Energy-Harvesting Distributed Machine Learning

Başak Güler
University of California, Riverside
Riverside, California
bguler@ece.ucr.edu

Aylin Yener
The Ohio State University
Columbus, Ohio
yener@ece.osu.edu

***Abstract*—This paper provides a first study of utilizing energy harvesting for sustainable machine learning in distributed networks. We consider a distributed learning setup in which a machine learning model is trained over a large number of devices that can harvest energy from the ambient environment, and develop a practical learning framework with theoretical convergence guarantees. We demonstrate through numerical experiments that the proposed framework can significantly outperform energy-agnostic benchmarks. Our framework is scalable, requires only local estimation of the energy statistics, and can be applied to a wide range of distributed training settings, including machine learning in wireless networks, edge computing, and mobile internet of things.**

I. Introduction

The environmental impact of large-scale machine learning is a major challenge against the sustainability of future smart ecosystems. For instance, the carbon emission of training a single machine learning model can get as large as the lifetime of five cars [1]. The environmental impact will be even greater with the emergence of machine learning in distributed environments, where millions of devices are expected to participate in training on a regular basis. This, combined with the fact that state-of-the-art machine learning models are trained over billions of parameters [2], calls for a novel design paradigm for large-scale machine learning.

In this paper, we propose energy harvesting [3] for the design of sustainable distributed machine learning systems. We consider a distributed training scenario with N clients (users), who wish to collaborate to train a machine learning model. Each user holds a local dataset $\mathcal{D}_i$, and the goal is to train a machine learning model over the joint dataset $\mathcal{D}_1, \ldots, \mathcal{D}_N$. Training is performed through distributed stochastic gradient descent (SGD) coordinated through a central server, who maintains a global model. At each iteration of training, the server sends the current estimate of the model parameters to the users. Users then locally update the global model by computing a local gradient on their local dataset, and send their local updates to the server. The server then aggregates the local updates from the users, updates the global model, and sends the updated model back to the users. Unlike the conventional distributed SGD setting, in this work, users receive energy through an energy harvesting process [3]–[15], and can only participate in training if they have energy available to do so.

Energy and resource efficiency in machine learning has been studied in various notable works [16], [17]. Broadly, these settings can be categorized into two. The first line of work focuses on minimizing the energy consumption of the compute or communication framework [16]. The second line of work, on the other hand, is focused on minimizing the training loss within a given energy budget, where all of the energy is available at the beginning of training [17]. In contrast, our work focuses on training with devices that can harvest small amounts of energy from the ambient environment, where energy arrivals are intermittent and non-homogeneous across different devices.

Prior to this work, user sampling for distributed machine learning has been primarily investigated in the context of improving communication efficiency or convergence rate [18]–[26]. In these works, the primary goal is to either select a small set of users to participate at a given training iteration in order to reduce the overall communication overhead or due to bandwidth limitations, or to select a few informative users to maximize the convergence rate of training, with the assumption that all users are available to participate in training if selected. In contrast, in our setting, users can only participate in training if they have available energy. Moreover, the energy availability of different users can be different. Several notable works have considered distributed learning when users have a chance to drop out, unlike the current setup, in these settings, user dropouts occur uniformly at random [27]–[29].

We demonstrate that energy-harvesting can be a good candidate for machine learning in distributed networks, through a practical distributed training framework with theoretical convergence guarantees. Our experiments show that the proposed framework significantly outperforms the alternative distributed SGD benchmarks that are agnostic to the energy arrival process. We hope our work to open up new research directions in leveraging energy-harvesting for sustainable machine learning in large-scale mobile and edge networks.

II. System Model

A. Training Setup

We consider a distributed training setup in a network with N devices (users). The users are connected through a central server who coordinates the training. User i has a local dataset $\mathcal{D}_i$, consisting of D_i data points. We define the total number of data points in the network as $D = \sum_{i\in[N]} D_i$. The goal is to train a model $\mathbf{w}$ that minimizes a global loss function

$$F(\mathbf{w}) = \frac{1}{D}\sum_{i=1}^{N}\sum_{j=1}^{D_i} l(\mathbf{w}, \mathbf{x}_{ij}) \tag{1}$$

where $l(\mathbf{w}, \mathbf{x}_{ij})$ denotes the loss of data point $\mathbf{x}_{ij}$ from the local dataset of user i. Note that the loss function in (1) is evaluated with respect to the entire set of data points that belong to the N users. As such, equation (1) can also be written as

$$F(\mathbf{w}) = \sum_{i=1}^{N} p_i F_i(\mathbf{w}) \tag{2}$$

978-1-5386-8209-8/21/$31.00 ©2021 IEEE

where $p_i = \frac{D_i}{D}$ such that $\sum_{i=1}^{n} p_i = 1$, and

$$F_i(\mathbf{w}) = \frac{1}{D_i} \sum_{j=1}^{D_i} l(\mathbf{w}, \mathbf{x}_{ij}) \tag{3}$$

represents the local loss function of user i.

Training is performed through distributed SGD, in which the model parameters are updated iteratively in the negative direction of the gradient. Each iteration is represented by a discrete time instant $t \in \{0, 1, 2, \ldots\}$. The current estimation of the model parameters at iteration t is represented by a d-dimensional vector $\mathbf{w}^{(t)} \in \mathbb{R}^d$, where d is the model size.

We now review the conventional distributed SGD protocol. In this setting, at the beginning of each iteration, the server sends $\mathbf{w}^{(t)}$ to the users. Then, user $i \in \{1, \ldots, N\}$ computes a local stochastic gradient,

$$g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) \triangleq \nabla F_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) \tag{4}$$

by using a (uniformly) random sample $\xi_i^{(t)}$ from the local dataset $\mathcal{D}_i$. Hence, the stochastic gradient is an unbiased estimator of the true gradient of user i,

$$\mathbb{E}_{\xi_i^{(t)}}[\nabla F_i(\mathbf{w}^{(t)}, \xi_i^{(t)})] = \nabla F_i(\mathbf{w}^{(t)}), \tag{5}$$

where $\nabla F_i(\mathbf{w}^{(t)})$ is the gradient of the local loss function in (3). The gradient of the global loss function in (1) is given by,

$$\nabla F(\mathbf{w}^{(t)}) \triangleq \sum_{i=1}^{N} p_i \nabla F_i(\mathbf{w}^{(t)}). \tag{6}$$

After the local computations, users send their local gradients from (4) to the server. The server then updates the model,

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \sum_{i=1}^{N} p_i g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) \tag{7}$$

where η is the learning rate (step size), and sends the updated model back to the users for the next iteration.

B. Energy Harvesting Profile of the Users

This work considers devices that are powered by the energy harvested from the ambient environment, such as RF, solar, or kinetic energy [3], [4]. We assume that one step of the SGD protocol costs a unit amount of energy at each user, which includes computing the local gradient from (4) and sending it to the server. It is also assumed that each user has a unit battery that can store enough energy for one step SGD.

We let E_i^t denote the energy arrival process at user i, in particular $E_i^t = 1$ if user i receives energy at time t and $E_i^t = 0$ otherwise. The specific distribution of the energy arrivals depends on the harvesting process. Our focus is on the following energy harvesting scenarios.

1) Deterministic Energy Arrivals: We first consider a deterministic energy harvesting scenario in which energy arrivals are known by each user in advance. We assume that energy may arrive at arbitrary non-overlapping time instances, and let $\mathcal{I}_i = \{t : E_i^t = 1\}$ denote the set of time instances at which user i receives energy. We also define $\underline{I}_i^t = \max_{t':t' \leq t,\ t' \in \mathcal{I}_i} t'$ for the time of the most recent energy arrival up to t, and $\bar{I}_i^t = \min_{t':t' > t,\ t' \in \mathcal{I}_i} t'$ for the time of the next energy arrival after time t. Finally, for a given t, we define the duration between $\underline{I}_i^t$ and $\bar{I}_i^t$ as $T_i^t = \bar{I}_i^t - \underline{I}_i^t$.

2) Stochastic Energy Arrivals: We next consider the stochastic energy harvesting scenario where energy arrivals are modeled through a stochastic process. Unlike the deterministic setting, users do not know the exact time instant at which energy will be received, but only the probabilistic model governing the underlying harvesting process. Our focus is on the following stochastic arrival scenarios.

(Binary Arrivals) In the binary energy arrival setup, at each time instant, user i receives a unit amount energy with probability β_i. More specifically, we let $E_i^t \sim \text{Bern}(\beta_i)$:

$$E_i^t = \begin{cases} 1 & \text{with probability} \quad \beta_i \\ 0 & \text{with probability} \quad 1 - \beta_i \end{cases} \tag{8}$$

where $\beta_i \in (0, 1]$, to represent whether or not user i receives energy at time t. Parameter β_i quantifies how frequent user i receives energy, and may vary from one user to another.

(Uniform Arrivals) We next consider a uniform energy arrival scenario in which device i receives a unit amount of energy at a uniformly random time instant every T_i time instants. Formally, for any t such that $t \mod T_i = 0$, user i receives a unit amount of energy at a uniformly random time instant within $\{t, \ldots, t + T_i - 1\}$.

Note that this is not an immediate generalization of the first setting, as in the former setup there is a non-zero probability that user i will never receive energy in T_i time instants. In contrast, in the second setting, user i receives a unit amount energy with probability 1 at every T_i time instants, but the exact time instant at which energy is received is unknown.

As we demonstrate in our experiments, the conventional distributed SGD strategy from Section II might bias the model towards users that have more frequent energy arrivals, causing a performance loss in training. As such, the training strategy should take into account the energy arrival patterns of the users.

Main Problem. Given the above training and energy harvesting settings, the main problem we study in our work is, *"How to design a distributed stochastic gradient descent framework for energy harvesting devices, where energy arrivals are intermittent and heterogeneous, while ensuring theoretical convergence guarantees?"*.

In the sequel, we provide a simple energy harvesting distributed learning strategy with provable convergence guarantees. The proposed strategy takes into account the intermittent energy availability due to the energy harvesting process of the individual users while ensuring that the model does not bias towards any particular user.

III. Energy Harvesting Distributed SGD

A. Distributed SGD with Deterministic Energy Arrivals

We first study the deterministic energy harvesting scenario and provide a simple distributed training framework with theoretical convergence guarantees. The individual steps of our framework is provided in Algorithm 1. Our framework consists of three main components, user scheduling, local gradient computations, and server-side model update.

1) User scheduling: The first component of our framework is user scheduling for training. Conventional user selection algorithms for distributed SGD are designed under the assumption that all users are inherently available to participate in the

Algorithm 1 Distributed SGD with Deterministic Energy Arrivals

input Number of devices N, local dataset $\mathcal{D}_i$ of device $i \in [N]$, number of iterations T, initial model parameters $\mathbf{w}^{(0)}$.
output Model parameters (weights) $\mathbf{w}^{(T)}$.

for user $i = 1, \dots, N$ **do**
  Initialize $U_i^t = 0$ for $t \in [T]$.
for iteration $t = 0, \dots, T-1$ **do**
  Users $i = 1, \dots, N$:
  if $E_i^t = 1$ **then**
    Sample an integer J uniformly random from $\{0, \dots, T_i^t - 1\}$.
    Update $U_i^{t+J} = 1$.
  if $U_i^t = 1$ **then**
    Compute the local gradient $g_i(\mathbf{w}^{(t)}, \xi_i^{(t)})$.
    Send $T_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)})$ to the server.
  Server:
  Update the model according to (10).
  Send the model parameters $\mathbf{w}^{(t+1)}$ to the users.

training process if selected, and employ a user sampling strategy to reduce the communication load or aim at selecting the users that will maximize the convergence rate for training [18]–[21]. In contrast, in our setup, not all users can participate in the training process at all rounds. This is due to the intermittent energy arrivals, if a user has no energy at a given time instant, they will not be able to participate in training.

A naive approach would be to utilize the conventional distributed SGD algorithm from (7). However, doing so may bias the trained model towards users who have more frequent energy availability. Another approach is to wait until all users become available, and then use the conventional distributed SGD algorithm from (7). However, waiting for all users to have enough energy can significantly increase the total training time needed to achieve a target performance level.

Instead, we propose a practical scheduling strategy that can be performed locally by the users, while ensuring that the model does not bias towards any user. In this setting, whenever a user receives energy, i.e., $E_i^t = 1$ for some t, the user samples an integer J uniformly at random from the set $\{0, \dots, T_i^t - 1\}$, and participates at iteration $t + J$.

2) Local gradient computation: At the beginning of each training iteration, the server sends the current estimate of the model parameters $\mathbf{w}^{(t)}$ to the users. If a user decides to participate in the current training iteration t, according to the scheduling strategy from Section III-A1, it computes the local gradient from (4). Then, the user sends to the server a scaled version of their local gradient,

$$T_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) = T_i^t \nabla F_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) \tag{9}$$

3) Server-side model update: After receiving the local computations from (9) from the participating users, the server updates the model as:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \sum_{i \in S_t} p_i \left(T_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) \right) \tag{10}$$

where $\mathcal{S}_t$ denotes the set of users who have participated at round t. Note that due to the stochastic nature of the user scheduling process, $\mathcal{S}_t$ is random.

As we demonstrate in Section IV, this process provides theoretical convergence guarantees for the model. Moreover, the user scheduling process does not require a central coordinator and can be performed locally by the users, solely based on local energy estimations, hence is scalable to large networks.

Algorithm 2 Distributed SGD with Stochastic Energy Arrivals

input Number of devices N, local dataset $\mathcal{D}_i$ of device $i \in [N]$, number of iterations T, initial model parameters $\mathbf{w}^{(0)}$.
output Model parameters (weights) $\mathbf{w}^{(T)}$.

for iteration $t = 0, \dots, T-1$ **do**
  Users $i = 1, \dots, N$:
  if $E_i^t = 1$ **then**
    Compute the local gradient $g_i(\mathbf{w}^{(t)}, \xi_i^{(t)})$.
    Send $\gamma_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)})$ to the server.
  Server:
  Update the model according to (11).
  Send the model parameters $\mathbf{w}^{(t+1)}$ to the users.

B. Distributed SGD with Stochastic Energy Arrivals

We next consider distributed training under the stochastic energy harvesting setting. The training strategy again consists of three main components, user scheduling, local gradient computation, and server-side model update. We employ a *best-effort* user scheduling strategy, where each user participates in training as soon as they receive energy, by computing the local gradient from (4), and sending to the server a scaled gradient $\gamma_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)})$, where $\gamma_i^t = \frac{1}{\beta_i}$ and $\gamma_i^t = T_i$ for the binary and uniform energy arrival settings, respectively.

After receiving the local computations from the participating users, the server updates the model as,

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \sum_{i \in S_t} p_i \left(\gamma_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) \right) \tag{11}$$

The individual steps of this process are provided in Algorithm 2.

IV. Convergence Analysis

We now state the convergence guarantees of our framework, by first reviewing a few common technical assumptions [19], [30] that will be needed for our convergence analysis.

Assumption 1. *(Bounded variance) The variance of the stochastic gradients from* (4) *are bounded:*

$$E_{\xi_i^{(t)}}[||g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) - \nabla F_i(\mathbf{w}^{(t)})||^2] \leq \sigma^2 \quad \text{for } i \in [N] \tag{12}$$

Assumption 2. *(Second moment bound) The expected squared norm of the stochastic gradients from* (4) *are bounded:*

$$E_{\xi_i^{(t)}}[||g_i(\mathbf{w}^{(t)}, \xi_i^{(t)})||^2] \leq G^2 \quad \text{for } i \in [N] \tag{13}$$

We also assume that the local loss functions $F_i(\mathbf{w})$ for $i \in [N]$ (and thus the global loss function $F(\mathbf{w})$) are μ-strongly convex and L-smooth, as in [19, Assumptions 1 and 2]. Next, we provide a key technical lemma.

Lemma 1. *(Unbiasedness) For distributed SGD with deterministic energy arrivals,*

$$\mathbb{E}_{S_t}\left[\sum_{i \in S_t} p_i T_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) \right] = \sum_{i=1}^{N} p_i g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}), \tag{14}$$

hence the user scheduling scheme is unbiased. Moreover, for distributed SGD with stochastic energy arrivals, the unbiasedness condition from (14) *holds by replacing* T_i^t *with* $\frac{1}{\beta_i}$ *and* T_i *for binary and uniform arrivals, respectively.*

Proof. We first define a Bernoulli random variable α_i^t to represent whether or not user i participates at iteration t:

$$\alpha_i^t = \begin{cases} 1 & \text{if user } i \text{ participates at time } t \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Then, for any given t,

$$P[\alpha_i^t = 1] = P[J = t - \underline{I}_i^t] = \frac{1}{T_i^t} \tag{16}$$

By letting $\alpha_t \triangleq (\alpha_1^t, \ldots, \alpha_N^t)$, we find that,

$$\mathbb{E}_{S_t}\left[\sum_{i\in S_t} p_i T_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)})\right] = \mathbb{E}_{\alpha_t}\left[\sum_{i=1}^{N} \alpha_i^t p_i T_i^t g_i(\mathbf{w}^{(t)}, \xi_i^{(t)})\right] \tag{17}$$

$$= \sum_{i=1}^{N} p_i T_i^t \frac{1}{T_i^t} g_i(\mathbf{w}^{(t)}, \xi_i^{(t)}) \tag{18}$$

where (17) follows from $S_t = \sum_{i=1}^N \alpha_i^t$, and (18) is from (16).

The proof for stochastic arrivals follows the same lines along with the observation that, for the best-effort user scheduling strategy $P[\alpha_i^t = 1] = P[E_i^t = 1]$. □

We now state our convergence guarantees.

Theorem 1. *For training a machine learning model from* (1)*, using the distributed SGD algorithm with deterministic energy arrivals and a constant learning rate* $\eta \leq \min\left\{\frac{1}{2\mu}, \frac{1}{L}\right\}$.

$$\mathbb{E}[F(\mathbf{w}^{(T)})] - F(\mathbf{w}^*) \leq \frac{L}{\mu}(1-\eta\mu)^T (F(\mathbf{w}^{(0)}) - F(\mathbf{w}^*) - \frac{\eta C}{2}) + \frac{\eta L C}{2\mu} \tag{19}$$

in T *iterations, where* $\mathbf{w}^*$ *denotes the optimal model parameters that minimize the global loss function in* (1)*, and*

$$C \triangleq \Big(\sum_{i=1}^{N}\Big(T_{i,max} - 1\Big)p_i^2 + \sum_{i=1}^{N}\sum_{j=1}^{N} p_i p_j\Big) G^2, \tag{20}$$

where $T_{i,max} \triangleq \max\{T_i^1, \ldots, T_i^T\}$ *for* $i = 1 \ldots, N$.

Remark 1. *The first term in the right hand side of* (19) *vanishes as* $T \to \infty$*, whereas the second term* $\frac{\eta L C}{2\mu}$ *represents a non-vanishing error term due to the constant learning rate. By using a decreasing learning rate as in [19], [21], this term can also be made vanishing as* $T \to \infty$.

Proof. (Sketch) The proof follows standard steps for the convergence analysis of distributed SGD algorithms [18], [19], [30], hence we provide a proof sketch in the sequel. By letting $g_i^t \triangleq g_i(\mathbf{w}^{(t)}, \xi_t)$, $\mathbf{w}^* \triangleq \arg\min_{\mathbf{w}} F(\mathbf{w})$, and $\xi_t \triangleq (\xi_1^{(t)}, \ldots, \xi_N^{(t)})$, from (10) we find that,

$$\mathbb{E}_{S_t,\xi_t}[\|\mathbf{w}^{(t+1)} - \mathbf{w}^*\|^2] = \mathbb{E}_{S_t,\xi_t}[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2] - 2\eta \mathbb{E}_{S_t,\xi_t}[\langle \mathbf{w}^{(t)} - \mathbf{w}^*, \sum_{i\in\mathcal{S}_t} p_i T_i^t g_i^t\rangle] + \eta^2 \mathbb{E}_{S_t,\xi_t}[\|\sum_{i\in S_t} p_i T_i^t g_i^t\|^2] \tag{21}$$

From Lemma 1, (5), and μ-strong convexity, we observe that,

$$\mathbb{E}_{S_t,\xi_t}[\langle \mathbf{w}^{(t)} - \mathbf{w}^*, \sum_{i\in\mathcal{S}_t} p_i T_i^t g_i^t\rangle]$$

$$= \mathbb{E}_{S_t,\xi_t}[\langle \mathbf{w}^{(t)} - \mathbf{w}^*, \sum_{i\in\mathcal{S}_t} p_i T_i^t g_i^t - \sum_{i=1}^{N} p_i \nabla F_i(\mathbf{w}^{(t)})\rangle] + \mathbb{E}_{S_t,\xi_t}[\langle \mathbf{w}^{(t)} - \mathbf{w}^*, \sum_{i=1}^{N} p_i \nabla F_i(\mathbf{w}^{(t)})\rangle] \tag{22}$$

$$= \langle \mathbf{w}^{(t)} - \mathbf{w}^*, \nabla F(\mathbf{w}^{(t)})\rangle \tag{23}$$

$$\geq F(\mathbf{w}^{(t)}) - F(\mathbf{w}^*) + \frac{\mu}{2}\|\mathbf{w}^* - \mathbf{w}^{(t)}\|^2 \tag{24}$$

We also have from Lemma 1 that,

$$\mathbb{E}_{S_t,\xi_t}[\|\sum_{i\in S_t} p_i T_i^t g_i^t\|^2] = \mathbb{E}_{S_t,\xi_t}[\|\sum_{i\in S_t} p_i T_i^t g_i^t - \sum_{i=1}^{N} p_i g_i^t\|^2] + \mathbb{E}_{S_t,\xi_t}[\|\sum_{i=1}^{N} p_i g_i^t\|^2] \tag{25}$$

By combining (21), (24), and (25), we find that,

$$\mathbb{E}_{S_t,\xi_t}[\|\mathbf{w}^{(t+1)} - \mathbf{w}^*\|^2] \leq (1-\eta\mu)\mathbb{E}_{S_t,\xi_t}[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2] - 2\eta(F(\mathbf{w}^{(t)}) - F(\mathbf{w}^*)) + \eta^2 \mathbb{E}_{S_t,\xi_t}[\|\sum_{i\in S_t} p_i T_i^t g_i^t - \sum_{i=1}^{N} p_i g_i^t\|^2] + \eta^2 \mathbb{E}_{S_t,\xi_t}[\|\sum_{i=1}^{N} p_i g_i^t\|^2] \tag{26}$$

By defining α_i^t as in (15) and $\alpha_t = (\alpha_1^t, \ldots, \alpha_N^t)$,

$$\mathbb{E}_{S_t,\xi_t}[\|\sum_{i\in S_t} p_i T_i^t g_i^t - \sum_{i=1}^{N} p_i g_i^t\|^2]$$

$$= \mathbb{E}_{\alpha_t,\xi_t}[\|\sum_{i=1}^{N} p_i(\alpha_i^t T_i^t g_i^t - g_i^t)\|^2] \tag{27}$$

$$= \sum_{i=1}^{N} p_i^2 \mathbb{E}_{\alpha_t,\xi_t}[\|\alpha_i^t T_i^t g_i^t - g_i^t\|^2] + \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} \mathbb{E}_{\alpha_t,\xi_t}[\langle p_i(\alpha_i^t T_i^t g_i^t - g_i^t), p_j(\alpha_j^t T_j^t g_j^t - g_j^t)\rangle] \tag{28}$$

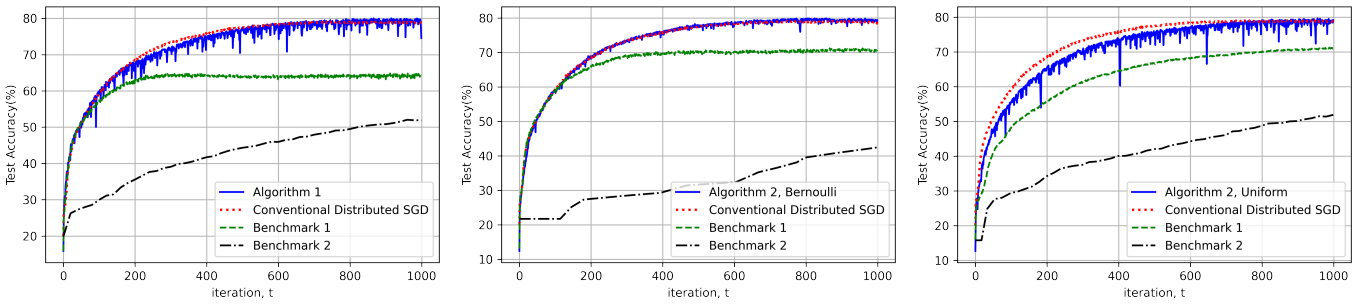
$$= \sum_{i=1}^{N} p_i^2 \mathbb{E}_{\alpha_t,\xi_t}[\|\alpha_i^t T_i^t g_i^t - g_i^t\|^2] \tag{29}$$

$$= \sum_{i=1}^{N} p_i^2 (T_i^t)^2 \mathbb{E}_{\xi_t}[\mathbb{E}_{\alpha_t|\xi_t}[(\alpha_i^t - \frac{1}{T_i^t})^2 \|g_i^t\|^2 | \xi_t]] \tag{30}$$

$$\leq \sum_{i=1}^{N} p_i^2 (T_{i,max} - 1) G^2 \tag{31}$$

where (29) holds from (16) and that (α_i^t, g_i^t) is independent from (α_j^t, g_j^t) for all $i \neq j$; (31) is from (16) and (13). Finally,

$$\eta^2 \mathbb{E}_{S_t,\xi_t}[\|\sum_{i=1}^{N} p_i g_i^t\|^2] \leq \sum_{i=1}^{N} p_i^2 \mathbb{E}_{\xi_t}[\|g_i^t\|^2] + \sum_{i=1}^{N}\sum_{\substack{j=1 \\ j\neq i}}^{N} p_i p_j \mathbb{E}_{\xi_t}[\|g_i^t\| \|g_j^t\|] \tag{32}$$



(a) (b) (c)

Fig. 1. Test accuracy of the proposed framework compared to the benchmark distributed SGD algorithms for $N = 40$ users on the CIFAR-10 dataset, for (a) deterministic, (b) Bernoulli, and (c) uniform energy arrivals.

$$\leq \sum_{i=1}^{N} p_i^2 \mathbb{E}_{\xi_t}[\|g_i^t\|^2] + \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{p_i p_j}{2} \mathbb{E}_{\xi_t}[\|g_i^t\|^2 + \|g_j^t\|^2] \quad (33)$$

$$\leq \sum_{i=1}^{N} \sum_{j=1}^{N} p_i p_j G^2 \quad (34)$$

where (32) is from the Cauchy-Schwarz inequality; (33) is from the AM-GM inequality; (34) is from (13). By combining (26) and (31) with (34) and noting that $-2\eta(F(\mathbf{w}^{(t)}) - F(\mathbf{w}^*)) \leq 0$,

$$\mathbb{E}_{S_t,\xi_t}[\|\mathbf{w}^{(t+1)} - \mathbf{w}^*\|^2] \leq (1-\eta\mu)\mathbb{E}_{S_t,\xi_t}[\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2] + \eta^2 \Big(\sum_{i=1}^{N} \Big((T_{i,max} - 1) \Big) p_i^2 + \sum_{i=1}^{N} \sum_{j=1}^{N} p_i p_j \Big) G^2 \quad (35)$$

The remainder of the proof follows from standard induction arguments as in [19], [21], hence is omitted. □

Corollary 1. *For distributed SGD with stochastic energy arrivals, Theorem 1 holds by replacing $T_{i,max}$ with $\frac{1}{\beta_i}$ for binary arrivals and with T_i for uniform arrivals, respectively. The convergence analysis follows the same steps.*

V. Experiments

In our experiments, we consider a conventional image classification task with 10 classes on the CIFAR-10 dataset [31], distributed over 40 users uniformly at random. Training is performed via distributed SGD using the convolutional neural network architecture from [27] (about 10^6 model parameters). To demonstrate the impact of non-homogeneous energy-arrivals, users are partitioned into 4 equal-sized groups $\mathcal{A}_0, \ldots, \mathcal{A}_3$ such that $\mathcal{A}_k = \{i : i \mod 4 = k\}$, and the energy profiles of users in group $\mathcal{A}_k$ are set as:

$$E_i^t = \begin{cases} 1 & \forall t \text{ such that } t \mod \tau_k = 0 \\ 0 & \text{otherwise} \end{cases} \quad (36)$$

for $i \in \mathcal{A}_k$, where $(\tau_0, \tau_1, \tau_2, \tau_3) = (1, 5, 10, 20)$. Therefore, users in group $\mathcal{A}_0$ receive energy at every time-instant t, whereas users in groups $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$ receive energy at every 5, 10, and 20 time-instants, respectively. We compare our framework with the following distributed SGD benchmarks:

Benchmark 1. We first implement the distributed SGD framework from Section II when users participate in training as soon as they have energy available, by computing the gradient from (4) and sending it to the server, and then wait for the next energy arrival. Note that in this setting users do not scale the gradients with respect to the energy arrivals.

Benchmark 2. We then consider the distributed SGD framework from Section II when the global model is updated only if all users have enough energy to participate in training. That is, the server waits until all users have energy, then sends the current model parameters to the users, users compute the stochastic gradient from (4) and send it back to the server, and then the server updates the model as in (7). Hence, in this case, the model is updated once every $t = 20$ iterations.

Finally, we also implement the conventional distributed SGD framework from Section II when all users are available at every iteration, which represents our target (desired) accuracy level. We demonstrate our results in terms of the test accuracy with respect to time t in Figure 1 (a). Our results show that Algorithm 1 achieves the same accuracy level (about 80%) as conventional distributed SGD, whereas the two benchmarks achieve an accuracy of 64% and 52%, respectively, within $t = 1000$ iterations. This is due to the fact that the first benchmark favors users with more frequent energy arrivals, hence the model is biased. The second benchmark waits for all users to have enough energy before making a single SGD update, hence, even though the training algorithm is unbiased, its convergence rate is very slow. In contrast, Algorithm 1 converges fast while achieving good accuracy. Figures 1 (b) and (c) demonstrate the test accuracy for the same group structure (4 equal-sized groups) in the stochastic arrivals scenario from Algorithm 2, where we consider Bernoulli and uniform arrivals, respectively, and set $\beta_i = (1, 1/5, 1/10, 1/20)$, and $T_i = (1, 5, 10, 20)$ for the users in the 4 groups. The proposed algorithm achieves the target accuracy level of standard SGD and outperforms the two benchmarks also in the stochastic arrivals scenario.

VI. Conclusion

We have studied distributed machine learning when users have intermittent energy availability, and demonstrated a simple distributed learning strategy with provable convergence guarantees. Future directions include exploring the impact of energy accumulation, quantization, and stragglers. We hope our study to open up further research on energy harvesting for sustainable learning in distributed and mobile networks.

REFERENCES

[1] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," pp. 265–284, Jul 2019.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[3] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, 2015.

[4] H. B. Radousky and H. Liang, "Energy harvesting: an integrated view of materials, devices and applications," *Nanotechnology*, vol. 23, no. 50, p. 502001, 2012.

[5] K. Tutuncuoglu, O. Ozel, A. Yener, and S. Ulukus, "The binary energy harvesting channel with a unit-sized battery," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4240–4256, 2017.

[6] O. Ozel, K. Tutuncuoglu, S. Ulukus, and A. Yener, "Fundamental limits of energy harvesting communications," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 126–132, 2015.

[7] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 1180–1189, 2012.

[8] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1732–1743, 2011.

[9] B. Varan and A. Yener, "Delay constrained energy harvesting networks with limited energy and data storage," *IEEE Journal on selected Areas in Communications*, vol. 34, no. 5, pp. 1550–1564, 2016.

[10] K. Tutuncuoglu, B. Varan, and A. Yener, "Throughput maximization for two-way relay channels with energy harvesting nodes: The impact of relaying strategies," *IEEE Transactions on Communications*, vol. 63, no. 6, pp. 2081–2093, 2015.

[11] K. Tutuncuoglu and A. Yener, "Energy harvesting networks with energy cooperation: Procrastinating policies," *IEEE Transactions on Communications*, vol. 63, no. 11, pp. 4525–4538, 2015.

[12] ——, "Sum-rate optimal power policies for energy harvesting transmitters in an interference channel," *Journal of Communications and Networks*, vol. 14, no. 2, pp. 151–161, 2012.

[13] B. Gurakan, O. Ozel, J. Yang, and S. Ulukus, "Energy cooperation in energy harvesting communications," *IEEE Transactions on Communications*, vol. 61, no. 12, pp. 4884–4898, 2013.

[14] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 220–230, 2011.

[15] O. Ozel and S. Ulukus, "Achieving AWGN capacity under stochastic energy harvesting," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6471–6483, 2012.

[16] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2020, pp. 1–6.

[17] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[18] N. Agarwal, A. T. Suresh, F. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," in *Advances in Neural Information Processing Systems (Neurips 2018), Montréal, Canada*, 2018, pp. 7575–7586.

[19] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on Non-IID data," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[20] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv:2010.13723*, 2020.

[21] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.

[22] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7690–7703, 2020.

[23] Y. Sun, S. Zhou, and D. Gündüz, "Energy-aware analog aggregation for federated learning with redundant data," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.

[24] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Update aware device scheduling for federated learning at the wireless edge," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 2598–2603.

[25] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2021–2031.

[26] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[28] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[29] J. So, B. Guler, and A. S. Avestimehr, "Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning," *IEEE Journal on Selected Areas in Information Theory: Privacy and Security of Information Systems*, 2021.

[30] S. U. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[31] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.