University of Glasgow

School of Computing Science
Knowledge & Data Engineering Systems

Cluster-based & Label-aware Federated Meta-Learning for On-Demand Classification Tasks

Tahani Aladwani, Christos Anagnostopoulos,
Shameem Parambath & Fani Deligianni

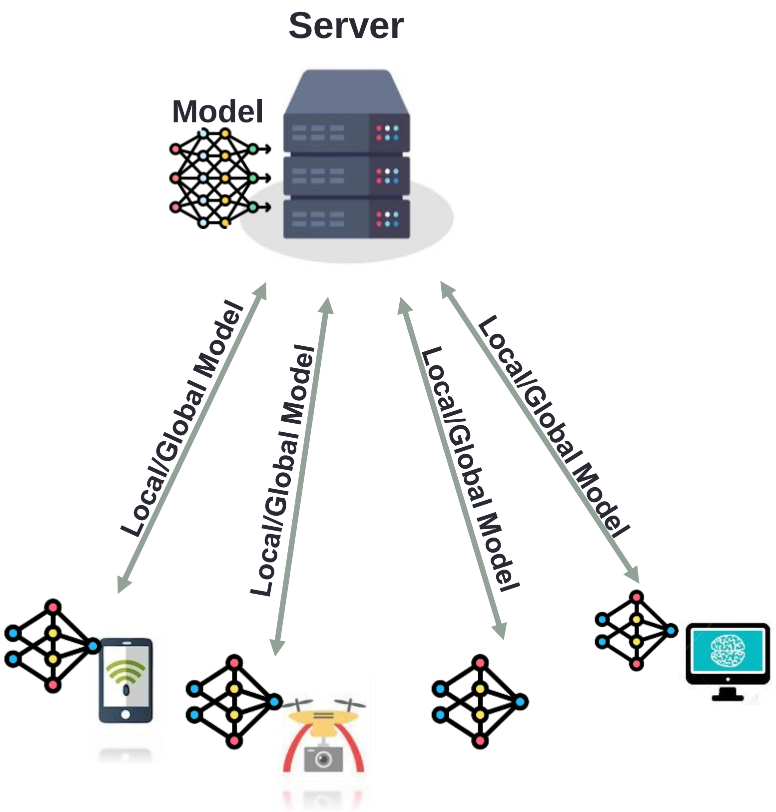
IEEE DSAA Conference 2024, Sunday, 6th Oct 2024 – Thursday, 10th Oct 2024, San Diego, US

Introduction

Federated Learning (FL): distributed learning paradigm that <u>collaboratively</u> trains a global model across clients <u>without data exchange</u>.

In many applications where <u>quick decisions are required</u>, or when a large number of alternatives has to be tested, <u>the predictions have to be performed in near real-time</u>.

Meta-Learning: accelerates model adaptation to arbitrary labels by allowing fine-tuning over small datasets when faced with previously unseen tasks.

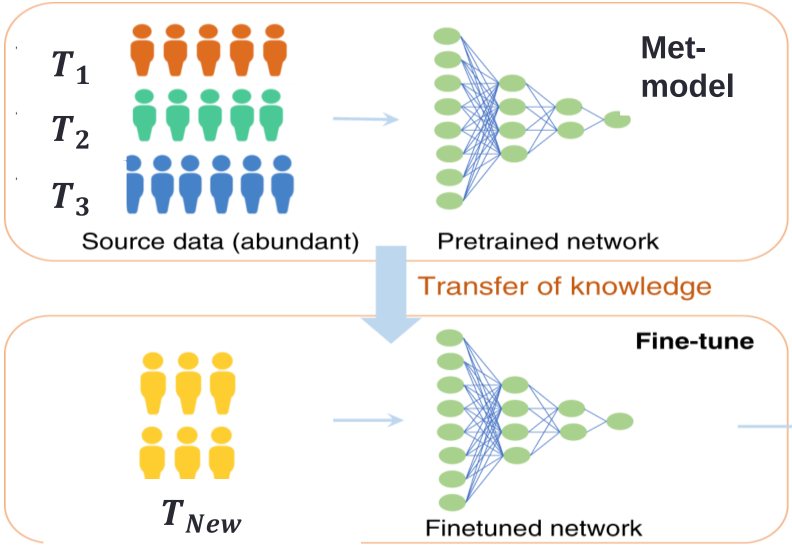


Introduction

Meta-Learning in FL relies on 'perfect setups', which are challenging to implement in real-world applications.

- Classification tasks share exactly the same set of labels ℓ and label distribution as those used in training meta-models.
- Not possible to deal with any arbitrary out-of-distribution classification requests.
- Labels are **equally** distributed among clients.

$$T_1(\mathcal{L}) \cap T_2(\mathcal{L}) \cap T_3(\mathcal{L}) \neq \emptyset$$



Therefore, they rely on a single, global Federated Meta- Learning (FML).
FML works well for homogeneous data and tasks, adapting to heterogeneous data and task distribution is challenging.

Challenges in FL Clients' Data

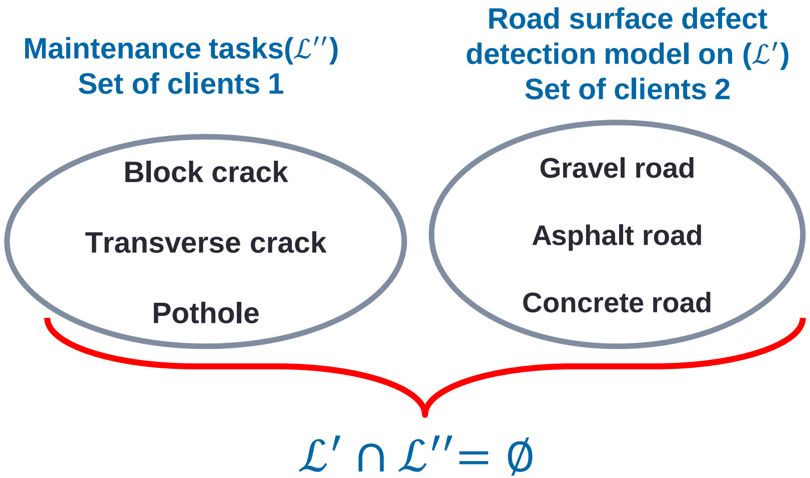
- Data & Class Labels are heterogeneous; due to shifts in **feature, label, and concept distributions**.
- A client may have **only a few classes** compared to total number of classes required for a specific task.
- Among the available classes on a client, there may be class imbalance.
- Such disparities in labels across clients impede the **<u>convergence of classifiers</u>** and **<u>degrading their performance</u>**.

Challenges in On-demand Tasks

On-demand classification task: requests training of a classifier over distributed clients' data, where data are labelled from a set $\mathcal{T} \subset \mathcal{L}$.

This set $\mathcal{T}$ can be: $\mathcal{T} \subset \mathcal{L}'$ or $\mathcal{T} \subset \mathcal{L}''$.

Note: in traditional FML and FL, we obtain the trivial case $\mathcal{T} \equiv \mathcal{L}$.



A single, global FML model proves to be inefficient and impractical to accommodate **(i)** any arbitrary classification tasks and **(ii)** out-of-distribution labels across clients.

Our CL-FML Solution

We introduce a **Cluster-based & Label-aware FML** framework (**CL-FML**) that addresses such challenges, departing from standard FL and FML paradigms.

Idea: CL-FML gathers clients together based on label shifting mitigating label imbalance per task.

Main goals:

- ✓ Study the cases of training more than one (reusable) meta- model tailored to available labels $\mathcal{L}_k \subset \mathcal{L}$ of a cluster of clients $\mathcal{C}_k$.
- ✓ Provide compact sized meta-models stored on clients temporarily, to be reused for future tasks
- ✓ CL-FML not only **adapts** meta-models solely to tasks with exactly the same distribution; it copes with **sharing** meta-models among clusters to further fine-tune <u>in case of out of distribution tasks</u>.

University of Glasgow

Centralized & Decentralized Federated Learning

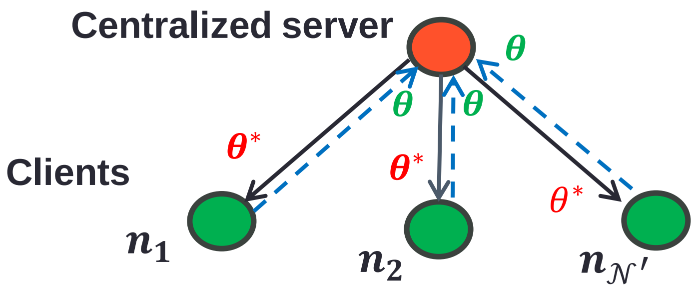
School of Computing Science
Knowledge & Data Engineering Systems

Centralized Federated Learning (CFL):
A distributed learning system with $\mathcal{N}$ clients $\mathcal{N}=n_1, n_2, \ldots, n_{\mathcal{N}}$. Let $\mathfrak{D}_i$ be the local dataset of a client $n_i \in \mathcal{N}$. In CFL , given a subset of $\mathcal{N}' < \mathcal{N}$ clients $\mathcal{N}' \subset \mathcal{N}$, **the local loss for each $n_i \in \mathcal{N}'$ is**:

$$\mathcal{R}_i(\theta) = \frac{1}{|\mathcal{D}_i|} \sum_{(x,y)\in\mathfrak{D}_i} \mathfrak{I}((\theta, x), y)$$

The global loss for selected clients $\mathcal{N}'$ is:

$$\mathcal{R}(\theta^*) = \sum_{n_i\in\mathcal{N}'} \rho_i \, \mathcal{R}_i(\theta) \quad , \textbf{\textit{where}} \ \rho_i = \frac{\mathfrak{D}_i}{\sum_{n_j\in\mathcal{N}'} |\mathfrak{D}_j|}$$

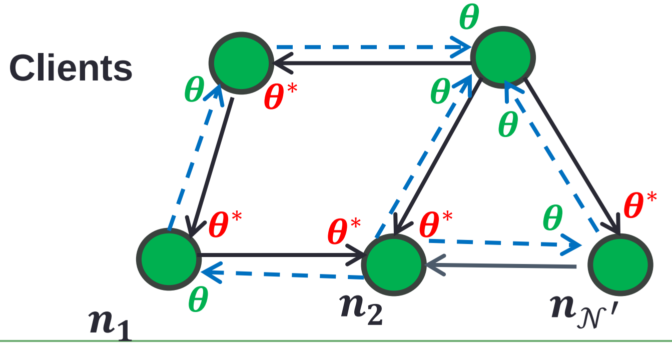


Decentralized Federated Learning (DFL):
In DFL each client n_i communicates only with its neighbors $\mathcal{N}_i \subset \mathcal{N}$ of clients with connections between them. Hence, there is no need for a centralized server to aggregate the locally updated models as in CFL. At round t, each client n_i first aggregates the models received from its neighbors $n_j \in \mathcal{N}_j$,

$$\theta_i^t = \sum_{n_j\in\mathcal{N}_i\cup\{n_i\}} \theta_i^t$$

Then, trains local model $\boldsymbol{\theta}$ using local data $\mathfrak{D}_i$.

$$\theta_i^{t+1} = \theta_i^t - \eta\nabla\mathcal{R}_i(\theta_i^t)$$



Problem Fundamentals

- Each client n_i collects local labelled data $\mathfrak{D}_i = \{X_i \times Y_i \sim \mathcal{P}_i : X_i \in \mathbb{R}^d; Y_i \in \mathcal{L}\}$ from an unknown joint probability distribution $\mathcal{P}_i$
- $\mathcal{L} = \{\ell_1, \dots, \ell_M\}$ all the available labels across all clients in the network.

For any pair of clients (n_i, n_j) with $n_i \neq n_j$, the joint probability distributions can be either **similar** ($\mathcal{P}_i \approx \mathcal{P}_j$) or **dissimilar** ($\mathcal{P}_i \neq \mathcal{P}_j$).

Clients have data with some labels from $\mathcal{L}$ and, in real cases, **not all of them**.

Note: all labels are not known to all clients in advance.

Before Clustering
(Label-aware Client Clustering)

To make the clients aware of the available labels, we introduce **a label-aware distributed mechanism.**

- We rely on **sharing only label distribution among clients** to approximate a prior label distribution per cluster.

Ring-based Label Dissemination:

Each client n_i disseminates only its local labels $\mathcal{L}_i \subset \mathcal{L}$ to neighbours.

In a ring topology, each client n_i sends a message to its neighbour n_j and receives a message from another neighbour $\mathcal{L}_l$.

At round t, client n_i expands its local label set $\mathcal{L}_i$ with the labels received from n_l, i.e., $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \mathcal{L}_l$ and sends $\mathcal{L}_i$ to n_j.

Before Clustering
(Label-aware Client Clustering)

Label-aware Clustering:

Based on the initial label set $\mathcal{L}_i$ and global label set $\mathcal{L}$, each client n_i represents its available labels with a probability $\boldsymbol{P}_i = [p_1, \ldots, p_M] \in [0,1]^M$.

Given $\mathcal{L}_i$ and $\mathcal{L}$, multi-hot encoding $z = [z_1, \ldots, z_M] \in [0,1]^M$ has $z_k = 1$ if $z_k \in \mathcal{L}_i$; $z_k = 0$, otherwise.

> Leader n_l initiates a Minimum Spanning Tree (MST) to incrementally gather all probability label vectors.

$\{P_i\}_{i=1}^{N}$, will be used for clustering the clients into $K < \mathcal{N}$.

- ✓ **Leader** groups nodes' label distributions into K clusters. Each cluster is represented by the cluster label distribution $\boldsymbol{w}_k = [w_{k1}, \ldots, w_{kM}]$ associated with the labels $\ell_1 \ldots, \ell_M$, respectively.
- ✓ Cluster label distributions $\boldsymbol{w}_k$ are incrementally updated upon receiving a client's label distribution $\boldsymbol{P}_i$.

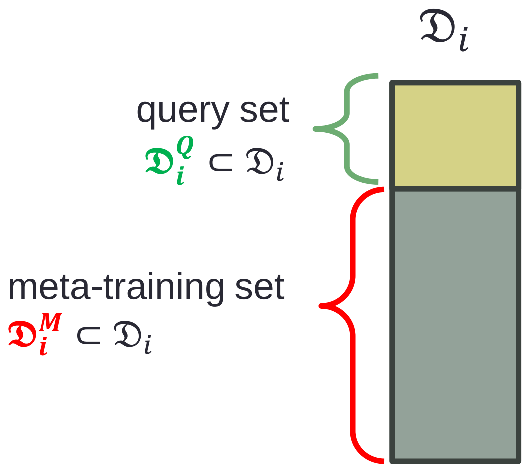
After Clustering

(Cluster-based Multiple Meta-model Learning)

Objective: train a tailored decentralized meta-model $\boldsymbol{f_k}$ for **each cluster** $\mathcal{C}_k, k \in [K]$, capable of fast and flexible adaptation to on-demand tasks with varying label sets $\mathcal{A} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots\}$. This is achieved via fine-tuning using selected samples from clients belonging to each cluster $\mathcal{C}_k$.

- $\boldsymbol{\mathfrak{D}_i^M}$ is used to train the cluster's meta- model $\boldsymbol{f_k}$. **It can be imbalanced**.
- $\boldsymbol{f_k}$ serves as the starting point to learn a generic representation of clients' data in $\mathcal{C}_k$, to use with future tasks' labels $\mathcal{T}_{1,} \in \mathcal{A}$ assigned to $\mathcal{C}_k$.
- $\boldsymbol{\mathfrak{D}_i^Q}$ refers to **labeled-balanced** samples eliminating class imbalances in the fine-tune stage of $\boldsymbol{f_k}$.

$$\boxed{\boldsymbol{\mathfrak{D}_i^Q} \cap \boldsymbol{\mathfrak{D}_i^M} = \emptyset}$$



Meta-models Learning

Within cluster $\mathcal{C}_k$: client $n_i \in \mathcal{C}_k$ locally updates $\theta_{k,i}$ along with neighbors $\theta_{k,j}$, $n_j \in \mathcal{N}_j$ deriving a new **local meta-model** from its meta-training set $\mathfrak{D}_i^M$ over local epochs E_M using SGD.

During round $t \in \{1, \ldots, T\}$, n_i aggregates its neighbors local meta-models as:

$$\boxed{\tilde{\theta}_{k,i}^t = \sum_{n_j \in \mathcal{N}_j} w_j \theta_{k,j}^t \quad , \quad w_i = \frac{|D_i^M|}{\sum_{n_j \in \mathcal{N}_j} |D_j^M|}}$$

Then, computes the gradient of the loss, $\nabla \mathcal{R}(\tilde{\theta}_{k,i}^t)$, updating the local meta-model as:

$$\theta_{k,i}^{t+1} \leftarrow \tilde{\theta}_{k,i}^t - \eta \nabla \mathcal{R}(\tilde{\theta}_{k,i}^t)$$

The cluster-based meta-model $\tilde{\theta}_{k,}^t = \tilde{\theta}_{k,i}^t$, $\forall\, n_i$ is then passed to all clients in the cluster.
This meta-model locally maintained on each client serves as an initial model.

Task-tailored Distributed Meta-model Learning

Based on the previous step, each client n_i **is allocated to a cluster** and is **equipped with a meta-model $\widetilde{\boldsymbol{\theta}}^{\boldsymbol{t}}_{\boldsymbol{k},}$**.

Consider a new incoming task $\mathcal{T}$ requesting the training of a classifier over distributed clients' data with labels $\mathcal{T} = \{\ell_{\mathcal{T}}\} \subseteq \mathcal{L}$.

The task assigned initially to a group $\boldsymbol{\mathcal{C}_k}$ of clients that have the majority of the labels requested in set $\mathcal{T}$ based on the closest group cluster distribution.

$$k = \arg min_{k \in [K]} \mathcal{H}(\boldsymbol{w_k}, \boldsymbol{q})$$

$\boldsymbol{q} = \{q_m\}_{m \in M}$ is the probability label distribution of the task's requested labels $\mathcal{T}$.

Task-tailored Distributed Meta-model Learning

We distinguish two cases:

Case I. If $\boldsymbol{\mathcal{T} \subseteq \mathcal{L}_k}$, then, group $\mathcal{C}_k$ is the most suitable to directly handle this task involving its clients in the training.

Case II. If $\boldsymbol{\mathcal{T} \supset \mathcal{L}_k}$, then:

- ✓ The group $\mathcal{C}_k$ initiates a process for handling the labels in $\mathcal{T} \cap \mathcal{L}_k$.
- ✓ Involve clients from other clusters $\{\mathcal{C}_m\}_{m=1}^{K}\{\mathcal{C}_k\}$ capable of handling the rest of the labels in $\mathcal{T}$= $\mathcal{T}\backslash\mathcal{L}_k$.
- ✓ Keep engaging clusters until all their labels are included in $\mathcal{T}$.
- ✓ Rank these clusters based on their label contribution to task $\mathcal{T}$ and engage the minimum number $m \leq K$ of those cluster whose $\bigcup_{k=1}^{m}\{\mathcal{L}_k\} \subseteq \mathcal{T}$.

Task-tailored Distributed Meta-model Learning

After selecting the most suitable cluster $\mathcal{C}^*$ (Case I) or most suitable clusters $\mathcal{C}^*_+$ (Case II), **the associated clients are engaged in the distributed training of the classifier** as the following.

✓ These clients use their cluster-based meta-models $\boldsymbol{f_k}$ from cluster $\mathcal{C} \in \mathcal{C}^*_+$ to start off the training process.

Note: Even though a substantial amount of relevant labeled-data may be available for $\mathcal{C}^*$or ($\mathcal{C}^*_+$), there might still be a need **for augmentation of data in group $\boldsymbol{\mathcal{C}_k} \in \boldsymbol{\mathcal{C}^*_+}$ with labels $\boldsymbol{\mathcal{T}/\mathcal{L}_k}$**, which are not present in k-th cluster's client data (**missing labels**).

This **facilitates the fine-tuning** of the requested task-tailored meta model.

Task-tailored Distributed Meta-model Learning (Data augmentation)

For each suitable cluster $\mathcal{C}_k \in \mathcal{C}_+^*$, the corresponding clients locally identify their missing labels required per task $\mathcal{T}$.

- These clients generate augmented data labelled by the missing labels using a MixUp meta-model g_ℓ from clients in cluster $\mathcal{C}_\ell \in \mathcal{C}_+^*$, $\ell \neq k$, **for which these labels are not missing.**

- MixUp g_ℓ generates labelled samples $(\boldsymbol{x}, \boldsymbol{y})$ conditioned on the labels $\boldsymbol{y}$ locally on a client $n_i \in \mathcal{C}_k$ such that $\{(\boldsymbol{x}, \boldsymbol{y}): \boldsymbol{y} \in \mathcal{T} \backslash \mathcal{L}_k\}$. Clients within the cluster individually use MixUp models.

Fine-tuning

As a result: a client $n_i \in \mathcal{C}_k$ can now construct its query set $\mathcal{D}_i^Q = \{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{y} \in \mathcal{T} \backslash \mathcal{L}_k\}$ including

(i) The actual data labelled with the requested task labels.

(ii) The augmented data labelled with the associated missing labels.

Subsequently, the task-tailored meta-model notated as $\hat{\theta}^{T'}$ is fine-tuned based on ***the query sets*** of the clients in the suitable clusters $\mathcal{C}_k \in \mathcal{C}_+^*$ after T' fine-tuning rounds.

The local update of the distributed task-tailored meta-model $\hat{\theta}_{k,i,}^{t}$ at fine-tuning round at client n_i from suitable cluster $\mathcal{C}_\ell \in \mathcal{C}_+^*$ uses batch SGD over the query set $\mathcal{D}_i^Q$ is given by:

$$\hat{\theta}_{k,i}^{t+1} \leftarrow \tilde{\theta}_{k,i}^{t} - \eta \nabla \mathcal{R}(\tilde{\theta}_{k,i}^{t})$$

Experimental Evaluation

Experimental Set-up:

- **Images:** MNIST, EMNIST, MEDMNIST, Fashion-MNIST, and CIFAR-100; classes $|\mathcal{C}|$ = (10, 47, 6, 10, 100), respectively.
- **Clients:** $|\mathcal{N}|$ ∈{50, 100, 200,100}.
- **On-demand tasks:** {600, 600, 500, 500}.
- **Fine-tuning data:** $1-\alpha$, α ∈{0.5, 0.6, 0.7}.
-

Baselines

- **Baseline 1: The decentralized FL** (**DFedAvg**).
- **Baseline 2: Cluster-based DFedAvg (C-DFedAvg)**.
- **Baseline 3: Group-based FML (G-FML).**

Note: CL-FML and G-FML, require fine-tuning for their meta-models over relatively small amount of data

Experimental Results

Comparison assessment with baselines (Meta-models)

MNIST								
Metric	**C-DFedAvg**	**DFedAvg**	**G-FML(0.7)**	**G-FML(0.6)**	**G-FML(0.5)**	**CL-FML(0.7)**	**CL-FML(0.6)**	**CL-FML(0.5)**
Training Rounds	10	16	9	10	10	3	4	3
Actual Data Access (%)	100	100	30	40	50	30	40	50
Augmented Data Generation (%)	48.83	61.14	30.588	20.19	18.35	24.4	19.53	14.65
Accuracy (%) / F_1 score	96.80/0.96	94.39/0.93	95.19/0.94	93.12/0.93	94.84/0.94	96.63/0.96	96.45/0.96	96.36/0.97

Fashion-MNIST								
Metric	**C-DFedAvg**	**DFedAvg**	**G-FML(0.7)**	**G-FML(0.6)**	**G-FML(0.5)**	**CL-FML(0.7)**	**CL-FML(0.6)**	**CL-FML(0.5)**
Training Rounds	15	20	10	10	9	6	5	6
Actual Data Access (%)	100	100	30	40	50	30	40	50
Augmented Data Generation (%)	38.46	47.82	14.34	19.12	23.91	11.53	15.38	19.23
Accuracy (%) / F_1 score	86.75/0.88	84.03/0.83	81.98/0.81	82.26/0.82	85.07/0.84	86.03/0.86	86.385/0.86	85.15/0.85

EMNIST								
Metric	**C-DFedAvg**	**DFedAvg**	**G-FML(0.7)**	**G-FML(0.6)**	**G-FML(0.5)**	**CL-FML(0.7)**	**CL-FML(0.6)**	**CL-FML(0.5)**
Training Rounds	15	25	20	18	20	7	6	6
Real Data Access (%)	100	100	30	40	50	30	40	50
Augmented Data Generation (%)	33.185	64	14.34	19.12	23.91	11.53	15.38	19.23
Accuracy (%) / F_1 score	90.85/0.89	89.16/0.88	88.15/0.87	88.22/0.881	88.71/0.88	89.53/0.89	91.29/0.91	89.72/0.89

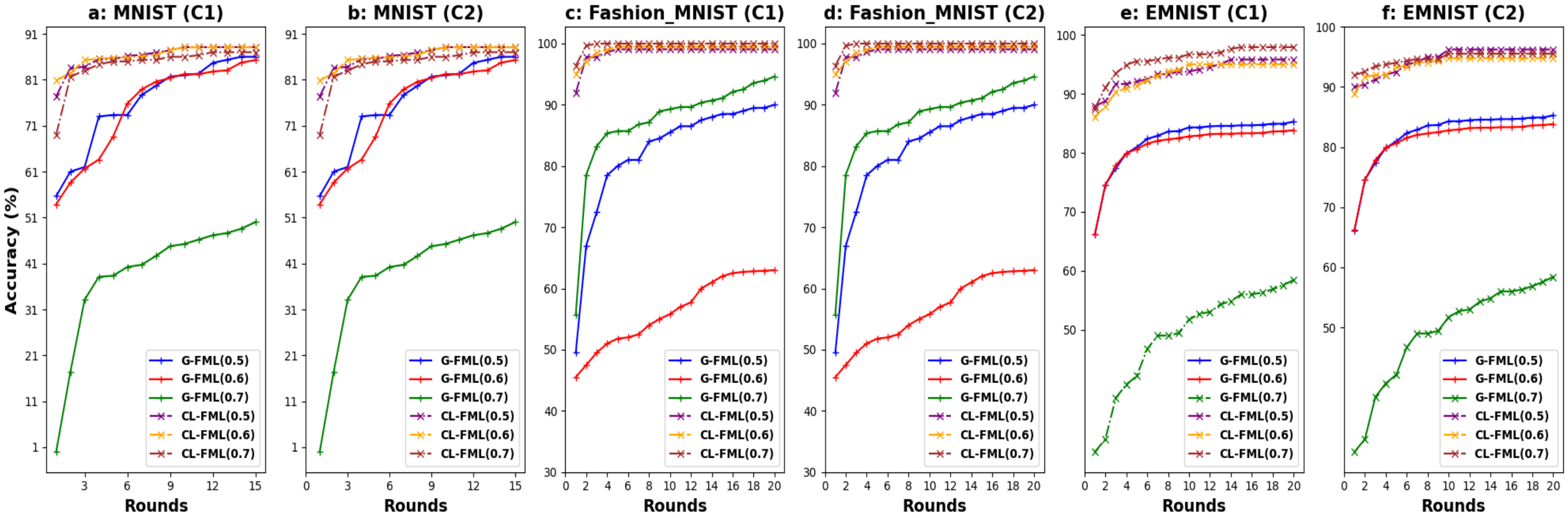
CIFAR-100								
Metric	**C-DFedAvg**	**DFedAvg**	**G-FML(0.7)**	**G-FML(0.6)**	**G-FML(0.5)**	**CL-FML(0.7)**	**CL-FML(0.6)**	**CL-FML(0.5)**
Training Rounds	40	85	25	24	27	12	10	14
Real Data Access (%)	100	100	30	40	50	30	40	50
Augmented Data Generation (%)	36.78	68.2	24.33	27.73	29.83	18.67	22.34	24.57
Accuracy (%) / F_1 score	71.40/0.71	67.59/0.67	67.53/0.67	68.28/0.67	68.67/0.68	70.89/0.74	71.18/0.74	71.15/0.73

University of Glasgow

School *of* Computing Science
Knowledge & Data Engineering Systems

Experimental Results

Multiple meta-models' top-1 accuracy (%) of CL-FML against global meta-model (G-FML) vs. convergence (samples of two groups).



Experimental Results

IMPACT OF OVERLAPPING/SIMILARITY BETWEEN TASKS & CLUSTERS ON FINE-TUNED MODELS PERFORMANCE.

$\mathcal{P}(\mathcal{T}\|sim)$	Similarity	C-DFedAvg	DFedAvg	G-FML(0.7)	G-FML(0.6)	G-FML(0.5)	CL-FML(0.7)	CL-FML(0.6)	CL-FML(0.5)
MNIST									
20%	80%	98.96%	98.65%	99.29%	99.10	99.34%	98.96%	99.55%	99.24%
30%	67%	97.11%	96.074	95.48%	95.88%	96.07%	96.01%	96.61%	96.81%
30%	50%	96.91%	96.56	97.13%	96.73%	96.91%	97.86%	97.51%	97.11%
20%	30%	97.01%	96.01	95.56%	96.55%	96.48%	97.01%	97.34%	97.23%
Fashion-MNIST									
20%	80%	91.96%	91.14%	87.80%	88.89%	90.03%	89.40%	92.11%	90.79%
30%	70%	90.04%	85.15%	86.13%	87.96%	88.77%	87.52%	89.36%	90.94%
30%	50%	86.48%	77.71%	71.18%	72.22%	83.52%	86.80%	86.92%	86.17%
20%	25%	79.78%	75.56%	75.77%	79.65%	79.55%	80.58%	79.55%	78.94%
EMNIST									
20%	80%	93.27%	89.38%	92.21%	93.25%	92.83%	93.81%	92.15%	93.35%
30%	70%	90.18%	90.41%	91.35%	88.80%	88.50%	88.59%	91.71%	89.61%
30%	50%	89.25%	89.20%	85.70 %	86.75%	87.33%	86.34%	86.47%	89.56%
20%	30%	79.64%	87.65%	83.34%	83.84%	85.19%	86.38%	84.84%	83.84%
CIFAR-100									
20%	80%	72.84%	65.60%	66.40%	67.60%	67.77%	71.20%	72.00%	72.82%
30%	70%	69.92%	67.60%	68.20%	67.86%	68.50%	69.54%	70.01%	70.07%
30%	50%	74.84%	71.13%	68.91 %	70.38%	70.67%	74.56%	73.78%	73.09%
20%	30%	68.43%	66.05%	66.63%	67.31%	67.77%	68.28%	68.95%	68.54%

Conclusions

- We introduced the **CL-FML** framework for classification tasks with label-shifting across distributed clients.

- **CL-FML** leverages decentralized federated meta-learning via a novel label-driven client clustering, where multiple cluster-based meta-learning models deal with any arbitrary classification tasks.

- **CL-FML** leverages data augmentation to train on- demand out-of-distribution classifier training.

- Comprehensive experiments against baselines showcase the superiority of **CL-FML**.

University of Glasgow

School *of* Computing Science
Knowledge & Data Engineering Systems

Thank you!

Tahani Aladwani