Journal of Circuits, Systems, and Computers
Vol. 32, No. 13 (2023) 2350220 (15 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218126623502201

World Scientific
www.worldscientific.com

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

WFLTree: A Spanning Tree Construction for Federated Learning in Wireless Networks*

Shuo Li

School of Mathematics and Statistics,
Shaanxi Normal University, Xi'an,
Shaanxi 710062, P. R. China

School of Mathematics and Data Sciences,
Changji University, Changji,
Xinjiang 831100, P. R. China
shuoli01001@foxmail.com

Yanwei Zheng† and Yifei Zou‡

School of Computer Science and Technology,
Shandong University, Qingdao,
Shandong 266237, P. R. China
†*zhengyw@sdu.edu.cn*
‡*yfzou@sdu.edu.cn*

Received 3 October 2022
Accepted 18 January 2023
Published 23 February 2023

Nowadays, more and more federated learning algorithms have been implemented in edge computing, to provide various customized services for mobile users, which has strongly supported the rapid development of edge intelligence. However, most of them are designed relying on the reliable device-to-device communications, which is not a realistic assumption in the wireless environment. This paper considers a realistic aggregation problem for federated learning in a single-hop wireless network, in which the parameters of machine learning models are aggregated from the learning agents to a parameter server via a wireless channel with physical interference constraint. Assuming that all the learning agents and the parameter server are within a distance Γ from each other, we show that it is possible to construct a spanning tree to connect all the learning agents to the parameter server for federated learning within $O(\log \Gamma)$ time steps. After the spanning tree is constructed, it only takes $O(\log \Gamma)$ time steps to aggregate all the training parameters from the learning agents to the parameter server. Thus, the server

*This paper was recommended by Regional Editor Takuro Sato.
†Corresponding author.

can update its machine learning model once according to the aggregated results. Theoretical analyses and numerical simulations are conducted to show the performance of our algorithm.

Keywords: Federated learning; edge intelligence; wireless networks; distributed computing.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

1. Introduction

The past decades have witnessed a rapid development of machine learning techniques and their wide applications in various scenarios, such as in the smart healthcare system,[1] intelligent transportation system,[2] cyberphysical system,[3–5] and some smart home applications.[6–9] In the machine learning techniques, the most standard approach is the centralized one, in which the training data is kept in a centralized database and trained with a single machine learning model.[10–12] However, this approach has the following concerns: (a) the efficiency of machine learning is limited by the computing resource of the centralized training device. In other words, if the centralized training device is not strong enough, it may take a long time to obtain a well-trained learning model; (b) such a database with all the training data is not available, not only due to the privacy concerns from the users but also because of the limited resources for storage and data transmission. To overcome the weaknesses mentioned above, the federated learning in Ref. 13 has been developed as a promising approach, in which agents collaboratively learn a global machine learning model while keeping the data locally on their devices.[14–18] Specifically, each learning agent holds a local storage for its private data, and trains its learning model from its own data. Then, the parameters of learning model will be aggregated to a parameter server. The parameter server updates the parameters according to the results of aggregation, and disseminates the new updated parameters to all learning agents. By doing this, a distributed learning framework has been implemented, in which the learning model of each agent is trained locally while a global optimization is obtained.

Due to its advanced distributed framework, in recent years, a series of federated learning algorithms have been designed over wireless networks, including those in Refs. 13 and 19–25, most of which rely on the reliable communications between devices. Specifically, the authors in Ref. 20 studied federated optimization in machine learning by considering that each device in the network has only a tiny fraction of the total data available. In Ref. 21, the authors advocated a federated learning method that leaves the training data distributed on the mobile devices, and learns a shared model by aggregating locally computed updates. In Ref. 22, a distributed learning algorithm based on the machine learning framework of deep echo state networks is proposed to predict the orientation and locations of VR users. In Ref. 23, a novel federated learning algorithm is proposed to minimize the communication cost. In Ref. 24, the problem of joint power and resource allocation for ultra-reliable low-latency communication in vehicular networks is studied, by proposing a novel distributed approach based on federated learning. However,

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

when the federated learning algorithms are designed in the wireless domain, most of the previous works have the assumption that the communications are reliable and sometimes the latency is ignored. A survey in Ref. 26 indicates that when the communication is unreliable or fails, the efficiency and correctness of the federated learning will be reduced. The research in Ref. 27 also discusses that federated learning will encounter training errors when the wireless channels are not reliable. In Ref. 28, an analytical model to characterize the effect of packet transmission errors on the performance of federated learning is given. However, the transmission error model is not comprehensive enough to depict the various failures of communications.

In this paper, we consider the parameter aggregation problem from the learning agents to the parameter server in an unreliable wireless network. The key problem to be solved in this paper is how to provide a reliable and efficient parameter aggregation process for federated learning. We answer this problem with a distributed spanning tree construction algorithm, by which all the learning agents can be connected to the parameter server within $O(\log \Gamma)$ time slots and also all parameters can be aggregated to the server within $O(\log \Gamma)$ time slots. Γ is the maximum distance between the agents and the server. The contributions of our work are summarized in the following.

To the best of our knowledge, our paper is one of the first considering the parameter aggregation problem for federated learning in wireless network within an unreliable communication model, especially with physical interference constraints. In our paper, a distributed algorithm is proposed, which can connect all the learning agents to the server by performing a spanning tree construction within $O(\log \Gamma)$ time slots. Additionally, we show that when the spanning tree has been constructed, it takes $O(\log \Gamma)$ time slots to aggregate all parameters of learning agents to the server. In other words, the global machine learning model of the whole federated learning system can be updated within every $O(\log \Gamma)$ time slots, which is an efficient result in distributed and wireless communication environment. Both theoretical analyses and extensive simulations are conducted to show the correctness and efficiency of our algorithm.

Roadmap. In Sec. 2, a detailed federated learning model in wireless channel is given. Our wireless federated learning tree construction and aggregation algorithm is presented in Sec. 3. In Sec. 4, a detailed theoretical analysis is given. The simulated results are presented in Sec. 5. Finally, in Sec. 6, we conclude our paper.

2. Federated Learning Model

We consider such a federated learning system in which n learning agents and a parameter server are arbitrarily deployed in a two-dimensional Euclidean space. Each learning agent has only a tiny fraction of the total data, which will be used to train its local machine learning model. The execution of federated learning algorithm in our system can be divided into successive epochs, each of which contains

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

four stages. The first stage is learning stage, in which each agent trains its machine learning model with its local database; the second stage is aggregation stage, in which all agents aggregate their new parameters of machine learning models to the parameter server; the third stage is the updating stage, in which the server updates its machine learning parameters according to the aggregated results; and the final stage is the dissemination stage. In this stage, the server disseminates the updated parameter to all learning agents. By repeating the epochs for sufficient times, a global optimal machine learning model is obtained on the parameter server and all the learning agents. In the first stage and the third stage, the learning rule and updating rule are determined by the machine learning algorithm itself and may vary in solving different problems. In the final stage, the downlinks from the server to the learning agents can be completed with a global broadcast. In this paper, we focus on the parameter aggregation in the second stage.

To support the parameter aggregation process, stable and fast communications between devices are necessary. However, in a wireless channel, it is not easy to guarantee the reliable and efficient communications because of collision and interference. The communications fail if two packets are transmitted simultaneously, and have the same destination, which is termed as a collision on the receiver. Besides, the packet receipt will fail if the interference at the receiver is very high.

SINR model. We assume that all learning agents and the server have a synchronized time clock, and a time slot is the unit for message transmission. Let W_1 and W_2 be the sets of devices which transmit or listen in a time slot. The devices in the sets W_1 and W_2 are also termed as transmitters and receivers for short, respectively. For each pair of nodes $u \in W_1$ and $v \in W_2$, its SINR value $\mathrm{SINR}(u, v)$ is defined as follows:

$$\mathrm{SINR}(u,v) = \frac{P_u/d^{\alpha}(u,v)}{\sum_{w \in W_1 \setminus \{u\}} P_w/d^{\alpha}(w,v) + N}. \tag{1}$$

In the above definition, P_u and P_w are the transmission powers of devices u and w. $d(u, v)$ and $d(w, v)$ are the Euclidean distances between devices (u, v) and devices (w, v), respectively. α is the path-loss exponent, which is a constant determined by the environment and is usually within $(2, 6]$. The formula $P_u/d^{\alpha}(u, v)$ indicates a fact that when the signal from u arrives at v, its strength gets weaker. For the transmissions from u to v, the signals from all the other transmitters $w \in W_1 \setminus \{u\}$ are regarded as interference. Only when the signal from u is β times larger than the sum of interference plus noise, can the receiver v receive and decode the packet from u. Constant β is a hardware-determined threshold, and N is the ambient noise of environment.

Capability and knowledge of devices. We assume that all the learning agents and the server have the following abilities in transmission: (1) each device is equipped with a half-duplex transceiver so that in each time slot, it can choose to transmit or listen, but cannot do both; (2) the minimum distance between devices is normalized as 1, and the maximum distance is denoted by Γ; (3) all devices have

the maximum transmission power P. From our SINR equation, it is necessary to have $P \geq \Gamma^{\alpha}\beta N$; and (4) each device knows its location information, which can be obtained from a GPS service.

3. Tree Construction and Parameter Aggregation

In this section, we show how a spanning tree is constructed to connect all the learning agents to the parameter server, and to provide efficient parameter aggregations. Considering that if all n learning agents are directly connected to the server, it takes at least n time slots for the server to receive all the parameters of agents because the server can at most receive one message per slot. Thus, it is significant to construct a tree structure to efficiently aggregate the parameters from learning agents. In the following, we show how the tree structure is built layer by layer by our algorithm.

Before the algorithm execution, we grid our network by the square cells of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$. Point $(0;0)$ is the grid origin. Each cell includes its left side without the top endpoint, and its bottom side without the right endpoint, and does not include its right and top sides. We say that $(i;j)$ is the coordinate of the cell with its bottom-left corner located at $(\frac{i}{\sqrt{2}};\frac{j}{\sqrt{2}})$ for integers i and j. A cell with coordinate $(i;j)$ is denoted as $c(i;j)$. Let (x_v, y_v) be the coordinate of node v. Then, $x_v = i$ and $y_v = j$ if agent v locates in the cell $c(i;j)$. For any two nodes u and v, we say that they are in the same window in terms of i if $x_v \mod 2^{i+1} = x_u \mod 2^{i+1}$ and $y_v \mod 2^{i+1} = y_u \mod 2^{i+1}$.

Our tree construction and parameter aggregation algorithm (Algorithm 1) consists of the following two parts: the WFLTree Construction (Algorithm 2) in part one, and the Parameter Aggregation with WFLTree (Algorithm 3) in part two.

Algorithm 1. WFLTree parameter aggregation for agent v

Initialization: $v \in V_0$;

$i = 0$;
while $i \leq \log\Gamma$ **do**
    Tree-Construction(i);
    $i = i + 1$;
end while
$i = 0$;
while $i \leq \log\Gamma$ **do**
    Parameter-Aggregation(i);
    $i = i + 1$;
end while

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

Algorithm 2. Tree-Construction(i) for agent v

if $v \in V_i$ **then**
  j=0;
  while $j \leq c * c$ **do**
    if $j = c * (x_v \mod c) + y_v$ **then**
      Transmit with power P_i;
      $V_{i+1} \leftarrow \{v\}$;
    else
      Listen;
      if receiving a message from u in the same window in terms of i **then**
        $\text{Parent}_v = u$;
        v keeps silent in the remaining slots;
      end if
    end if
    $j \leftarrow j + 1$;
  end while
end if

Algorithm 3. Parameter-Aggregation(i) for agent v

if $v \in V_i$ **then**
  j=0;
  while $j \leq c * c$ **do**
    if $j = c * (x_v \mod c) + y_v$ **then**
      Transmit $\{\text{Parent}_v + \mathcal{M}_v\}$ with power P_i;
    end if
    $j \leftarrow j + 1$;
  end while
end if
if $v \in V_{i+1}$ **then**
  $j = 0$;
  while $j \leq c * c$ **do**
    Listen;
    if receiving $\{\text{Parent}_v + \mathcal{M}_v\}$ from v and $\text{Parent}_v = u$ **then**
      Update its parameter $\mathcal{M}_u$ according to $\mathcal{M}_v$;
    end if
    $j \leftarrow j + 1$;
  end while
end if

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

In the first part, all agents are the leaves of a tree initially, and a spanning tree is constructed layer by layer from the bottom agents. Similarly, in the second part, all parameters are stored in the leaf agents, and will be aggregated layer by layer. The detailed executions of parts one and two are given in the following.

Part one (WFLTree Construction). Initially, all the n learning agents are the leaves of our tree structure, denoted by the set V_0. Our wireless federated learning tree (WFLTree) is constructed layer by layer by executing the function Tree-Construction(i) in Algorithm 1 by setting $i = 0, 1, \ldots, \log \Gamma$. In other words, by executing the function Tree-Construction(0), a set of agents are selected as the parents of agents in set V_0. Those parents constitute the set V_1. Then, the agents in set V_1 execute the function Tree-Construction(i) again to constitute a new set V_2. As proved in the next section, by repeating this process for at most $(\log \Gamma + 1)$ times, a spanning tree is constructed and the final root agent directly connects to the server. We say that only the server is in the set $V_{\log \Gamma + 1}$.

The detailed execution for an agent v in set V_i is given in the following: In the following $c * c$ time slots, v transmits at the jth slot with a transmission power P_i, and listens in the other slots. $j_v = c * (x_v \mod c) + y_v$ and $c = 2 * \lceil [(32 \cdot \frac{\alpha-1}{\alpha-2} + 4) \cdot 2\beta \cdot 2^{\alpha/2}]^{\frac{1}{\alpha}} \rceil + 2$. When v listens, if it receives a message from an agent u in the same window in terms of i, v becomes the child of u and keeps silent in the remaining slots. When v transmits in the slot j, it has the transmission power $P_i = 2N\beta * 2^{i\alpha}$ and becomes a member in the set V_{i+1}.

Part two (Parameter Aggregation with WFLTree). Our WFLTree helps a lot in the parameter aggregation process. Initially, all the parameters are generated by the leaf agents, all of which belong to the set V_0. Then, those parameters are aggregated layer by layer by executing the function Parameter-Aggregation(i) in Algorithm 1 by setting $i = 0, 1, \ldots, \log \Gamma$. Specifically, by executing the Parameter-Aggregation(i) once, those parameters will be aggregated from set V_i to V_{i+1} for $i = 0, 1, \ldots, \log \Gamma$. Since only the server is in the set $V_{\log \Gamma + 1}$, by repeating Algorithm 1 for $(\log \Gamma + 1)$ times, all parameters are aggregated to the server.

The detailed execution for the function Parameter-Aggregation(i) is given in the following, in which v is in the set V_i and u is in the set V_{i+1}. In the following $c * c$ time slots, v transmits at the jth slot with transmission power P_i. $j_v = c * (x_v \mod c) + y_v$, $c = 2 * \lceil [(32 \cdot \frac{\alpha-1}{\alpha-2} + 4) \cdot 2\beta \cdot 2^{\alpha/2}]^{\frac{1}{\alpha}} \rceil + 2$, and its transmitted message includes its parent's ID and its parameter; for the node u in set V_{i+1}, it listens in the following $c * c$ time slots, if it receives a message from agent v, and the parent of v is u, u updates its parameter according to the parameter of node v.

By Algorithm 1, our WFLTree is constructed layer by layer within the time complexity of $O(\log \Gamma)$, and the parameters in agents are aggregated to the server within $O(\log \Gamma)$ time slots. Figure 1 is an example of how our WFLTree is constructed and the parameters are aggregated.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

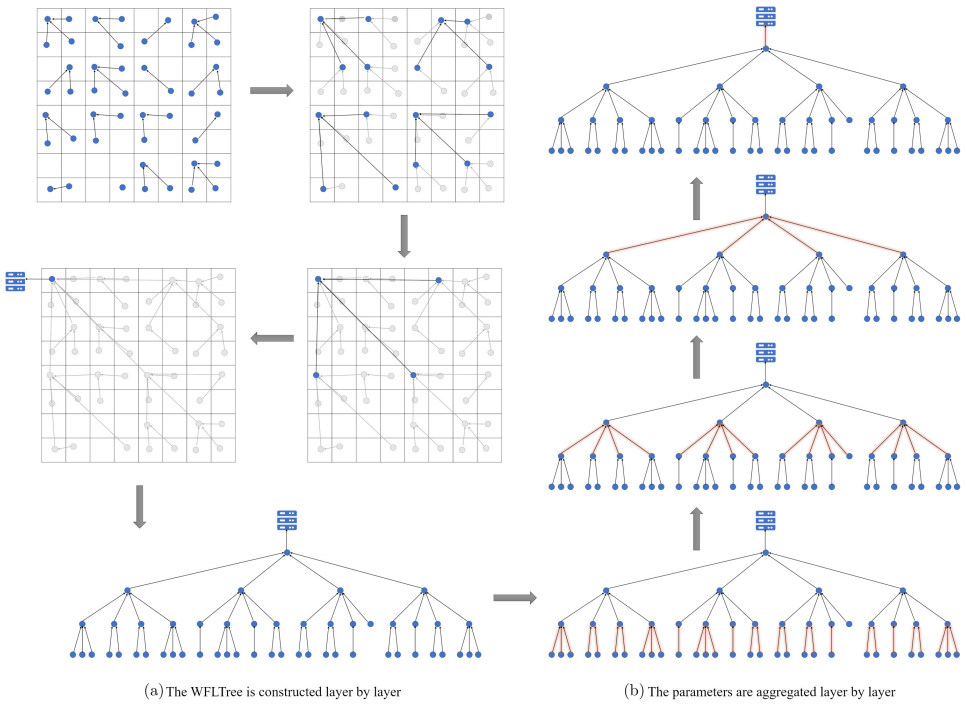

Fig. 1. An example of how our WFLTree is constructed and the parameters are aggregated.

4. Algorithm Analysis

Our analysis section consists of two parts. In the first part (Sec. 4.1), we show the efficiency and correctness of our wireless federated learning tree construction. In the second part (Sec. 4.2), we analyze the time complexity for parameter aggregation.

4.1. *Analysis for WFLTree construction*

We first show the time complexity of our tree construction.

Lemma 1. *The time complexity of our wireless federated learning tree construction is* $O(\log \Gamma)$.

Proof. Our wireless federated learning tree construction was completed by executing the function Tree-Construction(i) for $(\log \Gamma + 1)$ times. Each execution of the function Tree-Construction(i) requires $c * c$ time slots, in which c is a constant. Thus, the time complexity of our wireless federated learning tree construction is $O(\log \Gamma)$. □

In the next, we show that all the agents are connected to the server by our wireless federated learning tree construction. The analysis starts from executing

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

the function Tree-Construction(0). Initially, all agents are in the set V_0. After the execution of the function Tree-Construction(0), some agents move into the set V_1 while the others are left in the set V_0.

Lemma 2. *By executing the function* Tree-Construction(0), *all agents in the set* V_0 *are connected to the agents in the set* V_1 *if* V_0 *is not empty.*

Proof. From our algorithm design, for any node v, it becomes a member in set V_1 if it transmits in slot j_v, or is left in the set V_0 because of receiving a message from u before the slot j_v and u is in the same window in terms of $i = 0$ with v. Thus, let w be any of the agents left in the set V_0, after the execution of Tree-Construction(0). According to the above analysis, w must receive some messages from the agent u before the slot j_w and u is in the same window with w in terms of $i = 0$. In this case, w is connected to the agent u, and u becomes a member in the set V_1 after the slot j_u. □

Lemma 3. *By executing the function* Tree-Construction(0), *for any pair of nodes* u *and* v *in set* V_1, *it is impossible that* u *and* v *are in the same window in terms of* $i = 0$.

Proof. Before the execution of the function Tree-Construction(0), for any pair of nodes u and v in set V_0, it is impossible that u and v are in the same window in terms of $i = -1$. Otherwise, the minimum distance between u and v is smaller than 1, which violates our assumption in Sec. 2. Then, in the execution of the function Tree-Construction(0), for any window in terms of $i = 0$, the agents in that window must transmit in different slots. We prove that for any two agents u and v in a window in terms of $i = 0$, if u transmits, v can receive the message from u in the following claim. □

Claim 1. *For any two agents* u *and* v *in a window in terms of* $i = 0$, *if* u *transmits*, $\mathrm{SINR}(u, v) \geq \beta$.

Proof. According to our SINR model, we have the following equation:

$$\mathrm{SINR}(u, v) = \frac{P_u / d^{\alpha}(u, v)}{\sum_{w \in W_1 \setminus u} P_w / d^{\alpha}(w, v) + N}. \tag{2}$$

According to our transmission scheme, the node v with coordinates $(x_v; y_v)$ will transmit at the j_vth slot, in which $j_v = c * (x_v \mod c) + y_v$. With a similar area argument in Claim 2 of Ref. 31, we can prove that

$$\sum_{w \in W_1 \setminus u} \frac{P_w}{d^{\alpha}(w, v)} \leq \left(32 * \frac{\alpha - 1}{\alpha - 2} + 4\right) * 2N\beta * \frac{2^{\alpha/2}}{(c-1)^{\alpha}}. \tag{3}$$

By setting $c = 2 * \lceil [(32 \cdot \frac{\alpha-1}{\alpha-2} + 4) \cdot 2\beta \cdot 2^{\alpha/2}]^{\frac{1}{\alpha}} \rceil + 2$, we have $\mathrm{SINR}(u, v) \geq \beta$. □

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

Lemma 4. *By executing the function* Tree-Construction(0), *for any node u in the set V_1, it at most has three children in the set V_0.*

Proof. From our algorithm description, we know that an agent v in the set V_0 becomes a member in set V_1 if it transmits in the slot j_v, and is left in set V_0 if it receives a message from an agent in the same window in terms of $i = 0$. Note that the window in terms of $i = -1$ at most contains one agent. Thus, for a window in terms of $i = 0$, it at most contains four agents, in which one will transmit, become the parent of the other agents, and a member of set V_1; and the others will be the children and left in the set V_0. Thus, for any agent in the set V_1, it at most contains three children in the set V_0. □

With the above analysis, we know that (a) in a window in terms of $i = 0$, all agents in that window will transmit in different slots; (b) if u is the only transmitter in its window, all the other agents in the same window will receive it. Combining these two results together, we prove the result that for each window in terms of $i = 0$, it at most contains one agent in the set V_1.

With similar proofs, we can extend the results of Lemmas 2–4 to the more general cases with $i = 1, 2, \ldots, \log \Gamma$.

Lemma 5. *By executing the function* Tree-Construction(i) *with* $i = 0, 1, \ldots, \log \Gamma$:

(a) *all agents in the set V_i are connected to the agents in the set V_{i+1} if V_i is not empty*;
(b) *for any pair of nodes u and v in the set V_i, it is impossible that u and v are in the same window in terms of $i - 1$*;
(c) *for any node in the set V_{i+1}, it at most has three children, which belong to the set V_i.*

By setting $i = 0, 1, \ldots, \log \Gamma$ one by one, we can prove that all the agents in the set V_0 are connected to the agents in the set $V_{\log \Gamma}$. Additionally, from Lemma 5, we know that there is at most one agent in $V_{\log \Gamma}$ that locates in the window with a size of $\frac{\Gamma}{\sqrt{2}} * \frac{\Gamma}{\sqrt{2}}$. In other words, there is at most one agent in the set $V_{\log \Gamma}$. Finally, the only agent in the set $V_{\log \Gamma}$ connects to the parameter server. Until then, we prove the correctness of our WFLTree construction.

4.2. *Analysis for parameter aggregation*

After our WFLTree is constructed, those parameters can be aggregated layer by layer. First, we prove the time complexity of our parameter aggregation process.

Lemma 6. *The time complexity of parameter aggregation is $O(\log \Gamma)$.*

Proof. After our WFLTree is constructed, the parameter aggregation can be completed by executing the function Parameter-Aggregation(i) for $\log \Gamma$ times. Each execution of the function Parameter-Aggregation(i) requires $c * c$ time slots. Thus, the time complexity of the parameter aggregation is $O(\log \Gamma)$. □

In the next, we show the correctness of parameter aggregation. Initially, all the parameters are stored in the agents that belong to the set V_0.

Lemma 7. *After executing the function* Parameter-Aggregation(0), *all parameters are aggregated to the agents that belong to the set* V_1.

Proof. From our WFLTree construction, each node u that belongs to the set V_0 has a parent v in set V_1. When the function Parameter-Aggregation(0) is executed, v transmits in the slot j_v. With a similar proof in Claim 1, we show that $\text{SINR}(u, v) \geq \beta$, i.e., the parameter of v is aggregated to the agent u, which is the parent of v in the set V_1. □

With a similar analysis, we can extend the above result to the cases where $i = 1, 2, \ldots, \log \Gamma$. Finally, when $i = \Gamma$, all parameters are aggregated to the parameter server.

5. Simulation Results

In this section, we investigate the performance of our algorithm when the network parameters vary. Specifically, we focus on the running time of WFLTree construction, parameter aggregation, and total running time of the algorithm when the maximum distance of agents Γ, the number of nodes n, and the SINR parameters α, β vary. Note that we have already proved the correctness of our algorithm and showed that the time complexities for the WFLTree construction, parameter aggregation, and total algorithm are $O(\log \Gamma)$. In the following, we show the numerical running time of our algorithm in simulation. The maximum distance of agents Γ and the number of nodes n indicate the scope and size of the federated learning network, and the SINR parameters α, β represent the wireless communication environment.

Parameter setting. In simulation, we randomly and uniformly deploy n learning agents into a square area with a size of $\frac{\Gamma}{\sqrt{2}} \times \frac{\Gamma}{\sqrt{2}}$. The minimum distance between agents is 1 and the maximum distance Γ varies within $[100, 300]$; the number of agents varies within $[1000, 5000]$; and the SINR parameters are set as $\alpha = 3, \beta = 1.5$, $\alpha = 4, \beta = 1.5$, $\alpha = 3, \beta = 2$, and $\alpha = 4, \beta = 2$, respectively, to simulate different communication environments. Additionally, we set $c = 10$ and normalize $N = 1$. Our simulation is conducted in a Linux machine with Intel Xeon CPU E5-2670@2.60 GHz and 128-GB main memory, implemented in C++ programming language. For each reported result, the executions are repeated for over 20 times.

The performance of our algorithm. In this part, we investigate the running time of WFLTree construction, parameter aggregation, and total algorithm when

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

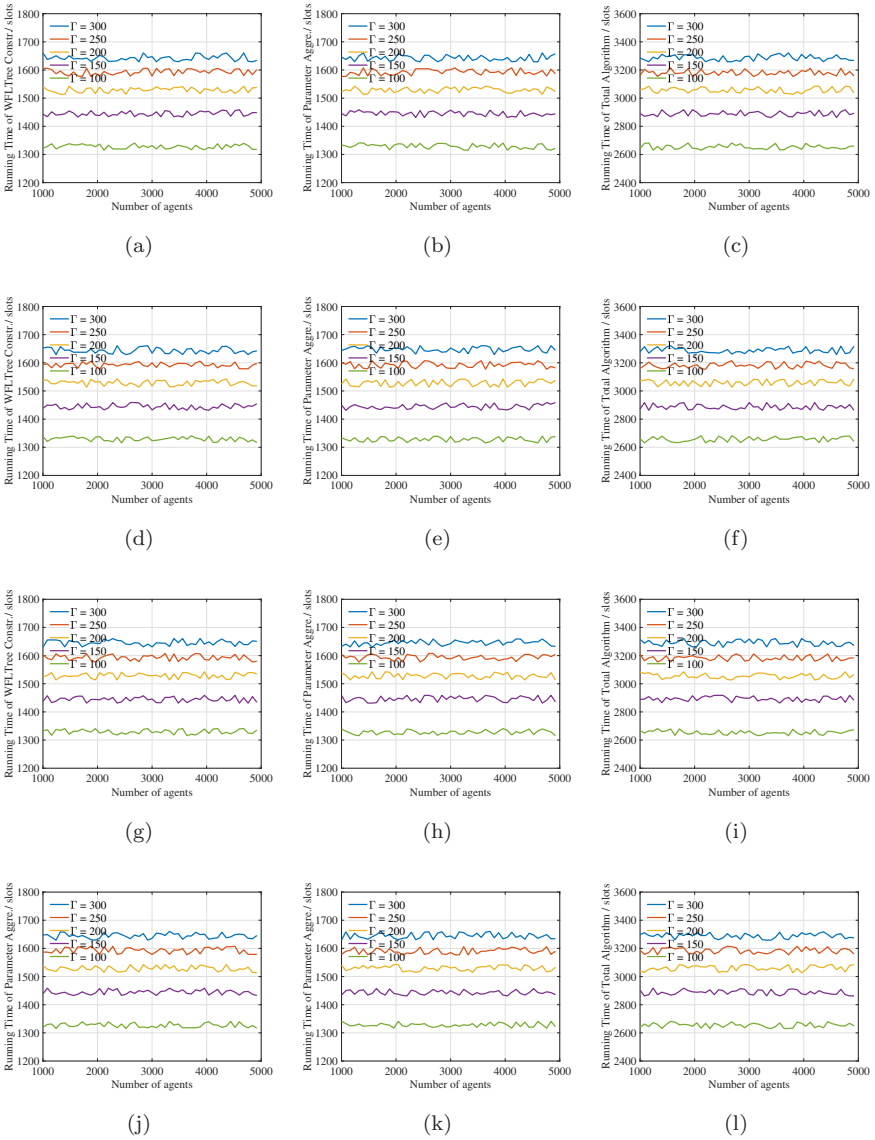

Fig. 2. The running times of WFLTree construction, parameter aggregation, and total algorithm when the maximum distance Γ, the number of agents n, and the SINR parameters α and β vary. (a) WFLTree construction with $\alpha = 3$, $\beta = 1.5$. (b) Parameter aggregation with $\alpha = 3$, $\beta = 1.5$. (c) Total running times with $\alpha = 3$, $\beta = 1.5$. (d) WFLTree construction with $\alpha = 4$, $\beta = 1.5$. (e) Parameter aggregation with $\alpha = 4$, $\beta = 1.5$. (f) Total running times with $\alpha = 4$, $\beta = 1.5$. (g) WFLTree construction with $\alpha = 3$, $\beta = 2$. (h) Parameter aggregation with $\alpha = 3$, $\beta = 2$. (i) Total running times with $\alpha = 3$, $\beta = 2$. (j) WFLTree construction with $\alpha = 4$, $\beta = 2$. (k) Parameter aggregation with $\alpha = 4$, $\beta = 2$. (l) Total running times with $\alpha = 4$, $\beta = 2$.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

the maximum distance $\Gamma \in [100, 300]$, the number of agents $n \in [1000, 5000]$, and the SINR parameters $\alpha = 3, 4$ and $\beta = 1.5, 2$. Specifically, all the curves are presented in Fig. 2, in which the x-axis and y-axis represent the number of agents and the number of slots. From the curves of Fig. 2(a) with $\alpha = 3$ and $\beta = 1.5$, we can see that the running time of WFLTree construction keeps stable when the number of agents n varies. However, it increases from $1,350$ to $1,650$ when Γ gets larger from 100 to 300. Besides, according to our theoretical result for $O(\log \Gamma)$, the constant behind O is about 200.

Figures 2(b) and 2(c) show the running times of parameter aggregation and total algorithm when $\alpha = 3$ and $\beta = 1.5$, from which we can see a similar tendency with the conclusions in WFLTree construction. Specifically, the running times of parameter aggregation and total algorithm get larger when Γ is larger, and keep stable for different n. Their constants behind O are about 200 and 400, respectively. Additionally, by comparing the running times of WFLTree construction, parameter aggregation, and total algorithm with different SINR parameters, we can see that they are nearly the same when α and β vary. In other words, our algorithm is insensitive to communication environment in terms of running time.

6. Conclusion

Different from the previous wireless federated learning works that are based on the reliable device-to-device communications, this paper considers the parameter aggregation problem from n learning agents to a parameter server under an unreliable wireless environment. Then, a wireless federated learning tree, termed as WFLTree, is constructed within $O(\log \Gamma)$ time slots to connect all the agents to the parameter server, in which Γ is the maximum distance between agents. Based on our WFLTree, the parameters from n agents can be aggregated to the parameter server within $O(\log \Gamma)$ time slots. Our work provides much faster result ($\log \Gamma$) compared with n slots, which is the lower bound if all learning agents are directly connected to the parameter server. Extending our research to mobile environments will be our work in the future.

Acknowledgments

This work is supported by the University Scientific Research Program Foundation of Xinjiang Province (Grant No. XJEDU2021I025).

References

1. Z. Liu, Y. Chen, Y. Zhao, H. Yu, Y. Liu, R. Bao, J. Jiang, Z. Nie, Q. Xu and Q. Yang, Contribution-aware federated learning for smart healthcare, *Proc. 36th AAAI Conf. Artificial Intelligence* (2022).
2. X. Yuan, J. Chen, N. Zhang, X. Fang and D. Liu, A federated bidirectional connection broad learning scheme for secure data sharing in Internet of Vehicles, *China Commun.* **18** (2021) 117–133.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.

3. Z. Cai and X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, *IEEE Trans. Netw. Sci. Eng.* **7** (2020) 766–775.
4. H. Zhang and S. Huang, EAT-ML: Efficient automatic tuning for machine learning models in cyber physical system, *J. Circuits Syst. Comput.* **30** (2021) 2150245.
5. J. S. Jayaprakash, M. J. P. Priyadarsini, P. B. Divakarachari, H. R. Karimi and S. Gurumoorthy, Deep q-network with reinforcement learning for fault detection in cyber-physical systems, *J. Circuits Syst. Comput.* **31** (2022) 2250158.
6. P. Ravichandran, C. Saravanakumar, J. D. Rose, M. Vijayakumar and V. M. Lakshmi, Efficient multilevel federated compressed reinforcement learning of smart homes using deep learning methods, *Proc. 2021 Int. Conf. Innovative Computing, Intelligent Communication and Smart Electrical Systems* (*ICSES*) (2021).
7. O. Aouedi, K. Piamrat, G. Muller and K. Singh, FLUIDS: Federated learning with semi-supervised approach for intrusion detection system, *Proc. 2022 IEEE 19th Annu. Consumer Communications & Networking Conf.* (*CCNC*) (2022).
8. Z. Cai and T. Shi, Distributed query processing in the edge-assisted IoT data monitoring system, *IEEE Internet Things J.* **8** (2021) 12679–12693.
9. C. Sun, L. Ji and H. Zhong, Speech emotion recognition on small sample learning by hybrid WGAN-LSTM networks, *J. Circuits Syst. Comput.* **31** (2022) 2250073.
10. M. Chen, U. Challita, W. Saad, C. Yin and M. Debbah, Artificial neural networks-based machine learning for wireless networks: A tutorial, *IEEE Commun. Surv. Tutor.* **21** (2019) 3039–3071.
11. Y. Sun, M. Peng, Y. Zhou, Y. Huang and S. Mao, Application of machine learning in wireless networks: Key techniques and open issues, *IEEE Commun. Surv. Tutor.* **21** (2019) 3072–3108.
12. Y. Liu, S. Bi, Z. Shi and L. Hanzo, When machine learning meets big data: A wireless communication perspective, *IEEE Veh. Technol. Mag.* **15** (2020) 63–72.
13. K. A. Bonawitz *et al.*, Towards federated learning at scale: System design, *Proc. 2nd SysML Conf.* (2019).
14. V. Smith, C. Chiang, M. Sanjabi and A. Talwalkar, Federated multi-task learning, *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates, 2017), pp. 4424–4434.
15. X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen and M. Chen, In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning, *IEEE Netw.* **33** (2019) 156–165.
16. W. Saad, M. Bennis and M. Chen, A vision of 6G wireless systems: Applications, trends, technologies, and open research problems, *IEEE Netw.* **34** (2020) 134–142.
17. E. Jeong, S. Oh, J. Park, H. Kim, M. Bennis and S. Kim, Multi-hop federated private data augmentation with sample compression, preprint, arXiv:1907.06426 [cs.LG] (2019).
18. Z. Yang, M. Chen, W. Saad, C. S. Hong and M. Shikh-Bahaei, Energy efficient federated learning over wireless communication networks, *IEEE Trans. Wirel. Commun.* **20** (2021) 1935–1949.
19. T. Li, A. K. Sahu, A. Talwalkar and V. Smith, Federated learning: Challenges, methods, and future directions, *IEEE Signal Process. Mag.* **37** (2020) 50–60.
20. J. Konečný, H. B. McMahan, D. Ramage and P. Peter Richtárik, Federated optimization: Distributed machine learning for on-device intelligence, preprint, arXiv:1610.02527 [cs.LG] (2016).
21. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, *Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS)* (2017).

22. M. Chen, O. Semiari, W. Saad, X. Liu and C. Yin, Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks, *IEEE Trans. Wirel. Commun.* **19** (2020) 177–191.
23. J. Konečný, B. McMahan and D. Ramage, Federated optimization: Distributed optimization beyond the datacenter, preprint, arXiv:1511.03575 [cs.LG] (2015).
24. S. Samarakoon, M. Bennis, W. Saad and M. Debbah, Distributed federated learning for ultra-reliable low-latency vehicular communications, *IEEE Trans. Commun.* **68** (2020) 1146–1159.
25. 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. Sel. Areas Commun.* **37** (2019) 1205–1221.
26. M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan and H. V. Poor, Distributed learning in wireless networks: Recent progress and future challenges, *IEEE J. Sel. Areas Commun.* **39** (2021) 3579–3605.
27. J. Park, S. Samarakoon, M. Bennis and M. Debbah, Wireless network intelligence at the edge, *Proc. IEEE* **107** (2019) 2204–2239.
28. H. H. Yang, Z. Liu, T. Q. S. Quek and H. V. Poor, Scheduling policies for federated learning in wireless networks, *IEEE Trans. Commun.* **68** (2020) 317–333.
29. D. Yu, Y. Zou, J. Yu, Y. Zhang, F. Li, X. Cheng, F. Dressler and F. C. M. Lau, Implementing the abstract MAC layer in dynamic networks, *IEEE Trans. Mob. Comput.* **20** (2021) 1832–1845.
30. D. Yu, Y. Zou, Y. Zhang, H. Sheng, W. Lv and X. Cheng, An exact implementation of the abstract MAC layer via carrier sensing in dynamic networks, *IEEE/ACM Trans. Netw.* **29** (2021) 994–1007.
31. L. Yang, Y. Zou, M. Xu, Y. Xu, D. Yu and X. Cheng, Distributed consensus for blockchains in Internet-of-Things networks, *Tsinghua Sci. Technol.* **27** (2022) 817–831.

J CIRCUIT SYST COMP Downloaded from www.worldscientific.com
by PEKING UNIVERSITY on 03/19/23. Re-use and distribution is strictly not permitted, except for Open Access articles.