Graph-Assisted Communication-Efficient Ensemble Federated Learning

Pouya M. Ghari and Yanning Shen*
Department of Electrical Engineering and Computer Science
University of California, Irvine
Irvine, CA, USA
E-mails: pmollaeb@uci.edu, yannings@uci.edu

***Abstract*—Communication efficiency arises as a necessity in federated learning due to limited communication bandwidth. To this end, the present paper develops an algorithmic framework where an ensemble of pre-trained models is learned. At each learning round, the server selects a subset of pre-trained models to construct the ensemble model based on the structure of a graph, which characterizes the server's confidence in the models. Then only the selected models are transmitted to the clients, such that certain budget constraints are not violated. Upon receiving updates from the clients, the server refines the structure of the graph accordingly. The proposed algorithm is proved to enjoy sub-linear regret bound. Experiments on real datasets demonstrate the effectiveness of our novel approach.**

***Index Terms*—federated learning, ensemble learning, graphs**

I. Introduction

Prevalence of distributed networks consisting of devices such as mobile phones and sensors with growing computational and storage capability enables distributing more network computation to the edge. *Federated learning* has emerged as a promising framework to train machine learning models under orchestration of a central server while training data remains distributed among the edge devices called *clients* [1]. In federated learning, a central server sends the current model to a set of clients at each learning round. Participating clients then compute updates of the current model based on their local data and send these updates to the server instead of their local data. The server then update the model. This procedure continues until convergence. In this context, communication efficiency is of utmost importance. To this end, clients-to-server communication efficiency in federated learning has been studied extensively in the context of model updates compression, see e.g., [2], [3]. On the other hand, server-to-clients communication bottleneck arises if the learning task involves large model, such that the required bandwidth to transmit the model exceeds the available bandwidth for server-to-clients communication. For example, base stations can be employed as an aggregating server in certain applications [4]. Often times the base station may only assign a limited portion of available bandwidth for server-to-client communication, while reserving most of the available bandwidth for other service required by users. Meanwhile, large models trained at the central server may exceed the clients' memory capacity. These challenges all motivate the study of server-to-clients communication-efficient federated learning, which is also the focus of the present work.

Ensemble learning methods are known to be effective for learning large-scale models, which combine several base predictors or experts to generate more accurate ensemble model. However, conventional ensemble learning methods (see e.g. [5]) are not directly applicable for communication efficient federated learning. To adapt ensemble learning to federated setting, FedBoost has been proposed by [6], where the server constructs a model by combining a subset of pre-trained models. In this way, the server only needs to transmit a subset of pre-trained models to the clients at each learning round. Pre-trained models can be trained on publicly available data without observing clients' data. FedBoost imposes a budget constraint on the number of pre-trained models that can be transmitted to the clients; however it cannot guarantees that the budget constraint is not violated at each learning round. Instead, it only guarantees the expected cost of model transmission satisfies the budget constraint. Moreover, ensemble learning techniques has been employed in vertical federated learning [7].

The present paper studies server-to-clients communication efficiency in federated learning. Specifically, we aim at selecting a subset of pre-trained models to construct an ensemble model. To this end, each pre-trained model is viewed as an expert lying on a graph. At each learning round, the server chooses a subset of pre-trained models based on the structure of the graph. Upon receiving the updates from the clients, the server refines the structure of the graph. In this context, the prediction provided by each pre-trained model can be viewed as feedback given by the expert. Hence, the constructed graph is named *feedback graph*. The cost of each pre-trained model is proportional to its parameter size and a budget constraint is set to transmit the models in order to construct the ensemble model. We develop an algorithm called **e**nsemble **f**ederated **l**earning with **f**eedback **g**raph (EFL-FG) which selects a subset of pre-trained models to be transmitted to clients, and guarantees the resulting communication cost does not exceed budget constraint at each learning round. EFL-FG is proved to enjoy sub-linear regret. Experiments on real datasets showcase the effectiveness of

*Work in this paper was partially supported by Microsoft Academic Grant Award for AI Research (PI : Y. Shen).

our proposed algorithms compared with state-of-art ensemble federated learning alternative.

II. Problem Statement and Preliminaries

This section introduces the problem of federated learning with ensemble method. Let there are a set of N clients that the server interacts with them to perform a learning task. Moreover, $\mathcal{X}$ and $\mathcal{Y}$ denote the input space and output space, respectively such that a data sample $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$. Furthermore, there are a set of K pre-trained models $f_1(\cdot), \ldots, f_K(\cdot)$ at the server. Each model $f_k(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$, $\forall k \in [K]$ is a mapping from the input space to the output space where $[K]$ denotes the set $\{1, \ldots, K\}$. At each round of learning, the server uniformly chooses a random subset of clients to send them the current model. The number of clients chosen by the server depends on the available bandwidth for clients-to-server communication. Specifically, at each learning round, the server constructs a model using pre-trained models and send it to the chosen subset of clients. Then, upon receiving new data samples, clients perform the learning task using the current model received from the server.

Let $\mathcal{S}_t := \{(\boldsymbol{x}_{i,t}, y_{i,t}), \forall i \in [N]\}$ represents a set of data samples observed by clients at learning round t. In this case, the problem of federated learning can be viewed as a function approximation problem. Specifically, given data samples $\{\mathcal{S}_t\}_{t=1}^{T}$, the goal is to find the function $\hat{f}(\cdot)$, such that the difference between $\hat{f}(\boldsymbol{x}_{i,t})$ and $y_{i,t}$ is minimized. When the i-th client performs the learning task on the received datum $\boldsymbol{x}_{i,t}$, it incurs the loss $\mathcal{L}(\hat{f}(\boldsymbol{x}_{i,t}), y_{i,t})$ where $\mathcal{L}(\cdot, \cdot)$ denotes the loss function. And $\mathcal{L}(f_k(\boldsymbol{x}_{i,t}), y_{i,t})$ denote the loss of each model $f_k(\cdot)$, $\forall k \in [K]$ for the datum $\boldsymbol{x}_{i,t}$ associated with the i-th client. In this context, the goal of the server is to construct the function $\hat{f}(\cdot)$ using pre-trained models in a way that the cumulative loss is minimized. In order to build the function $\hat{f}(\cdot)$ using the pre-trained models, the server employs the ensemble method. Using the conventional ensemble method, the model at the server is constructed as

$$\hat{f}(\boldsymbol{x}) = \sum_{k=1}^{K} w_k f_k(\boldsymbol{x}), \sum_{k=1}^{K} w_k = 1. \tag{1}$$

The ensemble method in (1) requires that the server sends all models to the clients. However, this may not be feasible due to e.g., insufficient bandwidth for server-to-clients communication, limited memory and computational capability of clients to store all models. The present paper proposes a novel algorithmic framework to choose a data-driven subset of models in a way that these limitations are taken into account.

III. Ensemble Federated Learning with Graphs

The present section first introduces a disciplined way to construct a graph based on the performance of pre-trained models. Then, a novel algorithm is proposed to construct an ensemble model employing a subset of pre-trained models chosen by the server based on the graph.

Let c_k be the cost incurred when the k-th pre-trained model is transmitted. Specifically, c_k can be the bandwidth required for transmission of the k-th model to clients. Let B_t denote the budget of the server which denotes the cumulative cost the server can afford for transmission at learning round t, e.g., available bandwidth for server-to-clients communication. In what follows, a principled algorithm to construct a graph is proposed which assists the learner to obtain an ensemble model.

A. Feedback Graph Generation

Let $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ be a directed graph at learning round t with a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}_t$. Each vertex $v_k \in \mathcal{V}$, $\forall k \in [K]$ represents the pre-trained model $f_k(\cdot)$. Let $\boldsymbol{w}_t$ be a weight vector, where the k-th element $w_{k,t}$ is the weight associated with the k-th model $f_k(\cdot)$, indicating the server's confidence about the performance of model $f_k(\cdot)$. At each learning round, the server updates $\boldsymbol{w}_t$ based on the observed loss of $f_k(\cdot)$ which will be specified later. Let $\mathbb{N}_{k,t}^{\text{out}}$, $\forall k \in [K]$ be the out-neighbor set of v_k. In order to construct the set $\mathbb{N}_{k,t}^{\text{out}}$, $\forall k \in [K]$, the server appends nodes v_j to $\mathbb{N}_{k,t}^{\text{out}}$ based on both weights and costs of models such that the cumulative cost of nodes in $\mathbb{N}_{k,t}^{\text{out}}$ does not exceed the budget B_t. At first, the server append v_k to $\mathbb{N}_{k,t}^{\text{out}}$ which means that there is a self loop for each $v_k \in \mathcal{V}$. Let

$$\begin{aligned} \mathbb{M}_{k,t} := \{v_i | \forall i : &\sum_{j \in \mathbb{N}_{k,t}^{\text{out}}} c_j + c_i \leq B_t, \\ &\sum_{j \in \mathbb{N}_{k,t}^{\text{out}}} w_j + w_i \leq \sum_{j \in \mathbb{N}_{k,t-1}^{\text{out}}} w_j, v_i \notin \mathbb{N}_{k,t}^{\text{out}}\} \end{aligned} \tag{2}$$

denote a set of vertices associated with v_k at learning round t. At learning round t, find

$$v_d = \arg \max_{v_i \in \mathbb{M}_{k,t}} \frac{w_{i,t}}{\sum_{v_j \in \mathbb{N}_{k,t}^{\text{out}}} c_j + c_i}. \tag{3}$$

the set $\mathbb{M}_{k,t}$ is then updated by appending v_d to $\mathbb{N}_{k,t}^{\text{out}}$. This procedure continues until $\mathbb{M}_{k,t}$ becomes an empty set, i.e., $|\mathbb{M}_{k,t}| = 0$, where $| \cdot |$ represents the cardinality of a set. This means there is no more node that can be appended to $\mathbb{N}_{k,t}^{\text{out}}$ such that the constraints in (2) are satisfied. Moreover, according to (3), the server appends v_d to $\mathbb{N}_{k,t}^{\text{out}}$ by considering the trade-off between the performance of nodes in prior rounds and the amount of cost that they might add to current cumulative cost of out-neighbors of v_k. When the server constructs $\mathbb{N}_{k,t}^{\text{out}}$, $\forall k \in [K]$, the set of edges $\mathcal{E}_t$ can be constructed. Specifically, $(k, j) \in \mathcal{E}_t$ if $v_j \in \mathbb{N}_{k,t}^{\text{out}}$. The procedure to construct the graph $\mathcal{G}_t$ is summarized in Algorithm 1. At each learning round, the server draws one node in $\mathcal{G}_t$ and transmits models which are out-neighbors of the chosen one. Then the learning task is carried out with a subset of models which are considered as nodes in $\mathcal{G}_t$. Thus, output of the selected models can be viewed as feedback collected from $\mathcal{G}_t$, which is henceforth named as *feedback graph*.

B. Ensemble Federated Learning

At each learning round t, the server selects one node in $\mathcal{G}_t$ and constructs the ensemble model with out-neighbors of the chosen node. To this end, the server assigns weight $u_{k,t}$

Algorithm 1 Feedback Graph Generation

```
Input:Models $f_k(.)$, weights $w_{k,t}$, costs $c_k$, $\forall k \in [K]$ and the budget $B_t$.
for $k = 1, ..., K$ do
    Append $v_k$ to $\mathbb{N}^{\text{out}}_{k,t}$.
    while $|\mathbb{M}_{k,t}| > 0$ do
        Node $v_k$ appends $v_d$ as in (3) to $\mathbb{N}^{\text{out}}_{k,t}$.
        Update $\mathbb{M}_{k,t}$ with respect to updated $\mathbb{N}^{\text{out}}_{k,t}$.
    end while
end for
Output:Feedback Graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$.
```

to node v_k which indicates the confidence in the accuracy of the obtained ensemble model when node v_k is chosen. Then, the server draws one of the nodes $v_k \in \mathcal{V}$ according to the probability mass function (PMF) $\boldsymbol{p}_t$ as follows

$$p_{k,t} = (1-\xi)\frac{u_{k,t}}{U_t} + \frac{\xi}{|\mathbb{D}_t|}\mathcal{I}(v_k \in \mathbb{D}_t) \tag{4}$$

where $\mathcal{I}(\cdot)$ denotes the indicator function, ξ is the exploration rate and $U_t := \sum_{k=1}^{K} u_{k,t}$. The set $\mathbb{D}_t$ denotes a dominating set for the feedback graph $\mathcal{G}_t$. A dominating set $\mathcal{D}_t$ of $\mathcal{G}_t$ is a subset of vertices such that there is an edge from at least one vertex in $\mathcal{D}_t$ to any vertex not in $\mathcal{D}_t$. The PMF in (4) constitutes a trade-off between exploitation and exploration. Let I_t denote the index of the drawn node at learning round t. Let $\mathbb{S}_t$ be a set of indices of nodes which are out-neighbors of the chosen node v_{I_t}. In this case, the server utilizes the weighting vector $\boldsymbol{w}_t$ to construct the ensemble model using models whose indices are in $\mathbb{S}_t$ as follows

$$\hat{f}_t(\boldsymbol{x}) = \sum_{k \in \mathbb{S}_t} \frac{w_{k,t}}{W_t} f_k(\boldsymbol{x}) \tag{5}$$

where $W_t := \sum_{k \in \mathbb{S}_t} w_{k,t}$. Then, the server transmits the ensemble model along with the chosen subset of pre-trained models to all clients. Upon receiving new datum the chosen subset of clients perform the learning task using the ensemble model sent by the server. Then, the i-th client incurs loss $\mathcal{L}(\hat{f}(\boldsymbol{x}_{i,t}), y_{i,t})$ associated with the received data sample $(\boldsymbol{x}_{i,t}, y_{i,t})$. Furthermore, the i-th client computes the loss $\mathcal{L}(f_k(\boldsymbol{x}_{i,t}), y_{i,t})$, $\forall k \in \mathbb{S}_t$. Then each client transmits the losses associated with the ensemble model and the chosen subset of models to the server. Upon receiving the losses, the server updates $w_{k,t}$ and $u_{k,t}$, $\forall k \in [K]$. To this end, the server employs the importance sampling loss estimate which results in an unbiased estimation of the incurred loss. The importance sampling loss estimate for the k-th model can be expressed as

$$\ell_{k,t} = \frac{\sum_{i=1}^{N} \mathcal{L}(f_k(\boldsymbol{x}_{i,t}), y_{i,t})}{q_{k,t}} \mathcal{I}(k \in \mathbb{S}_t) \tag{6}$$

where

$$q_{k,t} := \sum_{v_j \in \mathbb{N}^{\text{in}}_{k,t}} p_{j,t} \tag{7}$$

Algorithm 2 EFL-FG: Ensemble Federated Learning with Feedback Graph

```
Input:Models $f_k(\cdot)$, weights $w_{k,t}$, costs $c_k$, $\forall k \in [K]$.
Initialize: $w_{k,1} = 1$, $u_{k,1} = 1$, $\forall k \in [K]$.
for $t = 1, ..., T$ do
    The server generates $\mathcal{G}_t$ using Algorithm 1.
    The server draws one node in $\mathcal{G}_t$ according to the $\boldsymbol{p}_t$ in (4), with out-neighbors indexed by $\mathbb{S}_t$.
    The server sends models in $\mathbb{S}_t$ and the ensemble model to clients.
    Clients compute and send back to the sever the losses $\mathcal{L}(\hat{f}(\boldsymbol{x}_{i,t}), y_{i,t})$ and $\mathcal{L}(f_k(\boldsymbol{x}_{i,t}), y_{i,t})$, $\forall k \in \mathbb{S}_t$.
    The server computes the importance sampling loss estimates $\ell_{k,t}$ and $\hat{\ell}_{k,t}$, $\forall k \in [K]$.
    The server updates $w_{k,t+1}$ and $u_{k,t+1}$, $\forall k$ as in (9).
end for
```

represents the probability of $k \in \mathbb{S}_t$, where $\mathbb{N}^{\text{in}}_{k,t}$ denotes the in-neighbor set of v_k in $\mathcal{G}_t$. In addition, define the importance sampling loss estimate associated with incurred loss of ensemble model when v_k is drawn by the server as

$$\hat{\ell}_{k,t} = \frac{\sum_{i=1}^{N} \mathcal{L}(\hat{f}(\boldsymbol{x}_{i,t}), y_{i,t})}{p_{k,t}} \mathcal{I}(k = I_t). \tag{8}$$

Using the importance sampling loss estimates in (6) and (8), the weights $w_{k,t}$ and $u_{k,t}$ can the be updated as follows

$$w_{k,t+1} = w_{k,t} \exp(-\eta \ell_{k,t}) \tag{9a}$$

$$u_{k,t+1} = u_{k,t} \exp(-\eta \hat{\ell}_{k,t}) \tag{9b}$$

where η is the learning rate. The procedure that the server sends subset of models to clients is summarized in Algorithm 2. The algorithm is called EFL-FG which stands for Ensemble Federated Learning with Feedback Graph. In each learning round, clients only need to send their computed losses to the server while they do not have to reveal the loss function $\mathcal{L}(\cdot,\cdot)$ and observed data samples. Specifically, in some applications data samples $(\boldsymbol{x}_{i,t}, y_{i,t})$ may include some information about clients that they do not wish to share with the server.

Comparison with online learning. Online learning studies problems where a learner interacts with a set of experts such that at each learning round the learner makes decision based on advice received from the experts [8], [9]. The learner may observe the loss associated with a subset of experts after decision making, which can be modeled using a feedback graph [10], [11]. In EFL-FG, each pre-trained model is also viewed as an expert. However, there is a major innovative difference compared with online learning with feedback graph: the proposed EFL-FG constructs and refines the feedback graph to improve the performance while in online learning, the feedback graph is generated in an adversarial manner.

C. Regret Analysis

The present subsection studies the performance of EFL-FG in terms of cumulative regret. The difference between the loss incurred by the ensemble model and the loss of the best pre-trained model in the hindsight is defined as the regret of the ensemble model. In this context, the best pre-trained model in the hindsight is the one with minimum cumulative loss among all pre-trained models. Therefore, the cumulative regret of EFL-FG can be written as

$$\begin{aligned}\mathcal{R}_T =& \sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{E}_t[\mathcal{L}(\hat{f}(\boldsymbol{x}_{i,t}), y_{i,t})] \\ &- \min_{k\in[K]}\sum_{t=1}^{T}\sum_{i=1}^{N}\mathcal{L}(f_k(\boldsymbol{x}_{i,t}), y_{i,t}) \qquad (10)\end{aligned}$$

where $\mathbb{E}_t[.]$ denotes the conditional expectation given observed losses in prior learning rounds. In order to analyze the performance of EFL-FG, we assume that the following conditions hold:

(a1) The loss function $\mathcal{L}(f_k(\boldsymbol{x}_{i,t}), y_{i,t})$ is convex with respect to $f_k(\boldsymbol{x}_{i,t})$.

(a2) For each $(\boldsymbol{x}_{i,t}, y_{i,t})$, the loss is bounded $0 \leq \mathcal{L}(f_k(\boldsymbol{x}_{i,t}), y_{i,t}) \leq 1$.

(a3) The budget satisfies $B_t \geq c_k$, $\forall k \in [K]$, $\forall t$.

The following Theorem presents the regret bound for EFL-FG with respect to the best pre-trained model in hindsight.

Theorem 1. *Under (a1)–(a3), the expected cumulative regret of EFL-FG is bounded by*

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{i=1}^{N}\mathbb{E}_t[\mathcal{L}(\hat{f}(\boldsymbol{x}_{i,t}), y_{i,t})] - \sum_{t=1}^{T}\sum_{i=1}^{N}\mathcal{L}(f_{k^*}(\boldsymbol{x}_{i,t}), y_{i,t}) \\ \leq& \frac{\ln(K|\mathbb{N}_{k^*,1}^{out}|)}{\eta} \\ &+ \sum_{t=1}^{T}\left(\xi(1-\frac{\eta}{2}N^2) + \frac{\eta}{2}(K + \frac{1}{\bar{q}_{k^*,t}})N^2\right) \qquad (11)\end{aligned}$$

where

$$k^* = \arg\min_{k\in[K]}\sum_{t=1}^{T}\sum_{i=1}^{N}\mathcal{L}(f_k(\boldsymbol{x}_{i,t}), y_{i,t})$$

is the index of the best pre-trained model in the hindsight, and $\frac{1}{\bar{q}_{k^*,t}} := \sum_{j\in\mathbb{N}_{k^*,t}^{out}}\frac{w_{j,t}}{q_{j,t}W_{k^*,t}}$.

Proof. see Appendix in [12]. □

According to (4), $p_{k,t} > \frac{\xi}{|\mathbb{D}_t|}$, $\forall k \in \mathbb{D}_t$. In addition, each node $v_k \in \mathcal{V}$ is in-neighbor to at least one vertex in the dominating set $\mathbb{D}_t$. Therefore, based on (7), it can be concluded that $q_{k,t} > \frac{\xi}{|\mathbb{D}_t|}$, $\forall k \in [K]$ and as a result, we have $\frac{1}{\bar{q}_{k^*,t}} < \frac{|\mathbb{D}_t|}{\xi}$. If the greedy set cover algorithm (see e.g. [13]) is employed to find a dominating set for the feedback graph $\mathcal{G}_t$, EFL-FG obtains a dominating set with $|\mathbb{D}_t| = \mathcal{O}(\alpha(\mathcal{G}_t)\ln K)$ where $\alpha(\mathcal{G}_t)$ denotes the independence number of the feedback graph $\mathcal{G}_t$ [14]. In this case, if the server sets

$$\eta = \mathcal{O}\left(\sqrt{\frac{\ln K}{T}}\right), \xi = \mathcal{O}\left(\frac{(\ln K)^{\frac{3}{4}}}{T^{\frac{1}{4}}}\right) \qquad (12)$$

in (11), EFL-FG obtains sub-linear regret of

$$\mathcal{R}_T = \mathcal{O}\left(\sum_{t=1}^{T}(\ln K)^{\frac{3}{4}}\alpha(\mathcal{G}_t)T^{-\frac{1}{4}}\right). \qquad (13)$$

It is useful to point out that $\alpha(\mathcal{G}_t)$ depends on the budget B_t. Increase in B_t can result in more connected feedback graph $\mathcal{G}_t$ and as a result $\alpha(\mathcal{G}_t)$ decreases. Therefore, larger budget can assist EFL-FG to achieve tighter sub-linear regret bound. For example, when the budget is large enough such that at each learning round t, the feedback graph $\mathcal{G}_t$ is a densely connected graph as $\alpha(\mathcal{G}_t) = \mathcal{O}(1)$, the EFL-FG can achieve regret of $\mathcal{O}((\ln K)^{\frac{3}{4}}T^{\frac{3}{4}})$. By contrast, when the feedback graph $\mathcal{G}_t$ only includes self-loops, $\alpha(\mathcal{G}_t) = K$, the EFL-FG can achieve regret of $\mathcal{O}((\ln K)^{\frac{3}{4}}KT^{\frac{3}{4}})$. The budget B_t can denote server-to-clients communication bandwidth at round t, which implies larger server-to-clients communication bandwidth leads to obtaining tighter regret bound.

IV. Experiments

We tested the performance of different ensemble federated learning methods: our proposed EFL-FG and FedBoost [6] over the following real data sets downloaded from UCI machine learning repository [15]:

Bias Correction: This dataset includes $7,750$ samples of air temperature information with 21 features such as maximum or minimum air temperatures in the day. The goal is to predict the next-day minimum air temperature [16].

CCPP: The dataset contains $9,568$ samples, with 4 features including temperature, pressure, etc, collected from a combined cycle power plant. The goal is to predict hourly electrical energy output [17].

Energy: This dataset contains $19,735$ samples of 27 features of house temperature and humidity conditions were monitored with a wireless sensor network. The goal is to predict the energy use of appliances. [18].

Consider the case where there are 100 clients performing a regression task. The server stores 22 pre-trained models, including kernel based regression models with 5 Gaussian kernels, 5 Laplacian kernels, 5 polynomial kernels, 5 sigmoid kernels and 2 feed-forward neural networks. The bandwidth of Gaussian, Laplacian kernels, and the slope of sigmoid kernels are $0.01, 0.1, 1, 10, 100$. Also, the degree of polynomial kernels are $1, 2, 3, 4, 5$. The feedforward neural networks have 1 and 2 hidden layers respectively, where each hidden layer consists of 25 neurons with ReLU activation functions. Each model is trained with 10% of each dataset. Furthermore, the cost of sending each model is considered to be the number of parameters associated with model divided by the number of parameters associated with the model with maximum number of parameters. Hence, the maximum cost of sending a model is 1. The budget for sending models is $B = 3$. The learning rate η and exploration rate ξ for all methods are set to be $\frac{1}{\sqrt{T}}$. The performance of ensemble federated learning methods is

Table I
MSE ($\times 10^{-3}$) PERFORMANCE AND PERCENTAGE OF BUDGET VIOLENCE.

	MSE($\times 10^{-3}$)			budget violence (%)		
Algorithms	Bias	CCPP	Energy	Bias	CCPP	Energy
FedBoost	69.11	45.06	10.19	22.86%	13.79%	24.16%
EFL-FG	4.81	4.92	8.36	0%	0%	0%

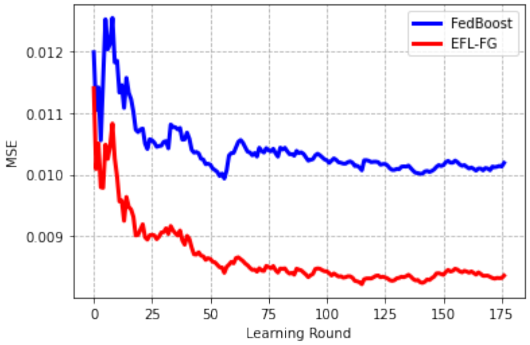

Figure 1. MSE performance on Energy dataset.

evaluated based on mean square error (MSE) at learning round t defined as

$$\text{MSE}_t = \frac{1}{t}\sum_{\tau=1}^{t}\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_{i,\tau} - y_{i,\tau})^2 \qquad (14)$$

where $\hat{y}_{i,t}$ denote the prediction made by the i-th client at learning round t. In order to derive dominating sets for feedback graphs in Algorithm 2, greedy set cover algorithm is employed [13]. Moreover, data samples are distributed i.i.d among clients. Note that in this experiments it is assumed that clients are not able to store their observed data in batch. Therefore, FedBoost implementation is modified to cope with this situation for fair comparison. Specifically, at learning round t, the i-th client transmits its update with respect to newly observed sample $\boldsymbol{x}_{i,t}$ instead of the whole batch of data.

Tables I shows the MSE and budget violence performance of different algorithms over all datasets, where *budget violence* indicates the proportion of learning rounds when the instantaneous budget constraint is violated. As can be seen from Table I, the proposed EFL-FG achieves lower MSE compared with FedBoost. Table I shows that FedBoost violates the instantaneous budget in at least 13% of learning rounds for all datasets, while EFL-FG guaratees that budget constraints is satisfied in every learning round.

Figure 1 illustrates the MSE versus learning rounds in Energy dataset. It can be observed that our proposed EFL-FG outperforms FedBoost. Note that both algorithms obtain different MSE performance starting from the first learning round since different subsets of pre-trained models are chosen and combined by different algorithms.

V. Conclusion

The present paper developed a federated learning approach to learn ensemble of pre-trained models when the server cannot transmit all models to clients due to limitations in communication bandwidth and clients' memory. Specifically, the server generates a graph at each learning round and constructs an ensemble model by choosing a subset of pre-trained models using the graph. This paper provided the algorithm EFL-FG which constructs an ensemble model whose size does not surpass a certain budget at each learning round. We proved that EFL-FG achieves sub-linear regret. Experiments on several real datasets reveal the merits of EFL-FG compared with other ensemble federated learning state-of-art.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of International Conference on Artificial Intelligence and Statistics*, vol. 54, Apr 2017, pp. 1273–1282.

[2] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2017.

[3] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez, and R. Arora, "FetchSGD: Communication-efficient federated learning with sketching," in *Proceedings of International Conference on Machine Learning*, vol. 119, Jul 2020, pp. 8253–8265.

[4] M. S. H. Abad, E. Ozfatura, D. GUndUz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8866–8870.

[5] P. Bühlmann, *Bagging, Boosting and Ensemble Methods*. Springer Berlin Heidelberg, 2012, pp. 985–1022.

[6] J. Hamer, M. Mohri, and A. T. Suresh, "FedBoost: A communication-efficient algorithm for federated learning," in *Proceedings of International Conference on Machine Learning*, vol. 119, Jul 2020, pp. 3973–3983.

[7] X. Chen, S. Zhou, B. Guan, K. Yang, H. Fao, H. Wang, and Y. Wang, "Fed-eini: An efficient and interpretable inference framework for decision tree ensembles in vertical federated learning," in *IEEE International Conference on Big Data (Big Data)*, 2021, pp. 1242–1248.

[8] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. USA: Cambridge University Press, 2006.

[9] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, p. 48–77, Jan 2003.

[10] S. Mannor and O. Shamir, "From bandits to experts: On the value of side-observations," in *Proceedings of International Conference on Neural Information Processing Systems*, 2011, pp. 684–692.

[11] C. Cortes, G. DeSalvo, C. Gentile, M. Mohri, and N. Zhang, "Online learning with dependent stochastic feedback graphs," in *Proceedings of International Conference on Machine Learning*, Jul 2020.

[12] P. M. Ghari and Y. Shen, "Graph-assisted communication-efficient ensemble federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2202.13447

[13] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, Aug 1979.

[14] N. Alon, N. Cesa-Bianchi, C. Gentile, S. Mannor, Y. Mansour, and O. Shamir, "Nonstochastic multi-armed bandits with graph-structured feedback," *SIAM Journal on Computing*, vol. 46, no. 6, pp. 1785–1826, 2017.

[15] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[16] D. Cho, C. Yoo, J. Im, and D.-H. Cha, "Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas," *Earth and Space Science*, vol. 7, no. 4, Mar 2020.

[17] P. Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," *International Journal of Electrical Power and Energy Systems*, vol. 60, pp. 126 – 140, 2014.

[18] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*, vol. 140, pp. 81 – 97, 2017.