International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Leveraging Organic Biomass for Advanced Cosmetics Formulations

Selvi[1], Abisha A[2]

[1,2]Panimalar Engineering College, Chennai

Abstract

The main element thought to be responsible for the decline the clients are distinct, they are typically connected. In FMTL, a **in federated learning (FL) performance is the dissemination of non-**regularization term that captures the relationships between the **independent and identically distributed (non-IID) data among clients.** clients' models is minimized to promote mutual impact between the **Research groups are very interested in a number of methods for** clients' models. Regretfully, the FMTL problem has not explicitly
handling non-IID data, including federated multitask learning taken these linkages into account. Furthermore, there is typically
(FMTL) and personalized FL. In order to explicitly utilize the less research done on FMTL algorithms that are nonconvex and
relationships between the client models for multitask learning, we communication-decentralized with guaranteed convergence.
first define the FMTL issue using Laplacian regularization. Next, we present a fresh perspective on the FMTL problem that demonstrates for the first time that the formulated FMTL problem is applicable to both conventional and customized FL. Additionally, we suggest two algorithms, FedU and decentralised FedU (dFedU), to address the FMTL problem in decentralised and communication-centralised schemes, respectively. In theory, we demonstrate that both methods' convergence rates result in sublinear speedup of order 1/2 for nonconvex objectives and linear speedup for strongly convex objectives. Through experiments, we demonstrate that our algorithms perform better than the standard algorithms pFedMe and Per-FedAvg in customized FL settings, MOCHA in FMTL settings, and FedAvg, FedProx, SCAFFOLD, and AFL in FL settings.

Keywords: Laplacian regularization, federated learning (FL), federated multi-task learning (FMTL), and personalized learning.

I. INTRODUCTION

A promising distributed and privacy-preserving technique for creating a global model from a large number of handheld devices is federated learning (FL), which has recently gained attention [1], [2], [3], and [4]. FL has several futuristic uses, including identifying potential disease symptoms (e.g., diabetes, heart attack, stroke). Most of the crime cases are not properly recorded and followed. Using the proper data only, government can initiate, prevention and mitigation methodologies through campaign and officials. using wearable technology in healthcare systems [5, 6, 7], or using Internet-of-things devices in smart cities [8,

9] to forecast the likelihood of disasters. The naturally non-independent and identically distributed (non-IID) data distributions among clients are one of the main obstacles in FL [10], [11]. The generalization error of the FL global model on each client's local data rises sharply with the number of data distribution discrepancies between clients [12], [13].

Federated multitask learning (FMTL) [16] and personalized FL [14], [15] have been suggested as ways to deal with non-IID data distributions among clients. The goal of Personalized FL is to create a global model that can be used to identify a "personalized model" for the local data of each client. In this case, the global model is regarded as the "agreed point" from which each client can begin customizing its model according to its diverse local data distribution. Motivated by multitask learning frameworks, FMTL seeks to concurrently learn distinct models, in contrast to personalized FL [17], [18]. Every client's data distribution is fit by one of these models. Therefore, without creating a global model like personalized FL, FMTL immediately tackles the problem arising from non-IID data distributions. However, it is noted from the perspective of local data at clients that clients with comparable characteristics (e.g., area, time, age, and gender) are likely to exhibit similar behaviors. As a result, even though the models of The following are this work's primary contributions.

1. Using Laplacian regularization, we create an FMTL issue that explicitly utilizes the relationships between the client models.
2. To address the defined FMTL problem, we provide the decentralized FedU (dFedU) and communication-centralized FedU algorithms. Additionally, we examine how quickly FMTL algorithms with convex and nonconvex objective functions converge.
3. Using actual datasets that represent the non-IID data distribution across clients, we empirically assess FedU and dFedU's performance. We demonstrate that FedU and dFedU perform better than the standard algorithm FedAvg in FL settings, the conventional algorithm MOCHA in FMTL conditions, and pFedMe and Per-FedAvg in customized FL settings with respect to local accuracy.

II.RELATED WORK

A. FEDERATED LEARNING:

FedAvg [1], one of the first FL works, creates the global model by averaging the local stochastic gradient descent (SGD) updates. An array of techniques [11], [19], [20], [21], and [22] are shown to enhance the global model's resilience in non-IID contexts. FedProx [19], for instance, addresses the statistical heterogeneity of clients by appending a proximal term to the local aim.

B. PERSONALIZED FL:

A number of customized FL strategies have been put out to address the problems caused by non-IID data in the traditional FL. While [24] used this mixing to jointly learn compact local representations on each client and a global model across all clients, [13], [23] tried to combine a local model with the global model. Motivated by the development of a globally applicable model that can rapidly adjust to the client's data following a few gradient descent steps, Moreau envelopes were employed by pFedMe [14], whereas Per-FedAvg [15] adopted model-agnostic meta-learning [25], an advancement in meta-learning techniques. To enhance FL customization, Jiang et al. [26] suggested combining FedAvg and Reptile [27].FedPer is an alternative customized FL method for deep neural network (DNN)training[28].

C. FEDERATED MULTITASK LEARNING:

Learning distinct models that fit each local data distribution is an additional strategy for handling non-IID data distributions at

clients. Accordingly, FMTL was initially presented in [16], where a systems-aware optimization framework called MOCHA is suggested for managing stragglers and fault tolerance in FL contexts. In addition, a number of other works have been written about FMTL. A framework for extended total variation minimization was presented by Sarcheshmehpour et al. [29], and it is helpful in FMTL networks. An FMTL algorithm was presented by Li et al. [30] to address the problems of robustness, accuracy, and fairness in FL. Shen et al. [31] used approximated variational inference to construct an FMTL method by treating the FL network as a star-shaped Bayesian network. An FMTL algorithm for online applications was the main emphasis of Li et al. [32]. Nevertheless, the convergence rate of FMTL with nonconvex objectives has not been examined in any of these research. Furthermore, the research has not yet examined the connections between the FMTL, standard FL, and tailored FL issues.

II.FTML: NEW VIEW

A. FORMULATION OF THE FMTL PROBLEM WITH LAPLACIAN REGULAZATION:

Fitting distinct models (i.e., wk Rd, k N) to the local data of customers while accounting for the relationships between these models is the aim of FMTL in this study. In a mobile network, for example, smart-device clients are attempting to learn about their behaviours by leveraging their private and personal information (such as image, text, voice, and sensor data). Their data may have non-IID distributions in FL situations since it may originate from many contexts, applications, and surroundings. Nevertheless, these customers are likely to act similarly in comparable situations or with similar characteristics (e.g., location, time, and age). Consequently, there are typically connections between the client models [33], [34], and [35]. A connected graph G N, E, A is used to show the relationships between the client models. N 1,..., N is the set of vertices that represent FL clients, E is the set of edges that represent relationships between the client models, and A RN is a symmetric, weighted adjacency matrix with ak4 A k4. Ak4 presents the reversible relationship between customers k and 4 (ak4 a4k, k, 4). In this case, ak4 0 indicates that clients k and 4 have no association with each other. The intensity of the association between the models of these two clients is determined by the value of ak4 > 0, which also indicates that client k is a neighbour of client 4. Let [D]kk = N ak4 and let D N be a diagonal matrix. Thus, L = 4 1 D − A is the graph's Laplacian matrix. Assume ,

W = [w T,..., w T]

Let L 1:= L ⊗ and T ∈ R d N be a collective model vector. N would be a matrix of Laplacian regularization. We now formulate the FMTL problem as follows:

where fk (wk ζk) is the regularized loss function corresponding to this sample and wk, and ζk is a random data sample taken from the client k distribution. When k < 4, the distribution of ζk and ζ4 can be different Please take note that we do not use any visual aids in our work to determine how similar the current client relationships are techniques to advance our suggested approach. Rather, we use a Laplacian regularization matrix L to show the current relationships between the clients' models, which we then insert into the Laplacian regularization term in the FMTL problem's objective function (1). Theoretically, in (1), the regularization hyperparameter η 0 regulates how each local model is affected by the models of nearby customers. Each client learns its own model each week using its own local data, independent of the server or other clients, if η = 0. This is known as an individual learning problem. The models of the nearby clients are encouraged to be near to one another if η > 0 by minimizing the Laplacian regularization term.

Section VI of the experiment will demonstrate how the performance of our suggested algorithms is affected by the current relationships between the client models.

Remark 1: Other regularization techniques can be used to encourage the models of the neighbouring clients to be close to one another. For example, Network Lasso uses wk w4 instead of w4 2 in (3) [36], [37], [38], while MOCHA uses tr(W ▲W) instead of (3) [16], where W < w1,..., wN Rd×N. However, the problem in [39], where a number of techniques are devised for strongly convex objectives, is generalized in problem (1). The generalized total variation minimization problem [29], which is resolved using a primal-dual approach for convex objectives, is comparable to issue (1). Problem (1) has a convex variant according to Vanhaesebrouck et al. [40]. which an alternate direction method of multipliers (ADMM) decentralized algorithm solves. In (1), we use the Laplacian regularization matrix L to introduce the FTML problem. We are able to successfully construct FMTL algorithms utilizing SGD by leveraging the unique qualities of L. Crucially, our algorithms can function in the following situations: 1) with both strongly convex and nonconvex goal functions, and 2) in both decentralized and centralized communication methods.

Assumption 1 (Smoothness) states that Fk is β-smooth for every k in N, meaning that for any w, wr ∈ Rd.

$\|\nabla F_k(w) - \nabla F_k(w^r)\| \leq \beta \| w - w^r \|$.

Assumption 2 (Strong Convexity): Fk is α-strongly convex for every k ∈ N, meaning that for every w, wr ∈ Rd.

$$F(w) \geq F(w^r) + \nabla F(w^r), w - w^r + \frac{\alpha}{2}\|w - w^r\|^2$$

The third assumption (limiting variance) states that the set of ∇ Fk(w, ζk), k ∈ N, is made up of unbiased stochastic gradients of ∇ Fk(w), k d∈N N, with σ 2 bounded the total variance, meaning that for each W ∈ R

$$\min_W J(W) = \underbrace{F(W)}_{\text{Global loss}} + \underbrace{\eta R(W)}_{\text{Laplacian regularization}} \qquad \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}_{\zeta_k} \|\nabla F_k(w_k, \zeta_k) - \nabla F_k(w_k)\|^2 \leq \sigma_1^2 \quad (1)$$

$$F(W) = \sum_{k=1}^{N} F_k(w_k) \quad (2)$$

$$R(W) = W^T L W = \frac{1}{2}\sum_{k=1}^{N} \sum_{4 \in N_k} a_{k4} ||w_k - w_4||^2 \quad (3)$$

Where,

The Euclidean norm is Nk N k, and. The expected loss function at client k is represented by Fk( ).

We see that the assumption of individual limited variance, which is applied to each client in FL and customized FL situations, is stronger than Assumption 3 [10], [14], and [15]. Additionally, (1) is somewhat comparable to the multitask learning problem in [41] and [42]. The latter, however, necessitates that each Fk (wk) be twice differential as well as evenly bounded from below and above.

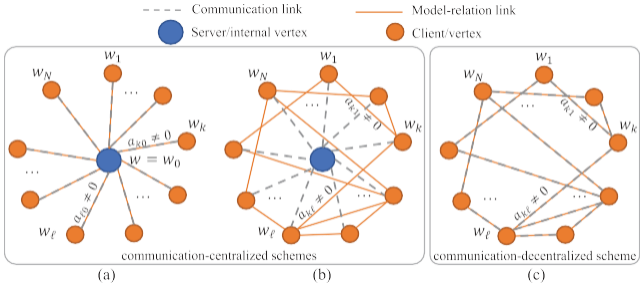


Figure 1 shows examples of FL's undirected weighted graphs. (a) A star graph with a server for both customized and conventional FL. Entity graph for FMTL with and without a server (b) and (c) problem does not take into account the issue of non-IID data distributions among clients, and thus, it is not formulated for FL settings.

B.NEW VIEW OF THE FMTL PROBLEM:

First, we note that in both traditional and customized FL, every client establishes a connection with a server using the communication-centralized method depicted in Fig. 1(a). A star graph is used to show the relationships between the client and server models. A server is represented as a virtually internal vertex 0 in this network, with a model w0 and a loss function F0

0. All of the client models in this case are only connected to the server model w0, that is, ak0 > 0, k, but not to one another, that is, ak4 0, k,4 0. This work focuses on developing FMTL algorithms to solve problem (1), assuming that the weights ak4 are known.

There are references to [43] and [44] regarding the discovery of ak4 in certain learning applications. In the following, we demonstrate that the FMTL problem (1) may be applied to both the normal FL and some customized FL types. We refer to LSGD-PFL [45] for a more broad optimization problem of customized FL.

1) FMTL's Relation to Conventional FL: The following problem's objective function can be viewed as a Lagrangian function:

$$\min_{W} \sum_{k=1}^{N} F_k(w_k) \quad \text{s.t. } w_1 = w_2 = \cdots = w_N \qquad (4)$$

This is comparable to the standard FL problem (FedAvg) [1]. Thus, (1) can be solved to obtain the solution of the typical FL problem.

2) FMTL's Relation to Customized FL Using pFedMe's Moreau Envelopes: The following is a simulation of the pFedMe [14] problem:

$$\min_{w} J(w) = \sum_{k=1}^{N} \tilde{J}_k(w) \qquad (5)$$

where $\tilde{J}_k(w) = \min F_k(z_k) + (\eta/2)||z_k - w||^2$. We observe that

$$J(w) = \sum_{k=1}^{N} \min_{z_k} F_k(z_k) + \frac{\eta}{2}\|z_k - w\|^2 = \min_{z_1,\ldots,z_N} \sum_{k=1}^{N} F_k(z_k) + \frac{\eta}{2}\|z_k - w\|^2 .$$

Consequently, the following issue with z0 = w and F0 ≡ 0 is equal to (5):

$$\min_{z_0, z_1, \ldots, z_N} \sum_{k=0}^{N} F_k(z_k) + \frac{\eta}{2} \sum_{k=0}^{N} \| z_k - z_0 \|^2$$

This, with the star graph topology and $a_{k0} = 1$, $\forall k \in N$, is a special case of (1).

3) Relation of FMTL to Meta-Learning-Based Personalized FL (Per-FedAvg): The problem of Per-FedAvg [15] is given by

$$\min_{w} J(w) = \sum_{k=1}^{N} F_k(w - \mu \nabla F_k(w)) \qquad (6)$$

ongoing. Set $4_k = (L_k/2)$, $k \in N$, and $w_k = w - \mu \nabla F_k(w)$.11: send the server w(t).By applying [46, Lemma 1.2.3] twice, we obtain that, for any $z_k \in R^d$ and for $\mu < \min_k 4_k$

$$\begin{aligned} F_k(w_k) &\le F_k(w) + (\nabla F_k(w), w_k - w) + 4_k \| w_k - w \|^2 \\ &= F_k(w) - (\mu - 4_k \mu^2) \| \nabla F_k(w) \|^2 \\ &\le F_k(z_k) + (\nabla F_k(w), z_k - w) + 4_k \| z_k - w \|^2 \\ &\quad - (\mu - 4_k \mu^2) \| \nabla F_k(w) \|^2 \\ &= F_k(z_k) + a_{k0} \| z_k - w \|^2 \\ &\quad - (\mu - 4_k \mu^2) \nabla F_k(w) - \frac{z_k - w}{\quad} \end{aligned}$$

where $4_k + (1/4(\mu - 4k\mu 2) = a_{k0}$. Therefore, $J(w) \le \min F$
$(z) + a\, z - w$, since $F_k(w_k) \le \min F_k(z_k) + a_{k0} z_k - w\, 2$.

$$\begin{aligned} J(w) &\le \sum_{k=1}^{N} \min_{z_k} F_k(z_k) + a_{k0} \| z - w \| \\ &= \min_{z_1, \ldots, z_N} \sum_{k=1}^{N} F_k(z_k) + a_{k0} \| z_k - w \|^2 \end{aligned}$$

With $z_0 = w$ and $F_0 = 0$, (6) can now be resolved using the following epigraph problem:

$$\min_{z_0, z_1, \ldots, z_N} \sum_{k=0}^{N} F_k(z_k) + \frac{\eta}{2} \sum_{k=0}^{N} a_{k0} \| z_k - z_0 \|^2$$

Therefore, with the star graph topology and $a_{k0} = 1$, $\forall k \in N$, is also a special case of (1).

III.FTML ALGORITHMS

A. FEDU CENTRALIZED ALGORITHM FOR COMMUNICATION:

In order to solve the defined FL problem (1) under the communication-centralized scheme, we provide an algorithm FedU in this part. This algorithm is introduced in Algorithm 1. In this case, an entity graph is used to record the connections between 1Every vertex in an entity graph represents a value of an entity (such as a person), and if two entities are (t+1) (t) seen to be similar, then there is an edge (such as friendship) between them [43]. The system uses the local updates from the participating entities to update the global model at each communication cycle in the FedU algorithm.

The main benefit of this method, nevertheless, is how the entity graph is used to improve the communication process's dependability and efficiency. The system can optimize which parties disclose their model updates and coordinate the aggregation process by examining the links between entities. This ensures that only the most pertinent information is shared, reducing needless communication cost. When working with large-scale and distributed systems, this graph-based method efficiently increases

convergence speed and lowers the communication costs related to federated learning. The two clients share a specific resemblance model and are neighbors of one another.

Algorithm 1 FedU

client k's input: local step-size μ
server's input: graph information $\{a_{k4}\}$, initial $w_k^{(0)}$, $\forall k \in N$, and global step-size $\mu = \mu R$
for each round $t = 0, \ldots, T-1$ **do**
  server uniformly samples a subset of clients $S^{(t)}$ of size S and sends $w_k^{(t)}$ to client k, $\forall k \in S^{(t)}$
  on client $k \in S^{(t)}$ **in parallel do**
    initialize local model $w^{(t)} \leftarrow w^{(t)}$
    for $r = 0, \ldots, R-1$ **do**
      compute mini-batch gradient $\nabla \widetilde{F}_k(w_{k,r})$
      $w_{k,r+1}^{(t)} \leftarrow w_{k,r}^{(t)} - \mu \nabla \widetilde{F}_k(w_{k,r}^{(t)})$
    end for
    send $w_{k,R}^{(t)}$ to the server
  end on client
  on server do
    $w_{k,R}^{(t)} \leftarrow w_k^{(t)}$, $\forall k \notin S^{(t)}$
    $w_k^{(t+1)} \leftarrow w_{k,R}^{(t)} - \mu\eta \sum_{4 \in N_k} a_{k4}(w_{k,R}^{(t)} - w_{4,R}^{(t)})$, $\forall k \in S^{(t)}$

The server then receives the most recent local update from the sampled clients to carry out model regularization for each local model following the completion of the R local update stages. It should be noted that, unlike the star graphs of the traditional FL and customized FL, the entity graph only shows relationships between client models and no server models. As a result, FedU differs significantly from both the personalized FL algorithms . Specifically, each client k ∈ S(t) copies the server's current local model, which is w(t) = w(t), and changes the form locally R times throughout each communication round.

$$w_{k,r+1}^{(t)} \leftarrow w_{k,r}^{(t)} - \mu \nabla \widetilde{F}_i \quad w_{k,r}^{(t)}$$

where the local step size is denoted by μ. Next, the server gets updates from sampled clients k ∈ S(t) and {wk,R}.

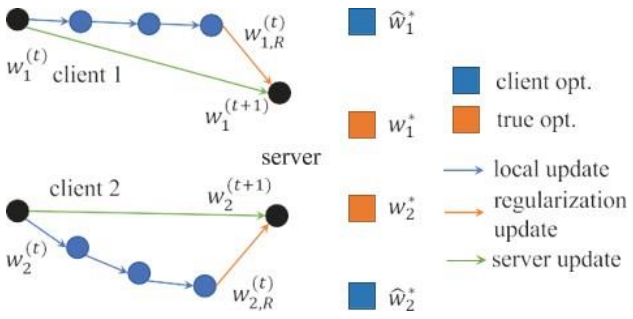
$$w_{k,R}^{(t)} \leftarrow w_k^{(t)}$$

for any client k ∈/ S(t) that is not sampled. Lastly, for any sampled client k ∈ S(t), the server does its regularization update as follows.

$$w_k^{(t+1)} \leftarrow w_{k,R}^{(t)} - \mu\eta \sum_{4 \in N_k \cap S^{(t)}} a_{k4}(w_{k,R}^{(t)} - w_{4,R}^{(t)})$$

and as follows for any non sampled client k ∈/ S(t):

$$w_k^{(t+1)} \leftarrow w_{k,l}^{(t)}$$



FedU's client and server side update processes are shown in Fig. 2 for two related jobs (clients) with three local steps (N = 2, R = 3) at round t. The client is approached by the local updates. where the global step size is $\mu = \mu R$. This step completes the communication round.Fig. 2 illustrates the FedU method using

N = 2 sample clients. The two clients share a specific resemblance model and are neighbors of one another. Let the global solution (also known as the real optimum or true opt.) be (w1∗, w2∗). Each round's local and regularization updates will result in the FedU convergent solution being (w1∗, w2∗).

B. dFedU: The server at the start of the learning process that uses a decentralized version of FedU. However, maintaining all of the graph's information (such as vertices and weighted edges) and storing all model updates on the server may not be feasible in a network with thousands of clients. This leads us to suggest dFedU, a decentralized FedU variant that is shown in Algorithm

2. In particular, each client of an entity graph [as seen in Fig. 1(c)] executes R local updates and transmits its updated model to its nearby clients in order to carry out the model regularization during each communication round. In this case, none of the numerous customers in the entire network need to speak with one another. All that any client has to do is talk to its neighbors.

IV FTML CONVERGENCE RATE

$$\sum_{k=1}^{N} \|\nabla F_k(w_k)\|^2 \le \sigma_2^2 + \sum_{k=1}^{N} \|\nabla_{w_k} J(W)\|^2$$

The FedU and dFedU convergence rates are shown in this section. Let the best answer to (1) be W ∗ w1∗,..., w∗N. Lemma 1: Assume that ηρ > 2β and that Assumption 1 is true, where ρ:= L. Then, for any W ∈ RdN, there exists σ2 ≥ 0, such that σ2 =∇ F(0) (ηρ/(ηρ − 2β))1/2.

(7)

where the gradient of J with respect to wk is denoted by wk J(W).

Therefore, in the event that each Fk is convex, then

$$\sum_{k=1}^{N} \|\nabla F_k(w_k^*)\|^2 \le \sigma_2^2.$$

(8)

By adjusting ηR, the requirement ηρ > 2β in Lemma 1 can always be met for any given value of ρ. As a result, η can manage the impact of the relationships between the client models toward w∗ and advance w(t) toward w∗, which ultimately results in the updated model after round t, i.e., (w(t+1), w(t+1)), convergence of FedU and dFedU. To meet this requirement, one can select a big η if ρ is small and vice versa.Keep in mind that (7) is rewritten as follows in the traditional FL setup, where wk = w, ∀k ∈ N:

$$\frac{1}{N}\sum_{k=1}^{N} \|\nabla F_k(w)\|^2 \le \frac{\sigma_2^2}{N} + \gamma^2 \|\nabla_w J(W)\|^2 \text{ with } \gamma = 1$$

(9)

which precisely matches the γ-local dissimilarity in [19] with σ2

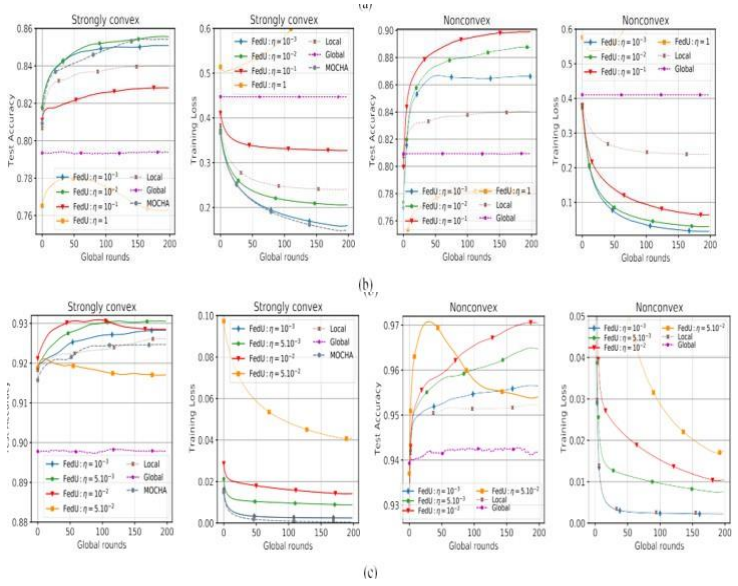
= 0 and the assumptions of (σ2/N,γ)-bounded gradient dissimilarity in [10] and [22]. In this scenario, σ2 0 and γ 1 correspond to IID cases, whereas σ2 0 and γ 1 correspond to non- IID cases.Let σ2 and ρ be defined as in Lemma 1 from now on, and W(t) w(t),..., w(t) be the collective vector produced by FedU (with client sampling) or dFedU (without client sampling, i.e., S N) at round t. Keep in mind that when S = N, the convergence rate of FedU is simply converted to the convergence rate of dFedU. We demonstrate in the following theorems that FedU admits sublinear speedup of order 1/2 for nonconvex goal functions and linear speedup for highly convex objective functions. Convergence in Strongly Convex Cases (Theorem 1): Assume that η > (2β/ρ) and that Assumptions 1–3 are true. Next,

exists $\mu \leq (\bar{p}_1 / R)$, such that, for any $T \geq (4N/\bar{p}_1 \alpha S)$

$$\mathrm{E}[J(\widetilde{W}^{(T)}) - J(W^*)] \leq \widetilde{\Gamma} \quad \alpha\alpha^{(0)} e^{-\frac{\bar{p}_1 \alpha S T}{4N}} + \frac{\sigma_1^2}{(\alpha T)^2 RS} + \frac{\sigma_2^2}{(\alpha T)^2 S} + \frac{\sigma_1^2}{\alpha T RS} + \frac{\sigma_2^2}{\alpha TS} \tag{9}$$

Algorithm 2 dFedU

1: **client** k**'s input**: $\{a_{k4}\}$, $\widetilde{N}_k$, initial w_k , $\forall$ $\in N$, local step-size μ, and global step-size $\mu = \mu R$
2: **for** each round t $= , \ldots ,$ 1 $- 1$ **do**
3: **on client** $k \in N$ **in parallel do**
4: initialize local model $w_{k,0}^{(t)} \leftarrow w_k^{(t)}$
5: **for** $r = 0, \ldots, R-1$ **do**
6: compute mini-batch gradient $\nabla \widetilde{F}_k(w_{k,r})$
7: $w_{k,r+1}^{(t)} \leftarrow w_{k,r}^{(t)} - \mu \nabla \widetilde{F}_k(w_{k,r}^{(t)})$
8: **end for**
9: send $w_{k,R}^{(t)}$ to its neighboring clients in $\widetilde{N}_k$
10: **end on client**
11: **on client** $k \in N$ **in parallel do**
12: $w_k^{(t+1)} \leftarrow w_{k,R}^{(t)} - \bar{p}\,\eta \quad_{4 \in \widetilde{N}_k} a_{k4}(w_{k,R}^{(t)} - w_{4,R}^{(t)})$
13: **end on client**
14: **end for**



(b)

(c)

where q = (128β ηρ/α) + 12(β + ηρ) + (96β2/α) + (32 pβ2/αηρ), p
= 2(β + ηρ) + (8η2ρ2/α)+ (64β2/α)+ (12(β + ηρ)2/ηρ)+6ηρ + (48β2/ηρ), where μ1min (1/q), (2/ηρ),

$$T = \widetilde{O}\frac{1}{\alpha S} + \frac{\sigma_1}{\alpha\sqrt{\varepsilon RS}} + \frac{\sigma_2}{\alpha\sqrt{\varepsilon S}} + \frac{\sigma_1^2}{\alpha RS\varepsilon} + \frac{\sigma_2^2}{\alpha S\varepsilon} \quad . \quad (10)$$

V.EXPERIMENTS

In this part, we assess FedU 1exists μ ≤ (˜μ1 /R) so that, for each T ≥ (4N/ ˜μ1 αS), in both strongly convex and nonconvex scenarios, the data are heterogeneous and non-IID. Comparing FedU with state-of-the-art learning algorithms, we demonstrate the improvements of FedU with Laplacian regularization in fed-erated multitask and personalized situations, including

A. EXPERIMENTAL SETTINGS:

Using genuine datasets created in federated contexts, such as MNIST, CIFAR-10, Vehicle Sensor, and Human Activity Recognition, we examine classification difficulties.

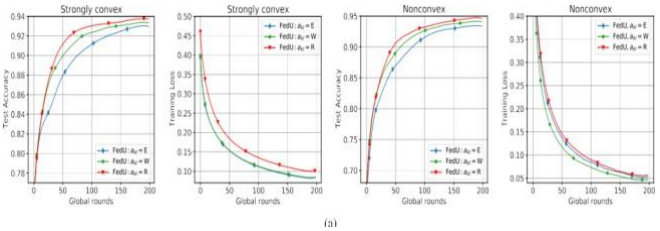
1. Human Activity Recognition: The collection of information from 30 people's cell phones' accelerometers and gyroscopes while they were engaged in six distinct activities, such as sitting, standing, walking, and lying down .
2. Vehicle Sensor: Information is gathered from a distributed wireless sensor network of 23 sensors, such as infrared, seismic, and acoustic sensors (microphones and geophones) of moving automobiles. To anticipate two vehicle types—a dragon wagon (DW) and an assault amphibian vehicle (AAV), we treat each sensor as a distinct assignment.
3. MNIST: A dataset of handwritten digits [53] with 70,000 instances and 10 labels. The entire dataset is dedicated to N 100 customers. Every customer has two over ten labels and varies in the extent of their local data.
4. CIFAR-10: A dataset for object identification [54] that comprises 60,000 color photos in ten classes. We contrast our rates in IID scenarios with those of FL and customized FL algorithms for illustrative purposes.

Some clients in real-world FL networks require collaborative learning with others since their data sizes are severely constrained. Therefore, we downsampled 80% of the data for each dataset that belonged to Fig. 3. Performance comparison among MOCHA, local model, global model, and FedU with the various sets of η in both strongly convex and nonconvex settings. (a) Human Activity. (b) Vehicle Sensor. (c) MNIST. to half of all clients in order to track how each algorithm behaves. In [47, Appendix F], we give all the information regarding datasets and findings without downsampling. 75% and 25% of each dataset are randomly assigned to training and testing, respectively.As the highly convex model for MNIST, Vehicle Sensor, and Human Activity Recognition, we employ a multinomial logistic regression (MLR) model with cross- entropy loss functions and an L2-regularization term. For the Human Activity and Vehicle Sensor datasets, we employ a straightforward DNN with a single hidden layer, a ReLU activation function, and a softmax layer at the network's end for nonconvex settings. The buried layer is 20 for the vehicle sensor and 100 for human activity. For MNIST, we employ a DNN with two hidden layers, each of which has a size of 100. We use

[1]'s CNN structure for CIFAR-10.Following the parameters of [16] and [24], the structural dependence matrix ▲ of MOCHA is selected as ▲ = (IN×N − (1/N)11T)2, where 1 is a vector of all ones size N and IN×N is the identity matrix with size N N. In this case, ▲ is precisely issue (1)'s Laplacian matrix L where every weight is ak4, k, and 4. Since FedU and dFedU perform equally in the absence of customer sampling,

we exclusively assess FedU's performance in our trials. To determine which combination of hyperparameters enables each algorithm to attain the maximum test accuracy, we do fivefold cross validation when comparing FedU with other algorithms. PyTorch version 1.6 [55] is used for all experiments. We adhere to the implementations of [24] for MOCHA and [14] for pFedMe, FedAvg, and Per-FedAvg. Every experiment is conducted on an NVIDIA Tesla T4 GPU. The location of all code and data is https://github.com/dual-grp/FedU_FMTL. Over ten runs, the accuracy is reported with the mean and standard deviation.

B) FEDU PERFORMANCE IN FMTL:



(a)

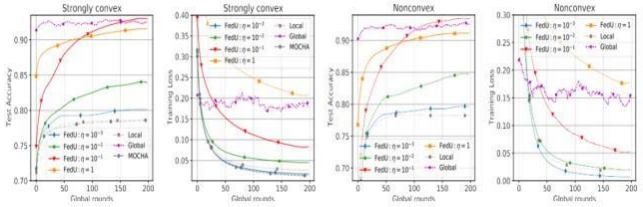
We first demonstrate FedU's advantages in FMTL by contrasting it with MOCHA, the traditional local model, which trains one distinct model for each client, and the global model, which trains one unique model on centralized data.

distinct {akl } situations and standardizing the values of {akl } within the interval [0, 1].

Random (R): Akl ∼ N(0, 1) is created at random for all values of akl.

Equal (E): We can select any value of akl between 0 and 1 when all clients have the same value for akl. Nonetheless, FedU will be able to get the maximum accuracy with a single value of ηakl. Therefore, we can select a small η whenever akl is large, and vice versa. We set akl to 0.5 in this experiment and modify η appropriately.

Weighted (W): We set akl 0 on the correlation between these customers because there are a number of them with quite little data quantities. Next, we put akl 0.5 on the relationship between clients with small and large data sizes, and akl.



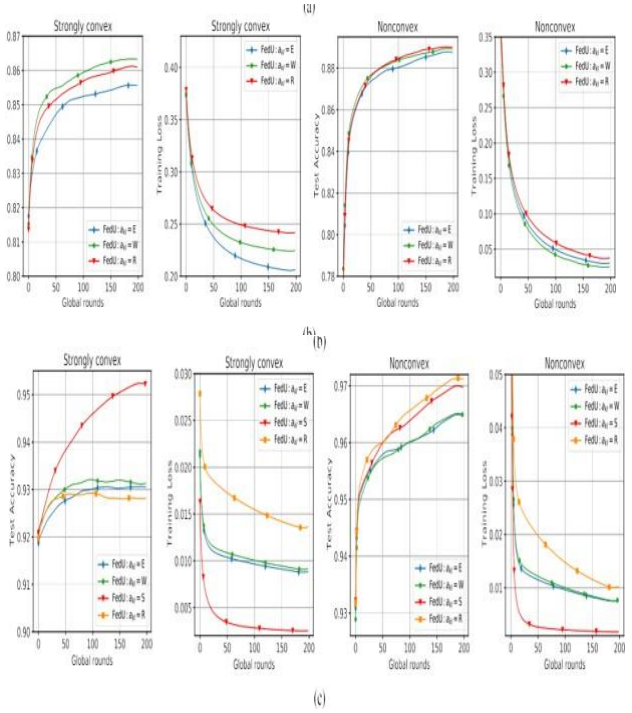

(a)
Strongly convex
Strongly convex
Nonconvex
Nonconvex
Test Accuracy
Training Loss
Global rounds
(b)
Strongly convex
Strongly convex
Nonconvex
Nonconvex
Test Accuracy
Training Loss
Global rounds
(c)

Fig. 4. Effects of graph information akl on{the}convergence of FedU in both convex and nonconvex settings. (a) Human Activity. (b) Vehicle Sensor.(c)

MNIST

We first demonstrate FedU's advantages in FMTL by contrasting it with MOCHA, the traditional local model, which trains one distinct model for each client, and the global model, which trains It should be noted that MOCHA and the FMTL algorithm's performance outcomes in [29] are comparable. We compare FedU with others using their finest fine-tuned parameters and evaluate it on a broad range of values of η 5.10−3, 10−3, 5.10−2, 10−2, 10−1, among others. Every client in FMTL stands for a distinct task. To ensure fair comparisons with local, global, and MOCHA models, all clients have the same weight connection with others and there is no client sampling. In Section VI-C, we also give instructions on how to select the various akl values. As indicated in Section III- A [16], we only provide the convex setting for MOCHA based on its assumption. e first demonstrate FedU's advantages in FMTL by contrasting it with MOCHA, the traditional local model, which trains one distinct model for each client, and the global model, which trains one unique model on centralized data.

FedU performs best, followed by MOCHA, local model, and global model, according to the results in Fig. 3. The global model performs a single job that is not adequately generalized on highly non-IID data, whereas the local model at each client learns only its own data without any input from the models of other clients. We also acknowledge that overfitting occurs in the local model when the client data size is small. On the other hand, FedU and MOCHA are able to to capture customer relationships and learn models for several related activities at once. In the case of FedU in particular, Laplacian regularization makes it possible to use more information about the models' structures to improve learning performance, and the contribution of clients with larger data sizes to those with smaller ones becomes more substantial.After observing various values of η, we discovered that the more η there is, the more coordination there is from other customers; hence, FedU operates better when η is raised.Nevertheless, the convergence of FedU is slowed down when η approaches a particular threshold, such as η 5.10−2 in Fig. 3. Depending on the dataset, η should then be carefully selected.

C) THE IMPACT OF THE GRAPH DATA:

We assume that all relationships between a client and its neighbors are equivalent for the purposes of the aforementioned tests. The connection weights must be determined beforehand, though, as they may have varying values in practice. Next, we assess the effect of graph data displayed in Figure 4 by simulating four FedU performs better with random akl than with equal akl in the majority of scenarios. FedU performs better when the akl values are weighted than when they are all equal. FedU performs best when compared to other scenarios, particularly for MNIST, when the akl values are weighted according to client similarity. Therefore, we can set higher values of weight connection for clients in the same geographic location than clients in different locations to take advantage of FedU, given the relationship between the clients' data distribution. For instance, in a weather forecasting application, clients in the same geographic location may have similar or close weather data.

TABLE I
PERFORMANCE COMPARISON OF CENTRALIZED SETTING ($R = 5$, $S = 0.1N$, $B = 20$, AND $T = 200$). THERE IS NO CONVEX MODEL FOR CIFAR-10; WE THEN ONLY REPORT THE NONCONVEX CASE

Dataset	Algorithm	Test Accurancy	
		Convex	Non Convex
CIFAR-10	FedU		**75.41** ± 0.29
	pFedMe		74.10 ± 0.89
	Per-FedAvg		64.70 ± 1.91
	FedAvg		34.48 ± 5.34
	FedProx		42.31 ± 4.21
	SCAFFOLD		45.12 ± 3.38
	AFL		49.07 ± 3.35
MNIST	FedU	**96.95** ± 0.11	97.81 ± 0.01
	MOCHA	96.18 ± 0.09	
	pFedMe	93.73 ± 0.40	**98.64** ± 0.17
	Per-FedAvg	90.33 ± 0.84	96.38 ± 0.40
	FedAvg	87.75 ± 1.31	91.48 ± 1.05
	FedProx	88.70 ± 1.18	91.60 ± 0.23
	SCAFFOLD	89.45 ± 0.37	92.15 ± 0.43
	AFL	89.79 ± 1.23	92.01 ± 1.21
Vehicle Sensor	FedU	**88.47** ± 0.21	**91.79** ± 0.31
	MOCHA	87.31 ± 0.23	
	pFedMe	81.38 ± 0.41	90.62 ± 0.41
	Per-FedAvg	81.07 ± 0.71	86.92 ± 1.3
	FedAvg	79.84 ± 0.91	84.04 ± 2.69
	FedProx	82.06 ± 0.91	87.65 ± 2.34
	SCAFFOLD	81.97 ± 0.91	88.48 ± 0.34
	AFL	82.25 ± 0.91	87.88 ± 1.08
Human Activity	FedU	**95.76** ± 0.46	**95.86** ± 0.36
	MOCHA	92.33 ± 0.67	
	pFedMe	95.41 ± 0.38	95.72 ± 0.32
	Per-FedAvg	94.78 ± 0.37	94.80 ± 0.60
	FedAvg	93.41 ± 0.95	93.74 ± 1.01
	FedPro	93.69 ± 0.84	94.65 ± 0.72

D) COMPARISON WITH FL PERSONALIZED ALGORITHM:

Lastly, we contrast FedU with the state-of-the-art personalized FL algorithms pFedMe and Per-FedAvg, as well as the traditional FL algorithms FedAvg, FedProx, SCAFFOLD, AFL, and MOCHA. Table I presents the findings. We compare all four genuine datasets after fixing the subset of clients S 0.1N. Overall, FedU nearly consistently performs at the top in every situatio

VI.CONCLUSION

Laplacian regularization has been used in this study to design an FMTL issue that captures the relationships between the client models. It has been demonstrated that the problem formulation may be applied to both classic and personalized FL. In order to solve the formulated problem with guaranteed convergence to the best solution, we have additionally suggested both decentralized and communication-centralized algorithms. Our algorithms FedU and dFedU reach the state-of-the-art convergence rates, according to theoretical results. The suggested algorithms outperform the vanilla FedAvg in FL settings, the standard MOCHA in FMTL settings, and pFedMe and Per-FedAvg in customized FL settings, according to experimental results using real datasets in both convex and nonconvex objectives.

REFERENCES

1. T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," Int. J. Med. Inform., vol. 112, pp. 59–67, Jan. 2018.
2. J. C. Jiang, B. Kantarci, S. Oktug, and T. Soyata, "Federated learning in smart city sensing:

Challenges and opportunities," Sensors, vol. 20, no. 21, p. 6230, Oct. 2020.
3. L. Ahmed, K. Ahmad, N. Said, B. Qolomany, J. Qadir, and
A. Al-Fuqaha, "Active learning based federated learning for waste and natural disaster image classification," IEEE Access, vol. 8,
4. pp. 208518–208531, 2020.
5. S. P. Karimireddy et al., "SCAFFOLD: Stochastic controlled averaging for federated learning," in Proc. Int. Conf. Mach. Learn., vol. 119, 2020,
6. pp. 1–12.
7. F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," 2019, arXiv:1910.14425.
8. D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, arXiv:1910.03581.
9. Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, arXiv:2003.13461.
10. C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with Moreau envelopes," in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 1–12.
11. A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 1–12.
12. V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi- task learning," in Proc. Int. Conf. Neural Inf. Process. Syst., 2017,
13. pp. 1–11.
14. A. Kumar and H. Daumé, "Learning task grouping and overlap in multi- task learning," in Proc. Int. Conf. Mach. Learn., 2012, pp. 1–15.
15. Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in Proc. 26th Conf. Uncertainty Artif. Intell., 2010, pp. 733–742.
16. T. Li et al., "Federated optimization in heterogeneous networks," in Proc. Mach. Learn. Syst., 2020, pp. 429–450.
17. Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, arXiv:1806.00582.
18. X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in Proc. Int. Conf. Learn. Represent., Apr. 2020, pp. 1–26.
19. A. Khaled, K. Mishchenko, and P. Richtarik, "Tighter theory for local SGD on identical and heterogeneous data," in Proc. Int. Conf. Artif. Intell. Statist., vol. 108, Aug. 2020, pp. 26–28.
20. F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," 2020, arXiv:2002.05516.
21. P. P. Liang et al., "Think locally, act globally: Federated learning with local and global representations," 2020, arXiv:2001.01523.
22. C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in Proc. Int. Conf. Mach. Learn., 2017, pp. 1–12.
23. Y. Jiang, J. Konecˇný, K. Rush, and S. Kannan, "Improving feder- ated learning personalization via model agnostic meta learning," 2019, arXiv:1909.12488.
24. A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, arXiv:1803.02999.

25. M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Fed- erated learning with personalization layers," 2019, arXiv:1912.00818.
26. Y. SarcheshmehPour, Y. Tian, L. Zhang, and A. Jung, "Networked federated learning," 2021, arXiv:2105.12769.
27. T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in Proc. 38th Int. Conf. Mach. Learn., Jul. 2021, pp. 1–12.
28. J. Shen, X. Zhen, M. Worring, and L. Shao, "Variational multi-task learning with Gumbel-Softmax priors," in Proc. Adv. Neural Inf. Process. Syst., 2021, pp. 1–12.
29. R. Li, F. Ma, W. Jiang, and J. Gao, "Online federated multitask learning," in Proc. IEEE Int. Conf. Big Data (Big Data), Dec. 2019, pp. 215–220.
30. A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," Mach. Learn., vol. 73, no. 3, pp. 243–272, 2008.
31. R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," J. Mach. Learn. Res., vol. 6, pp. 1817–1853, Nov. 2005.
32. R. Caruana, "Multitask learning," Mach. Learn., vol. 28, no. 1, pp. 41–75, Jul. 19