Federated Learning Enabled Link Scheduling in D2D Wireless Networks

Tianrui Chen, Xinruo Zhang, Minglei You, Gan Zheng, *Fellow, IEEE,* and Sangarapillai Lambotharan, *Senior Member, IEEE*

Abstract

Centralized machine learning methods for device-to-device (D2D) link scheduling may lead to a computing burden for a central server, transmission latency for decisions, and privacy issues for D2D communications. To mitigate these challenges, a federated learning (FL) based method is proposed to solve the link scheduling problem, where a global model is distributedly trained at local devices, and a server is used for aggregating model parameters instead of training samples. Specially, a more realistic scenario with limited channel state information (CSI) is considered instead of full CSI. Despite a decentralized implementation, simulation results demonstrate that the proposed FL based approach with limited CSI performs close to the conventional optimization algorithm. In addition, the FL based solution achieves almost the same performance as that of the centralized training.

Index Terms

Federated learning, Device-to-device (D2D), Link scheduling

I. Introduction

In the emerging Internet of Things (IoT) ecosystem, device-to-device (D2D) communication becomes an important technology which enables direct communications between devices [1]. One of the challenging problems in the D2D networks is the link scheduling problem, however, such problem is a tricky combinatorial and nonconvex optimization problem [2]. Conventional algorithms usually cannot meet the increasingly stringent time requirements in wireless networks.

Tianrui Chen and Minglei You are with the Department of Electrical and Electronic Engineering, University of Nottingham, Nottingham, NG7 2RD, UK (Email: {Tianrui.Chen1, Minglei.You}@nottingham.ac.uk), Xinruo Zhang is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, UK (Email: xinruo.zhang@essex.ac.uk), Gan Zheng is with the School of Engineering, University of Warwick, Coventry, CV4 7AL, UK (Email: gan.zheng@warwick.ac.uk), and Sangarapillai Lambotharan is with the Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK (Email: s.lambotharan@lboro.ac.uk).

The state-of-the-art machine learning (ML) methods such as [3], [4], and [5] have made great efforts to solve the D2D link scheduling problem. These works not only achieve performance close to the conventional optimization algorithms but also accelerate the approximation of this link scheduling problem. Nevertheless, these methods require centralized implementations where massive training samples, e.g., channel state information (CSI) of all D2D pairs, are transmitted from distributed devices to a central server and trained there. This could lead to a huge computational burden, especially for large-scale networks. CSI is of paramount importance from the perspective of physical layer security [6]. If CSI is obtained by eavesdroppers, they can exploit it to decode the confidential transmitted data [7] and perform various attacks. Moreover, sending the scheduling decisions from a central server to local devices may result in transmission delay. Due to the increasing computing power of IoT devices, the workload of the centralized training is promising to be moved to edge devices. Federated learning (FL) has emerged as a distributed ML solution, wherein clients train their models locally, and a server aggregates the local model parameters instead of their raw training data [8]. Therefore, the FL can alleviate the workload of the central server by moving model training to local devices and preserve the data privacy by keeping it locally which will also reduce the security risk induced by CSI exposure. Besides, decisions can be made locally hence reducing latency.

Regarding FL related scheduling, existing works mainly focus on device scheduling policies for facilitating the convergence of the FL, e.g., [9], rather than using the FL to solve the optimization problem itself. Besides, the FL approach [8] adopts the stochastic gradient descent (SGD) as the optimizer for updating local models in parallel at clients, which is usually difficult to tune and results in undesirable convergence performance [10]. Moreover, it is assumed that the channel matrix of all D2D pairs is available in [3], which is difficult to acquire in practical wireless networks.

To mitigate the aforementioned challenges, the FL is used to tackle the D2D link scheduling problem in a distributed manner under the assumption of limited CSI instead of full CSI. The contributions of this work are summarized as follows,

- An FL based method is proposed to facilitate distributed training for D2D link scheduling in the presence of only limited CSI. To the best of our knowledge, this is the first work to provide a distributed implementation of this problem.
- Compared to the conventional centralized training, the FL based approach has great features including computing offloading, local decision-making and privacy issue mitigation.

Simulation results demonstrate that the proposed decentralized solution achieves almost the same performance as that of the centralized training method.

The remainder of this paper is organised as follows. Section II presents the system model and the problem formulation for the D2D link scheduling problem. An FL enabled D2D link scheduling is introduced in Section III. Section IV evaluates the proposed approach via numerical results followed by conclusions in Section V.

II. System Model and Problem Formulation

A D2D wireless network with L unidirectional pairs in a shared channel is considered as the system model. The set of all D2D pairs is represented by $\mathcal{D} = \{D_1, D_2, \ldots, D_L\}$, and the set of their indexes is denoted by $\mathcal{L} = \{1, 2, \ldots, L\}$. Besides, the transmitter and receiver of D_l are denoted by T_l and R_l, respectively. The transmit power of D_l is denoted by p_l. The locations of all D2D pairs are randomly generated in a square area with an edge length of d_{area}, and the distance between T_l and R_l is randomly selected within a pairwise distance between $d_{\min}$ and $d_{\max}$.

Let h_{ll} denote the communication channel between the T_l and R_l, and h_{kl} denote the interference channel from T_k to R_l, where $l, k \in \mathcal{L}$ and $l \neq k$. Additionally, let x_l denote the binary decision variable of D_l indicating the on and off status of the D2D pair. If D_l is active, $x_l = 1$, otherwise $x_l = 0$. Let σ^2 denote additive white Gaussian noise power level. The signal-to-interference-plus-noise ratio (SINR) of D_l denoted by ξ_l is written as

$$\xi_l = \frac{|h_{ll}|^2 p_l x_l}{\sum_{k=1, k\neq l}^{L} |h_{kl}|^2 p_k x_k + \sigma^2}. \tag{1}$$

The objective of the D2D link scheduling problem is to maximize the sum rate of the entire system via optimizing the binary scheduling decisions, which can be formulated as

$$\max_{\boldsymbol{x}} \quad \sum_{l=1}^{L} \log_2(1 + \xi_l(\boldsymbol{x})), \quad \text{s. t.} \quad x_l \in \{0, 1\}, \ \forall l \in \mathcal{L}, \tag{2}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_L]^T$ denotes the binary scheduling vector. The data rate is normalized by the channel bandwidth.

III. Federated learning for D2D Link Scheduling

In this section, an FL based approach is presented for learning the mappings from the channel vectors to the binary scheduling decisions in D2D wireless networks.

A. Federated Learning for D2D Link Scheduling

For the implementation of the FL based method in D2D networks, each D2D pair is treated as a client. The FL [8] is employed to learn the input-output mapping of the link scheduling problem in the D2D wireless networks, which works as follows. Firstly, the parameters of the global model are randomly generated as the initial state at the server. Hereafter, at each round of the training, a fraction of clients are randomly selected, then the server sends its current model parameters to the selected clients. After this, each of the selected clients trains its local model using its local dataset based on the global parameters, and then transmits the updated local model parameters back to the server which will compute the average values based on the models of the clients and update its global model. The aforementioned steps are repeated until convergence. The FL over a D2D network is illustrated in Fig. 1, where $w_l^t, l \in \mathcal{L}$ and w^t denote the local and the global model parameters of the t-th round, respectively.

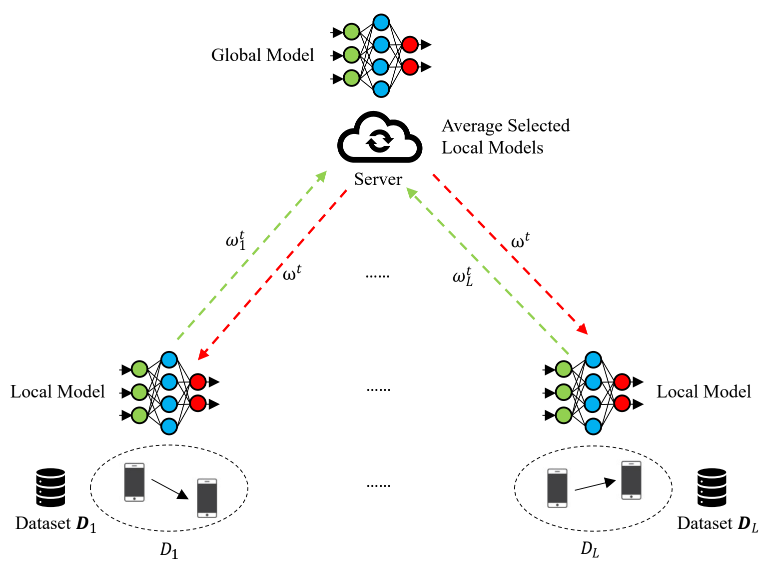


Fig. 1 Illustration of the FL over a D2D wireless network.

In the state-of-the-art works, it is common to assume that the full CSI of the entire D2D networks are available. For each D2D pair, it is difficult to acquire all CSI including the communication channel of itself, the communication channels of other D2D pairs, the interference channels from a particular D2D pair to other devices and vice versa. In this work, it is assumed that each D2D pair knows only its direct channel and the interference channels from itself to other devices. Let $\boldsymbol{h}_l = [h_{l1}, h_{l2}, \ldots, h_{lL}]^T \in \mathbb{R}^{L\times 1}$ denote the channel vector from T_l to R_k, $k \in \mathcal{L}$, and $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_L]^T \in \mathbb{R}^{L\times L}$ denote the channel matrix of a D2D network. For the centralized training, $\{\boldsymbol{H}, \boldsymbol{x}\}$ is a training sample. For the FL based solution, $\{\boldsymbol{h}_l^T, x_l\}$ is a training sample for the client D_l, $\forall l \in \mathcal{L}$, since each client can only access to its local data.

In practical applications, each client collects its local CSI from the real world and generates simulated CSI for other pairs to make up full CSI, and runs the cross-entropy (CE) algorithm [3] to get scheduling decisions. Local training can be conducted on each client and more samples can be collected to join the training. Let M_l denote the total number of samples that the client D_l generated. The local CSI and the corresponding decisions are selected as the local training dataset for D_l, i.e., $\boldsymbol{D}_l = \{(\boldsymbol{h}_l^T)^j, x_l^j\}_{j=1}^{M_l}$. For D_l, $Y_l^j = \{y_{lc}^j\}_{c=0}^1$ denotes the one-hot representation of x_l^j, where $c = \{0, 1\}$ because the D2D scheduling problem can be treated as a binary classification problem. Besides, $y_{l0}^j = 1$ and $y_{l1}^j = 0$ if $x_l^j = 0$, otherwise $y_{l0}^j = 0$ and $y_{l1}^j = 1$. Let $\tilde{Y}_l^j = \{\tilde{y_{lc}}^j\}_{c=0}^1$ denote the activation probabilities generated by the local model for the j-th sample of client D_l. At each client, the cross-entropy loss is adopted as the criterion to measure the distance between the output of the local model and the target. The local model of each client is updated by minimizing this loss function, which is written as follows,

$$\ell_l = -\sum_{j=1}^{M_l}\sum_{c=0}^{1} y_{lc}^j \ln \tilde{y_{lc}}^j. \tag{3}$$

Let B_l denote the batch size for training at the client D_l. The number of local updates can be denoted by $G = M_l/B_l$, which is indexed by g. Let C^t denote the set of the randomly selected ρL clients at the t-th round, where ρ represents the quantile, and $0 < \rho \leqslant 1$. For the t-th round of the FL, the global model parameters w^t are broadcast to the selected clients who will update their local models with $w_{i,g=0}^t = w^t$, $i \in C^t$. Next, each client performs G updates for its local model via gradient descent algorithms, which is written as

$$w_{i,g+1}^t = w_{i,g}^t - \lambda_i^t \nabla \ell_i(w_{i,g}^t, \xi_n^t),\ g = 0, \ldots, G-1, \tag{4}$$

where λ_i denotes the learning rate for updating model at the D_i. $\xi_n \in \boldsymbol{D}_{i,g}$ represents a training sample from the batch g of $\boldsymbol{D}_i$. After the local training at the selected clients, their local model parameters are updated as $w_i^{t+1} = w_{i,G}^t$ and they are sent to the central server where the Federated Averaging (FedAvg) algorithm [8] is performed for aggregation, and the global model is updated as

$$w^{t+1} = \frac{\sum_{i \in C^t} M_i w_i^{t+1}}{\sum_{i \in C^t} M_i}, \tag{5}$$

where M_i denotes the number of training samples of the selected clients $i \in C^t$.

To evaluate our proposed FL based method, a 4-layer feedforward deep neural network (DNN) is employed as the shared model in this work, where a Softmax function is adopted as the activation function at the last layer. Considering successful applications of adaptive moment estimation (ADAM) [11] optimizer in non-federated scenarios [3] [5], it is adopted to update the parameters of local models for faster convergence in this work. For each local model, the input is the concatenation of the communication channel of the D2D pair and the interference channels caused to other pairs. The output layer consists of two neurons which represent probabilities of the binary status of the D2D pair.

B. Data Sharing Analysis

This section provides the analysis of the data sharing of the FL based method and the centralized training. The total number of communication rounds is denoted by R. The number of layers of the shared model is denoted by Q, which is indexed by q. The number of neurons of the q-th layer is represented by E_q. Let M_{C} denote the number of training samples for the centralized learning.

For the FL based solution, the total number of parameters exchanged during the training is $W_{\mathrm{F}} = 2\rho LR(\sum_{q=1}^{Q-1} E_q E_{q+1} + \sum_{q=2}^{Q} E_q) + F\rho LR$ with the negligence of activation layers, where F denotes the number of other parameters that each selected client sends to the server along with the model parameters at each round. In this case, $F = 1$, i.e., the value of M_i, $i \in C^t$. The centralized training requires $W_{\mathrm{C}} = (L^2 + L)M_{\mathrm{C}}$ data to be shared with a central server due to CSI and solution sharing. As observed from the two equations, the overall number of data shared by the FL based approach mainly depends on the dimensions of the shared model and the number of communication rounds, while the number of data shared by the centralized training is related to the network scales and the number of training samples.

IV. Numerical Results

For fairness and convenience, the training data is generated by simulations to compare the performance between different methods. The FL is implemented by the Flower framework [12]. Rayleigh fading channel with zero mean and unit variance is adopted to model the small scale fading. The main system parameters and DNN parameters are given in Table I.

TABLE I System and DNN parameters.

Parameters	Values	Parameters	Values
Edge length	500 m	Path loss model	$148 + 40\log_{10}(d[\text{km}])$
Pairwise distance	[2m, 65m]	Dimensions of DNN	$\{L$+1, 50, 50, 2$\}$
Transmit power of D_l	20 dBm	Learning rate	0.0005
Noise density	-174 dBm/Hz	Batch size	10
Bandwidth	5 MHz	Samples $M_C = M_l$	2000

A. Benchmarks

The proposed FL based method with ADAM optimizer is compared with benchmarks as follows:

- **Cross-entropy (CE) Algorithm [3]:** The performance of this conventional algorithm based on importance sampling serves as an upper bound of the ML based methods in terms of the accuracy and the sum rate. The simulation results are normalized with respect to the CE algorithm to demonstrate the effectiveness of the proposed design.
- **Centralized Training:** This method trains a DNN model at a central server, where the training dataset contains the CSI of all D2D pairs and their scheduling decisions, i.e., $\boldsymbol{D}_{\mathrm{C}} = \{\boldsymbol{H}^j, \boldsymbol{x}^j\}_{j=1}^{M_{\mathrm{C}}}$. For fair comparisons, it adopts the same DNN model as the FL based method, and $M_{\mathrm{C}} = M_l$, $\forall l \in \mathcal{L}$.
- **FL with SGD:** This method simply changes the ADAM optimizer of the FL based approach to the SGD.

Unless specifically stated otherwise, the proposed FL based method adopts the ADAM optimizer in the simulations. The global model is evaluated with 1000 test samples at the server.

B. The Number of D2D Pairs

The performance of the FL based method and the benchmark with $L \in \{5, 10, 15, 20\}$ is presented in Fig. 2.

As indicated in Fig. 2, the FL based method achieves almost identical performance as that of the centralized training method. When L increases from 5 to 20, the accuracy generated by the FL based solution degrades from 89% to 83%, and the sum rate increases from 43 to 103 bits/s/Hz approximately which is close to the conventional CE algorithm, e.g., for $L = 20$, it is around 94% of the CE algorithm.

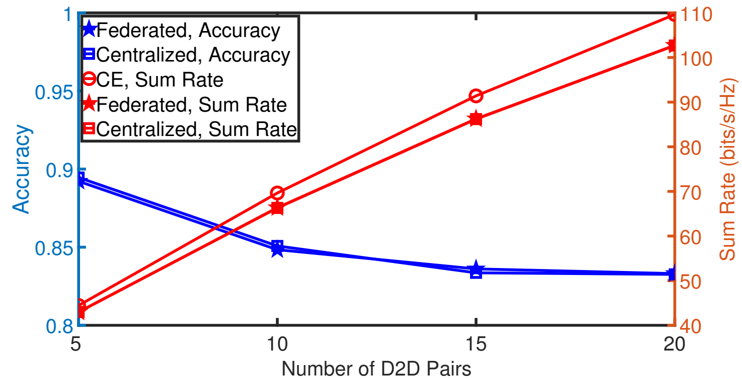


Fig. 2 Performance with different numbers of D2D pairs.

C. Convergence

The convergence performance of the FL based method is compared between the ADAM and the SGD optimizers, and the results are presented in Fig. 3. The learning rates for the FL with the SGD optimizer ranges from 0.0002 to 0.001 for networks with different L for better performance. For the FL based method with the ADAM as shown in Fig. 3a, the global models converge to their best performance within dozens of communication rounds for all the tested cases. As a comparison, the FL based approach with the SGD takes several hundred of rounds to converge as demonstrated in Fig. 3b. In this case, the FL with the ADAM is favorable in applications in wireless networks since it requires less communication rounds between the server and the clients than that with the SGD, thus the former is more efficient.

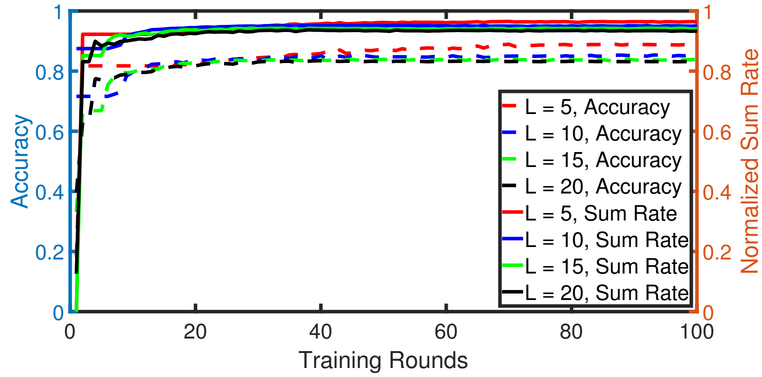
D. The Fraction of Selected Clients

The performance of the FL based method with different proportions of clients sampled randomly at each round is evaluated on $L = 10$ as shown in Table II.

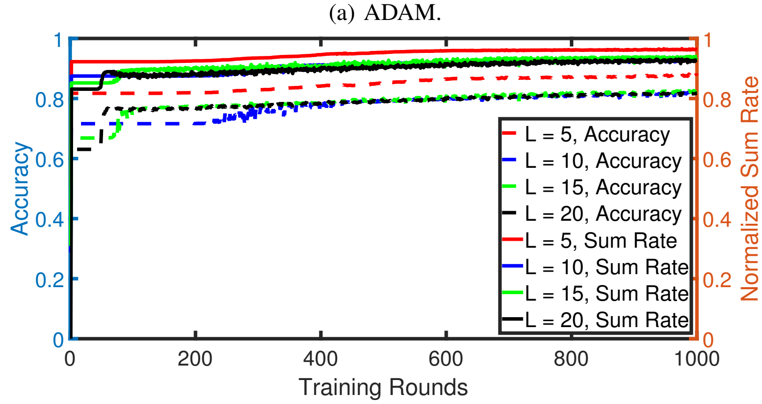
TABLE II Performance with different ρ of selected clients.

ρ	0.2	0.4	0.6	0.8	1
Accuracy	0.8505	0.8520	0.8453	0.8507	0.8484
Sum Rate	0.9519	0.9520	0.9524	0.9524	0.9527

As shown in Table II, when the server sampled different numbers of clients for parameter aggregation during training, the performance remains stable where all tested cases achieve an accuracy of around 0.85 and a sum rate of over 0.95. Consequently, it is recommended to select a small fraction of clients at each round of training for efficiency.



(a) ADAM.



(b) SGD.

Fig. 3 Convergence performance of the FL based method with different optimizers at clients.

E. Running Time

The time performance of the proposed approach is evaluated on the pretrained models with the processor Intel Core i5-9600KF CPU. It is compared to the conventional CE algorithm as shown in Table III, where the time performance of the model trained by the FL approach is ranging from 0.44 μs to 1.23 μs for increasing L from 5 to 20. The proposed approach significantly decreases the running time of the CE algorithm from second level to microsecond level. Therefore, it is a potential candidate for real-time applications in D2D networks.

TABLE III Comparisons of the average running time in μs.

L	5	10	15	20
CE	1.6070×10^5	9.4370×10^5	2.3110×10^6	4.7089×10^6
FL	0.4387	0.9430	0.9456	1.2319

F. Data Sharing Comparison

The equations of the data sharing for the FL based solution and the centralized training are given in Section III. B. Let us assume $M_{\mathrm{C}} = 2000$, $\rho = 0.1$, and $R = 30$. The parameters of the local model shown in Table I are adopted for data sharing calculations. For $L = 20$, $W_{\mathrm{F}} \approx 4.5 \times 10^5$, and $W_{\mathrm{C}} = 8.4 \times 10^5$. In this case, the FL based method reduces nearly half of the data to be shared compared to the centralized training. The FL based approach will achieve greater advantages in some scenarios, e.g., larger system scales, more training samples, and faster convergence. On the opposite, the FL based method cannot outperform the centralized training in some cases, e.g., small scale networks or small training datasets.

V. Conclusion

This work proposes an FL based method to approximate the D2D link scheduling problem with limited CSI. This decentralized approach not only mitigates the computing burden of a central server via local training, but also reduces transmission delay via local decision-making, as well as avoids exposing CSI to eavesdroppers. Simulation results demonstrate that the proposed federated learning method achieves almost the same performance as that of the centralized training. Additionally, the data sharing comparison between the FL and the centralized training has been analyzed. Improving scalability and generalizability will be studied in future works.

References

[1] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1172–1182, Jan. 2014.

[2] K. Shen and W. Yu, "FPLinQ: A cooperative spectrum sharing strategy for device-to-device communications," in *IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, 2017, pp. 2323–2327.

[3] T. Chen, X. Zhang, M. You, G. Zheng, and S. Lambotharan, "A GNN-based supervised learning framework for resource allocation in wireless IoT networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1712–1724, Jun. 2021.

[4] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1248–1261, Mar. 2019.

[5] M. Lee, G. Yu, and G. Y. Li, "Graph embedding based wireless link scheduling with few training samples," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2282–2294, Dec. 2020.

[6] J. D. V. Sánchez, L. Urquiza-Aguiar, M. C. P. Paredes, and D. P. M. Osorio, "Survey on physical layer security for 5g wireless networks," *Annals of Telecommunications*, vol. 76, pp. 155–174, 2021.

[7] H. Tan, N. Xie, and A. X. Liu, "Physical layer security in distributed antenna systems using one-bit feedback information," *IEEE Network*, vol. 36, no. 3, pp. 191–196, Jul. 2022.

[8] 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 the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54. PMLR, Apr. 2017, pp. 1273–1282.

[9] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE transactions on communications*, vol. 68, no. 1, pp. 317–333, Sep. 2019.

[10] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, Feb. 2020.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015, pp. 1–15.

[12] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão *et al.*, "Flower: A friendly federated learning framework," 2022.