Continual Local Updates for Federated Learning with Enhanced Robustness to Link Noise

Ehsan Lari[1], Vinay Chakravarthi Gogineni[2], Reza Arablouei[3], Stefan Werner[1]
[1]Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway
[2]The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark
[3]CSIRO's Data61, Pullenvale QLD 4069, Australia

***Abstract*—Communication errors caused by noisy links can negatively impact the accuracy of federated learning (FL) algorithms. To address this issue, we introduce an FL algorithm that is robust to communication errors while concurrently reducing the communication load on clients. To formulate the proposed algorithm, we consider a weighted least-squares regression problem as a motivating example. We recast this problem as a distributed optimization problem over a federated network, which employs random scheduling to enhance communication efficiency, and solve the reformulated problem via the alternating direction method of multipliers. Unlike conventional FL approaches employing random scheduling, the proposed algorithm grants the clients the ability to continually update their local model estimates even when they are not selected by the server to participate in FL. This allows for more frequent and ongoing client involvement, resulting in performance improvement and enhanced robustness to communication errors compared to when the local updates are only performed when the respective clients are selected by the server. We demonstrate the effectiveness and performance gains of the proposed algorithm through simulations.**

I. Introduction

Federated learning (FL) [1]–[6] is a distributed machine-learning approach that enables the training of models across multiple decentralized devices without transferring the data to any global server. FL allows devices to share their knowledge in terms of model estimates or gradients without revealing their raw data, thereby improving data privacy and security [7]. FL holds significant potential in a range of critical applications, including healthcare [8], finance [9], and industrial IoT [10], [11]. These data-sensitive domains demand utmost priority on data privacy, making FL an ideal approach to safeguard sensitive information while enabling effective machine learning.

The literature on FL encompasses a diverse array of methods that have been extensively investigated to address a multitude of aspects and challenges such as preserving privacy [9], [12]–[14], handling Byzantine attacks [15], [16], and improving communication efficiency [7], [17]. However, a significant number of these works operate under the assumption of ideal communication links, neglecting the presence of communication errors or noise [18]–[22]. In practical real-world applications, however, the communication links connecting clients and the server are susceptible to noise corruption, posing a potential threat to the performance of the model [23]. To tackle this challenge, several studies have introduced different techniques aimed at enhancing the accuracy of FL models when confronted with communication noise. Some works focus on the uplink noise and neglect the downlink noise [24], [25] while some other also consider the effects of the downlink noise [26].

Distributed algorithms that are based on the alternating direction method of multipliers (ADMM) can exhibit a certain degree of robustness to additive communication noise [27]. This is due to the inherent characteristics of ADMM that allow it to alleviate the impact of noise present in the communication channel. However, it requires complete collaboration among all clients, which stands in contrast to the objectives of FL, particularly those associated with system heterogeneity, such as varying computation and storage capacities across different clients. Hence, there exists a need for noise-robust ADMM-based algorithms that achieve convergence even when only a subset of clients participate in each iteration.

In this paper, we introduce a new FL algorithm that exhibits both communication efficiency and robustness to communication noise/errors. Additionally, our algorithm facilitates continual local updates at the clients, even when they are not selected by the server. This results in improved accuracy with no extra communication. We derive the proposed algorithm by using ADMM to solve a weighted least-squares (WLS) regression problem. We consider the communication links in both uplink and downlink to be noisy. To achieve noise robustness, we eliminate the dual variable update step at each client and transmit a linear combination of the last two global model updates. Our extensive simulation results corroborate the effectiveness of the proposed algorithm.

II. Background

Let us consider a federated network with K clients and a server. Each client k has an exclusive dataset denoted by $\mathcal{D}_k = \{\mathbf{H}_k, \mathbf{y}_k\}$ where $\mathbf{y}_k \in d_k$ is a column vector and $\mathbf{H}_k$ is a matrix of size $d_k \times L$. Each client independently trains a local model on its respective dataset using an FL algorithm. The process entails exchanging model updates with the global server, allowing for collaborative model training while preserving data privacy. For client k, a linear regression model relating $\mathbf{H}_k$ to $\mathbf{y}_k$ can be described as

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}^* + \boldsymbol{\nu}_k, \tag{1}$$

where $\mathbf{x}^* \in \mathbb{R}^L$ is the global regression parameter vector and $\boldsymbol{\nu}_k$ represents noise or perturbation.

The goal of FL is to find an estimate of $\mathbf{x}^*$ that is the optimal solution to

$$\min_{\{\mathbf{x}_k\}} \sum_{k=1}^{K} \mathcal{J}_k(\mathbf{x}_k) \;\text{ s.t. } \mathbf{x}_k = \mathbf{x},\; k = 1, 2, \cdots, K, \tag{2}$$

where $\mathcal{J}_k(\mathbf{x}) = \|\mathbf{y}_k - \mathbf{H}_k\mathbf{x}\|^2_{\mathbf{W}_k}$ is the local objective function at client k and $\mathbf{W}_k$ is the appropriate weight matrix of client k. In addition, $\mathbf{x}_k$ represents the local model estimate at client k, and $\mathbf{x}$ denotes the global model estimate.

We use an ADMM-based approach to solve (2). In this approach, primal and dual updates are computed by each client and transmitted to the global server via a noisy communication channel. Each client implements local iterations to update its local model and shares its updated local model estimate with the global server. The global server aggregates the received model estimates from the clients and updates the global model estimate. The server then sends the updated global model estimate to the clients over a noisy channel. The clients utilize the received global model estimate in updating their local estimates and the process continues until a convergence criterion is met.

To solve (2) using ADMM, we express the related augmented Lagrangian function as

$$\sum_{k=1}^{K} \mathcal{L}_k(\mathbf{x}, \mathbf{x}_k, \boldsymbol{\pi}_k) = \sum_{k=1}^{K} \mathcal{J}_k(\mathbf{x_k}) + \langle \mathbf{x}_k - \mathbf{x}, \boldsymbol{\pi}_k \rangle + \frac{\rho_k}{2}\|\mathbf{x}_k - \mathbf{x}\|_2^2, \tag{3}$$

where $\boldsymbol{\pi}_k \in \mathbb{R}^L$ and $\rho_k > 0$ are, respectively, the Lagrange multiplier vector and the penalty parameter associated with client k. Hence, the corresponding ADMM iterations at each client k are given as

$$\boldsymbol{\pi}_{k,n} = \boldsymbol{\pi}_{k,n-1} + \rho_k(\mathbf{x}_{k,n} - \mathbf{x}_n) \tag{4a}$$

$$\mathbf{x}_{k,n+1} = \hat{\mathbf{x}}_k - \mathbf{N}_k^{-1}(\boldsymbol{\pi}_{k,n} - \rho_k \mathbf{x}_n) \tag{4b}$$

and at the global server as

$$\mathbf{x}_{n+1} = \frac{1}{\sum\limits_{k=1}^{K} \rho_k} \sum_{k=1}^{K} (\rho_k \mathbf{x}_{k,n+1} + \boldsymbol{\pi}_{k,n}) \tag{5}$$

where we define $\mathbf{N}_k = 2\mathbf{H}_k^{\intercal}\mathbf{W}_k\mathbf{H}_k + \rho_k\mathbf{I}$ and $\hat{\mathbf{x}}_k = 2\mathbf{N}_k^{-1}\mathbf{H}_k^{\intercal}\mathbf{W}_k\mathbf{y}_k$, and the index n denotes the iteration number.

In the above iterations, each client shares the local estimate $\rho_k\mathbf{x}_{k,n+1} + \boldsymbol{\pi}_{k,n}$ with the global server after computing (4a) and (4b). The global server then aggregates the received estimates to obtain the global estimate as in (5).

As it is evident from the recursions (4) and (5), both primal and dual model updates are sent to the server for it to estimate the global model update. However, by making a careful selection of the initial values, the information of the dual update can be incorporated into the primal update. Therefore, using the initial values $\mathbf{x}_{-1} = \mathbf{0}$, $\boldsymbol{\pi}_{k,-1} = \mathbf{0}$, and $\mathbf{x}_{k,0} = \hat{\mathbf{x}}_k$, we can eliminate the Lagrange multiplier vectors $\boldsymbol{\pi}_{k,n}$ and restate the recursions (4)-(5) as

$$\mathbf{x}_{k,n+1} = (\mathbf{I} - \rho_k\mathbf{N}_k^{-1})\mathbf{x}_{k,n} + \rho_k\mathbf{N}_k^{-1}(2\mathbf{x}_n - \mathbf{x}_{n-1}) \tag{6a}$$

$$\mathbf{x}_{n+1} = \frac{1}{\sum\limits_{k=1}^{K} \rho_k} \sum_{k=1}^{K} \rho_k \mathbf{x}_{k,n+1}. \tag{6b}$$

By defining $\mathbf{s}_{k,n} = 2\mathbf{x}_{k,n} - \mathbf{x}_{k,n-1}$ and $\mathbf{s}_n = 2\mathbf{x}_n - \mathbf{x}_{n-1}$, we can further rewrite (6) as

$$\mathbf{x}_{k,n+1} = (\mathbf{I} - \rho_k\mathbf{N}_k^{-1})\mathbf{x}_{k,n} + \rho_k\mathbf{N}_k^{-1}\mathbf{s}_n \tag{7a}$$

$$\mathbf{s}_{k,n+1} = 2\mathbf{x}_{k,n+1} - \mathbf{x}_{k,n} \tag{7b}$$

$$\mathbf{s}_{n+1} = \frac{1}{\sum\limits_{k=1}^{K} \rho_k} \sum_{k=1}^{K} \rho_k \mathbf{s}_{k,n+1}. \tag{7c}$$

In this algorithm, the clients share $\mathbf{s}_{k,n+1}$ with the server, and the server broadcasts $\mathbf{s}_{n+1}$ to the clients. When the exchange of model parameters occurs over noisy communication channels, the iterations (7) are more robust to communication noise compared to (6) as they utilize a single noisy global estimate received from the server to update the local estimate, i.e., $\mathbf{s}_n$, rather than two in (6), i.e., $\mathbf{x}_n$ and $\mathbf{x}_{n-1}$.

In the downlink, noisy versions of the aggregated global model updates are received by the clients as $\tilde{\mathbf{s}}_n^k = \mathbf{s}_n + \boldsymbol{\zeta}_n^k$ where $\boldsymbol{\zeta}_n^k$ denotes the downlink noise of client k at iteration n. In the uplink, a noisy version of the local model update of each client is received by the server as $\tilde{\mathbf{s}}_{k,n} = \mathbf{s}_{k,n} + \boldsymbol{\eta}_{k,n}$ where $\boldsymbol{\eta}_{k,n}$ denotes the uplink noise of client k at iteration n.

In recursions (7), all clients are required to take part in each global model update. However, FL clients often have limited and diverse communication capabilities. Hence, participation of all clients in each global update round can be costly. Therefore, FL servers usually employ random scheduling of their clients by selecting only a subset of the clients, denoted by $\mathcal{S}_n$, to participate in model aggregation during each iteration n. The scheduling reduces the communications required at each iteration, leading to improved efficiency and better resource management. Utilizing random scheduling and considering noisy communications, we can rewrite the recursions (7) as mentioned in [28] as

$$\mathbf{x}_{k,n+1} = \begin{cases} \left(\mathbf{I} - \rho_k\mathbf{N}_k^{-1}\right)\mathbf{x}_{k,n} + \rho_k\mathbf{N}_k^{-1}\tilde{\mathbf{s}}_n^k, & k \in \mathcal{S}_n \\ \mathbf{x}_{k,n}, & \text{otherwise} \end{cases} \tag{8a}$$

$$\mathbf{s}_{k,n+1} = 2\mathbf{x}_{k,n+1} - \mathbf{x}_{k,n}, \quad k \in \mathcal{S}_n \tag{8b}$$

$$\mathbf{s}_{n+1} = \frac{1}{\sum\limits_{k\in\mathcal{S}_n} \rho_k} \sum_{k\in\mathcal{S}_n} \rho_k \tilde{\mathbf{s}}_{k,n+1}. \tag{8c}$$

III. Proposed Algorithm

It is important to enhance the energy efficiency of FL by reducing its communication load, particularly considering that clients often have constraints on their energy resources in real-world scenarios. Lowering the communication burden on the clients can also improve the scalability and cost-effectiveness of

FL algorithms. The efficiency of communication during model training is related to the amount of data exchanged between the clients and the server. Thus, minimizing the exchanges of models, parameters, gradients, or other relevant information while upholding high accuracy levels presents a significant challenge in FL.

Random scheduling of clients is a practical method to enhance communication efficiency within FL, using which the overall communication load can be effectively reduced and resource utilization can be optimized more efficiently. It helps alleviate the potential bottlenecks that can arise from simultaneous communication by all clients. As a result, it improves the efficiency of data transmissions and plays a role in enhancing the overall performance of FL. However, in the conventional random scheduling approach as in (8), the clients that are not selected during each iteration do not carry out any local update and their most recent local model estimates are not incorporated into the global model aggregation process.

Here, we propose to allow all clients, including those not selected through random scheduling, to update their local model estimates during every iteration. As we will show later, this can improve the performance without introducing any additional communication overhead or imposing any significant increase in computations on the clients or the server. To realize the proposed algorithm, we let the clients store the most recent global model estimate received from the server and the server store the latest local model estimates received from the clients. Hence, the clients continually update their local models using their most recent global model estimate and the server updates the global model using the latest local updates from all clients. When a client is selected at iteration n, its latest local model estimate is made up-to-date at the server and the global model estimate received from the server replaces its older version at the client.

Therefore, the recursions of the proposed resource-efficient and noise-robust FL algorithm featuring continual local updates are given by

$$\begin{aligned}\mathbf{x}_{k,n+1} &= (\mathbf{I}-\rho_k\mathbf{N}_k^{-1})\mathbf{x}_{k,n} \\ &\quad + \rho_k\mathbf{N}_k^{-1}\left[a_{k,n}\tilde{\mathbf{s}}_n^k + (1-a_{k,n})\tilde{\mathbf{s}}_m^k\right] \quad (9a)\\ \mathbf{s}_{k,n+1} &= 2\mathbf{x}_{k,n+1} - \mathbf{x}_{k,n}, \quad k\in\mathcal{S}_n \quad (9b)\\ \mathbf{s}_{n+1} &= \frac{1}{\sum_{k=1}^{K}\rho_k}\sum_{k=1}^{K}\rho_k\left[a_{k,n}\tilde{\mathbf{s}}_{k,n+1} + (1-a_{k,n})\tilde{\mathbf{s}}_{k,m}\right], \quad (9c)\end{aligned}$$

where $a_{k,n}$ represents random scheduling, i.e., $a_{k,n}=1$ when $k\in\mathcal{S}_n$ and $a_{k,n}=0$ otherwise. In addition, $\tilde{\mathbf{s}}_m^k$ represents the most recent global model estimate received from the server and stored in client k, which is utilized when the client is not chosen by the server. Moreover, $\tilde{\mathbf{s}}_{k,m}$ denotes the most recent local model estimate associated with client k, which is stored at the server and utilized during iterations when the client is not selected through random scheduling. We summarize the proposed algorithm in Algorithm 1.

Algorithm 1 The proposed communication-efficient and noise-robust FL algorithm with continual local updates.

Parameters: penalty parameters ρ_k, number of clients K, set of clients $\mathcal{S}$

Initialization: global model $\mathbf{x}_0 = \mathbf{x}_{-1} = \mathbf{0}$, local models $\mathbf{x}_{k,0} = \hat{\mathbf{x}}_k$

For $n = 1, \cdots, N$

The server randomly selects a subset $\mathcal{S}_n$ of its clients and sends the aggregated global model $\mathbf{s}_n$ to them.

Client Local Update:

If $k\in\mathcal{S}_n$

Receive $\tilde{\mathbf{s}}_n^k$, a noisy version of $\mathbf{s}_n$, from the server.

Store the latest global model $\tilde{\mathbf{s}}_m^k = \tilde{\mathbf{s}}_n^k$.

Update the local model as

$$\mathbf{x}_{k,n+1} = (\mathbf{I}-\rho_k\mathbf{N}_k^{-1})\mathbf{x}_{k,n} + \rho_k\mathbf{N}_k^{-1}\tilde{\mathbf{s}}_n^k.$$

Send $\mathbf{s}_{k,n+1} = 2\mathbf{x}_{k,n+1} - \mathbf{x}_{k,n}$ to the server.

Else

Update the local model as

$$\mathbf{x}_{k,n+1} = (\mathbf{I}-\rho_k\mathbf{N}_k^{-1})\mathbf{x}_{k,n} + \rho_k\mathbf{N}_k^{-1}\tilde{\mathbf{s}}_m^k.$$

EndIf

Aggregation at the Server:

The server receives $\tilde{\mathbf{s}}_{k,n+1}$, noisy versions of the locally updated models from the selected clients $k\in\mathcal{S}_n$ and aggregates them with $\tilde{\mathbf{s}}_{k,m}$, the stored local model estimates of the non-selected clients via

$$\mathbf{s}_{n+1} = \frac{1}{\sum_{k=1}^{K}\rho_k}\sum_{k=1}^{K}\rho_k\left[a_{k,n}\tilde{\mathbf{s}}_{k,n+1} + (1-a_{k,n})\tilde{\mathbf{s}}_{k,m}\right].$$

EndFor

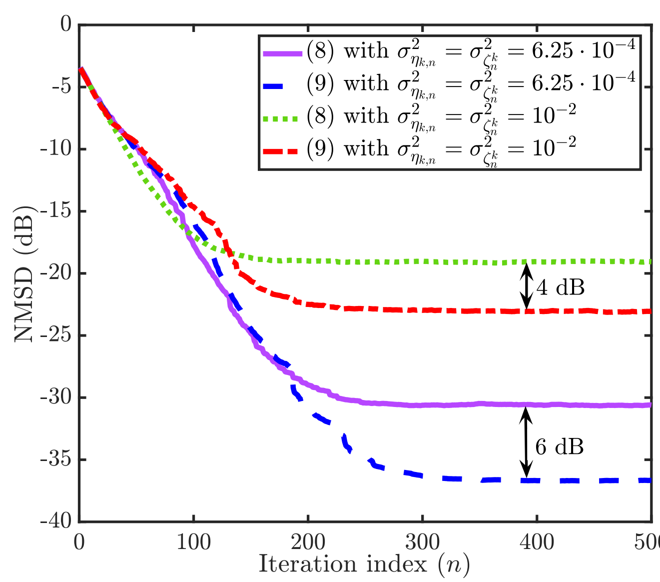

Fig. 1. NMSD of (8) and (9) versus iteration number for $|\mathcal{S}_n| = 4$ and different link noise variances.

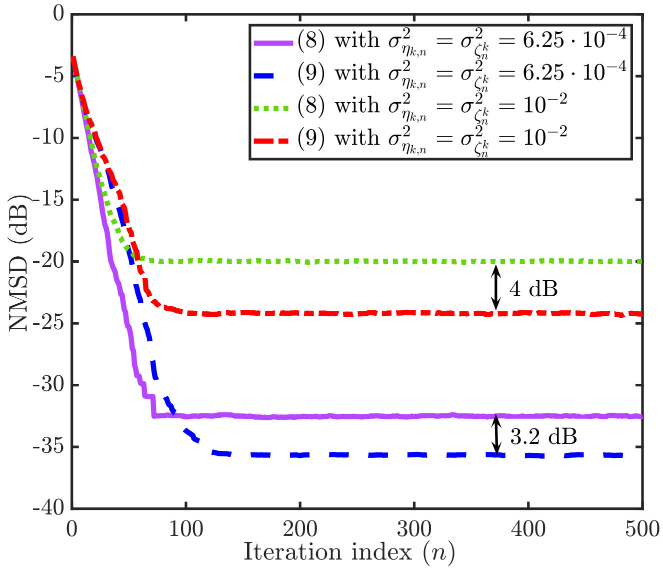

Fig. 2. NMSD of (8) and (9) versus iteration number for $|\mathcal{S}_n| = 10$ and different link noise variances.

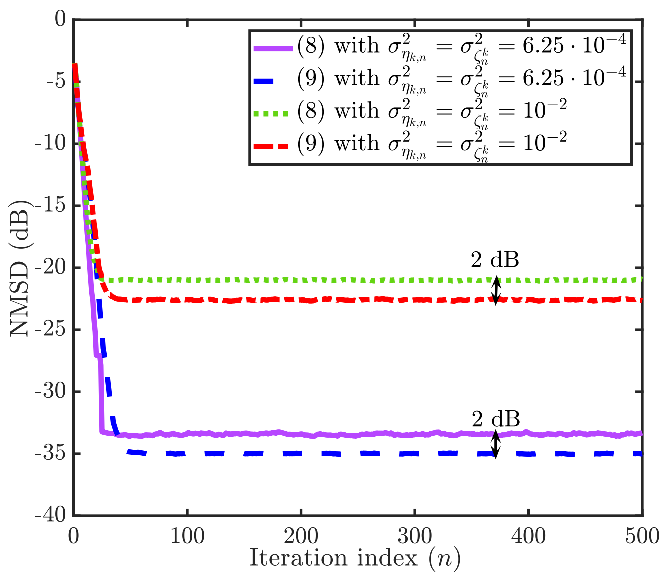

Fig. 3. NMSD of (8) and (9) versus iteration number for $|\mathcal{S}_n| = 25$ and different link noise variances.

IV. Simulation Results

In this section, we conduct a series of numerical experiments to examine the performance of the proposed algorithm. We consider a scenario with $K = 100$ clients directly connected to a global server. The goal of the federated network is to estimate a global model $\mathbf{x}^*$ of dimension $L = 128$. To induce data imbalance among the clients, we draw the size of each client's local dataset, d_k, from a uniform distribution, i.e., $d_k \in \mathcal{U}(50, 90)$. Each client k has synthetic and imbalanced non-IID data $\{\mathbf{H}_k, \mathbf{y}_k\}$ with the matrices $\mathbf{H}_k$ drawn from a multivariate normal distribution $\mathcal{N}(\mu_{\mathbf{H}_k}, \sigma^2_{\mathbf{H}_k})$ where $\mu_{\mathbf{H}_k} \in \mathcal{U}(-0.5, 0.5)$ and $\sigma^2_{\mathbf{H}_k} \in \mathcal{U}(0.5, 1.5)$. The weight matrices are set to the inverse covariance matrix of $\mathbf{y}_k$ at each client k, i.e., $\mathbf{W}_k = \Sigma^{-1}_{\mathbf{y}_k} = \mathbb{E}[(\mathbf{y}_k - \mathbb{E}[\mathbf{y}_k])\,(\mathbf{y}_k - \mathbb{E}[\mathbf{y}_k])^\intercal]^{-1}$. We set the global parameter vector $\mathbf{x}^*$ arbitrarily by drawing each entry from a standard normal distribution $\mathcal{N}(0, 1)$. The observation noise $\boldsymbol{\nu}_k$ at each client k is zero-mean IID Gaussian with variance 10^{-4}. The additive noise in both uplink and downlink are zero-mean IID white Gaussian. In all experiments, we set the penalty parameter to $\rho_k = 1$ for all clients. At each iteration n, the server selects a subset of the clients with equal probability of selection assigned to each client. We evaluate the performance using the network-wide average normalized mean-square deviation (NMSD) defined at each iteration n as

$$\mathrm{NMSD}(n) = \frac{1}{K} \sum_{k=1}^{K} \frac{\|\mathbf{x}_{k,n} - \mathbf{x}^\star\|_2^2}{\|\mathbf{x}^\star\|_2^2}. \tag{10}$$

We obtain the learning curves (i.e., NMSD in dB vs. iteration number n) by averaging over 100 independent trials.

In Figs. 1-3, we present a performance comparison between the proposed algorithm, i.e., (9), and its closest contender, i.e. (8), which does not feature continual local updates. We obtain the results by including noise in the communication links and utilizing random scheduling for communication efficiency. We set the number of clients selected at each iteration to $|\mathcal{S}_n| \in \{4, 10, 25\}$ and use different link noise variances of $\sigma^2_{\boldsymbol{\eta}_{k,n}} = \sigma^2_{\boldsymbol{\zeta}^k_n} \in \{6.25 \cdot 10^{-4}, 10^{-2}\}$. We plot the corresponding learning curves in the figures. We observe that the proposed algorithm exhibits robustness against communication noise/error even when a small portion of the clients participate in every FL round. It also outperforms (8) significantly in terms of the steady-state NMSD in all considered cases. In summary, the results demonstrate that permitting the clients, which are not selected by random scheduling, to continually update their local models leads to a notable reduction in the steady-state NMSD without compromising the convergence rate. Moreover, as expected, when the variance of the noise in the communication links increases, the learning accuracy decreases.

V. Conclusions

We proposed a new FL algorithm that leverages random scheduling to enhance communication efficiency while being robust to additive noise in the communication links, in part, due to eliminating the dual parameters and minimizing the use of noisy estimates in update equations. The key novel feature of the proposed algorithm is that it allows the clients to update their model estimates locally even when they are not selected by the global server for participation in FL as per random scheduling. Our simulation results showed that the continual local updates lead to performance improvements in terms of both learning accuracy and robustness to link noise.

Acknowledgement

The Research Council of Norway supported this work.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. and Stat.*, Apr. 2017, pp. 1273–1282.

[2] V. Smith, C. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Proc. Adv. Neural Inf. Process. Sys.*, vol. 30, Jan. 2017.

[3] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9441–9455, Oct. 2020.

[4] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Trans. Automat. Contr.*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[5] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for IoT devices," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1817–1829, Feb 2021.

[6] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[7] H. Zhao, W. Du, F. Li, P. Li, and G. Liu, "FedPrompt: Communication-efficient and privacy-preserving prompt tuning in federated learning," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, 2023, pp. 1–5.

[8] T. Xia, J. Han, A. Ghosh, and C. Mascolo, "Cross-device federated learning for mobile health diagnostics: A first study on COVID-19 detection," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, 2023, pp. 1–5.

[9] Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: Challenges and solutions," *IEEE Consumer Electronics Mag.*, vol. 9, no. 3, pp. 8–16, 2020.

[10] R. Xie, C. Li, X. Zhou, and Z. Dong, "Asynchronous federated learning for real-time multiple licence plate recognition through semantic communication," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, 2023, pp. 1–5.

[11] Z. Zheng, Y. Zhou, Y. Sun, Z. Wang, B. Liu, and K. Li, "Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges," *Connection Science*, vol. 34, no. 1, pp. 1–28, 2022.

[12] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, 2020.

[13] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 603–618.

[14] Y. Chang, K. Zhang, J. Gong, and H. Qian, "Privacy-preserving federated learning via functional encryption, revisited," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 1855–1869, 2023.

[15] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, Aug. 2018.

[16] X. He, J. Zhang, and Q. Ling, "Byzantine-robust and communication-efficient personalized federated learning," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, 2023, pp. 1–5.

[17] V. C. Gogineni, S. Werner, Y. Huang, and A. Kuh, "Communication-efficient online federated learning strategies for kernel regression," *IEEE Internet Things J.*, pp. 4531–4544, Nov 2022.

[18] ——, "Communication-efficient online federated learning framework for nonlinear regression," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2022, pp. 5228–5232.

[19] ——, "Decentralized graph federated multitask learning for streaming data," in *Proc. 56th Annu. Conf. Inf. Sci. Syst.*, Apr. 2022, pp. 101–106.

[20] 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 J. on Selected Areas in Commun.*, vol. 37, no. 6, pp. 1205–1221, Mar. 2019.

[21] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Federated learning for ultra-reliable low-latency v2v communications," in *Proc. IEEE Global Commun. Conf.*, Feb. 2018, pp. 1–7.

[22] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. on Commun.*, Jul. 2019, pp. 1–7.

[23] F. Ang, L. Chen, N. Zhao, Y. Chen, W. Wang, and F. R. Yu, "Robust federated learning with noisy communication," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3452–3464, Mar. 2020.

[24] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[25] H. Guo, A. Liu, and V. K. N. Lau, "Analog gradient aggregation for federated learning over wireless networks: Customized design and convergence analysis," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 197–210, Jan. 2021.

[26] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of federated learning over a noisy downlink," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1422–1437, Mar. 2022.

[27] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links—part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[28] E. Lari, V. C. Gogineni, R. Arablouei, and S. Werner, "Resource-efficient federated learning robust to communication errors," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, 2023, pp. 265–269.