Optimized and Adaptive Federated Learning for Straggler-Resilient Device Selection

Sourasekhar Banerjee
Dept. of Computing Science
Umeå University
Umeå, SE-907 81, Sweden
sourasb@cs.umu.se

Xuan-Son Vu
Dept. of Computing Science
Umeå University
Umeå, SE-907 81, Sweden
sonvx@cs.umu.se

Monowar Bhuyan
Dept. of Computing Science
Umeå University
Umeå, SE-907 81, Sweden
monowar@cs.umu.se

***Abstract*—Federated Learning (FL) has evolved as a promising distributed learning paradigm in which data samples are disseminated over massively connected devices in an IID (Identical and Independent Distribution) or non-IID manner. FL follows a collaborative training approach where each device uses local training data to train local models, and the server generates a global model by combining the local model's parameters. However, FL is vulnerable to system heterogeneity when local devices have varying computational, storage, and communication capabilities over time. The presence of stragglers or low-performing devices in the learning process severely impacts the scalability of FL algorithms and significantly delays convergence. To mitigate this problem, we propose Fed-MOODS, a Multi-Objective Optimization-based Device Selection approach to reduce the effect of stragglers in the FL process. The primary criteria for optimization are to maximize: (i) the availability of the processing capacity of each device, (ii) the availability of the memory in devices, and (iii) the bandwidth capacity of the participating devices. The multi-objective optimization prioritizes devices from fast to slow. The approach involves faster devices in early global rounds and gradually incorporating slower devices from the Pareto fronts to improve the model's accuracy. The overall training time of Fed-MOODS is $1.8\times$ and $1.48\times$ faster than the baseline model (FedAvg) with random device selection for MNIST and FMNIST non-IID data, respectively. Fed-MOODS is extensively evaluated under multiple experimental settings, and the results show that Fed-MOODS has significantly improved model's convergence and performance. Fed-MOODS maintains fairness in the prioritized participation of devices and the model for both IID and non-IID settings.**

***Index Terms*—Federated learning, Adaptive device selection, Statistical heterogeneity, Multi-objective optimization, Straggler-resilient device**

I. Introduction

Federated learning (FL) is a paradigm in distributed machine learning where multiple devices collaboratively train a model without sharing raw data [1]. Apart from privacy, it reduces the communication burden by sending only the model parameters instead of sending terabytes of data to the server. Implementation of federated learning is very challenging, as it suffers from system (device) heterogeneity and statistical (data) heterogeneity. System heterogeneity refers to devices with varying computation capacity, memory capacity, bandwidth, etc., [2]–[4], and statistical heterogeneity means Identical and Independent Distribution (IID) and non-Identical and Independent Distribution (non-IID) of data [4], [5]. Due to stragglers in the federated learning system, keeping statistical accuracy high and dealing with system heterogeneity simultaneously is very challenging. Straggler devices are low performing devices that are incompetent in processing, communicating and storage. Involving stragglers causes significant delays from learning to inference [6].

Federated learning is of two types based on how devices take part in learning: cross-device and cross-silo [6]. In cross-silo FL, every device takes part in every round of the learning process. Compared to that, in cross-device FL, millions of devices are attached to the edge. Since devices are dynamic, all devices cannot be available for the entire process. Therefore, only a few devices participate in the learning process in each round. The server selects a subset of devices randomly for every round of training. However, the random selection of devices works better for straggler-free FL settings. In the presence of huge stragglers, mainly on non-IID data, the random selection based learning approach converges very slowly, and a high impact of randomness is present in the model training [7], [8]. The server's interest is most preferred in client selection i.e., the devices that respond quickly to the server only take part in the learning. As a result, stragglers can never contribute to the FL. Moreover, removing straggler devices and only training models based on the non-straggler devices may not generalize the final model properly and cause huge information loss, which may lead to unfairness in the learning process and jeopardize the sustainability of the FL system. Therefore, it is essential to choose devices such that the model converges quickly, produces sufficient accuracy, and maintains fairness.

To mitigate these problems, Reisizadeh et al. [9] proposed an approach called FLANP that leverages the interplay between model accuracy and device heterogeneity. The algorithm includes faster devices based on computation capability in the early learning rounds and later involves stragglers. However, they only ranked devices based on their computational ability. We considered computation, communications, and storage characteristics of devices altogether and introduced Fed-MOODS, a multi-objective optimization-based adaptive device selection approach. We inferred multi-objective optimization to rank devices based on system performance, i.e., available processing, memory, and bandwidth capacity. Devices are selected adaptively from the Pareto fronts to contribute in every global rounds. Multiple devices with varying computing and storage capabilities constitute a typical federated learning environment. Due to slow devices or stragglers, applying standard federated learning algorithms such as FedAvg [10] on highly heterogeneous devices result in significant and unanticipated delays. Our focus in this work is to mitigate these problems that aggravates from system heterogeneity in the FL framework while keeping the performance of the model stable. We employ interaction between statistical accuracy and system heterogeneity to design a straggler-resilient federated learning approach that selects a subset of available devices adaptively in each global round of training. Our main contributions are as follows.

- We introduce Fed-MOODS, a straggler-resilient multi-objective optimization-based adaptive prioritized device selection approach to mitigate the system heterogeneity problems in federated learning.
- Fed-MOODS considers computation, communications, and storage heterogeneity and formulates them as multi-objective functions to optimize and generate the rank of the local devices.
- Fed-MOODS minimizes the overall wall-clock training time of the model, improves the model's performance, maintains fairness in device selection, and generalizes the final model.
- We experimented Fed-MOODS across multiple benchmark datasets (MNIST, FMNIST, and CIFAR-10) and baseline models (FedAvg, FedProx) with random-device selection. We show that the proposed approach is superior to other baselines models for both IID and non-IID settings.

We assumed that (i) Devices would share the system level information with the server. (ii) Local devices and the server are both trustworthy. (iii) During the learning process, the device's local data remains unchanged, and (iv) all participating devices remain active for the whole learning.

Organization. The rest parts of paper is organized as follows: we provide a brief literature survey in Section II. The proposed approach is given in Section III. Experiments, results, and analyses are reported in Section IV. Finally, the conclusion and future work are discussed in Section V.

II. Related Work

Federated Learning [10] allows users to learn a predictive model collaboratively while maintaining privacy, ownership, and data localization. Each participating device produces a model update during local training, which is sent to the server and aggregated with other devices' models to produce the global update [2]. This global update is subsequently distributed to all participating devices, allowing them to improve their local models in next consecutive rounds. The participating devices are heterogeneous from the system and data perspective. Federated learning causes system heterogeneity problems for devices' having different processing, communication, and storage capacities. Asynchronous approaches have shown considerable benefits in distributed or decentralized learning [11], but these approaches are not very attractive in FL for the staleness of slow devices [12], [13]. FLANP [9], a straggler-resilient adaptive device participation algorithm to reduce the stragglers' effect in FL. The learning begins with computationally faster devices and then adaptively includes the slower devices. This process continues until the model converges. In [2], the authors analyzed the impact of statistical heterogeneity on the device selection, the convergence of the model, and fault tolerance in FL settings. In [14], the authors showed that the existing federated algorithms suffer from a speed-accuracy problem in presence of statistical and system heterogeneity. The algorithm finds global minima at a sub-linear rate. To solve that issue, they proposed FedLin, which guarantees linear convergence to the global minima. In [3], the authors carried out empirical studies on the effect of system heterogeneity in the FL system. They built a heterogeneity-aware FL framework that compiles standard federated algorithms while considering the system heterogeneity. In [15], the authors proposed a FL model with attention transfer that reduces the effect of stragglers. A few more works on adaptive FL are in [16], [17].

Multi-objective optimization-based solutions in federated learning is interesting in finding model optimality by satisfying the fairness constraints of every participating device [18], [19]. Fairness in federated learning is a challenging task to accomplish [20]. Unfairness may arise in different phases of the FL process, starting from local device selection [21], [22] to model optimization [23], [24]. The notions of fairness in FL can be categorized in different ways, such as, accuracy parity [24], good-intent fairness [23], selection fairness [21], [22], contribution fairness [25], and many more. In this paper, we only considered good-intent fairness and selection fairness.

Here, we employ the advantages of multi-objective optimization to achieve optimal performance of the learned model in presence of stragglers.

III. Proposed Approach

This section starts by describing the system model of the proposed approach. Then, we formulate the adaptive FL problem and the objectives of device selection. After that, we describe the Fed-MOODS algorithm and analyze the computation time for learning.

A. System model

Many heterogeneous devices are distributed across the edge of the network and connected to the global server in typical FL settings. A set of devices is selected adaptively from the Pareto fronts in every global round (see Figure 1). On the other end, a global server is present to orchestrate the learning process and build the global model. The server broadcasts the learned model to all the devices. The procedure continues until the model converges.

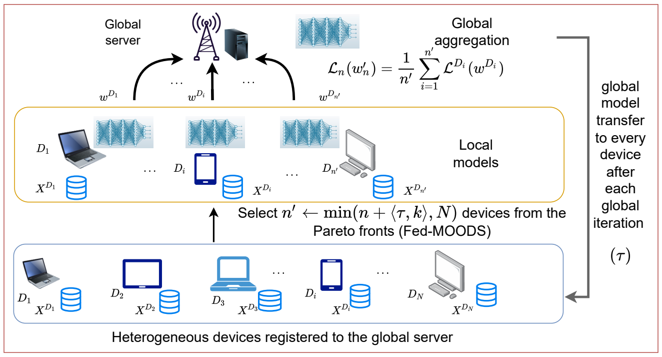

Global server
Global aggregation
$\mathcal{L}_n(w'_n) = \frac{1}{n'} \sum_{i=1}^{n'} \mathcal{L}^{D_i}(w^{D_i})$
w^{D_1}
w^{D_i}
$w^{D_{n'}}$
global model transfer to every device after each global iteration
(τ)
D_1
X^{D_1}
D_i
X^{D_i}
$D_{n'}$
$X^{D_{n'}}$
Local models
Select $n' \leftarrow \min(n + \langle\tau, k\rangle, N)$ devices from the Pareto fronts (Fed-MOODS)
D_1
X^{D_1}
D_2
X^{D_2}
D_3
X^{D_3}
D_i
X^{D_i}
D_N
X^{D_N}
Heterogeneous devices registered to the global server

Fig. 1. Overview of Fed-MOODS framework - an adaptive straggler-resilient device selection.

B. Problem formulation

The problem formulation is divided into two categories: (i) Multi-objective formulations of device heterogeneous properties, such as computation or processing, communication, and storage capacity. The primary goal is to optimize these functions or properties, generate Pareto fronts, and rank devices based on them. (ii) The formulation of the empirical loss function for the adaptive device selection based on Pareto fronts to mitigate the statistical heterogeneity problem in FL.

*1) **Multi-objective formulations**:* We formulate three objective functions based on the available device computation or processing, memory, and bandwidth capacity of each device.

a) Maximize available processing capacity: Low level performance can measure the instructions per cycle (IPC) of each device processing capacity [26]. In a multi-core environment, a processor can handle multiple instructions per clock cycle. Multiple devices have CPU (central processing unit) with different processing capacities in a heterogeneous federated environment. The CPU utilization (u) is estimated as: $u = 1 - p^a$ [26], where a is the number of processes currently running, p is the average percentage of waiting time [26]. If the device contains c multiple cores then the overall CPU utilization of each device can be defined as:

$$\begin{aligned} D_u &= \frac{1}{c}\sum_{i=1}^{c}(1 - p^a) \\ where \quad D_u &\in [0, 1] \end{aligned} \tag{1}$$

A device contains GPU (Graphics Processing Unit) along with CPU. Based on Eq. (1), GPU utilization is D_{gu} and CPU utilization is D_{cu}. The operating system (OS) of each device checks how much GPU (D_g) and CPU(D_c) are free using Eqs. (2) and (3), respectively.

$$D_g = (1 - D_{gu})(\%) \tag{2}$$

$$D_c = (1 - D_{cu})(\%) \tag{3}$$

Suppose N devices are participating in FL. The server attempts to maximize the available processing capacity (D_i^{PA}) as in Eq. (4).

$$\begin{aligned} \max_{i=1}^{N} D_i^{PA} &= \frac{1}{2}(D_g + D_c) \\ \text{s.t. } 0 \leq D_g &\leq 100, \\ 0 \leq D_C &\leq 100 \end{aligned} \tag{4}$$

b) Maximize available memory: Memory requirement (MR) for device D_i is the amount of memory required to train a neural network model[1]. While training a ConvNet, the total required memory includes storage for parameters, intermediate layers, and the gradient of each parameter. An extra memory is needed if the learning uses optimizers like momentum, RMSprop, Adams, etc. Hence, the memory requirement (D_i^{MR}) to learn a neural network is calculated as follows.

$$D_i^{MR} = B \times \sum_{l=1}^{L} MR_l \times Byte$$

where the neural network has L layers (including input and fully connected layers) and B is *Batch size*. Suppose the i^{th} device, D_i has current total memory D_i^{TM}. The available memory, D_i^{AMR} is computed as follows.

$$D_i^{AMR} = D_i^{TM} - D_i^{MR}$$

Now server collects D_i^{AMR} from the N devices and maximize in the following Eq. (5).

$$\begin{aligned} &\max_{i=1}^{N} D_i^{AMR} \\ &\text{s.t.} \frac{D_i^{TM}}{D_i^{MR}} \geq 1, \\ &0 \leq D_i^{TM} \leq 100, \\ &0 \leq D_i^{MR} \leq 100 \end{aligned} \tag{5}$$

c) Maximize available bandwidth: Network bandwidth can be estimated based on the amount of data transferred between devices and the server. Each device calculates the total amount of data (D_i^{TD}) to be replicated in gigabytes, data duplication ratio (D_i^{DR}), and length of the replication window time (D_i^{RWT}) in seconds. The server collects these information from each device and calculates the required network bandwidth[2] (D_i^{RNB})in Gbps for N devices using Eq. (6).

$$D_i^{RNB} = \frac{D_i^{TD} * (100/D_i^{DR})}{(D_i^{RWT})} \tag{6}$$

[1] https://cs231n.github.io/convolutional-networks/#case
[2] https://bit.ly/ibm-itsm-srv-doc

TABLE I
FED-MOODS: AN EXAMPLE TO ILLUSTRATE HOW THE MULTI-OBJECTIVE OPTIMIZATION PROCESS WORKS ACROSS 5 DEVICES.

Device(s)	PA	MA	AB	(PA_d)	MA_d	AB_d	$PA_d + MA_d$ $+AB_d$
D_1	97	56	25	3	0	4	7
D_2	99	76	10	4	3	3	10
D_3	82	81	3	2	4	1	7
D_4	56	60	5	0	1	2	3
D_5	70	61	2.5	1	2	0	3

Server computes the required network bandwidth for N local devices and perform the objective in the following Eq. (7).

$$\begin{aligned} &\max_{i=1}^{N} D_i^{RNB} \\ \text{s.t. } &D_i^{RWT} \geq 1, \\ &D_i^{DR} \geq 100, \\ &D_i^{TD} \geq 0 \end{aligned} \tag{7}$$

Multi-objective optimization of these three functions generate the Pareto fronts where devices are arranged in ascending order from best to worst performing devices.

Illustration. We showcase a scenario in which there are 5 heterogeneous mobile devices for smooth understanding. These devices vary in configurations, i.e., different available processing capacities, memory, and bandwidth of the communication channels. According to Table I, let 5 devices are: $D = \{D_1, D_2, D_3, D_4, D_5\}$. The server first computes the *available processing capacity (PA)*, *available memory (MA)*, and *available bandwidth (AB)* of each device. Next, the server calculates the *domination count* $\{(PA_d)$, (MA_d), and $(AB_d)\}$ of each device. *Domination count* of a device, D_i signifies how much better the device is in terms of PA, MA, and AB compared to the other participating devices. For example, in Device D_2, PA_d is 4, MA_d is 3, and AB_d is 3, i.e., Device D_2 has maximum available processing capacity. It has more available memory than 3 devices except for D_3 and the available bandwidth is also better than 3 devices except for D_1. We add all the domination counts $(PA_d + MA_d + AB_d)$ and rank each device based on them. The final list contains $\{\{D_2\}, \{D_3, D_1\}, \{D_5, D_4\}\}$. If there is a tie in domination count, we select the device with the highest available processing capacity to break the tie. To estimate the domination counts, we employ a multi-objective optimization approach, NSGA-II [27] to obtain an optimal solution. NSGA-II is an evolutionary multi-objective optimization approach that can optimize three defined objectives efficiently.

2) ***Federated learning formulation**:* Considering a federated learning system consists of N local devices, $D = \{D_1, D_2, \ldots, D_N\}$, and a server. Each device $D_i \in D$ has access to m data samples denoted by $X^{D_i} = \{x_1^{D_i}, x_2^{D_i}, \ldots, x_m^{D_i}\}$, and $X^{D_i} \in \mathbb{R}$. The empirical loss function of device D_i is defined as:

$$\mathcal{L}^{D_i}(w^{D_i}) = \frac{1}{m}\sum_{j=1}^{m} l(w_j, x_j^{D_i})$$

where $l(w, x_j^{D_i})$ is the empirical loss of the model w trained on the j^{th} data sample of the i^{th} device D_i. For any global iteration, suppose the participating devices are n' ($n' \leftarrow \min(n + \langle \tau, k \rangle, N)$, and $n' \in [1, N]$) then the empirical loss for each global round ($\mathcal{L}(w_\tau)$) is defined as:

$$\mathcal{L}(w_\tau) \equiv \mathcal{L}_{n'}(w_{n'}) = \frac{1}{n'}\sum_{i=1}^{n'} \mathcal{L}^{D_i}(w^{D_i})$$

where $\mathcal{L}_{n'}(w_{n'})$ denotes the average empirical loss over n' devices. Number of devices are increasing adaptively in each global round (τ) until the global model converges. The adaptiveness is denoted as k. Here, the objective is to minimize the global loss. The empirical loss for G global rounds is defined as:

$$\mathcal{L}(w^*) = \min_{\tau=1}^{G} \mathcal{L}(w_\tau)$$

C. Algorithm

We propose Fed-MOODS, a multi-objective optimization-based adaptive device selection approach for FL that maximizes available processing capacity, memory, and bandwidth among N devices. Based on the three objectives mentioned in Eqs. (4, 5, and 6), Fed-MOODS ranks devices according to their performance and selects devices adaptively in each global round. We describe Fed-MOODS in two parts, Algorithm 1 for adaptive device selection and learning; and Algorithm 2 for multi-objective optimization based device ranking.

a) Algorithm 1: It has two phases. *Phase I* (steps 2 to 3) is to rank devices according to the Pareto fronts. *Phase I* is described in detail in Algorithm 2. In *Phase II* (steps 4 to 18), the server adaptively selects local devices from D (step 5) for each global iteration until the model converges. The algorithm is adaptive (steps 5 to 15), i.e., in every global round of learning, devices get an opportunity to contribute to the global model. At first, the algorithm selects the first n devices from the set D and then adaptively adds k devices in each global round (τ) until the model converges. In the worst case, all devices participate in the learning process.

b) Algorithm 2: The server initially collects meta-data from devices regarding the processing capacity, memory, and bandwidth, respectively. Then calculate D_i^{PA}, D_i^{AMR}, and D_i^{RNB} for each device ($i \in N$)(steps 3 to 5). Later, we employ NSGA-II to find the domination count of the devices and rank them accordingly to their Pareto fronts (step 6). Finally, the server generates a list of devices, D', based on their ranks (steps 7 to 9) and returns to the Fed-MOODS (step 10).

D. Computational time analysis

We characterize and compare the computational run-time of Fed-MOODS with random participation of devices as a baseline. Suppose the computation time of the N available devices are $\{T_{Cl_1}, T_{Cl_2}, \ldots T_{Cl_N}\}$. In Algorithm 1, each local device performs E local iterations and G global rounds until convergence. n is the initial set of devices, k is the adaptiveness factor. System heterogeneity causes different computation time for each device. Therefore, server waits until

Algorithm 1 Fed-MOODS - Adaptive Device Selection and Training

```
Input: $D$    ▷ Collect meta-data to compute available processing capacity, memory, and bandwidth from N number of total devices
$X = \{\forall_{i=1}^{N} X^{D_i}\}$
Output: $\mathcal{L}(w^*)$    ▷ The optimal model, and loss function
    initialize: $w_\tau = w_0$    ▷ Initialize global model weight
procedure FED-MOODS($D$)
    Phase 1:
    $D \leftarrow$ call DEVICERANK($D$)    ▷ Rank all devices
    Phase 2:
    Select first $n'$ devices from $D$
    for each global iteration $\tau = 1, 2, \ldots G$ do
        Broadcast global model $w_\tau$ to $n'$ devices
        for each selected devices $D_i$ in parallel do
            for each local epoch $E$ do
                for batch $b \in X^{D_i}$ and $b \leq m$ do.    ▷ Data divided in to $m$ batches
                    $w^{D_i} \leftarrow w^{D_i} - \eta l(w_b, x_b^{D_i})$.    ▷ Local model at device $D_i$
                    $\mathcal{L}^{D_i}(w^{D_i}) = \frac{1}{m}\sum_{j=1}^{m} l(w_j, x_j^{D_i})$    ▷ Empirical local loss function at device $D_i$
        $w_\tau \leftarrow \frac{1}{n'}\sum_{i=1}^{n'} w_\tau^{D_i}$    ▷ Global model at round $\tau$
        $\mathcal{L}(w_\tau) \equiv \frac{1}{n'}\sum_{i=1}^{n'} \mathcal{L}^{D_i}(w^{D_i})$    ▷ Empirical loss at global round $\tau$
        $n' \leftarrow \min(n + \langle \tau, k \rangle, N)$    ▷ $k$ devices are added from the Pareto fronts in every global iteration.
    $w^* = \min_w\{\mathcal{L}(w) \equiv \sum_{\tau=1}^{G} w_\tau \mathcal{L}(w_\tau)\}$    ▷ Optimal global model
    $\mathcal{L}(w^*) = \min_{\tau=1}^{G} \mathcal{L}(w_\tau)$    ▷ $\mathcal{L}(w^*)$ is the minimum global empirical loss among $\tau$ global models.
    return $\mathcal{L}(w^*)$
```

Algorithm 2 DeviceRank - Algorithm for Ranking Devices

```
Input: $D = \{\forall_{i=1}^{N} D_i < D_g, D_c, D_i^{TM}, D_i^{MR}, D_i^{TD}, D_i^{DR}, D_i^{RWT} >\}$
▷ Server collects meta-data to compute available processing capacity, memory, and bandwidth from N number of total devices
Output: $\mathcal{D}'$    ▷ List of devices according to the maximum to minimum domination count.

procedure DEVICERANK($D$)
    for i =1 to N do
        Compute $D_i^{PA}(D_g, D_c)$    ▷ Compute available processing capacity of the $i^{th}$ device.
        Compute $D_i^{AMR}(D_i^{TM}, D_i^{MR})$    ▷ Compute available memory of the $i^{th}$ device.
        Compute $D_i^{RNB}(D_i^{TD}, D_i^{DR}, D_i^{RWT})$    ▷ Compute availble bandwidth of the $i^{th}$ device.
    Compute $\forall_{i=1}^{N}\ Dom(D_i(D_i^{PA}, D_i^{AMR}, D_i^{RNB}))$    ▷ Compute domination count of every devices using NSGA-II. and rank them according to the Pareto fronts
    for i = 1 to N do
        Select the device $D_i$ successively from the Pareto fronts.
        $D' = D' \cup D_i$    ▷ List of devices according o their Pareto fronts
    Return $D'$
```

the slowest device responds. The computational run-time of $T_{Fed-MOODS}$ is defined below.

Definition 1. *For constants N, n, k, G, and E, the time required to train global model is $\mathcal{O}(G{*}E{*}(T_n{+}T_{n+k}{+}\ldots{+}T_{n+\langle\tau,k\rangle}))$, where $0 \leq \tau \leq G$.*

$T_{n+\langle\tau,k\rangle}$ is the maximum unit computation time of the slowest device at the τ^{th} global round. $T_{n+\langle\tau,k\rangle}$ can be defined as, $T_{n+\langle\tau,k\rangle} = \max_{j=1}^{n+\langle\tau,k\rangle} T_{Cl_j}$. From the Figure 5, we can observe that the computational run-time is exponential in nature, what we represent as e^{λ}, where λ is a constant. The average run-time of $T_{Fed-MOODS}$ can be written as, $\bar{T}_{Fed-MOODS} = \frac{1}{G}\sum_{\tau=1}^{G}(\max_{j=1}^{n\ +\ \langle\tau,k\rangle} T_{Cl_j}) \approx \mathcal{O}(e^{\lambda})$.

For the same settings, random device participation takes G' global rounds to converge. The computational run-time T_{Random} as the baseline is defined below.

Definition 2. *For constants N, G', and E, the time required to train global model is $\mathcal{O}(G' * E * (T_1 + T_2 + \ldots + T_{G'}))$.*

$T_{G'}$ is the maximum unit computation time of the slowest device (select n' from N) in random selection for the G^{th} global round. Similarly, from the Figure 5, we can observe that the computational run-time for T_{Random} is exponential in nature. Therefore, the average run-time for learning by randomly selecting devices is $\bar{T}_{Random} = \frac{1}{G'}\sum_{\tau=1}^{G'}(\max_{j=1}^{n'} T_{Cl_j}) \approx \mathcal{O}(e^{\lambda'})$, where λ' is a constant.

In the worst case, $\lambda = \lambda'$, then $\bar{T}_{Fed-MOODS} = \bar{T}_{Random}$, otherwise, $\lambda < \lambda'$, and $\bar{T}_{Fed-MOODS} < \bar{T}_{Random}$. Experimentally, we have shown $T_{Fed-MOODS} \leq T_{Random}$ and $\bar{T}_{Fed-MOODS} \leq \bar{T}_{Random}$ in Section IV-G. To support our analysis of computational run-time for $T_{Fed-MOODS}$ when compares with T_{random} as baseline, we prove a lemma below.

Lemma 1. *The average run-time of Fed-MOODS ($\bar{T}_{Fed-MOODS}$) is always less than or equal to the baseline federated learning with random device selection ($\bar{T}_{Random}$) iff $\lambda \leq \lambda'$.*

Proof 1. *Let $e^{\lambda'} < e^{\lambda}$ as estimated run-time for T_{random} and $T_{Fed-MOODS}$, respectively. According to the Definition 1 and 2, the following conditions $\lambda \leq \lambda'$ and $\lambda : \lambda' \leq 1$ always hold. Therefore, the assumption is false. Hence, proved by contradiction.* □

IV. Experiments and Analysis

A. Simulation setup

We simulate a FL environment in our local machine to reflect the effect of stragglers. In order to model the device heterogeneity, we incorporated two different approaches to validate Fed-MOODS. At first, we employ different local rounds to different devices to incorporate system heterogeneity as we simulated the federated network in a computer. We allowed a maximum of 10 local epochs to a non-straggler device and less than 10 local epochs to stragglers. Assuming that, for a synchronous federated learning, a straggler will perform less epoch than the non-straggler device. The simulation setup is given in Table II. We compared the performance of Fed-MOODS with the randomly selected devices. Secondly, to measure the wall clock run-time of Fed-MOODS, we assumed the run-time of non-straggler devices is in the range of 10^2 ms to 10^3 ms, and for stragglers, it is in the range of 10^3 ms to 10^4 ms. All devices complete a fixed set of local rounds in each global round. For validation of run-time, we only used FedAvg as a baseline method. We used the early stopping mechanism to terminate the learning process if there is no improvement in loss function for 10 consecutive rounds.

B. Datasets & network

We used three benchmark datasets, MNIST [28] (60,000 samples for training and validation, and 10,000 testing sam-

TABLE II
SIMULATION SETUP: PARAMETERS, VALUES, AND THEIR DESCRIPTION

Parameter(s)	Value	Description
local devices	100	Devices for a local update of the model
Server	1	For performing multi-objective optimization, model aggregation
Federated algorithm	2	FedAvg [14], FedProx [7]
local device's participation	Adaptive and random	Adaptive participation of devices for Fed-MOODS, by random, frequency of participation is 10%
Dataset	IID and non-IID	IID and non-IID division of MNIST, CIFAR-10, and FMNIST dataset
Local iteration	Maximum 10	Number of local iteration at each device for each global iteration.
Global iteration	Maximum 100, 500	100 global iterations for learning on MNIST and FMNIST dataset. 500 global iterations for learning model on CIFAR-10 datasets.
Presence of stragglers	10%, 50%, 70%, 90%	Presence of stragglers in each global iteration for different experiments.
Training network	3	Three Convolutional Neural Network (CNN) having two hidden layers for training on MNIST, CIFAR-10, and FMNIST datasets, respectively.
Optimizer	1	Stochastic Gradient Descent (SGD)
Performance metrics	2	Test accuracy, F1-score

TABLE III
NEURAL NETWORK ARCHITECTURE

Neural Network	Number of Convolutional layer	In channel	Out channel	Kernel size	Number of Fully connected layer	In features	Out features	Activation function
CNNMnist	2	1	10	5	2	320	50	softmax
		10	20			50	10	
CNNFMnist	2	1	6	5	4	192	120	ReLU
						120	60	
		6	12			60	40	
						40	10	
CNNCifar10	2	3	6	5	3	400	120	softmax
		6	16			120	84	
						84	10	

ples), CIFAR-10 [29] (50,000 samples for training and validation, and 10,000 testing samples), and FMNIST [30] (60,000 samples for training and validation, and 10,000 testing samples) to validate Fed-MOODS. All datasets are distributed to devices in IID and non-IID manner.

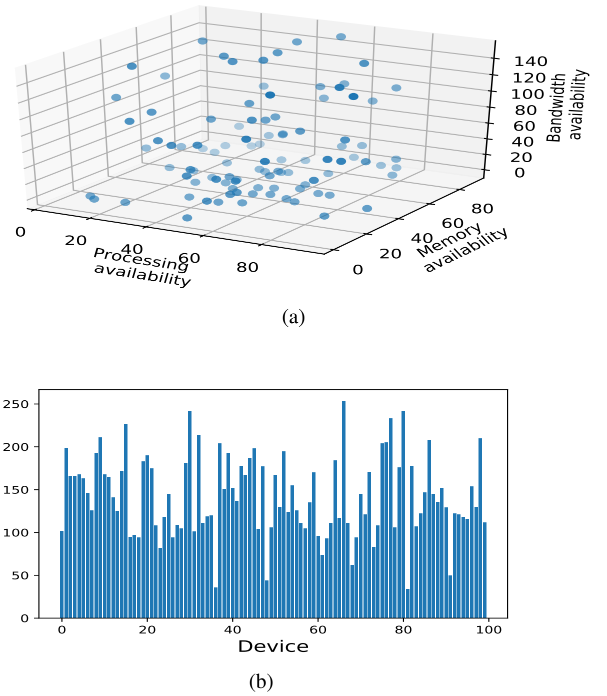
To implement Fed-MOODS, we created a federated network consisting of 100 heterogeneous devices. Each device trains a 2 layered convolutional neural network (CNN). The details of the neural network is given in the Table III.

C. Baseline algorithms

We consider FedAvg [10] and FedProx [7] as federated algorithms integrated with Fed-MOODS to compare the performance of adaptive device selection with random partial participation of devices. We evaluated the performance of Fed-MOODS with these baselines, both with respect to global rounds and wall clock time simulations.

D. Rank devices based on NSGA-II

In *Phase-I* of the Fed-MOODS, we attempt to maximize three objective functions (see Fig. 2(a)) mentioned in subsection III-B1 to characterize each device and obtain the rank of devices (see Fig. 2(b)) based on their domination counts. A device with the highest domination count is the strongest device. Similarly, a device with the lowest domination count is the weakest device concerning its system heterogeneity.



(a)

(b)

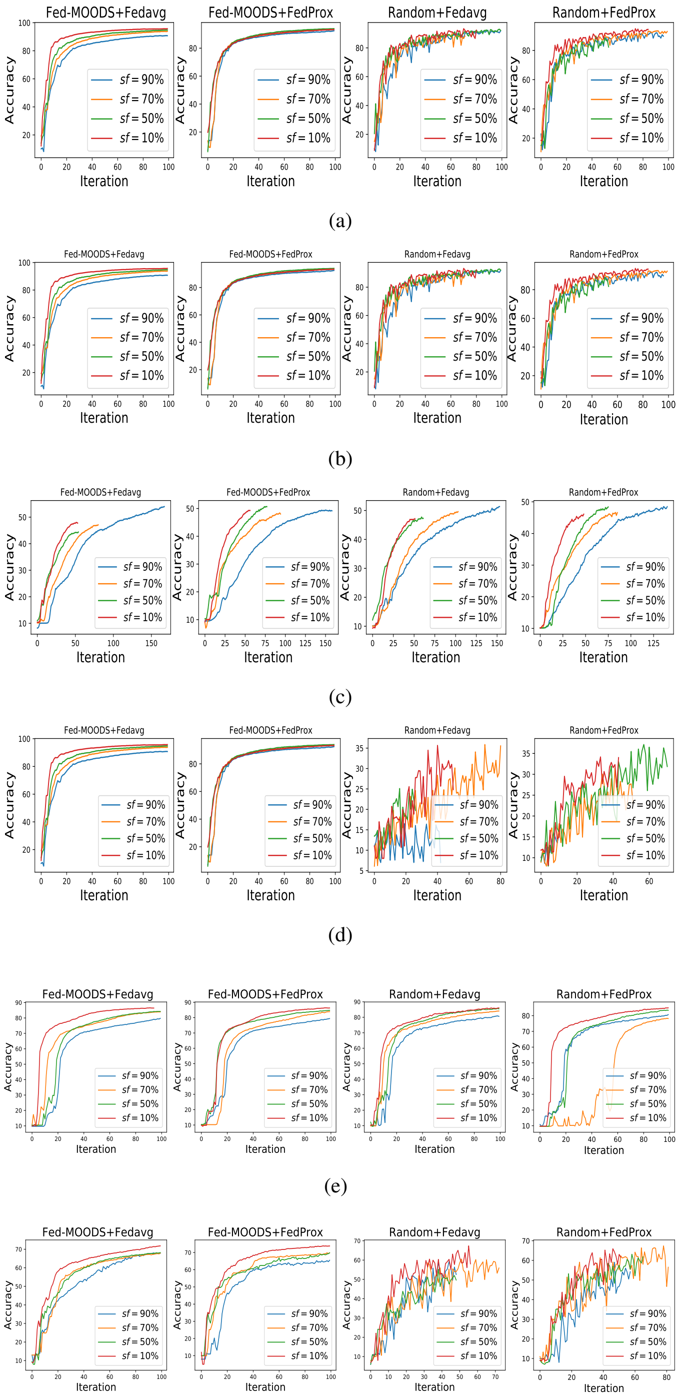
Fig. 2. (a) Device characteristics - devices based on the three objective functions: available processing capacity, memory, and bandwidth. (b) Devices with the domination counts.

E. Comparison with random device participation

1) Convergence comparison: We verified the convergence of Fed-MOODS integrated with FedAvg [10] and FedProx [7] separately with random device selection (selecting 10% of the total devices in each global round) in the presence of different fractions of stragglers (10%, 50%, 70%, and 90%) in Fig. 3 (left to right). The convergence curves are similar for the IID datasets (see Fig. 3(a), 3(c), and 3(e)). Fed-MOODS converges quickly; therefore, it maintains the model's fairness without involving all stragglers in the learning process. But for non-IID (see Fig. 3(b), 3(d), 3(f)), Fed-MOODS takes more global rounds to converge, but it is faster than random device selection. The convergence curves are more stable compared to learning with random device selection.

2) Fairness in terms of performance: We compared the performance (see Table IV) of Fed-MOODS with baseline models with partial device selection based on the F1-score at a frequency of 90% stragglers in both IID and non-IID settings.

We compared the performance of Fed-MOODS both involving stragglers and without involving stragglers for the IID data. Here, without incorporating stragglers imply that we first divide data into 100 devices and then remove stragglers. We only kept 10 non-straggler devices for learning. We observed that Fed-MOODS gives a 97.6% F1-score for the MNIST IID dataset even if we do not incorporate stragglers. Similarly, for CIFAR-10 (51.79%) and FMNIST (78.16%), the performance is almost equivalent to the models incorporating stragglers. Even though we omit 90% of the devices, Fed-MOODS still



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3. Convergence comparison of Fed-MOODS and baseline models with random device participation across (a) MNIST-IID, (b) MNIST-non-IID, (c) CIFAR-10 IID, (d) CIFAR-10 non-IID (e) FMNIST IID, (f) FMNIST non-IID, with different straggler fractions (sf).

maintains model fairness by giving an equivalent performance with baselines.

For non-IID division of data, the performance of Fed-MOODS (94.27% for MNIST, 49.33% for CIFAR-10, and 70% for FMNIST) is also better than the baseline models with randomly selected devices ((93% for MNIST, 9.37% for CIFAR-10, and 50% for FMNIST) at 90% straggler frequency. Even with high frequency of stragglers, Fed-MOODS can often achieve maximum performance. It also maintains fairness, as Fed-MOODS allows every device to contribute. Here, the performances of the models on CIFAR-10 and FMNIST are deficient because we used a simple 2-layer CNN for training and used the early stopping mechanism to terminate. However, since the main purpose of the experiment is to examine the behaviour of Fed-MOODS and the baselines regarding stragglers, the simple architecture suits the needs as well.

TABLE IV
PERFORMANCE (F1-SCORE) COMPARISON BETWEEN FED-MOODS AND BASELINE MODELS WITH RANDOM DEVICE PARTICIPATION IN PRESENCE OF 90% STRAGGLERS. ♥ AND ♢ DENOTE *involving stragglers* AND *without involving stragglers*, RESPECTIVELY.

Dataset	Fed-MOODS + FedAvg	Fed-MOODS + FedProx	Random device selection + FedAvg ♥	Random device selection + FedAvg ♢	Fed-MOODS + + FedAvg ♢
MNIST IID	94.7	93.5	94.00	96.28	**97.00**
CIFAR-10 IID	48.65	**52.92**	49.51	49.67	51.79
FMNIST IID	78.66	78.48	80.48	79.01	78.19
MNIST non-IID	93.41	**94.27**	93.00	NA	NA
CIFAR-10 non-IID	**49.33**	48.79	9.37	NA	NA
FMNIST non-IID	63.12	**65**	50.25	NA	NA

3) Fairness of probability of devices' appearance (PoA):

We assumed that N heterogeneous devices are available in the FL system, and all participate in learning. For random selection, if we select n' devices randomly from N devices in each global round, then the PoA of a device $p(D_i)$ for the G global rounds is $(1 - (\frac{N-1}{N})^{n'})^G$. Even though the straggler devices are present, random selection gives an equal PoA to each device. Fed-MOODS is biased toward non-straggler devices and does not assign equal PoA to every device. According to Algorithm 1, the PoA of a device in training rounds is $0 \leq P(D_i) \leq 1$, where $P(D_i) = \frac{\bar{G}}{G}$. Here, $\bar{G}$ is the number of appearances of a device (D_i) in total global rounds, and G is the total global rounds. Fed-MOODS adaptively incorporates devices in each training round, so that the PoA of a non-straggler device is always greater than the PoA of a straggler. For example, the first n devices appear in every global round ($\bar{G} = G$). Therefore, the PoA = 1 for the first n devices. Fed-MOODS is adding k devices adaptively in each round, so that the following k devices appear in $G-1$ global rounds. Therefore, PoA = $1 - \frac{1}{G}$. Similarly, the devices added in the $(G-1)^{th}$ global round, the PoA will be $\frac{1}{G}$. If $N = n + \langle G, k \rangle$, i.e., for G^{th} global round, all devices are participating in learning. If converging round $G^* > G$, then for the $G^* - G$ rounds, all devices participate in training. As Fed-MOODS considers the participation of every device until convergence (G^*), every device will get a chance to contribute its information to maintain high statistical accuracy. Therefore,

TABLE V
COMPARISON OF MODELS AMONG TEST ACCURACY

Dataset		SF %	Fed-MOODS + FedAvg	Fed-MOODS + FedProx	Random + FedAvg	Random + FedProx
MNIST	IID	90	**97.2**	96.31	97.2	96.89
		70	97.54	97.49	97.61	97.5
		50	**97.94**	97.76	97.74	97.61
		10	98.11	**98.39**	98.05	98.11
	Non-IID	90	92.31	91.93	92.04	91.47
		70	**93.91**	92.79	89.18	93.43
		50	**94.69**	93.47	93.05	89.61
		10	**95.74**	93.59	93.17	93.86
CIFAR-10	IID	90	**53.43**	50.20	49.15	48.86
		70	46.3	47.15	48.62	47.17
		50	43.59	**49.42**	46.25	48.9
		10	46.71	**47.33**	45.48	44.72
	Non-IID	90	49.23	**49.55**	15.84	10
		70	**48.75**	47.68	33.99	29.75
		50	**46.56**	45.93	24.98	38.44
		10	45.86	**47.81**	33.75	34.0
FMNIST	IID	90	78.66	78.48	80.48	79.44
		70	82.63	82.81	83.04	77.63
		50	83.32	83.89	85.17	82.59
		10	**85.39**	85.22	84.44	84.68
	Non-IID	90	63.22	**65.33**	50.26	58.18
		70	**67.16**	65.54	56.92	64.07
		50	70.0	**70.97**	55.56	61.81
		10	**71.76**	67.58	58.18	59.26

fairness in device selection is maintained here in the presence of stragglers. If $n + \langle G^*, k\rangle < N$, i.e., partial participation of devices can produce an equivalent model performance to that of total involvement of devices. Therefore, Fed-MOODS is straggler-resilient as well as maintains fairness.

F. Test accuracy

In Table V, we compared the test accuracy of the models for a different fraction of stragglers. Fed-MOODS produces similar results with a random selection of devices for IID datasets. Accordingly, in non-IID settings, Fed-MOODS outperforms the random selection approach by a maximum of 1.88% for MNIST, 34% for CIFAR-10, and 15% for FMNIST datasets, respectively.

G. Wall-clock time comparison

We compared the wall clock learning time of a neural network model using Fed-MOODS and baseline models on the MNIST and FMNIST datasets. We compared two cases (See Fig. 4 and 5) where the straggler frequency is very high (90%), and the straggler frequency is low (10%). From Fig. 4 we see that Fed-MOODS gradually involve straggler devices in each global round. Whereas in random partial device participation, the effect of randomness is clearly visible. From table VI, to complete 100 global rounds, Fed-MOODS is $1.8\times$ and $1.48\times$ faster than the baseline model (FedAvg) with random device participation on the MNIST and FMNIST non-IID dataset, respectively. In Fig. 5, we measured validation loss with wall clock time to train a federated model. We observed that Fed-MOODS takes less time to converge than any baseline model with random device participation for the MNIST and FMNIST non-IID datasets.

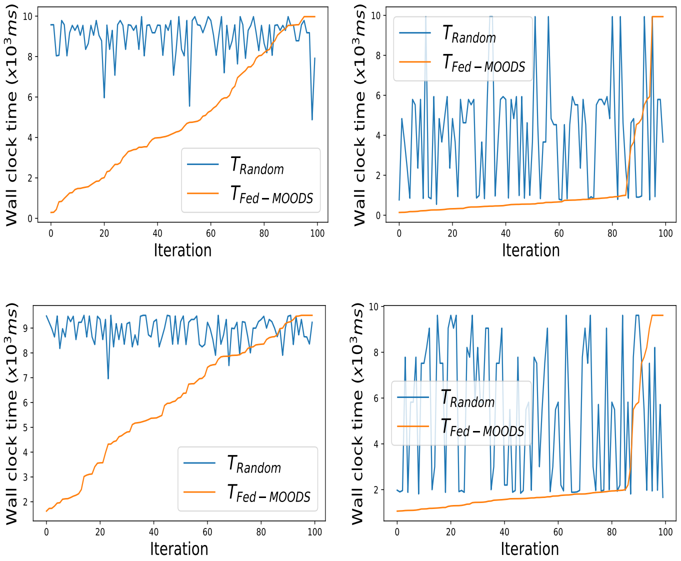


Fig. 4. Compare wall-clock time vs global iterations between Fed-MOODS and baseline model with random device participation in the presence of 90% (left top and bottom) and 10% (right top and bottom) stragglers on MNIST-nonIID (top) and FMNIST-nonIID (bottom) datasets, respectively.

TABLE VI
TOTAL AND AVERAGE WALL CLOCK TIME COMPARISON BETWEEN FED-MOODS AND BASELINE MODEL WITH RANDOM DEVICE SELECTION AT PRESENCE OF 90% STRAGGLERS ON NON-IID DATA.

Datasets	Random Device selection		Fed-MOODS	
	$T_{Random}(ms)$	$\bar{T}_{Random}(ms)$	$T_{Fed-MOODS}(ms)$	$\bar{T}_{Fed-MOODS}(ms)$
MNIST	9×10^5	9×10^3	$\mathbf{4.9 \times 10^5}$	$\mathbf{4.9 \times 10^3}$
FMNIST	8.9×10^5	8.9×10^3	$\mathbf{6 \times 10^5}$	$\mathbf{6 \times 10^3}$

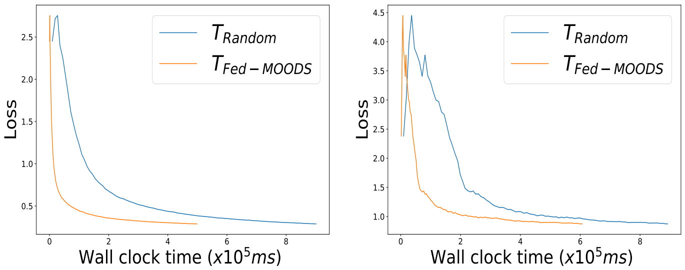


Fig. 5. Training loss vs wall-clock time comparison of Fed-MOODS and baseline model with random device participation in presence of 90% stragglers on MNIST non-IID (left) and FMNIST non-IID (right) datasets, respectively.

H. Effect of adaptiveness

In Fig. 6, we compared the convergence of Fed-MOODS by adapting devices from the Pareto front in each round in the presence of 90% stragglers for the MNIST, FMNIST, and CIFAR-10 non-IID datasets, respectively. We observed a significant increase performance in adaptiveness, making the model converge quickly, but it also incorporates more stragglers in the learning process.

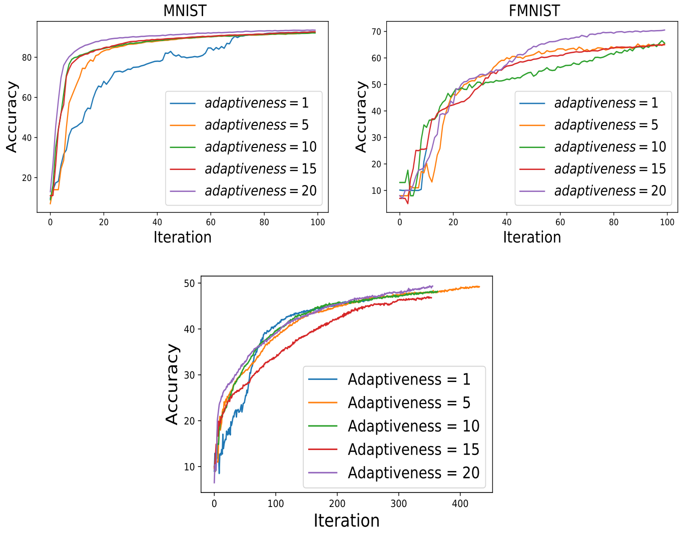


Fig. 6. Convergence comparison of different adaptiveness level in presence of 90% stragglers on MNIST non-IID dataset (top left), FMNIST non-IID (top right) and, CIFAR-10 non-IID (bottom) dataset, respectively.

V. Conclusion and Future Work

In this work, we proposed Fed-MOODS, a multi-objective optimization-based adaptive device selection approach to minimize the effect of stragglers in federated learning. We formulated every device's available processing capacity, memory, and bandwidth as a multi-objective optimization problem. We generate the rank of devices from the Pareto fronts by solving the multi-objective functions. The algorithm adaptively selects devices for training according to their ranking. We verified the Fed-MOODS on three baseline datasets (MNIST, CIFAR-10, and FMNIST), considering both IID and non-IID divisions of data among 100 devices. Fed-MOODS is straggler-resilient with the ability to maintain the model's fairness and reduce overall training time by 1.8× and 1.48× faster than the baseline model (FedAvg) with random device participation on the MNIST and FMNIST non-IID dataset, respectively. Our work suggests several exciting directions, including the theoretical convergence analysis of Fed-MOODS, and understanding the trade-off between fairness and robustness issues in device selection in scalable FL.

Acknowledgment

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation.

References

[1] S. Banerjee et. al., "Multi-diseases classification from chest-x-ray: A federated deep learning approach," *Australasian Joint Conference on Artificial Intelligence*, pp. 3–15, 2020.

[2] Y. Wang et.al., "Accelerated training via device similarity in federated learning," *"Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking"*, pp. 31–36, 2021.

[3] C. Yang et. al, "Characterizing Impacts of Heterogeneity in Federated Learning upon Large-Scale Smartphone Data," *Proceedings of the Web Conference*, pp. 935–946, 2021.

[4] V. Smith et. al, "Federated Multi-task Learning," *Advances in neural information processing systems*, vol. 30, 2017.

[5] S. Banerjee et. al., "Fed-FiS: a Novel Information-Theoretic Federated Feature Selection for Learning Stability," *International Conference on Neural Information Processing*, pp. 480–487, 2021.

[6] P. Kairouz et. al, "Advances and Open Problems in Federated Learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[7] T. Li et. al, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[8] X. Li et. al, "On the convergence of fedavg on non-iid data," *arXiv:1907.02189*, 2019.

[9] A. Reisizadeh et. al., "Straggler-Resilient Federated Learning: Leveraging the Interplay Between Statistical Accuracy and System Heterogeneity," *arXiv:2012.14453*, 2020.

[10] B. McMahan et. al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Artificial intelligence and statistics*, pp. 1273–1282, 2017.

[11] X. Lian et. al, "Asynchronous Decentralized Parallel Stochastic Gradient Descent," *International Conference on Machine Learning*, pp. 3043–3052, 2018.

[12] S. Stich, "Local SGD Converges Fast and Communicates Little," *arXiv:1805.09767*, 2018.

[13] C. Xie et. al, "Asynchronous Federated Optimization," *arXiv:1903.03934*, 2019.

[14] A. Mitra et. al, "Achieving Linear Convergence in Federated Learning under Objective and Systems Heterogeneity," *arXiv preprint arXiv:2102.07053*, 2021.

[15] H. Shi et. al., "Towards Federated Learning with Attention Transfer to Mitigate System and Data Heterogeneity of Clients," *proceedings. of the 4th Int. Workshop on Edge Sys., Analytics and Networking*, pp. 61–66, 2021.

[16] G. Canonaco et. al., "Adaptive federated learning in presence of concept drift," *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2021.

[17] L. Li et. al., "Fedsae: A novel self-adaptive federated learning framework in heterogeneous systems," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2021.

[18] S. Cui et. al., "Addressing Algorithmic Disparity and Performance Inconsistency in Federated Learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[19] Z. Hu et. al, "Fedmgda+: Federated learning meets multi-objective optimization," *arXiv:2006.11489*, 2020.

[20] Y. Shi et. al., "A survey of fairness-aware federated learning," *arXiv preprint arXiv:2111.01872*, 2021.

[21] P. Zhou et. al., "Loss tolerant federated learning," *arXiv preprint arXiv:2105.03591*, 2021.

[22] T. Huang et. al., "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1552–1564, 2020.

[23] M. Mohri et. al., "Agnostic federated learning," *International Conference on Machine Learning*, pp. 4615–4625, 2019.

[24] T. Li et. al., "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[25] L. Lyu et. al., "Collaborative fairness in federated learning," *Federated Learning, Springer*, pp. 189–204, 2020.

[26] A. Yadin, *Computer Systems Architecture*. CRC Press, 2016.

[27] K. Deb et. al, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[28] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[29] A. Krizhevsky et. al., "Learning multiple layers of features from tiny images," *Citeseer*, 2009.

[30] H. X. et. al., "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv:cs.LG/1708.07747*, 2017.