

Communication-Efficient Federated Learning for Digital Twin Systems of Industrial Internet of Things

Yunming Zhao*, Li Li ****, Ying Liu**, Yuxi Fan*, Kuo-Yi Lin****

*College of Electronic and Information Engineering, Tongji University, Shanghai, China 201804 (e-mail: 2011604@tongji.edu.cn, lili@tongji.edu.cn, 1830749@tongji.edu.cn, 19603@tongji.edu.cn).

** Department of Mechanical Engineering, School of Engineering, Cardiff University, Cardiff CF24 3AA, UK. (e-mail: LiuY81@Cardiff.ac.uk).

***Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai China 201804, (e-mail: lili@tongji.edu.cn, 19603@tongji.edu.cn).

Abstract: With the rapid development and deployment of Industrial Internet of Things technology, it promotes interconnection and edge applications in smart manufacturing. However, challenges remain, such as yet-to-improve communication efficiency and trade-offs between computing power and energy consumption, which limits the application and further development of IIoT technology. This paper proposes the digital twin systems into the IIoT to build model between physical objects and digital virtual systems to optimize the structure of IIoT. And we further introduce federal learning to train the digital twins model and to improve the communication efficiency of IIoT. In this paper, we first establish the digital twins model of IIoT based on industrial scenario. Moreover, to optimize the communication overhead allocation problem, this paper proposes an improved communication-efficient distribution algorithm, which speeds up the training performance of federated model and ensures the performance of industrial system model by changing the update training mode of client and server and allowing some industrial equipment to participate in federated training. This paper simulates the real-word intelligent camera detection to validate the proposed method. Comparing our proposed method with the existing traditional methods, the results show the advantages of the proposed method can improve the communication performance of the training model.

Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Federated Learning, Digital Twins, Industrial Internet of Things, Communication-Efficient, Intelligent Manufacturing.

1. INTRODUCTION

Industry 4.0 was first proposed by Germany to use information technology to promote the transformation and upgrading of productivity and reach the era of intelligent (King and Grobbelaar, 2020). However, the development of Industry 4.0 is inseparable from the application of technologies such as Cyber-Physical Systems, Internet of Things, and Big Data (Pereira and Romero 2017). Industry 4.0 has also promoted the emergence of the Industrial Internet of Things (IIoT). The IIoT, as a new type of industrial technology that combines a new generation of information technology with industrial production, fully connects the operating dynamics of a series of industrial equipment, products, analysis modules, and practitioners, and then constructs an industrial-grade intelligent manufacturing system (Jeschke et al., 2017). The IIoT integrates sensors and controllers with perception and monitoring capabilities through different communication methods and advanced learning and analysis technologies, so as to greatly improve manufacturing efficiency, product quality, and resource costs in traditional industries. Therefore, the research of IIoT technology needs to make full use of technologies such as digital twins, machine learning, and intelligent manufacturing to better realize the intelligence and interconnection of

industries, and finally realize industry 4.0 (Song et al., 2021). However, the implementation of the IIoT relies to a large extent on dynamic perception and intelligent decision-making in industrial scenarios. In actual situations, it is difficult to obtain information about different industrial equipment and complex environments. The digital twin technology provides a new solution to the problems encountered by the IIoT. Digital twin technology (Ma et al., 2019) can support the model establishment of the actual system and the synchronous update of the status information of the industrial equipment. Using digital twins, an established virtual system can perform real-time update and interaction, and complete the analysis and planning of industrial system equipment based on the actual input data.

The digital twin systems established under the typical IIoT scenario, as shown in Figure 1, mainly includes the following three layers: user layer, edge computing layer, and digital twin layer (Lu et al., 2020). The user layer mainly includes industrial equipment, products, and industrial workers, which are used for data generation and input at different industrial levels. The edge computing layer mainly includes a series of edge computing servers. The use of such devices can accelerate the processing of industrial data, achieve the results of low latency, efficient sensing operation, and data

security protection, and save the energy of mobile devices (Wang et al., 2019). The digital twin layer includes real industrial systems and digital twin systems, and these two systems interact in real time through inputting the status information of equipment and products. However, in the IIoT, a large amount of equipment data is needed to complete the analysis and learning tasks of the industrial system. In the industrial production of different industries, as the security and privacy protection of industrial data are particularly important, it is particularly difficult to establish an effective digital twin technology based on the IIoT technology.

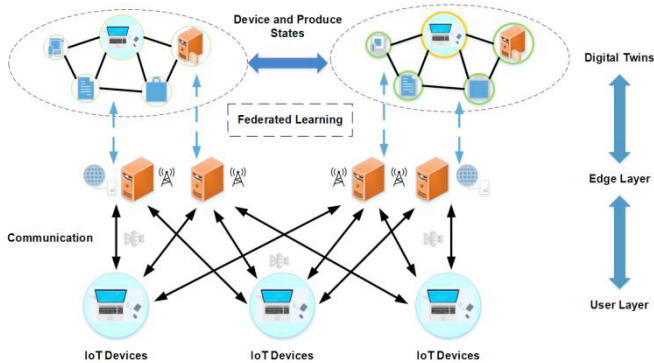


Figure 1. Digital Twin Systems Model (Lu et al., 2020)

Federated learning can establish a distributed learning model without requiring local data access to multiple devices (McMahan et al., 2017). This avoids the data leakage of industrial data to a certain extent. However, when using federated learning for learning and training, the circular communication between the server and the client is required. There are a large number of communication processes. Generally, in actual industrial situations, there are a large amount of industrial data, such as temperature, humidity, real-time data of equipment, and other parameters such as product manufacturing. It has multiple physical quantities, multiple scales, and multiple dimensions (Lade et al., 2017). And the complexity of the selected training model is much higher than the learning model of the general system. In addition, industrial systems include multi-dimensional information such as production equipment, manufactured products, and process knowledge. The distributions of generated industrial data do not have consistent characteristics. In the actual industrial digital twin system, it faces the limitation of communication and computing resources (Gao et al., 2021), that is, the communication delay between different industrial equipment is inconsistent, the generation time of the different industrial data does not have synchronization consistency, and then a good synchronization state cannot be achieved. Therefore, the communication and computing costs of the training system are further increased. For the above practical challenges, we propose a communication-efficient distribution algorithm to ensure the performance of industrial system model by changing the update training mode of client and server and allowing some industrial equipment to participate in federated training.

The rest of the work is organized as follows: section 2 presents the related studies on federated learning of the IIoT. Section 3 sets federal learning model established in the digital

twin systems. Section 4 develops communication-efficient distribution algorithm. Section 5 conducts the experiments and performance comparisons. Finally, Section 6 summarizes the conclusions of the work.

2. RELATED STUDIES

Research on the IIoT based on federated learning has attracted the attention of researchers from different industries. The author used federated learning for proactive content caching in edge computing, which alleviated the difficulty of processing large amounts of data in mobile devices (Yu et al., 2018). Lu et al. solve the data security problem in the IIoT by introducing the federal learning of privacy protection, and provides a distributed industrial equipment data sharing mechanism (Lu et al., 2020). Tran et al. use federated learning to solve the optimal problem of communication energy consumption and training cost under wireless network (Tran et al., 2019). Other research work (Chen et al., 2020, Triastcyn et al., 2019 and Lu et al., 2020) also focuses on user data privacy and the leakage of model parameter under federated learning, and designs reasonable security privacy algorithms.

However, in the IIoT system, federated learning faces the huge consumption of communication-cost. For example, when model training is performed on different industrial data, the obtained model parameters occupy more memory, and the required communication capacity and communication frequency will be increase. In order to get a good-performance model, the many iterations times is required, and the number of communications between the client and the server will increase. Therefore, many researchers begin to focus on how to reduce the communication-cost of the federated learning. The research work by McMahan et al. is considered to be a pioneering work in federated learning, which improves communication efficiency by increasing the local computation time of each participant client in each global communication round (McMahan et al., 2017). In addition, they also propose to increase the parallelism degree to increase the speed of federated training. Nashio and Yonetani developed the FedCS framework for mobile edge devices, which maximize the integration of available clients in each global epoch to achieve high efficiency in the federated learning (Nashio and Yonetani, 2019). Zhu proposed a sparse training mode through the sketch update method, so that only some particular parameters need to be uploaded to the server to reduce the volume of transmitted data, and further reducing the communication overhead (Zhu and Jin, 2020).

Moreover, changing the aggregation training mode between the client and the server can also reduce the communication-cost of the federated learning system. In terms of training mode, Sprague et al. combined the asynchronous aggregation method with the FedAvg algorithm based on the asynchronous gradient descent to obtain the good training results and verify the effectiveness of local asynchronous training in federated learning (Sprague et al., 2019). Wu et al. proposed a semi-asynchronous protocol that allows each participating client to participate in training at a moderate time and accelerate the training convergence of the federated

model by using the information of delayed local model (Wu et al., 2021).

3. FEDERATED LEARNING SYSTEM MODEL

This section will introduce in detail the federal learning model established in the digital twin systems of IIoT, and explain the IIoT digital twins model, communication model and federated learning model of IIoT respectively.

3.1 IIoT Digital Twins Model

This paper establishes the system model of the typical industrial digital twins system described in Figure 1. Assuming that the client of the user layer is $U = \{u_1, u_2, \dots, u_n\}$, and each client device holds its own dataset, and can be denoted as $D = \{D_1, D_2, \dots, D_n\}$. And its sample capacity is represented as $N = \{n_1, n_2, \dots, n_n\}$. Mobile edge servers are distributed in the edge computing layer. They can communicate with the client equipment of the user layer through the wireless network and carry out a series of data processing tasks with low latency and high computing performance. Let the edge computing layer be $S = \{S_1, S_2, \dots, S_m\}$. In this paper, the edge computing layer can be used to construct a digital twin system of the real physical model. Let each client u_i performs a digital twins model through the closest edge server be expressed as DT_i , which specifically includes the following information: the client's local training model M_i , the historical production data D_i , the operation state s_i , and interaction status with the client us_i . Then, at time t , the digital twin model of client k is recorded as:

$$DT_i(t) = [M_i, s_i, us_i] \quad (1)$$

In equation (1), the client needs to use the generated operation data for training and learning to obtain the local model M_i , and the operation state of each client also needs to continuously interact with its DT_i to maintain the consistent convergence of the system. Generally, the generated digital twin system of one client can also interact with the other clients.

3.2 Federated Learning Model of IIoT

The main goal of federated learning is to train the data on distributed industrial equipment to obtain the learning model of the entire system M . Let the single data sample in the client be (x_i, y_i) , and the loss function of the training model be $f(w, x_i, y_i)$, which defines the difference value between the real label and the predicted result of train sample. $F_i(w)$ is defined as the model loss function of the whole data set in client i , which is the average value of the sum of the loss functions of all samples in client i , can be expressed as:

$$F_i(w) = \frac{1}{n_i} \sum_{(x_i, y_i) \in D_i} f(w, x_i, y_i) \quad (2)$$

The total number of data samples for all clients is $N = \sum_{i=1}^n n_i$, and the loss function value after the aggregation of the entire training model can be calculated as:

$$F_g(w) = \frac{1}{N} \sum_{i=1}^n F_i(w) \quad (3)$$

Therefore, the optimal training goal of the entire federation learning system is:

$$\min_w F_g(w) \quad (4)$$

In federated learning, all clients in an industrial system have the same machine learning model w . Each client participating in the training will update the global model according to the local data, and obtain the local model of client i on the edge computing server $w_i(t)$:

$$w_i(t) = w(t-1) - \eta \nabla F_i(w(t-1)) \quad (5)$$

Where, η is the learning rate, and $w(t-1)$ is the global model parameter at time $t-1$. In order to obtain the global model of federated learning, we need to select an edge computing server as the aggregator to globally aggregate the models of all the clients participating in the training. And the resulting global model is:

$$w(t) = \frac{1}{N} \sum_{i=1}^n n_i w_i(t) \quad (6)$$

Therefore, the optimal federated learning training model can be obtained by repeatedly training the optimization formula (6).

4. COMMUNICATION-EFFICIENT FEDERATED LEARNING DISTRIBUTION ALGORITHM

In order to reduce the communication overhead of real industrial systems and establish the excellent-performance digital twin systems, this section will propose a communication-efficient distribution algorithm to obtain the best federated learning training model.

In the existing communication distribution algorithm FedAvg, federated learning needs to follow the synchronization training mode of all clients, that is, the local model of each participating client uploads to the aggregation server after completing the local training, and then performs federated aggregation update task. However, in the real industrial system, when iterative training is carried out in this way, the following problems will be encountered. 1) There are the different communication states in the industrial system. 2) As the federated learning follows the client-server training mode, there are many upload and download communication processes between client and server. For the above practical problems, this paper proposes an efficient communication distribution algorithm by improving the federal training mode and client participation.

4.1 Preliminaries

The FedAvg algorithm is a communication algorithm that accelerates and optimizes distributed equipment resources composed of a parameter server and multiple working nodes. The FedAvg algorithm is a communication algorithm that accelerates and optimizes distributed equipment resources composed of a parameter server and multiple working nodes. In order to reduce the communication overhead of the federated system, when the FedAvg algorithm is used for training, the working node will perform multiple iterations of local model training on the local device, and then upload the

local model to the parameter server. In this way, the communication-cost of upload between the client and the server can be reduced. In the industrial system, the industrial equipment client i conducts more times local model parameter training on the edge computing server side close to client i , that is, it will execute the formula (7) k times:

$$w_i^k(t) = w_i^k(t-1) - \eta \nabla F_i(w_i^k(t-1)) \quad (7)$$

When $t = 0$, the local model parameters of industrial equipment can be calculated: $w_i^0(t) = w(t-1)$.

4.2 Partial participation of industrial equipment clients

This article combines the real industrial system to improve the efficiency of the federal system by selecting only some proportion of industrial equipment for participate in model training, thereby can reduce the upload and download communication number between the client and the server, and making full use of the limited computing power of the edge computing server.

In the proposed method, we set C to be the selected proportion of clients to participate in training at each global epoch, E to be the iteration times when one client performs local training, and B to be the sample batches randomly selected when one client performs local training. Therefore, for a client including n_k data samples, the local model update times $u_k = E \frac{n_k}{B}$ at one global epoch. However, for the communication status of real industrial equipment, as comprehensively considering the convergence of global aggregation, we also set an aggregation parameter G , which represents the selected proportion of clients that need to be uploaded during global aggregation training, and needs to satisfy that G is less than or equal to C . In this way, when the aggregator collects the local models of G clients, it can perform global aggregation tasks without waiting for the training of $C-G$ local models to set a certain tolerance degree for federated learning.

4.3 Semi-synchronous update mode

In a real industrial environment, since it is easy to increase the unnecessary waiting time of the client and server by using the synchronous training update mode in the federated learning, the proposed algorithm introduces the semi-synchronous training update mode. In the industrial digital twin system model, the aggregation server does not need to wait for the client equipment with slow training speed when aggregating the local model. This method not only sets a certain tolerance degree, but also introduces a lag parameter τ and builds a cloud buffer. In federated training, when the delayed time of local training of selected clients does not exceed τ , the delay local model will be stored in the cloud buffer. In a new round global training, it can be moderately added to the global model aggregation task without additional communication cost. But when the delayed time of local training of selected clients exceeds τ , the local model of the client will be discarded to prevent the stale model of selected clients from negatively affecting the convergence of the global model. The implementation process of selecting the client and participating in the global training is shown in Figure 2.

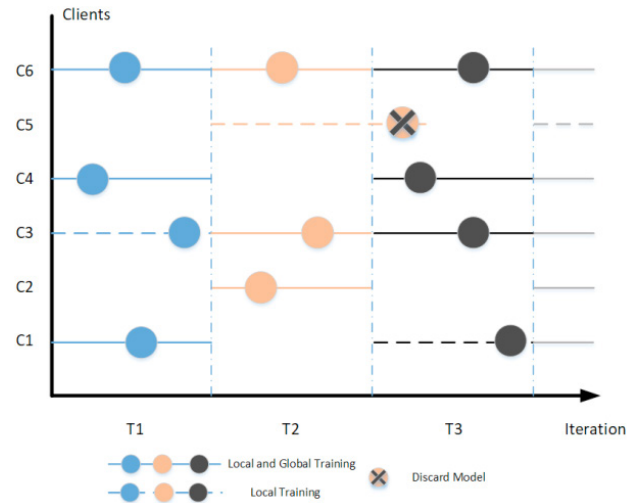


Figure 2. Update process of Clients

5. EXPERIMENTS

This section will conduct the experiments to evaluate the learning performance and communication-cost of the proposed method. We set the data distribution and experimental configuration, and compare the results of different methods.

5.1 Data Sets and Models

In this experiment, we proposed the MNIST dataset(Yan 1998) to verify the proposed algorithm. As a typical public pattern recognition dataset, the MNIST dataset can simulate intelligent image generated by different real industrial camera. The MNIST includes a total of 60,000 training samples and 10,000 test samples. Combined with the industrial situation, we use the MNIST data set as data generated by some industrial equipment, such as industrial cameras, or the input data of industrial workers. Therefore, in this experiment, we set the number of industrial equipment to 60, and randomly set the communication status of industrial equipment. In addition, we select an edge computing server as the aggregation server device. When distributing the data, we randomly shuffle the training samples and distribute them to 60 devices. The specific number of samples on each industrial device is different, which satisfies the Gaussian distribution that the mean value is 1000, and variance value is 50. The Multilayer Perceptron (MLP) model is selected as the training model. The MLP classification model trained on the federated learning client is composed of three hidden layers, including 20, 10 and 5 units respectively, and uses ReLu function as the activation function. Softmax is set as the output layer.

5.2 Experimental Configuration

In this experiment, we determine that all clients use the same model parameter, such as mini-batch sample size B and local training iteration times E . In order to weigh the influence between the local training time of the client and the training accuracy of the global model, this paper selects the number of local iterations of the client E to 5 and the mini-batch sample size B to 50. In addition, the termination condition of global model training is that the global training times reach 50, or

the training results can reach the predefined accuracy and training loss.

Firstly, we determine the values of the model parameters C and G . By selecting different values of C and G , the training results curve of C with G is shown in Figure 3.

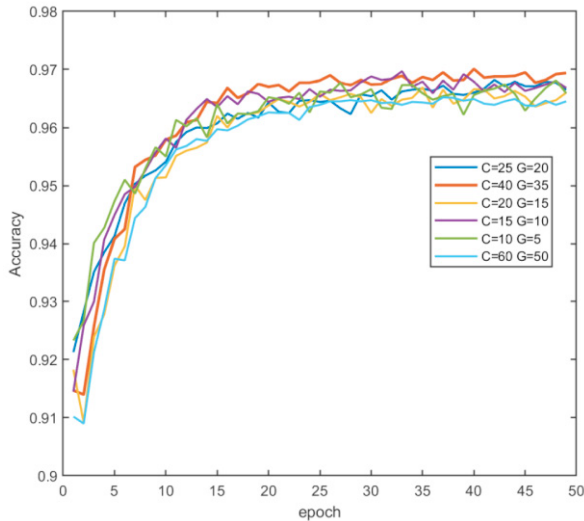


Figure 3 The training results of different values of C and G

From Figure 3 and Table 1, it can be seen that increasing the values of C_1 and C_2 does not always improve the accuracy of the global training model. In federated learning, as the values of C_1 and C_2 increase, the number of clients participating in training also increases, and the communication-cost of the downlink is increased, which may cause communication congestion. Therefore, choosing appropriate values of C_1 and C_2 is very important to improve the communication efficiency of the system. In this paper, we evaluated the balance between the calculation efficiency and convergence speed, and set C_1 and C_2 to 25 and 20 respectively.

Then, In order to simulate the heterogeneity of client devices and industrial data, this experiment assumes that all clients have a certain probability of crashing, and a certain number of crash clients is preset at each global training, and the delay time of crash clients should be set to simulate their slow state.

In this experiment, semi-synchronous training was performed under different delay conditions, and the results of global model training are shown in Figure 4. When the delay time τ is set to 0, the system is in the client-server synchronous training mode, the larger the delay time τ , the greater the tolerance degree of the aggregator for delayed clients.

As can be seen from Fig. 4, the accuracy of the training model is greater than 96% by using the semi synchronous communication protocol. In the case of low latency federated learning, the semi synchronous training mode has great advantages in the accuracy, convergence speed and communication cost. But when the value of the delay time τ is too large, the local model versions of different industrial equipment clients are too different, and the stale information may be introduced and cause the gradient update of the global model to be outdated. Therefore, in practical

application, we comprehensively consider the delay time of the system and set the delay time τ to 4.

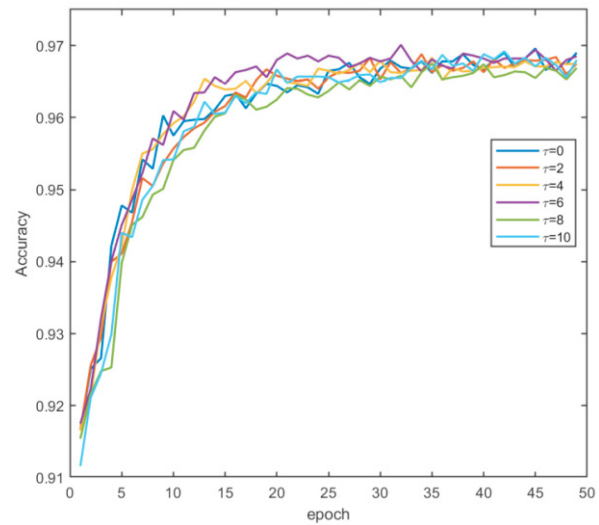


Figure 4 The training results under the different delay time

5.3 Performance of our proposed method

This experiment will compare the proposed algorithm and the existing FedAvg algorithm to verify the effect of our algorithm for the data distribution of typical industrial equipment. In this article, the system is set to allow some crashed clients and delayed clients among all the industrial equipment. In our experiments, Communication-cost coefficient represents a normalized value of the communication round between the server and the client. The smaller the value, the lower the communication cost required. The model performances are shown in Table 1.

Table 1 Model performances of different algorithm

Model	Accuracy	Loss	Communication-cost coefficient
FedAvg	0.9627	0.1695	1
Our Algorithm	0.9648	0.1597	0.375

It can be seen from Table 1 that the performance of the proposed federated learning method is better than the existing FedAvg method, and it can save communication costs and improve the training effect.

6. CONCLUSIONS

Aiming at optimizing the communication overhead allocation problem in federated learning of the model of digital twins, we propose a communication-efficient distribution algorithm to ensure the performance of industrial system model by changing the update training mode of client and server and allowing some industrial equipment to participate in federated training. This paper sets up the actual experimental configuration to analyze the effectiveness of the proposed method. Comparing our proposed method with the existing FedAvg method, the results show that our proposed method can improve the communication performance of the training model and reduce the communication-cost. Our proposed

method is more suitable for the environmental configuration of the digital twin system of complex industrial situations.

ACKNOWLEDGMENT

This research was supported by National Key R&D Program of China, No. 2018YFE0105000, 2018YFB1305304, Shanghai Municipal Science and Technology Major Project under grant 2021SHZDZX0100, the Shanghai Municipal Commission of Science and Technology No. 1951113210, 19511132101, and National Natural Science Foundation of China (No. 62173253).

REFERENCES

- Chen Y, Sun X, Jin Y (2020). Communication-Efficient Federated Deep Learning with Asynchronous Model Update and Temporally Weighted Aggregation. *IEEE Transactions on Neural Networks and Learning Systems*. 31(10), 4229-4238.
- Gao Weifeng, Zhao Zhiwei, Min Geyong, et al. (2021). Resource Allocation for Latency-aware Federated Learning in Industrial Internet-of-Things. *IEEE Transactions on Industrial Informatics*. 17(12), 8505-8513.
- Jeschke S, Brecher C, Meisen T, et al. (2017). *IIoT and Cyber Manufacturing Systems*. Springer International Publishing.
- King Susan, Grobbelaar S S (2020). Industry 4.0 and Business Model Innovation: A Scoping Review. 26th ICE/IEEE International Technology Management Conference. IEEE, 1-8.
- Lade P, Ghosh R, Srinivasan S (2017). Manufacturing Analytics and IIoT. *IEEE Intelligent Systems*. 32(3), 74-79.
- Lu Yunlong, Huang Xiaohong, Zhang Ke, et al. (2020). Communication-Efficient Federated Learning for Digital Twin Edge Networks in Industrial IoT. *IEEE Transactions on Industrial Informatics*, 17(8), 5709-5718.
- Lu Y., Huang X., Dai Y., Maharjan S., and Zhang Y. (2020). Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, *IEEE Trans. Ind. Informat.* 16(6), 4177–4186.
- Lu Yunlong, Huang Xiaohong, Zhang Ke, et al. (2020). Blockchain Empowered Asynchronous Federated Learning for Secure Data Sharing in Internet of Vehicles. *IEEE Transactions on Vehicular Technology*, 69(4), 4298-4311.
- Ma Y, Zhou H., He H., Jiao G., Wei S. (2019). A digital twin-based approach for quality control and optimization of complex product assembly. *International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM, Ireland)*. 762–767.
- McMahan H. Brendan, Moore Eider, et al. (2017). “Communication-Efficient Learning of Deep Networks from Decentralized Data.” *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 1273–1282.
- Nishio T, Yonetani R (2019). Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. *IEEE International Conference on Communications (ICC)*. 1–7.
- Pereira A C, Romero F (2017). A review of the meanings and the implications of the Industry 4.0 concept. *Procedia Manufacturing*, 13, 1206-1214.
- Song Qiang, Lei Shiyu, Sun Wen, Zhang Yan (2021). Adaptive Federated Learning for Digital Twin Driven IIoT. 2021 *IEEE Wireless Communications and Networking Conference (WCNC, Nanjing)*. IEEE, 1-6.
- Sprague M R, Jalalirad A, Scavuzzo M, et al. (2019). Asynchronous Federated Learning for Geospatial Applications. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 21–28.
- Tran N. H., Bao W., Zomaya A., et al. (2019). Federated learning over wireless networks: Optimization model design and analysis, *Proc. IEEE Conf. Comput. Commun.* 1387–1395.
- Triastcyn A. and Faltings B. (2019). Federated learning with Bayesian differential privacy. *Proc. IEEE Int. Conf. Big Data*. 2587–2596.
- Wang S, Tuor T, Saloniemi T, et al. (2019). Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications*. 37(6), 1205-1221.
- Wu W, He L, Lin W, et al. (2021). SAFA: a Semi-Asynchronous Protocol for Fast Federated Learning with Low Overhead. *IEEE Transactions on Computers*. 70(5), 655-668.
- Yann Lecun, Leon Bottou, Bengio, Y, Haffner (1998), Patrick Gradient-based learning applied to document recognition, *Proceedings of the IEEE*. 86(11):2278-2324.
- Yu Z, Hu J, Min G, et al. (2018). Federated Learning Based Proactive Content Caching in Edge Computing. *IEEE Global Communications Conference (Abu Dhabi)*. 1-6.
- Zhu H, Jin Y (2020). Multi-Objective Evolutionary Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*. 31(4), 1310-1322.