Adaptive Federated Learning for Digital Twin Driven Industrial Internet of Things

Qiang Song*, Shiyu Lei*, Wen Sun†, Yan Zhang‡
*School of Cyber Engineering, Xidian University, Xi'an, 710126, China.
†School of Cybersecurity, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an Shaanxi, 710072, China.
‡University of Oslo, Oslo, Norway, and also with the Simula Metropolitan Center for Digital Engineering, Oslo, Norway.
Email: *qiangsong@stu.xidian.edu.cn

2021 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-7281-9505-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/WCNC49053.2021.9417370

***Abstract*—Industrial Internet of Things (IoT) enables distributed intelligent services varying with the complex industrial environment to achieve the benefits of Industry 4.0. In this paper, we consider a new architecture of digital twin empowered Industrial IoT, in which digital twins capture characteristics of industrial devices to assist federated learning tasks of industrial scenarios. A trust-based aggregation is proposed in federated learning to alleviate the effects of digital twins deviation and emphasize the contribution of high-performance clients. Based on Lyapunov dynamic deficit queue and deep reinforcement learning, we propose a federated learning framework that adaptively adjusts the aggregation frequency to improve the learning performance under resource constraints. Numerical results show that the proposed framework outperforms the benchmark in terms of learning accuracy, convergence, and energy saving.**

***Index Terms*—digital twin, federated learning, learning efficiency, communication efficiency.**

I. Introduction

Industrial Internet of Things (IoT), as an extension use of IoT, interconnects numerous industrial devices, analytical sections, and people at work. Through exquisite cooperation of machinery devices, Industrial IoT enables automatic manufacturing of applications ranging from automotive to agriculture, and finally realizes Industry 4.0 [1]. The success of Industrial IoT hinges on dynamic perception and intelligent decision, which is difficult to capture due to the heterogeneous Industrial IoT devices and the complex industrial environment. [2] As a key technology for the intelligent and digitization of the Industrial IoT, digital twins (DTs) support the precise modeling and synchronous updates of industrial equipment. [3] It realizes real-time interaction, analysis, learning and prediction based on actual data. In reality, decisions in the Industrial IoT usually require a large amount of data to support. [4] However, security and privacy issues caused by the distributed characteristics of data in the Industrial IoT pose challenges to the application of DTs [5].

Recently, the advantages of federated learning have provided new possibilities for solving Industrial IoT privacy protection [6] and data security [7]. Federated learning can build a distributed machine learning system under the conditions of multi-party participation and data confidentiality. These features enable federated learning to ensure privacy while reducing communication costs. Therefore, when it comes to issues such as privacy protection, regulatory requirements, data islands, and expensive or unreliable connections, federated learning is a viable solution. [8]

The current research work on the influencing factors of the federated learning aggregation process focuses on data size, computing power, communication cost, reputation value, etc. [9-12]. *Nishio et al.* [13] comprehensively consider the impact of heterogeneity in data scale, computing capabilities, and communication channels of mobile devices on aggregation and convergence when resources are limited. *Pandey et al.* [14] measure the reliability of equipment through communication efficiency. In order to improve the learning effect, *Lu et al.* [15] introduce an asynchronous learning framework, which accelerates the convergence speed of learning. However, the existing works are not applicable in Industrial IoT as they ignore the dynamic and complex industrial environment. In federated learning, with the increase of aggregation rounds, the gain obtained by the global model is nonlinear, and the network environment is time-varying, which means that the design of the architecture of federated learning in Industrial IoT needs to consider the frequency and timing of aggregation.

To address above issues, we study adaptive calibration of the global aggregation frequency to improve training efficiency under resource constraints. We introduce DTs of Industrial IoT to map the real-time state and operation of physical entities to the digital world. By considering the DT deviation in the trust-based aggregation strategy, the contribution of devices to the global model of federated learning is quantified. Based on *Deep Q Network* (*DQN*), we develop an adaptive calibration of global aggregation frequency to minimize the loss function of federated learning under a given resource budge, enabling a dynamic tradeoff between energy consumption and communication energy in changing communication environments.

The rest of this paper is organized as follows. Section II introduces DT-based system model. Section III formulates DQN-based federated learning scheme. In Section IV, numerical results show that the proposed framework outperforms the benchmark in terms of learning accuracy, convergence, and energy saving. Finally, conclusions are drawn in Section V.

II. System Model

A. DTs for Industrial IoT

As shown in Fig. 1, we introduce a DT empowered Industrial IoT, consisting of various types of industrial devices,

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:28:12 UTC from IEEE Xplore. Restrictions apply.

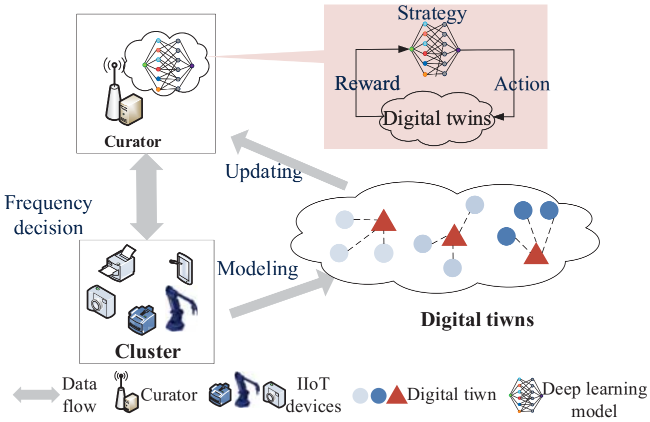

Fig. 1: DTs for federated learning in a heterogeneous Industrial IoT scenario.

curator servers, and DTs of industrial devices. Devices with limited communication and computing resources establish communication with servers through wireless communication links. DTs map the physical status of heterogeneous devices into digital space and update them in real time.

The DT of an Industrial device is established by the server it belongs to, where the historical and current behavior of the device are dynamically presented in a digital form by collecting and processing the existing key physical state of the device. Within time t, the DT of the training node i can be expressed as

$$DT_i(t) = \{F(w_i^t), f_i(t), E_i(t)\}, \tag{1}$$

where w_i^t is the current trained parameter of the node i, $F(w_i^t)$ represents the current training state of the node i, $f_i(t)$ is the current computational capability of the node i, and $E_i(t)$ indicates the energy consumption.

Note that there is a deviation between the mapped value of DT and the actual value. We use the CPU frequency deviation $\hat{f}_i(t)$ to represent the deviation between the actual value of the device and its DT mapping value, and $\hat{E}_i(t)$ to represent energy consumption deviation. Therefore, the DT model after calibration can be expressed as

$$\hat{DT}_i(t) = \{F(w_i^t), f_i(t) + \hat{f}_i(t), E_i(t) + \hat{E}_i(t)\}. \tag{2}$$

DTs can receive the physical state data of the device and conduct self-calibration according to the empirical deviation value, maintaining consistency with the device and feeding back information in real time, to realize dynamic optimization of the physical world and enhance the ability of trend perception and decision-making.

B. Federated learning in Industrial IoT

In Industrial IoT, industrial devices (excavators, sensors, monitors, etc.), located in different geographical locations or even owned by different companies, need collaboratively complete production tasks based on federated learning. As shown in Fig. 1, an excavator with sensors collects a large amount of production data and is in a real-time monitoring environment. Through collaboration of clients to perform federated learning and intelligent analysis, system can make better decisions on quality control and predictive maintenance, without transmitting a large amount of generated data collected by sensors.

Task initialization, in which the curator broadcasts the task and the initialized global model w_0, is the first step of federated learning. Next, after receiving w_0, the training node i uses its data D_i to update the local model parameters w_i^t to find the optimal model parameter that minimizes the loss function

$$F(w_i^t) = \frac{1}{D_i} \sum_{x_j, y_j \in D_i} f(w_i^t, x_i, y_i), \tag{3}$$

where t denotes the current local iteration index, $f(w_i^t, x_i, y_i)$ quantifies the difference between estimated and true values for instances of running data D_i, and x_i, y_i are the samples of training data. After T rounds of local training, the updated local model parameters are sent to the curator, where T is a preset frequency. Then, the curator is responsible for the global model aggregation to obtain the parameters of the k-th aggregation w_k according to the preset aggregation strategy (see Subsection III-C). The loss value after the k-th global aggregation is $F(w_k)$. After global aggregation, the curator broadcasts the updated global model parameter back to each training node. Local model training and global model aggregation need to be repeated until the global loss function converges or the model reaches the preset accuracy.

C. Trust-based Aggregation in DT-driven Industrial IoT

In federated learning in Industrial IoT, to improve the learning performance and resist malicious attacks, the parameters uploaded by clients with high reputation should have greater weight in the aggregation. Unlike the traditional reputation model that only considers security threats, the effects of DT deviation, learning effect, and dataset quality on learning is comprehensively considered.

Note that DTs have inevitable deviations in the mapping of node states in terms of CPU frequency $\hat{f}_i$, and the mapping deviations for different clients are different. The parameters uploaded by clients with low mapping deviation should occupy more updated weight in the aggregation. In addition, in a *Byzantine attack*, a malicious client user may provide the curator with incorrect or low-quality updates to abate the accuracy of the global model. Therefore, learning quality and interaction records are introduced to count for malicious updates to weaken the threat of malicious clients. Based on the subjective logic model, the belief of the curator j for the node i in the time slot t can be expressed as

$$b_{i\to j}^t = (1 - u_{i\to j}^t)\hat{f}_i(t) q_{i\to j}^t \frac{\alpha_i^t}{\alpha_i^t + \beta_i^t}, \tag{4}$$

where $u_{i\to j}^t$ represents the failure probability of packet transmission, $\hat{f}_i(t)$ indicates the DT deviation of the curator j to the node i, $q_{i\to j}^t = \frac{w_i^t - \bar{w}}{\sum_{i=1}^n w_i^t - \bar{w}}$ denotes quality of learning based

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:28:12 UTC from IEEE Xplore. Restrictions apply.

on the honesty of most training devices, α_i^t is the number of positive interactions, and β_i^t is the number of malicious actions such as uploading lazy data. Specifically, the curator uses $FoolsGold$ scheme that identifies unreliable clients according to the gradient update diversity of local model updates in non-IID federated learning in which the training data of each node has a unique distribution. The reputation value of the curator i for the node j is expressed as

$$T_{i\to j} = \sum_{t=1}^{T} b_{i\to j}^t + \iota u_{i\to j}^t, \tag{5}$$

where $\iota \in [0,1]$ is a coefficient indicating the degree of uncertainty affecting the reputation. In the global aggregation, the curator retrieves the updated reputation value and aggregates the local model w_i^k of participating clients into a weighted global model, i.e.,

$$w_k = \frac{\sum_{i=1}^{N_d}\sum_{t=1}^{T} T_{i\to j} w_i^t}{\sum_{i=1}^{N_d} T_{i\to j}}, \tag{6}$$

where w_k represents the global parameter after the k-th global aggregation, N_d is the number of training device. Through such a trust-based aggregation, the deviation of DT is considered and the security threats caused by malicious participants are effectively resisted, which can increase the learning convergence rate while enhanceing the robustness of the framework.

D. Energy Consumption Model in Federated Learning

In federated learning, the energy consumption of a training node is composed of the computational energy consumed by local training and the communication energy consumed by global aggregation. The computational energy consumed by training node i to perform a round of training are expressed as

$$E^{cmp} = n_{cmp} F / f_i, \tag{7}$$

where F is defined as the CPU frequency required for a round of training and n_{cmp} is the normalization factor of the consumed computing resources. To formulate the communication energy consumed, M is used to represent the number of bits of the neural network. In order to transmit the gradient to the curator, all training clients share uplink sub-channels based on orthogonal frequency division multiple access (OFDMA), which is expressed as a set $\mathcal{C} = \{1, 2, ..., C\}$. Therefore, there is no co-channel interference between training clients. After collecting gradients from all training clients, the curator broadcasts the global average gradient to training clients on all sub-channels. The communication resources consumed by training node i performing an aggregation are expressed as

$$E^{com} = \frac{n_{com} M}{\sum_{c=1}^{C} l_{i,f} W log_2(1 + \frac{p_{i,c} h_{i,c}}{I})}, \tag{8}$$

where $l_{i,f}$ represents the time fraction allocated by the training node i on sub-channel f, W denotes the sub-channel bandwidth, $p_{i,f}$ is the uplink transmission power of the training node i on sub-channel f, h is the uplink channel power gain between the training node i and the curator, I is the noise power and n_{com} is the normalization factor of the consumed communication resources.

III. Deep Reinforcement Learning for Aggregation Frequency Based on Digital Twin

A. Problem Formulation

The objective of this paper is determining the best tradeoff between local update and global parameter aggregation in a time-varying communication environment to minimize the loss function within a given resource budget. The aggregation frequency problem can be formulated as

$$\mathbf{P1}: \underset{\{a_0, a_1, ..., a_k\}}{arg\min}\ F(w_k) \tag{9}$$

$$s.t. \sum_{i=1}^{n} a_i E^{cmp} + k E^{com} \le \beta R_m \tag{9a}$$

where w_k represents the global parameter after the k-th global aggregation, $F(w_k)$ is the loss value after the k-th global aggregation, $\{a_0, a_1, ..., a_k\}$ is a set of strategies for the frequency of local updates, a_i indicates the number of local updates required for the i-th global update. Constraint (9a) represents the given budget on available resources, and β represents the upper limit of the resource consumption rate in the entire learning process. In Eqn. (9) and Eqn. (9a), the loss value $F(w_k)$ and the computational energy consumption E^{cmp} include the training state $F(w_i^t)$ and computational capability f_i, respectively, which is estimated by DTs to enable the curator to grasp the key status of the entire federated learning. The deviation in computational energy consumption E^{cmp} caused by the mapping deviation of DT in the computational capability of the node is calibrated by trust-based aggregation.

B. Problem Simplification

Due to the long-term resource budget constraints, if the energy consumption of the current aggregation is too high, the node will not be able to participate in aggregation in the future. In addition, **P1** is a nonlinear programming problem, and the complexity of solving it increases exponentially with the increase of federated learning rounds. Therefore, it is necessary to simplify **P1** and the long-term resource budget constraints.

The effect of training after k global aggregations can be written as

$$F(w_0) - F(w_k) = \sum_{i=1}^{k} [F(w_{i-1}) - F(w_i)]. \tag{10}$$

For optimal training results, i.e.,

$$\min[F(w_k) - F(w_0)] = \max \sum_{i=1}^{k} [F(w_{i-1}) - F(w_i)]. \tag{11}$$

Based on the *Lyapunov optimization*, a dynamic resource deficit queue is established to simplify **P1** by dividing the long-term resource budget into available resource budget for each time slot. The difference between the resources used and

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:28:12 UTC from IEEE Xplore. Restrictions apply.

the resources available is the definition of the length of the resource deficit queue. R_m is the limit of the total resource, and $\beta R_m/k$ is the resource available in the k-th aggregation. The evolution of the resource deficit queue is as follows

$$Q_{i+1} = \max\{Q_i + (a_i E^{cmp} + E^{com}) - \beta R_m/k, 0\}, \quad (12)$$

where $(a_i E^{cmp} + E^{com}) - \beta R_m/k$ is the deviation of resources in the k-th aggregation. According to the above Eqn. (8) and Eqn. (11), the original problem **P1** can be transformed into

$$\mathbf{P2}: \underset{\{a_0,a_1,\dots,a_k\}}{arg\max} \sum_{i=1}^{k} [v(F(w_{i-1}) - F(w_i)) - Q_i(a_i E^{cmp} + E^{com})] \quad (13)$$

$$s.t. \sum_{i=1}^{n} a_i E^{cmp} + kE^{com} \le \beta R_m \quad (13a)$$

where v and Q_i are positive control parameters that dynamically balance training performance and resource consumption. It should be noted that the accuracy of federated learning can be easily improved at the beginning of the training, while it is costly to improve the accuracy at the later stage of training. Therefore, v increases with the increase of training rounds and is motivated towards the goal of maximizing the ultimate benefits.

C. Deep reinforcement learning (DRL) Model

DQN, which is a successful application of Deep reinforcement learning (DRL) and is well known through AlphaGo, is used to solve the frequency problem of local updates. DT learns models by interacting with the environment, without pre-training data and model hypothesis. The optimization problem is formulated as a DRL denoted by $\mathcal{M}$, which includes the system state $\mathcal{S}(t)$, action space $\mathcal{A}(t)$, policy $\mathcal{P}$, reward function $\mathcal{R}$ and next state $\mathcal{S}(t+1)$. The detailed description of parameters is as follows:

- **System State** The system state illustrates the characteristics and training state of each node, including the current training state of all clients $\varsigma(t)$, the current state of the resource deficit queue Q_i, and the average value output from the hidden layer of the neural network of each node $\tau(t)$, i.e.,

$$\mathcal{S}(t) = \{\varsigma(t), Q_i, \tau(t), \mathcal{A}(t-1)\}. \quad (14)$$

- **Action Space** The action is defined by a vector $\mathcal{A}(t) = \{a_1^t, a_2^t, ..., a_n^t\}$ where a_i^t indicates the number of local updates, which need to be discretized. For simplicity, we use a_i to denote a_i^t since subsequent statements are based on a specific time t.
- **Policy** Policy $\mathcal{P}$ is the mapping of state space to action space, i.e., $\mathcal{A}(t) = \mathcal{P}\{\mathcal{S}(t)\}$. $\mathcal{P}$ is used to ensure that the model is credible.
- **Reward Function** It is noted that our goal is to determine the best tradeoff between local update and global parameter aggregation to minimize the loss function, the reward function is naturally related to the decline of the overall loss function and the state of the resource loss queue. The action is assessed by

$$\mathcal{R} = [v(F(w_{i-1}) - F(w_i))] - Q_i(a_i E^{cmp} + E^{com}). \quad (15)$$

- **Next State** The current state $\mathcal{S}(t)$ is provided by the DT real-time mapping, and the next state $\mathcal{S}(t+1)$ is the prediction obtained by the DT without actually running DQN in the physical world, which can be expressed as $\mathcal{S}(t+1) = \mathcal{S}(t) + \mathcal{P}(\mathcal{S}(t))$.

Algorithm 1: Adaptive calibration of the global aggregation frequency.

Input: eval_net O, target_net O', update frequency F_u of target_net parameters, greed coefficient ϵ, greed coefficient growth rate r;
Output: The parameters of the trained DQNs;

```
Randomly initialize the parameters of evaluation eval_net O and target_net O' ;
for each episode do
    Initialize the parameters in environment setup ;
    for each time slot t do
        select a_i^t = max_A O(S(t), A; w_i) with probability ε and select a random action a_i^t with probability 1 − ε;
        Perform federated learning training;
        Calculate immediate reward R with Eqn. (15) and update the system state S(t + 1);
        Store the experience tuples (S, A, R, S(t + 1));
        if the experience relay is full then
            if t%F_u = 1 then
                Update the target_net parameters
            end
            Learning all samples from the experience relay;
            Calculate q-eval value by eval_net;
            Calculate q-target value according to Eqn. (17);
            Perform a gradient descent step according to Eqn. (18);
        end
    end
end
return The parameters of the trained DQNs.
```

D. DQN-based Optimization Algorithm for Aggregation Frequency

To solve the MDP problem, we use *DQN*-based optimization algorithm. As shown in Fig. 1, in the Industrial IoT environment, all aspects of the physical objects are mapped to the virtual space by DTs in real time, forming a digital mirror image. At the same time, the DRL agent interacts with the DTs of the devices to learn the global aggregation frequency decision. The Federated learning module makes frequency decisions based on the trained model and the DT status of

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:28:12 UTC from IEEE Xplore. Restrictions apply.

the training clients. Through DTs, the agent achieves the same training effect as the real environment at a lower cost.

1) Training Step: When using *DQN* to achieve adaptive calibration of the global aggregation frequency, initial training samples are allocated to the training clients, and initial parameters are set for the target_net and the eval_net to maintain their consistency. The state array consists of the initial resource value and the corresponding loss value obtained by training each node. In each iteration, the state of the experience relay needs to be judged. If it is full, the action is selected with probability according to the $\epsilon-$greedy strategy, otherwise the action is randomly selected with probability ϵ.

After the action is selected, the reward is calculated according to Eqn. (15) and the system status is updated. Next, the current state, selected actions, rewards, and the next state are recorded in the experience relay. Then we sample from the experience relay to train target_net, which randomly interrupts the correlation between the states by randomly sampling several samples in the experience replay as a batch. By extracting the state, the eval_net parameters are updated according to the loss function as follows:

$$F(w_i) = \mathbb{E}_{\mathcal{S},\mathcal{A}}[y_i - O(\mathcal{S},\mathcal{A};w_i)^2], \tag{16}$$

where $O(\mathcal{S},\mathcal{A},w_i)$ represents the output of the current network eval_net, and y_i is the q-target value calculated according to the parameters of the target_net, which is independent of the parameters in the current network structure. The target_net is used to calculate the q-target value according to the following formula:

$$y_i = \mathbb{E}_{\mathcal{S},\mathcal{A}}[r + \gamma \max_{\mathcal{A}'} O(\mathcal{S}',\mathcal{A}';w_{i-1})|\mathcal{S},\mathcal{A}], \tag{17}$$

where $\{\mathcal{S}',\mathcal{A}'\}$ is the sample from the experience relay, $O(\mathcal{S}',\mathcal{A}',w_{i-1})$ represents the output of target_net. In this way, the entire objective function can be optimized by the stochastic gradient descent method:

$$\begin{aligned}\nabla_{w_i} F(w_i) =& \mathbb{E}_{\mathcal{S},\mathcal{A}}[(r + \gamma \max_{\mathcal{A}'} O(\mathcal{S}',\mathcal{A}';w_{i-1}) \\ &- O(\mathcal{S},\mathcal{A};w_i))\nabla_{w_i} O(\mathcal{S},\mathcal{A};w_i)].\end{aligned} \tag{18}$$

After a certain number of iterations, eval_net parameters need to be copied to target_net. Namely, the updates of loss and target_net are performed in time intervals and experience replay is updated in real time. Repeat the above steps until the loss value reaches the preset value. The complete frequency algorithm for global update is presented in **Algorithm 1**.

2) Running Step: After the training is completed, the proposed frequency decision agent is deployed on the curator, and the adaptive frequency calibration is performed according to the DTs of devices. Firstly, DTs provides the state of the training clients and channels as inputs to the trained *DQNs*. Then output action probability distribution is obtained through the eval_net, and the appropriate action is selected as the execution action according to the $\epsilon-$greedy strategy. The selected action is then executed in federated learning, and the received environmental feedback value is stored in the experience replay to facilitate the agent's retraining.

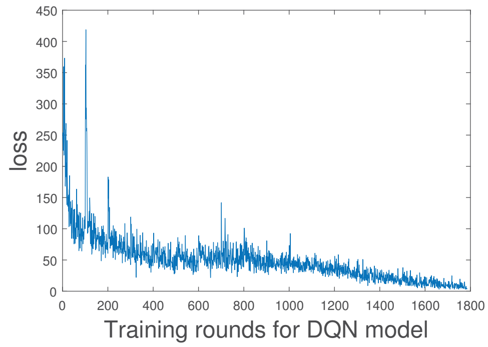

Fig. 2: The convergence performance of DQN.

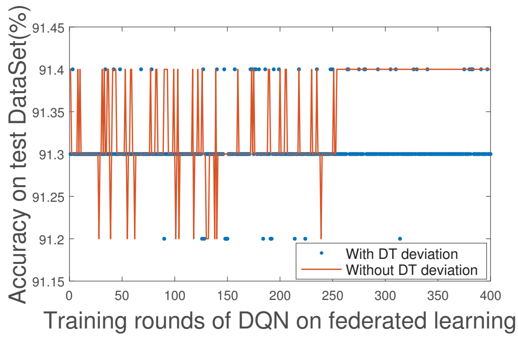

Fig. 3: Comparison of DT deviation and without DT deviation.

IV. Numerical Results

In this section, we show the performance of the proposed *DQN* federated learning scheme and compare it with benchmark scheme under different environment settings. We apply the proposed scheme to the widely used image dataset MNIST, and implement the simulation of federated learning and *DQN* in PyTorch. We employ two identical neural networks to initialized *DQN*, where the each network consists of three fully connected layers deployed in sequence with the size of $48 \times 200 \times 10$. The fixed aggregation frequency scheme is chosen as the benchmark scheme to illustrate the performance of the proposed scheme.

Fig. 2 depicts the convergence trend of the loss function of *DQN*. It can be seen that the value of the loss function decreases rapidly and then tends to be stable with the training rounds increasing, and converges after 400 rounds. Therefore, the proposed *DQN* is feasible for federated learning in Industrial IoT, which can effectively reduce the energy consumption of federated learning scenarios and improve the learning effect.

Fig. 3 compares the federated learning accuracy that can be achieved in the presence of DT deviation and after calibrating DT deviation. The federated learning with DT deviation calibration based on the trust weighted aggregation strategy can reach better learning performance than the federated learning with the DT deviation. When these two algorithms have not converged, the federated learning with calibration deviation

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:28:12 UTC from IEEE Xplore. Restrictions apply.

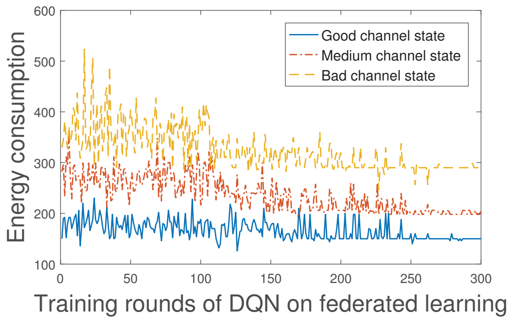

Fig. 4: Comparison of energy consumed by federated learning in different channel states.

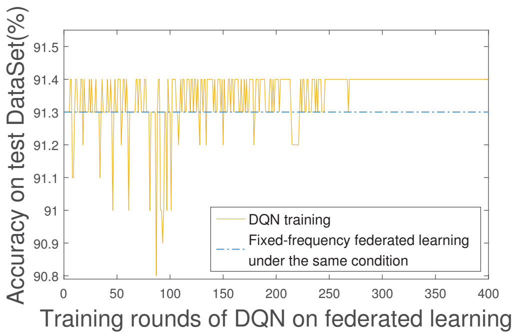

Fig. 5: Comparison of the accuracy achieved of federated learning between adaptive frequency and fixed frequency.

is also superior. Moreover, it is observed that DQN with DT deviation cannot converge.

Fig. 4 compares the energy consumed by DQN-based federated learning in different channel state. With the improvement of the wireless channel state, the energy consumption of clients decreases. It is noted that the communication energy consumption of clients increases, when the channel state decays. Since DQN scheme can adaptively adjust the aggregation frequency, federated learning will aggregate in rounds with good learning effect and low energy consumption. Therefore, with the convergence of DQN, the energy consumption of the three channel states gradually decreases.

Fig. 5 compares the accuracy by DQN-based federated learning with fixed frequency federated learning. When DQN converges gradually, the accuracy of DQN-based federated learning is significantly higher than the fix one. This is because the gain of the global aggregation to the federated learning accuracy is non-linear and the fixed frequency scheme may does not necessarily aggregate in rounds with less energy consumption and better learning effect.

V. Conclusions

In this paper, we have studied adaptive federated learning for digital twin driven Industrial IoT. Thanks to the DT that can agilely capture the divergence of the time-varying network, the proposed scheme can adaptively adjust the aggregation frequency of the federated learning with the channel state changing. The numerical results show that the proposed scheme outperforms the benchmark scheme in terms of learning accuracy, convergence effect and energy saving.

Acknowledgment

This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 824019, in part by the National Natural Science Foundation of China (61771373), in part by Special Funds for Central Universities Construction of World-Class Universities (Disciplines), and in part by China 111 Project (B16037).

References

[1] Y. Lu, X. Huang, K. Zhang, S. Maharjan and Y. Zhang, "Communication-Efficient Federated Learning for Digital Twin Edge Networks in Industrial IoT," in IEEE Transactions on Industrial Informatics.

[2] W. Sun, H. Zhang, R. Wang and Y. Zhang, "Reducing Offloading Latency for Digital Twin Edge Networks in 6G," in IEEE Transactions on Vehicular Technology.

[3] Y. Lu, X. Huang, K. Zhang, S. Maharjan and Y. Zhang,"Low-latency Federated Learning and Blockchain for Edge Association in Digital Twin empowered 6G Networks," in IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2020.3017668 .

[4] W. Sun, S. Lei, L. Wang, Z. Liu and Y. Zhang, "Adaptive Federated Learning and Digital Twin for Industrial Internet of Things," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2020.3034674.

[5] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen and H. Yu, "Federated Learning," in Federated Learning, Morgan & Claypool, 2019.

[6] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang and D. I. Kim, "Incentive Design for Efficient Federated Learning in Mobile Networks: A Contract Theory Approach," 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS), Singapore, 2019, pp. 1-5.

[7] Z. Chen, P. Tian, W. Liao and W. Yu, "Zero Knowledge Clustering Based Adversarial Mitigation in Heterogeneous Federated Learning," in IEEE Transactions on Network Science and Engineering, PP(99):1-1.

[8] F. Sattler, S. Wiedemann, K. Müller and W. Samek, "Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data," in IEEE Transactions on Neural Networks and Learning Systems, 2020.

[9] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep Reinforcement Learning for Stochastic Computation Offloading in Digital Twin Networks", IEEE Transactions on Industrial Informatics.

[10] F. Ang, L. Chen, N. Zhao, Y. Chen, W. Wang and F. R. Yu, "Robust Federated Learning With Noisy Communication," in IEEE Transactions on Communications, vol. 68, no. 6, pp. 3452-3464, June 2020.

[11] S. Savazzi, M. Nicoli and V. Rampa, "Federated Learning With Cooperating Devices: A Consensus Approach for Massive IoT Networks," in IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4641-4654, May 2020.

[12] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient Federated Learning and Permissioned Blockchain for Digital Twin Edge Networks", IEEE Internal of Things, DOI: 10.1109/JIOT.2020.3015772.

[13] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-7.

[14] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor and C. S. Hong, "A Crowdsourcing Framework for On-Device Federated Learning," in IEEE Transactions on Wireless Communications, vol. 19, no. 5, pp. 3241-3256, May 2020.

[15] Y. Lu, X. Huang, Y. Dai, S. Maharjan and Y. Zhang, "Differentially Private Asynchronous Federated Learning for Mobile Edge Computing in Urban Informatics," in IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 2134-2143, March 2020.

Authorized licensed use limited to: Temple University. Downloaded on August 10,2021 at 21:28:12 UTC from IEEE Xplore. Restrictions apply.