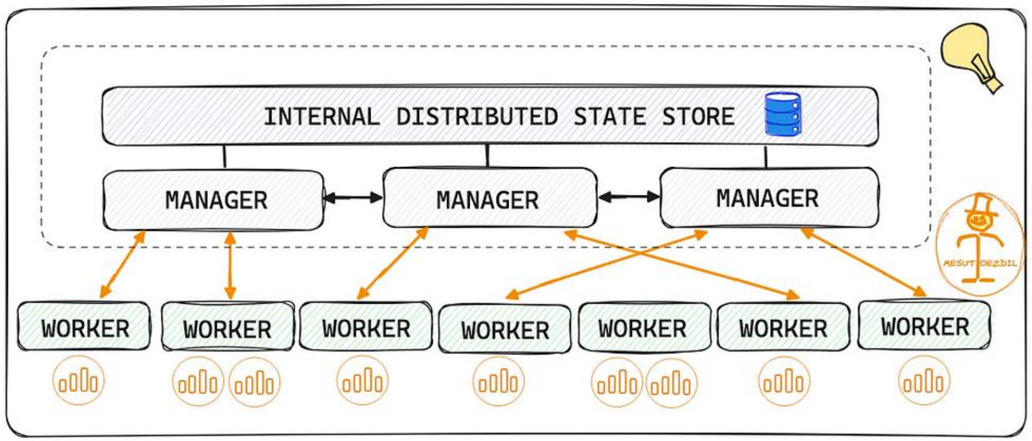


Performance Evaluation

- Using the Docker swarm







Performance Evaluation

Baselines:

- **FedAvg**
- Uses labeled data for training and averages model updates from clients.
- Does not account for heterogeneity or knowledge sharing among clients.
- **pFedMe**
- Allows clients to update local models independently without diverging far from a global reference model.
- Supports training on both labeled and unlabeled data.
- **UM-pFSSL**
- Enables clients to share knowledge by migrating fixed numbers of models across clients.
- Lacks dynamic optimization for migration and does not explore confidence thresholds for pseudo-labeling.

Performance Evaluation

Convergence speed:

- Ferrari has the fastest convergence

- AlexNet on CIFAR-10
- CNN on SVHN
- CNN an HAR

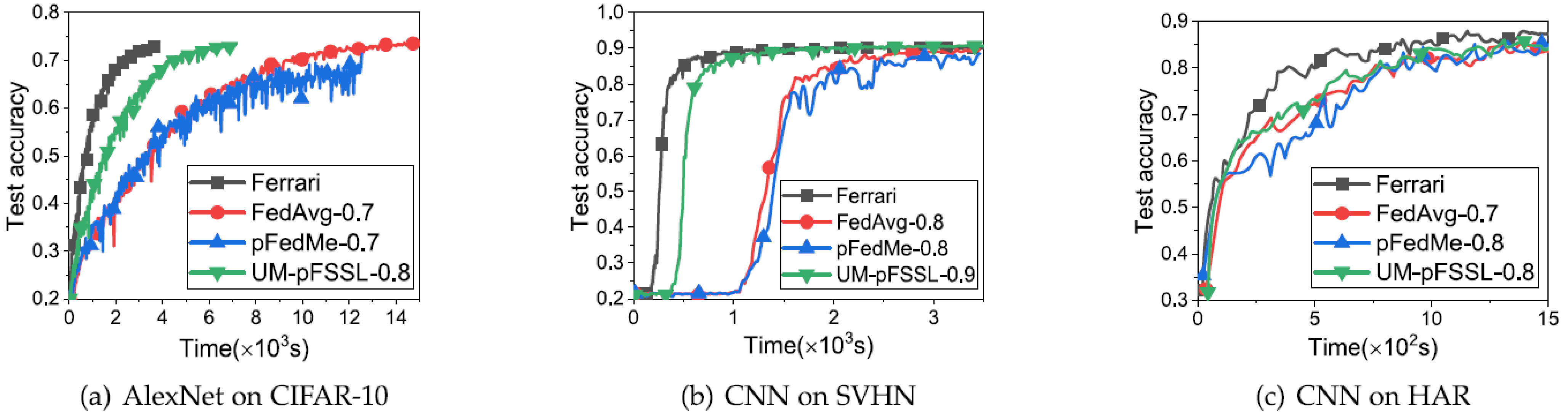


Fig. 5. Test accuracy of four systems on three datasets with non-IID level $\zeta = 20\%$.

Performance Evaluation

TABLE III
EFFECT OF DIFFERENT CONFIDENCE THRESHOLDS C ON TEST ACCURACY(%) OF THREE BASELINES ($\zeta = 20\%$)

	The confidence threshold C							
No.	0.5	0.6	0.7	0.8	0.9	0.95	0.98	1.0
FedAvg	71.6	73.0	**74.0**	73.5	72.5	73.7	71.3	70.1
pFedMe	70.3	71.0	**71.0**	70.3	69.5	68.4	67	67.4
UM-pFSSL	68.8	72.2	72.9	**73.0**	72.5	72.0	70.5	68.9

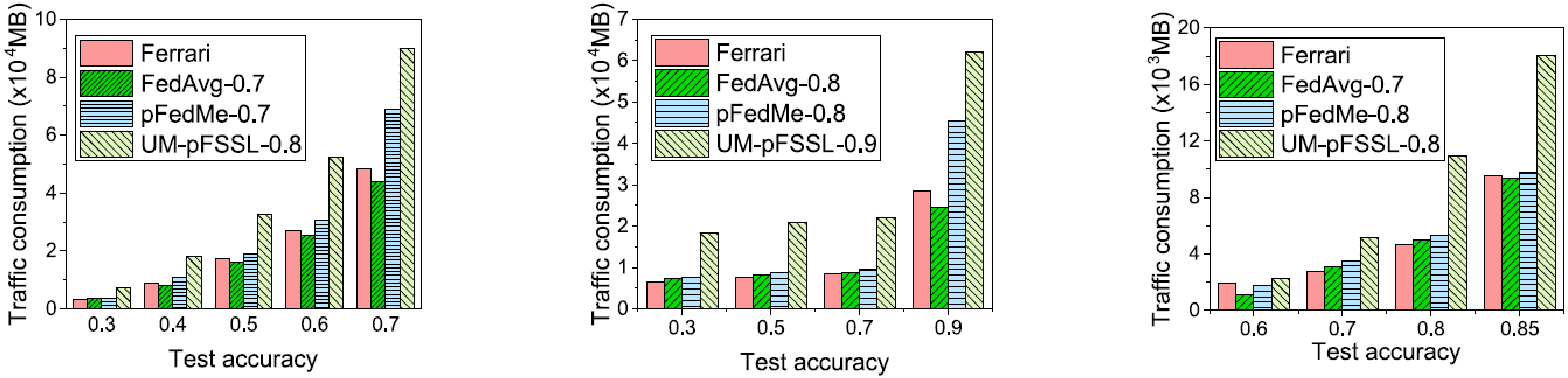
TABLE IV
ACCURACY(%) OF FOUR SYSTEMS ($\zeta = 20\%$)

	Different systems			
	Ferrari	FedAvg	pFedMe	UM-pFSSL
CIFAR-10	74.2	74.0	71.0	73.0
SVHN	90.7	90.4	90.0	90.1
HAR	89.0	86.4	88.4	87.2

Performance Evaluation

Communication cost:

- Ferrari achieves higher accuracy with less communication cost
- Effect of adaptive model mitigation within LANs

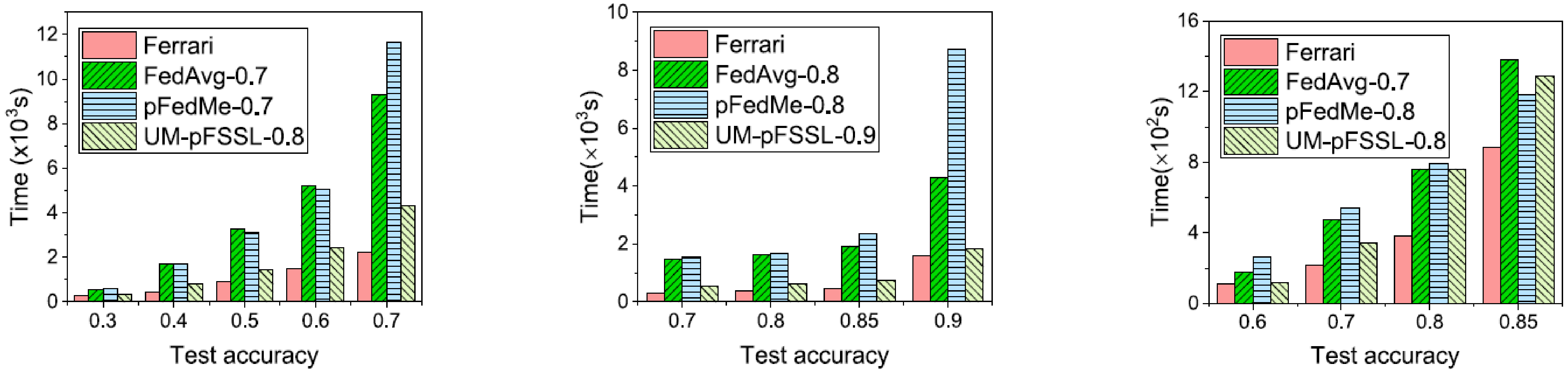


(a) AlexNet on CIFAR-10 (b) CNN on SVHN (c) CNN on HAR

Fig. 6. Comm. cost to achieve the target accuracy.

Performance Evaluation

- Ferrari has the fastest convergence



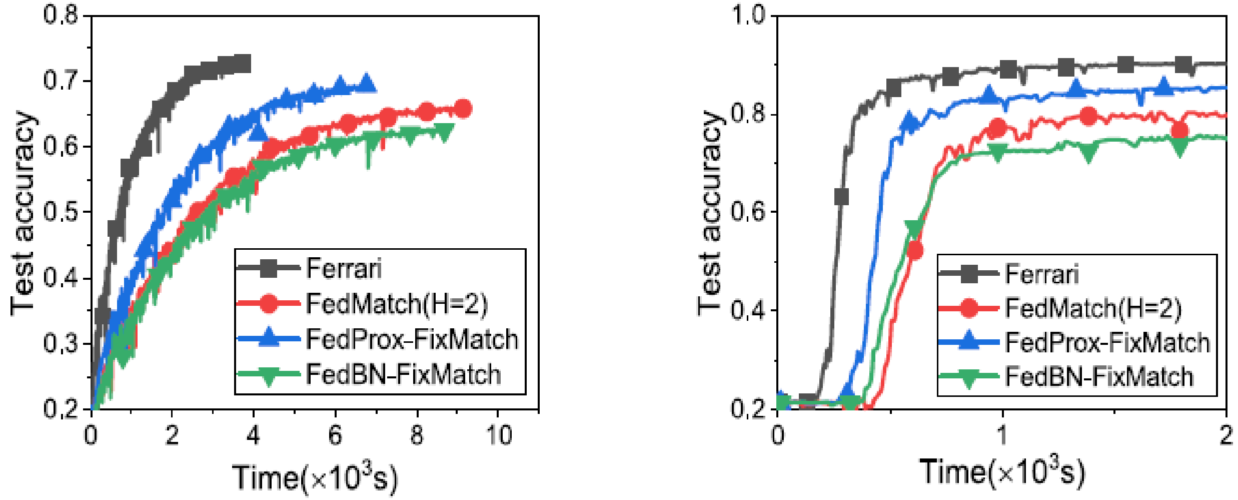
(a) AlexNet on CIFAR-10 (b) CNN on SVHN (c) CNN on HAR

Fig. 7. Time cost to achieve the target accuracy.

Performance Evaluation

pFSSL v.s Traditional FSSL:

- H=2 → The selection of two helper agents for each client at each epoch.
- In FedMatch, the migration number is fixed without considering the impact of confidence threshold.

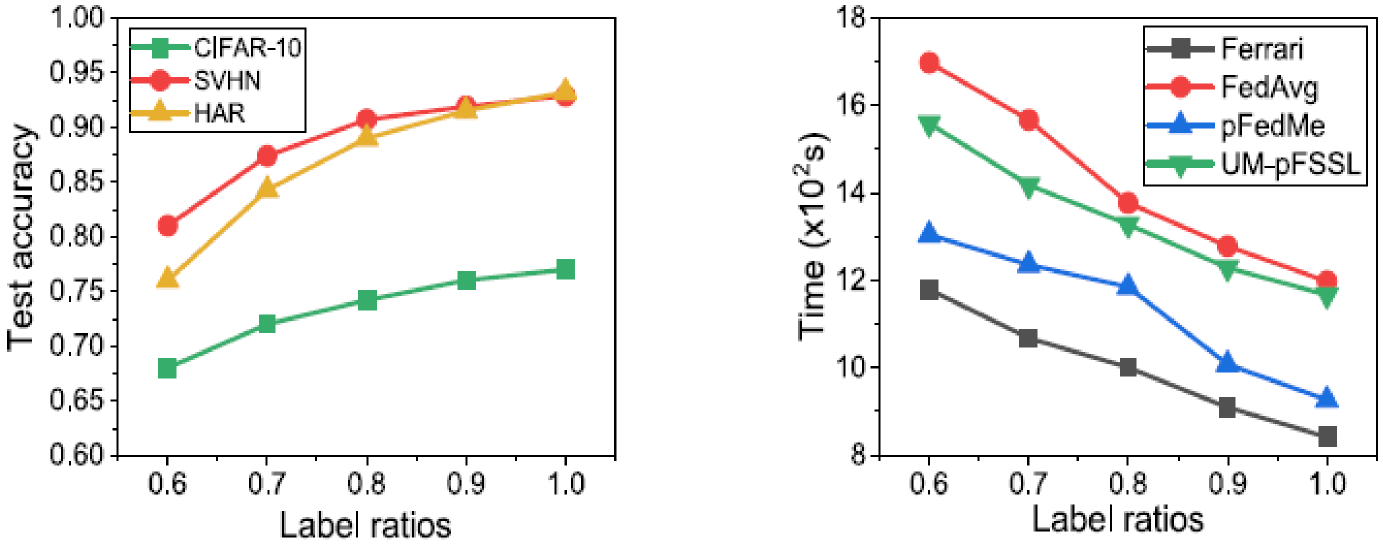


(a) AlexNet on CIFAR-10 (b) CNN on SVHN

Fig. 8. Test accuracy of Ferrari and traditional FSSL systems with non-IID level $\zeta = 20\%$.

Performance Evaluation

Effect of different label ratios:



(a) Test accuracy

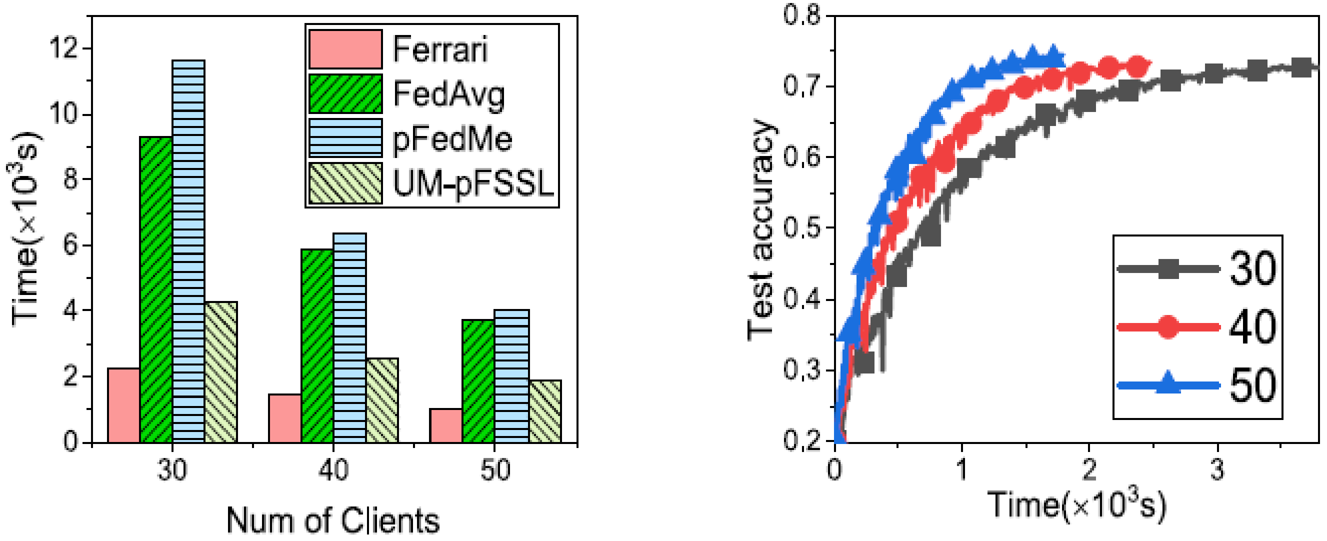
(b) Time costs to achieve the target accuracy on HAR

Fig. 9. Performance of models trained with different label ratios.

Performance Evaluation

Effect of system scales:

- By increasing the number of clients, the model convergence in all systems becomes faster because of more training data generated by clients.

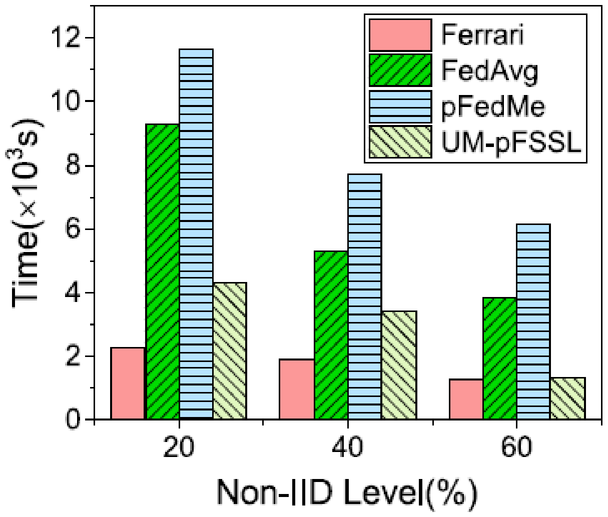


(a) Time to reach 70% accuracy (b) Accuracy v.s. Time on Ferrari

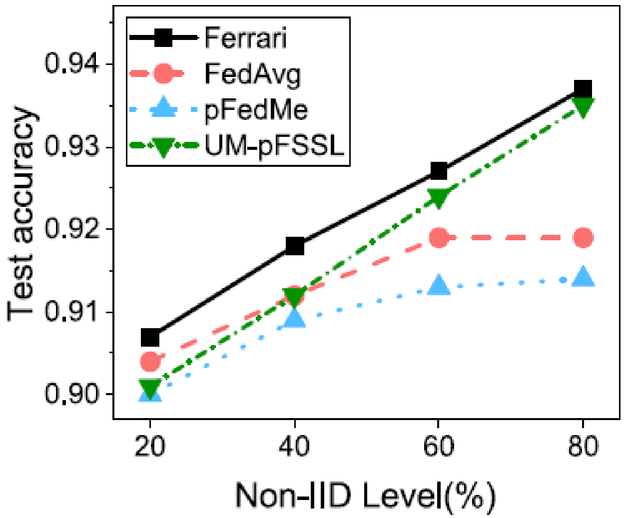
Fig. 10. Training with different number of clients on CIFAR-10 ($\zeta = 20\%$).

Performance Evaluation

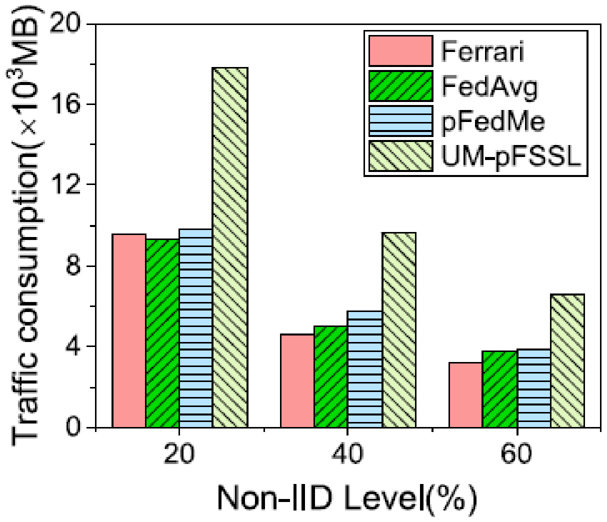
Impact of data distributions (non-IID data on training performance):



(a) Time to reach 70% accuracy on CIFAR-10



(b) Accuracy within 4,000s on SVHN



(c) Comm. cost to reach 85% accuracy on HAR

Fig. 11. Model training with different non-IID levels.

Conclusions

- Presented the designed and implementation of novel **FSSL** called **Ferrari**, to accelerate **model training** and boost the **pseudo-labeling** among clients under **resource limitation** and **data heterogeneity**.
- Utilized **model migrations within** LANs to allow knowledge sharing among clients.
- The **trade-off** between the **quantity** and the **quality** of pseudo-labels to enhance **model performance** with **fewer** communication resources (resources limitation and similar models' potential).
- Ferrari achieved with the number of model migrations and the confidence thresholds for heterogeneous clients during training.
- Ferrari **outperforms** benchmarks on **three** world datasets and models **without** sacrificing model accuracy.

Intelligent networking and computing for Media and Energy Systems Lab

Thank you

faranak1995@gachon.ac.kr

가천대학교 소프트웨어학과
Gachon University