EDT-SaFL: Semi-Asynchronous Federated Learning for Edge Digital Twin in Industrial Internet-of-Things

Ming Tao, *Member, IEEE,* Lingling Liao, *Student Member, IEEE,* Yin Zhang, *Senior Member, IEEE,* Lei Liu, *Member, IEEE,* Geyong Min, *Member, IEEE,* Dusit Niyato, *Fellow, IEEE,* Schahram Dustdar, *Fellow, IEEE*

***Abstract*—Through conducting equivalent model training within the paradigm of edge intelligence, the Digital Twin Edge Networks (DITEN) have been widely employed in the Industrial Internet-of-Things (IIoT) to facilitate the cost-effective execution without the operational disruption. However, due to the insufficient consideration of heterogeneity in computing and communication capabilities of distinct industrial terminals in the Digital Twin (DT) model training, the existing approaches of DT construction/update have unbalanced model training cost and loss in the whole life cycle of DT model, hindering the abilities of quick responding to complex and dynamic productions and ensuring the data consistency of virtual-real space. To address this issue, we define a global loss minimization problem with constraint, and propose an original approach of semi-asynchronous federated learning, named EDT-SaFL, as a promising solution. Considering the collaborative utilization of heterogeneous resources, and the contribution of local data quantity and quality to the global model update, the EDT-SaFL consists of three important operations, *Terminal Selection for Model Training*, *Self-Adaptation of Local Training Iterations*, and *Semi-asynchronous Global Aggregation*. With the analysis of convergence, complexity and communication overhead, the experiments have evidently demonstrated the superiority of EDT-SaFL on the datasets of CIFAR-10 and Industrial-Equipment.**

***Index Terms*—Industrial Internet-of-Things, Digital Twin, Edge Intelligence, Semi-Asynchronous Federated Learning.**

I. Introduction

ADDRESSING the intelligent manufacturing needs and avoiding the critical time and resource consumptions caused by traditional trial-and-error methods, as well as the unpredictable risks arisen by operational disruption, the Digital Twin (DT) as a popular technology has been widely employed in the Industrial Internet-of-Things (IIoT) owing to the inherent advantages in virtual-real collaboration [1–4]. Nevertheless, as a typical deterministic network, the IIoT has a strict requirement of service response delay during the interaction between DT models and physical systems. With the advantage of low-latency service, the edge computing can provide a strong support for the DT implementation in complex IIoT scenarios, addressing the extremely high requirements of real-time and high-reliability data processing [5, 6]. Therefore, with the combination of advantages inherent in DT and edge computing, the Digital Twin Edge Networks (DITEN) as a research hotspot have attracted extensive attention from both academia and industry, being widely employed in the IIoT scenarios including product design, fault diagnosis, decision support, and predictive maintenance [7–9].

In the DITEN for IIoT, to improve the overall service performance, the DT construction/update as an involved operation is of paramount importance, which should employ appropriate modeling methods to accurately map the practical physical scenarios in terms of the deep understanding of characteristics and behaviors of the industrial systems. However, there are still no standard definition, unified framework, or implementation approach for DT construction/update [10, 11]. As a popularity, the physical-principle based modeling method is incompetent in guaranteeing the accuracy and reliability of DT model, while the data-driven modeling method can achieve higher precision and more intelligent DT model through artificial intelligence technologies, e.g., machine learning, data mining and deep learning [12, 13]. The machine learning techniques can realize the DT model training and prediction on a large amount of data, and improve the efficiency and accuracy while ensuring the DT reliability. The data mining techniques can automatically discover the data association and the hidden rules in the DT, providing the accurate prediction and decision support. The deep learning techniques can enable higher levels of data abstraction and autonomous decision-making [14, 15].

In complex and dynamic intelligent manufacturing scenarios, the industrial systems often consist of subsystems with multiple variables and interrelated factors [16]. The constructed DT models should accurately represent the corresponding multi-dimensional physical characteristics including geometry,

This work was supported in part by the Guangdong Key Construction Discipline Research Ability Improvement Project under Grant 2021ZDJS086; in part by the DGUT Research Team Project under Grant TDY-B2019009; in part by the Guangdong Natural Science Foundation (Key Project) under Grant 2022B1515120059.

Ming Tao and Lingling Liao are with the School of Computer Science and Technology, Dongguan University of Technology, China (Email: taom@dgut.edu.cn, liaolingling@dgut.edu.cn).

Yin Zhang is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China (Email: zhangyin123@uestc.edu.cn).

Lei Liu is with the Guangzhou Institute of Technology, Xidian University, Guangzhou, China (Email: leiliu@xidian.edu.cn).

Geyong Min is with the Department of Computer Science, University of Exeter, U.K. (Email: g.min@exeter.ac.uk).

Dusit Niyato is with the College of Computing and Data Science, Nanyang Technological University, Singapore (Email: dniyato@ntu.edu.sg).

Schahram Dustdar heads the Distributed Systems Group, TU Wien, Austria and is a part-time ICREA Professor at UPF, Barcelona (Email: dustdar@dsg.tuwien.ac.at)

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3595117

physics, behavior and rules, and be constantly updated in real-time to maintain the data consistency of virtual-real space. Therefore, with the ability of necessitating the model training based on physical data, the data-driven method is more applicable for the DT construction/update in the DITEN for IIoT [17]. Nevertheless, with the data-driven method, given the limited number of model parameters on edge servers, the heterogeneous industrial terminals with a large amount of local data are necessitated to participate in the DT model training to provide the abundant local trained model parameters, while addressing the distinct and strict requirements on privacy, delay and accuracy in intelligent manufacturing scenarios [18]. Therefore, with the advantages of protecting data privacy, reducing communication cost, improving model quality, and conducting multi-party cooperation, the federated learning (FL) is an appropriate option to aggregate a global DT model through training local models [19, 20].

Before the global model update, the synchronous FL requires the edge parameter server to wait for all terminals completing the local updates, in which, it is unnecessary to execute the frequent local model transmission, but would impose a great increase in waiting delay while updating the global model [21–24]. To reduce the waiting delay of global update and improve the global model performance, the asynchronous FL also has garnered significant interest. Although the asynchronous FL can mitigate the waiting delays of transmitting local model parameters while conducting the global update, the aggregation of global model in a completely asynchronous manner might lead to the obsolescence of some local models, reducing the accuracy of the global model. Meanwhile, the frequent transmission of local model parameters is necessary in the asynchronous FL, which would cause the substantial communication overhead [25–29].

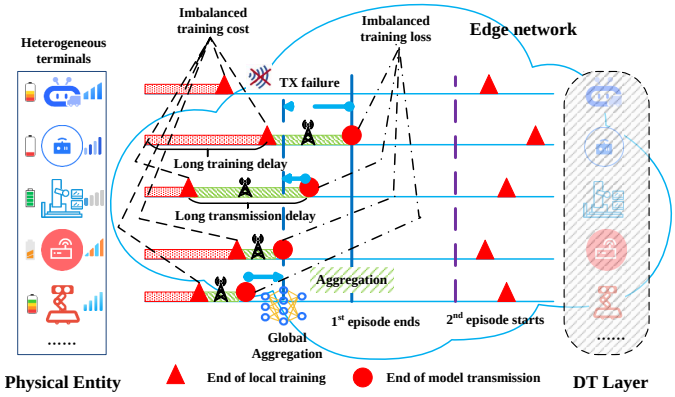


Fig. 1. Influence of the heterogeneity on DT model training cost and loss.

With the advantages of FL, acknowledging the aforementioned solutions are efficient in specific scenarios of DT construction/update, however, the critical heterogeneity in computing and communication capacities on the distinct industrial terminals participating in the DT model training, has not been fully addressed in either the synchronous FL or the asynchronous FL [30]. As shown in Fig. 1, in a DITEN for industrial scenario, due to the heterogeneity, the industrial terminals have different local training delays. With the synchronous FL, the global update in each training episode needs to wait for the moronic terminals, also known as straggler terminals, completing the local training, which would lead to the lag-effect imposing the negative impact on the global model. Although the asynchronous FL does not require the long waiting delay, the global model is updated frequently. As a result, compared to the latest global model, the local models trained at the moronic terminals would be outdated, leading to the obsolescence negatively influencing the global model as well. Therefore, the heterogeneity has a crucial impact on the insufficient balance on DT model training cost and loss due to the lag-effect and obsolescence, and the DT can neither effectively address the IIoT service requirements with strict response delay nor ensure the data consistency.

With this motivation, addressing the heterogeneity of industrial terminals, we investigate an approach of semi-asynchronous FL for DT construction/update in the DITEN for IIoT. To improve the balance of training cost and loss for the DT construction/update in the whole life cycle of DT model, the optimization objective is defined as minimizing the training loss of the global model with the constraint of maximum acceptable global training delay, and then, through combining the advantages of both synchronous FL and asynchronous FL, and mitigating the lag-effect inherent in the synchronous FL and the obsolescence issue in the asynchronous FL, we propose a new approach of semi-asynchronous FL, named EDT-SaFL, to offer a promising solution. The main contributions of this paper are summarized as follows.

- To effectively coordinate the heterogeneity for achieving the efficient utilization of heterogeneous computing and communication resources on training industrial terminals, a semi-asynchronous global update mode is designed with the definition of staleness. In which, a theorem is presented and proven to deduce the number of synchronous/asynchronous updates in a training episode, on the basis, the model parameters transmission delay, and the global aggregation delay are derived. Further, the global optimization is formulated as a minimization problem with constraint.
- For selecting the appropriate industrial terminals to participate in the DT model training, the ***Terminal Selection for Model Training*** is designed in the EDT-SaFL, in which, the Upper Confidence Bound (UCB) based optimistic estimation strategy is developed to determine the distinct capability of heterogeneous terminals, the Gradient Norm Value (GNV) indicating the importance of local data for global model update and convergence is also defined, and the weighted sum of terminal capability and GNV is used to provide the decision basis.
- With the selected terminals, for adaptively adjusting the number of local training iterations to alleviate both lag-effect and obsolescence of local models on the training accuracy in the global aggregation, and balance the model training cost and loss, the ***Self-Adaptation of Local Training Iterations*** is designed in the EDT-SaFL, where the number of local training iterations on a training terminal is adapted to the heterogeneity of inherent computing

power, the local data volume and quality.

- With local trained model parameters, for alleviating the problem of lag-effect and obsolescence while conducting the global aggregation, the ***Semi-asynchronous Global Aggregation*** is designed, in which, the staleness is introduced to modify the local trained model parameters for mitigating the inconsistency between the local model parameters and the global ones, and with the stated semi-asynchronous global update mode, the weighted method is employed to aggregate the global model parameters mapping in the DT layer to form the global DT model.

The rest of this paper is organized as follows. In Section II, the schema of EDT-SaFL is presented, and the semi-asynchronous global update mode is elaborately designed. In Section III, the deduced delays caused in a training episode, and the optimization problem of global loss minimization with constraint are formulated. In Section IV, the proposed EDT-SaFL is thoroughly presented. In Section V, the convergence, complexity and communication overhead of EDT-SaFL are analyzed. In Section VI, the experimental results and comparative analysis are discussed. Finally, we summarize this paper and discuss future research issues in Section VII.

II. Schema

For the DT construction/update in the considered DITEN for IIoT, there are $|\Re|$ heterogeneous industrial terminals, $\Re = \{r_1, r_2, ..., r_{|\Re|}\}$. To address the heterogeneity of industrial terminals participating in the DT model training, as shown in Fig. 2, a semi-asynchronous FL based solution, named EDT-SaFL, is proposed to balance the training cost and loss in the whole life cycle of DT model, in which, the involved symbols and the corresponding explanations are summarized in Table I.

TABLE I
INVOLVED SYMBOLS AND EXPLANATIONS.

Symbols	Explanations
$\Re_N$	The set of terminals selected for the local training, $\{r_1, ..., r_i, ..., r_N\}$
$\Re_a$	The set of selected terminals conducting the asynchronous global update
$\Re_s$	The set of selected terminals conducting the synchronous global update
η	The learning rate
λ_0	The initial training iterations assigned to each selected terminal
D_i	The local data volume on r_i
$\|w\|$	The data volume of trained model parameters
$F(w^k)$	The loss function of global model parameters w^k in the $(k-1)$-th episode
$f_i(w_i^k)$	The loss function of local model parameters w_i^k on r_i
g_i^{k-1}	The gradient of $f_i(w_i^{k-1})$
ψ^g	The global model version number
ψ_i^l	The local model version number on r_i
θ_i	The difference between ψ^g and ψ_i^l, named staleness
$s(\theta_i)$	The inconsistency between the local model and the global model
ψ	The predefined threshold for staleness
$\mathcal{T}^k$	The total time cost in the k-th training episode
$\mathcal{T}_k^{LT}, \mathcal{T}_k^{TR}, \mathcal{T}_k^{AGG}$	The local training delay, the model parameters transmission delay and the global aggregation delay in the k-th training episode
$\mathcal{T}_{k,i}^{LT}$	The local training delay on r_i
$\mathcal{T}_{max}$	The maximal acceptable global training time cost
λ_i	The local training iterations obtained by ***Self-Adaptation of Local Training Iterations***
$\mathcal{C}_i$	The computing power of r_i
c_i	The number of CPU cycles required to train a data sample on r_i
v_i	The data transmission rate on r_i
n	The number of synchronous updates
c_u	The number of CPU cycles required to aggregate one unit of data
$\mathcal{C}_e$	The computing power of the edge server
β_i^τ	A binary variable representing whether r_i is selected for the local model training in the τ-th episode
$\hat{c}_i^k$	The UCB-based computing power of r_i
$\tilde{c}_i^k$	The relative computing power of r_i
Δ_i^k	The confidence radius of UCB
$\hat{v}_i^k$	The UCB-based communication capability of r_i
$\tilde{v}_i^k$	The relative communication capability of r_i

In each training episode, the global model parameters w at the edge server are firstly distributed to N superior industrial terminals selected by the ***Terminal Selection for Model Training***, that is, $\Re_N \subset \Re$. On the selected terminal, r_i ($r_i \in \Re_N$) iteratively trains w with the local private data based on the iterations determined by the ***Self-Adaptation of Local Training Iterations***. After the local training, the local model parameters are sent back to the edge server, and the global model parameters are updated by the ***Semi-asynchronous Global Aggregation***. With the updated global model parameters, the decision-making of global aggregation is finally made in the DT layer to form the global DT model.

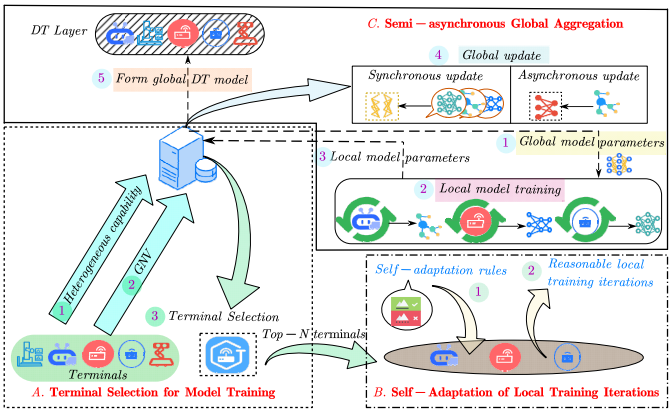


Fig. 2. Semi-asynchronous FL based solution.

In the solution, to address the issues of lag-effect and obsolescence caused by the heterogeneity of industrial terminals, a global model version number ψ^g at the edge server, a local model version number ψ_i^l on r_i, and the difference between ψ^g and ψ_i^l, $\theta_i = \psi^g - \psi_i^l$, named the staleness, are introduced. In the k-th training episode, assuming that $\Re_a$ and $\Re_s$ respectively are the set of selected terminals conducting the asynchronous global update and conducting the synchronous global update, after the local training on r_i, the local trained model parameters will be immediately uploaded to the edge server, and θ_i will be checked with the predefined threshold ψ. Accordingly, the edge server will conduct the global update with the semi-asynchronous mode stated as follows.

- If $\theta_i < \psi$, with the asynchronous global update, the global model parameters would be immediately updated with the local ones uploaded from r_i, and the global model version number would be updated to $\psi^g \leftarrow \psi^g + 1$. During the process, $r_i \rightarrow \Re_a$, and the other selected terminals, $r_j \in \Re_N - \Re_a - \Re_s$, still perform the local model training with the previous global model parameters.
- If $\theta_i \geq \psi$, to cooperate with r_i for completing the synchronous update, the other selected terminals, $r_{j \setminus i} \in \Re_N - \Re_a - \Re_s$, whose training iterations have not been ended, would temporarily suspend the local training and upload the local model parameters to the edge server. After completely receiving the local model parameters from r_i and r_j, the global model parameters would be updated with the synchronous global update, and $\psi^g \leftarrow \psi^g + 1$, $r_i \rightarrow \Re_s$. Then, the updated global model parameters would be distributed to those suspended terminals, restarting the local training to finish

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3595117

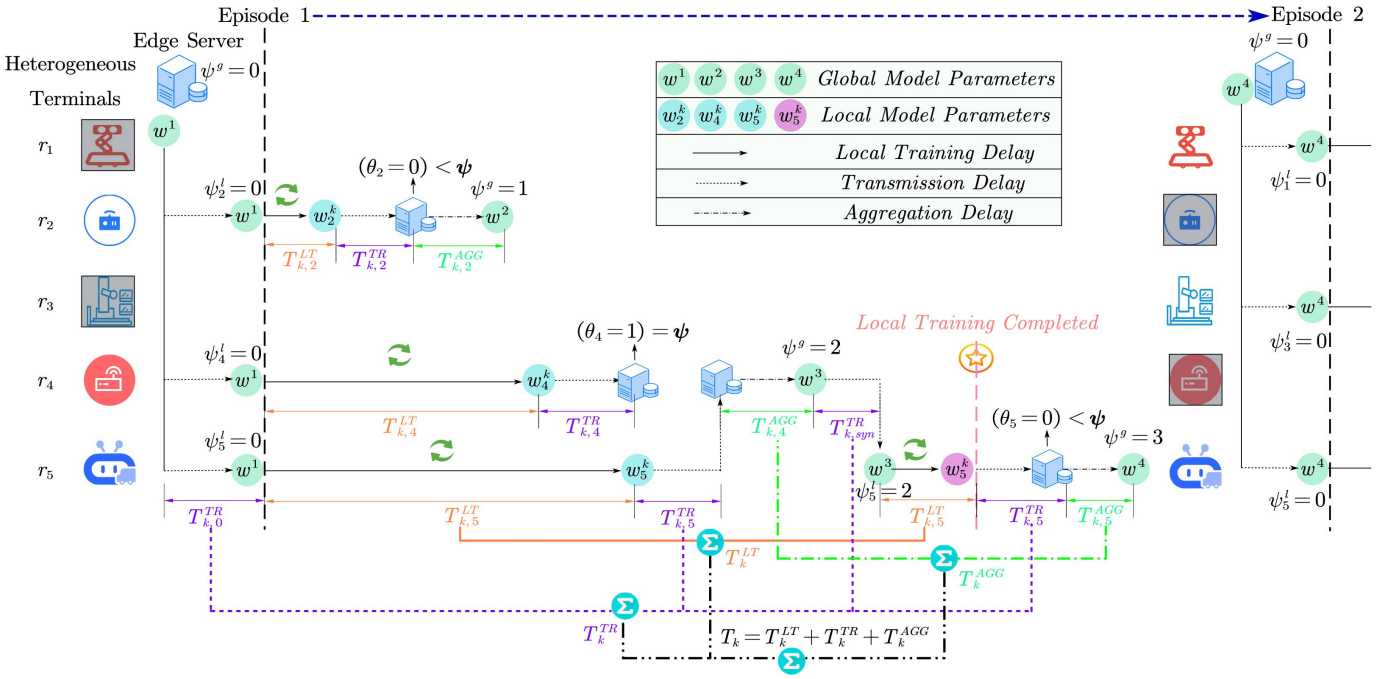


Fig. 3. An illustration of conducting a training episode.

the preset training iterations. Correspondingly, the local model version number on the suspended terminal would be updated to $\psi_j^l \leftarrow \psi^g$.

Notedly, within the executive process of the global update determined by the checked staleness, those certain factors, e.g., real-time connectivity quality, would influence the process. However, to ensure the robustness, stability and reliability of the overall operation of the designed semi-asynchronous global update mode, the determination should not be arbitrarily changed with the operating environment. In fact, in the designed ***Terminal Selection for Model Training***, the complete evaluation of the heterogeneity in computing and communication capacities on industrial terminals also can overcome the immediate changes in the operating environment.

For clearly understanding the operation process of designed semi-asynchronous global update mode, we give an illustration shown in Fig. 3. For the sake of simplicity, $\psi = 1$. Meanwhile, to ensure full discussions of all possible global updates and full capture of the complete flow of solution, it is necessary to configure sufficient number of terminals in the illustration, and the number of terminals is set to 5. In the k-th training episode, with the ***Terminal Selection for Model Training***, r_2, r_4 and r_5 are selected for the local model training. To alleviate both lag-effect and obsolescence of local models on the training accuracy, the ***Self-Adaptation of Local Training Iterations*** is adopted to determine the local training iterations of selected terminals. Initially, $\psi^g = 0$ and $\psi_i^l = 0$, and the global model parameters w^1 are sent to the selected terminals with the distribution delay $T_{k,0}^{TR}$. On the superior r_2, the local model parameters w_2^k are first obtained and uploaded to the edge server. With the semi-asynchronous mode, on the edge server, through checking $\theta_2 = \psi^g - \psi_2^l$, that is $\theta_2 = 0$ and $\theta_2 < \psi$, the global model parameters would be updated to w^2 with an asynchronous global update, and $\psi^g \leftarrow 1$, $r_2 \rightarrow \Re_a$.

Along with r_4 finishing the local training, the local model parameters w_4^k are uploaded to the edge server. Similarly, through checking $\theta_4 = \psi$, a synchronous global update would be conducted. In this process, the training r_5 would temporarily suspend the local training and upload the local model parameters to the edge server. After completely receiving the local model parameters from r_4 and the suspended terminal r_5, the global model parameters would be updated to w^3, and $\psi^g \leftarrow 2$, $r_4 \rightarrow \Re_s$. Next, with the distribution delay $T_{k,syn}^{TR} = T_{k,0}^{TR}$, w^3 would be distributed to r_5 for restarting the local training to finish the preset training iterations, and $\psi_5^l \leftarrow 2$.

After receiving the local trained model parameters w_5^k on the edge server, $\theta_5 < \psi$ is checked, and the global model parameters are updated to w^4 using the asynchronous global update as the last one, $\psi^g \leftarrow 3$, $r_5 \rightarrow \Re_a$. As soon as the final global model parameters are obtained through all selected terminals completing the local training, the k-th training episode is completely conducted. To begin the next training episode, $k \leftarrow k+1$, $\psi^g \leftarrow 0$, $\psi_i^l \leftarrow 0$ and $\theta_i \leftarrow 0$.

III. Formulation

As the stated semi-asynchronous global update mode, $\mathcal{T}_k$ as the total time cost in the k-th episode that consists of the local training delay $\mathcal{T}_k^{LT}$, the model parameters transmission delay $\mathcal{T}_k^{TR}$, and the global aggregation delay $\mathcal{T}_k^{AGG}$, that is, $\mathcal{T}_k = \mathcal{T}_k^{LT} + \mathcal{T}_k^{TR} + \mathcal{T}_k^{AGG}$. The specific definitions of $\mathcal{T}_k^{LT}$, $\mathcal{T}_k^{TR}$ and $\mathcal{T}_k^{AGG}$ are stated as follows.

1) Local Training Delay: Given the local training iterations on the selected terminals, the distinct terminals would have different local training delays. Since the selected terminals perform the local training in a parallel manner, the local

training in the k-th episode is completely finished only if all selected terminals have finished the distinct local training iterations, and the time cost, $\mathcal{T}_k^{LT}$, would be the maximum in the local training delays of selected terminals.

For r_i, to assume that $\mathcal{C}_i$ and c_i respectively denote the computing power of r_i and the number of CPU cycles required to train a data sample on r_i. The local training delay on r_i therefore can be defined as $T_{k,i}^{LT} = \frac{\lambda_i \cdot D_i \cdot c_i}{\mathcal{C}_i}$, where λ_i is the local training iterations obtained by ***Self-Adaptation of Local Training Iterations***, and D_i is the local data volume on r_i. Therefore, $\mathcal{T}_k^{LT}$ can be defined as

$$\mathcal{T}_k^{LT} = \max\left\{ T_{k,i}^{LT} = \frac{\lambda_i \cdot D_i \cdot c_i}{\mathcal{C}_i} \,|r_i \in \Re_N \right\}. \tag{1}$$

2) Model Parameters Transmission Delay: For transmitting the trained local model parameters to the edge server, the Orthogonal Frequency-Division Multiple Access (OFDMA) technology is employed to minimize the channel interference among r_i ($r_i \in \Re_N$), with which, r_i is assigned a distinct sub-channel ensuring the equal bandwidth. With the Shannon formula, the data transmission rate can be defined as $v_i = B \cdot \log_2\left(1 + \frac{P_i \cdot |g_i|^2}{N_0}\right)$, where B is the sub-channel bandwidth, P_i is the uplink transmission power, g_i is the channel gain between r_i and the edge server, and N_0 is the Gaussian noise power. In a training episode shown in Fig. 3, the communication delays include the delay of distributing the initial global model parameters to the selected terminals, the delay of distributing the updated global model parameters to the suspended terminals after a synchronous update, and the transmission delay of trained local model parameters. Since the data volume of model parameters is denoted as $|w|$, both transmission delay and distribution delay can be defined as $|w|/v_i$.

In the k-th training episode, after completing the local training on r_i, it will upload the trained local model parameters to the edge server that would trigger a global update with the stated semi-asynchronous mode. Therefore, the total number of global update is N. To deduce the model parameters transmission delay and the global aggregation delay, it is necessary to derive the number of synchronous/asynchronous updates, and ***Theorem* 1** is introduced here.

***Theorem* 1**: In a training episode, if there are N terminals selected for the local model training, with the predefined ψ, there would be $\lfloor N/(\psi+1) \rfloor$ synchronous updates, and there would be $N - \lfloor N/(\psi+1) \rfloor$ asynchronous updates.

Proof: Before the φ-th global update, assuming that the set of selected terminals finishing the local training is $\Re_{\varphi-1}$. With the stated semi-asynchronous mode, if the φ-th global update is synchronous, after the update, there is $\psi_g = \varphi$, and for the suspended terminal r_i ($r_i \in \Re_N - \Re_\varphi$), there is $\psi_i^l = \varphi$. On this basis, in the next ψ updates, the model version difference on the selected terminals triggering the global update would be always less than ψ, and the triggered global updates are always conducted in an asynchronous manner. After the $(\varphi+\psi)$-th update, there is $\psi^g = \varphi + \psi$, and for the terminal r_j ($r_j \in \Re_N - \Re_{\varphi+\psi}$), there is still $\psi_j^l = \varphi$. Therefore, at the start of the next update, for the selected terminal triggering this update, the model version difference would be equal to ψ, and a synchronous update would be triggered. Accordingly, we can come to a conclusion that there are ψ asynchronous updates between any two synchronous updates, that is, if there are n synchronous updates in a training episode, there would be $(n-1)\cdot\psi$ asynchronous updates. Additionally, with the stated semi-asynchronous mode and the predefined threshold ψ, there would be ψ asynchronous updates before the first synchronous update, and a synchronous update would be performed after every $\psi+1$ updates, that is, if $\varphi\%(\psi+1) = 0$, a synchronous update would be performed, otherwise, an asynchronous update would be performed.

To sum up, in a training episode that the N selected terminals conduct N global updates, if there are n synchronous updates, there are $\psi+(n-1)\cdot\psi = n\cdot\psi$ asynchronous updates, and there is

$$n + n\cdot\psi \le N \rightarrow n = \lfloor N/(\psi+1) \rfloor. \tag{2}$$

■

With the stated semi-asynchronous mode and ***Theorem* 1**, the transmission delays involved in a synchronous update consist of the upload delay of the local model parameters on the suspended terminals and the delay of distributing the updated global model parameters to the suspended terminals. Therefore, for the n synchronous updates, the transmission delays can be defined as $\lfloor N/(\psi+1) \rfloor \cdot \frac{2|w|}{v_i}$.

Additionally, we would find that, if $N\%(\psi+1) = 0$, that is, the remainder of N over $\psi+1$ is 0, the last global update triggered by the selected terminal finally completing the local training is a synchronous one, and $\mathcal{T}_k^{TR}$ should consist of the distribution delay of initial global model parameters $\frac{|w|}{v_i}$ and $\lfloor N/(\psi+1) \rfloor \cdot \frac{2|w|}{v_i}$. Otherwise, if $N\%(\psi+1) \neq 0$, the last triggered global update is an asynchronous one, which should consider the upload delay of local model parameters again. Therefore, $\mathcal{T}_k^{TR}$ can be defined as

$$\mathcal{T}_k^{TR} = \begin{cases} \lfloor N/(\psi+1) \rfloor \cdot \dfrac{2|w|}{v_i} + \dfrac{|w|}{v_i}, & N\%(\psi+1) = 0, \\ \lfloor N/(\psi+1) \rfloor \cdot \dfrac{2|w|}{v_i} + 2\cdot\dfrac{|w|}{v_i}, & N\%(\psi+1) \neq 0. \end{cases} \tag{3}$$

3) Global Aggregation Delay: To assume that the number of CPU cycles required to aggregate one unit of data is c_u, and the computing power of the edge server is $\mathcal{C}_e$. With the stated semi-asynchronous mode, being similar to the definition of $\mathcal{T}_k^{TR}$, $\mathcal{T}_k^{AGG}$ should be only the global aggregation delay caused by the synchronous updates in the condition of $N\%(\psi+1) = 0$, which is defined as

$$\mathcal{T}_k^{SYN} = \sum_{m=1}^{n} \frac{(N - m\cdot\psi - (m-1))\cdot|w|\cdot c_u}{\mathcal{C}_e}, \tag{4}$$

in which, in terms of ***Theorem* 1**, since there are $m\cdot\psi$ asynchronous updates and $(m-1)$ synchronous updates before the m-th synchronous update, $N - m\cdot\psi - (m-1)$ is the number of selected terminals participating in the m-th synchronous update.

If $N\%(\psi+1) \neq 0$, $\mathcal{T}_k^{AGG}$ also considers the aggregation delay of the last asynchronous update represented by $\frac{|w|\cdot c_u}{\mathcal{C}_e}$.

Therefore, $\mathcal{T}_k^{AGG}$ is defined as

$$\mathcal{T}_k^{AGG} = \begin{cases} \mathcal{T}_k^{SYN} + \dfrac{|w| \cdot c_u}{\mathcal{C}_e}, & N\%(\psi+1) \neq 0, \\ \mathcal{T}_k^{SYN}, & N\%(\psi+1) = 0. \end{cases} \tag{5}$$

With the solution, to balance the training cost and loss for the DT construction/update in a training episode, addressing the collaborative and efficient utilization of heterogeneous computing and communication resources, we introduce a loss function $F\left(w^k\right)$ to measure the model training cost and loss, which is defined as

$$\begin{aligned} F\left(w^k\right) &= \sum_{i=1}^{N} \left(\frac{D_i}{D} \cdot f_i\left(w_i^k\right) \right), \\ f_i\left(w_i^k\right) &= \frac{1}{D_i} \cdot \sum_{j=1}^{D_i} f_i\left(w_i^k, x_{ij}, y_{ij}\right), \end{aligned} \tag{6}$$

where, as a logistic regression model [31], $f_i\left(w_i^k\right)$ is the loss function of local model parameters w_i^k in the data sample (x_{ij}, y_{ij}), and $D = \sum_{i=1}^{N} D_i$ is the total data volume of all selected terminals. Accordingly, to minimize the global training loss with the constraint of maximum acceptable global training time cost in the whole life cycle of DT model, the optimization objective of this solution is defined as

$$\begin{aligned} \mathcal{P}1: \min & \sum_{k=1}^{K} F\left(w^k\right) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \mathcal{T}_k \leq \mathcal{T}_{max}, \end{aligned} \tag{7}$$

where $\mathcal{T}_{max}$ is the maximal acceptable global training time cost and K is the total training episodes.

IV. Solution

The EDT-SaFL consists of the three following important operations, and the details are shown in Algorithm 1.

A. Terminal Selection for Model Training

Given the heterogeneity of industrial terminals in the considered scenarios and the data distribution across terminals is Non-Independent and Identically Distributed (Non-IID), the terminal selection for model training is an important factor affecting the DT model construction/update. Therefore, a terminal selection strategy is proposed as an essential component executed in each training episode, which should dynamically select terminals to participate in the model training. Terminals that have greater positive impact on global updates, would have higher possibility of being selected, improving training efficiency and convergence speed. In general, for the terminal selection in the Non-IID context, since the communication capability related to P_i affects the speed of data synchronization and the efficiency of global updates, terminals with high transmission power are more suitable for transmitting local model parameters, and since the computing power of terminal $\mathcal{C}_i$ determines the ability to efficiently process local data, terminals with strong computing power are more suitable for

Algorithm 1 Semi-Asynchronous Federated Learning

Initialize w^1, ψ, λ_0;
for $k = 1, 2, \cdots, K$ **do**
  $\psi^g = 0$, $\varphi = 0$;
  for $i = 1, 2, \cdots, |\Re|$ **do**
    Calculate $\hat{c}_i^k, \hat{v}_i^k$, and δ_i^k using Eq. 8, Eq. 9, Eq. 10 and Eq. 11;
    Calculate ϑ_i^k with Eq. 12;
    Set $\psi_i^l = 0$;
  end for
  $\Re_N \leftarrow \varnothing, \Re_a \leftarrow \varnothing, \Re_s \leftarrow \varnothing$;
/* Terminal Selection for Model Training */
  Select the Top-N terminals with ϑ_i^k to form $\Re_N$;
/* Self-Adaptation of Local Training Iterations */
  for $r_i \in \Re_N$ **do**
    Calculate λ_i;
    Distribute $(w^k, k, \psi_i^l, \lambda_i)$ to r_i;
  end for
/* Semi-asynchronous Global Aggregation */
  /* Terminal side: */
  for $r_i \in \Re_N - \Re_a - \Re_s$ **do**
    Train the local model with λ_i;
    Calculate $\bar{g}_i^k$ with Eq. 14;
    Update w_i^k with Eq. 15;
    Calculate $s\left(\theta_i\right) = \frac{1}{1+e^{-\theta_i}}$;
    Update $w_{i,new}^k$ with Eq. 16;
    /* With the proven ***Theorem* 1** that a synchronous update would be performed after every $\psi+1$ updates. */
    if $\varphi\%(\psi+1)! = 0$ **then**
      /* Asynchronous update: */
      Calculate $f_i\left(w_{i,new}^k\right)$ with Eq. 6;
      Upload $(w_{i,new}^k, f_i(w_{i,new}^k), \psi_i^l)$;
    else
      /* Synchronous update: */
      Upload $(w_{i,new}^k, f_i(w_{i,new}^k), \psi_i^l)$;
      Suspend the local training;
      for $r_{j \setminus i} \in \Re_N - \Re_a - \Re_s$ **do**
        Upload $(w_j^k, f_j(w_j^k), \psi_j^l)$;
      end for
      Restart the local training with w^{k+1};
    end if
  end for
  /* Edge Server side: */
  for $r_i \in \Re_N - \Re_a - \Re_s$ **do**
    if $\theta_i < \psi$ **then**
      Update to w^{k+1} with an asynchronous mode;
      $\psi^g \leftarrow \psi^g + 1$, $\varphi = \varphi + 1$, $r_i \rightarrow \Re_a$;
    end if
    if $\theta_i \geq \psi$ **then**
      Update to w^{k+1} with a synchronous mode;
      $\psi^g \leftarrow \psi^g + 1$, $\varphi = \varphi + 1$, $r_i \rightarrow \Re_s$;
      Update $\psi_i^l = \psi^g$ for suspended terminals;
      Distribute (w^{k+1}, ψ_i^l) to suspended terminals;
    end if
  end for
  Calculate $F(w^k)$ with Eq. 6;
end for

handling complex data. Therefore, it is necessary to model the computing/communication capability of each terminal here.

- The computing power of r_i is defined as the ratio of the training data volume to the training delay T_i^{train} in a training episode, that is, $\mathcal{C}_i = \frac{\lambda_i \cdot D_i}{T_i^{train}}$, where the processed data volume related to the training iterations and the local data volume can be defined as $\lambda_i \cdot D_i$.
- The communication capability of r_i is defined as the ratio of the transmitted data volume of local model parameters $D_i^{param.}$ to the time cost $T_i^{trans.}$ required for the data transmission in a training episode, that is, $v_i = \frac{D_i^{param.}}{T_i^{trans.}}$.

In the complex and changeable industrial scenarios, to dynamically adapt to the changes in terminal performance, the UCB-based optimistic estimation strategy is adopted to conduct the terminal selection for model training [32].

In the k-th training episode, the UCB-based computing power of r_i is defined as the sum of the relative computing power and the confidence radius of UCB [33], $\hat{c}_i^k = \tilde{c}_i^k + \Delta_i^k$,

$$\tilde{c}_i^k = \begin{cases} \dfrac{\mathcal{C}_i \cdot \sum_{\tau=1}^{(k-1)} \beta_i^\tau}{k-1}, & r_i \in \Re_N^{k-1} \\ \mathcal{C}_i \cdot (k-1) \cdot (1 - \dfrac{\sum_{\tau=1}^{(k-1)} \beta_i^\tau}{k-1}), & r_i \notin \Re_N^{k-1} \end{cases} \tag{8}$$

is the relative computing power of r_i. In which, $\sum_{\tau=1}^{(k-1)} \beta_i^\tau$ is the frequency of r_i selected for the local model training in the previous $(k-1)$ episodes, β_i^τ is a binary variable representing whether r_i is selected for the local model training in the τ-th episode, $\Re_N^{k-1}$ is the set of terminals selected for the local model training in the $(k-1)$-th episode, and Δ_i^k is the confidence radius of UCB, which is defined as

$$\Delta_i^k = \begin{cases} \sqrt{\dfrac{N \cdot \ln Z^{k-1}}{\sum_{\tau=1}^{k-1} \beta_i^\tau}}, & r_i \in \Re_N^{k-1}, \\ \Delta_i^{k-1}, & r_i \notin \Re_N^{k-1}, \end{cases} \tag{9}$$

where Z^{k-1} is the sum of frequencies of the terminals selected both in the $(k-1)$-th episode and in the previous $(k-1)$ episodes, that is, $Z^{k-1} = \sum_{i=1}^{N} \sum_{\tau=1}^{(k-1)} \beta_i^\tau$ and $r_i \in \Re_N^{k-1}$.

Similarly, the UCB-based communication capability of r_i is defined as $\hat{v}_i^k = \tilde{v}_i^k + \Delta_i^k$, where

$$\tilde{v}_i^k = \begin{cases} \dfrac{v_i \cdot \sum_{\tau=1}^{(k-1)} \beta_i^\tau}{k-1}, & r_i \in \Re_N^{k-1} \\ v_i \cdot (k-1) \cdot (1 - \dfrac{\sum_{\tau=1}^{(k-1)} \beta_i^\tau}{k-1}), & r_i \notin \Re_N^{k-1} \end{cases} \tag{10}$$

is the relative communication capability of r_i.

Additionally, given the local data of heterogeneous terminals is Non-IID, the local data quality would affect the stability and reliability of the global model update to some extent, and the poor data quality would lead to a large gradient fluctuation of the trained model, influencing the model convergence [34]. Meanwhile, the execution efficiency of heterogeneous terminals would also affect the stability and reliability of the model update, and if the execution efficiency is low, the training instability and unreliability would be increased. That is, the local model trained with the local data that has significant gradient can effectively promote the global model convergence and aggregation. Accordingly, in the k-th training episode, the GNV of the trained local model that indicates the importance of local data on the $(k-1)$-th model training [35], is also adopted to conduct the terminal selection for model training, which is defined as

$$\delta_i^k = ||g_i^{k-1}||^2, \; g_i^{k-1} = \nabla f_i\left(w_i^{k-1}\right), \tag{11}$$

where $f_i\left(w_i^{k-1}\right)$ is the loss function of r_i.

Accordingly, through defining the ratio as

$$\vartheta_i^k = \frac{\rho_{com} \cdot \hat{c}_i^k + \rho_{comm} \cdot \hat{v}_i^k + \rho_{gnv} \cdot \delta_i^k}{\sum_{i=1}^{|\Re|} \left(\rho_{com} \cdot \hat{c}_i^k + \rho_{comm} \cdot \hat{v}_i^k + \rho_{gnv} \cdot \delta_i^k\right)}, \tag{12}$$

where ρ_{com}, ρ_{comm} and ρ_{gnv} respectively are the weights of computing power, communication capability and GNV, the terminals with Top-N ratios would be selected for the local model training.

B. Self-Adaptation of Local Training Iterations

The heterogeneous computing powers of selected terminals result in imbalanced training cost under the same training intensity. Therefore, to alleviate both lag-effect and obsolescence of local model on the global update and balance the model training cost, it is necessary to adaptively adjust the number of local training iterations on the selected terminal. Since the training capacity of a selected terminal is proportional to the computing power, the self-adaptation rules can be defined as follows. In the k-th episode, if the computing power of the selected terminal exceeds the average $\bar{c}^k = \frac{\sum_{i=1}^{N} \mathcal{C}_i}{N}$, the terminal should be assigned more training iterations. Conversely, the training iterations should be reduced on the selected terminals whose computing powers are below the average. Additionally, the local data volume and quality on the selected terminal also impose an important influence on the local model training, which further affects the global model update. Therefore, the selected terminals with more local data and better data quality would be assigned more training iterations. Here, in terms of the definition of GNV shown in Eq. 11, the impact of local data quality on the model training also can be indicated by the GNV.

Accordingly, the optimization of local training iterations is guided by the computing power of terminal, the local data volume, and the GNV. To ensure superior performance and balance the local computation load, the self-adaptation local training iterations on r_i can be defined as

$$\lambda_i = \lambda_0 \cdot \left(\frac{\mathcal{C}_i}{\bar{c}^k}\right) \cdot \left(\frac{D_i}{\sum_{r_i \in \Re_N} D_i}\right) \cdot \left(\xi \cdot \delta_i^k\right), \tag{13}$$

where λ_0 is the initial training iterations assigned to each selected terminal in the first episode. For avoiding the obsolescence and lag-effect, both $\frac{D_i}{\sum_{r_i \in \Re_N} D_i}$ and ξ ($\xi \in (0, 1]$) are adopted to balance the difference of the local training iterations assigned on different selected terminals.

C. Semi-asynchronous Global Aggregation

Although the existing proposals attempt to address terminal heterogeneity through adaptive aggregation or staleness

reduction, the factor, e.g., staleness or local data volume, is individually considered, overlooking the critical influence of data quality on model aggregation. Moreover, in the considered dynamic IIoT environments, the varied data distribution and terminal conditions also have significant effect on the efficiency of existing proposals. Therefore, to effectively address the lag-effect inherent in the synchronous FL and the obsolescence issue in the asynchronous FL, a semi-asynchronous global aggregation mechanism is designed by comprehensively considering local gradient, staleness, and local data volume, achieving high-quality global updates.

Specifically, in the k-th episode, after the local model training on each selected terminal, to mitigate fluctuations of global model parameters caused by the heterogeneity in training iterations, ensuring stability and effectiveness of global model update, the average local gradient on r_i defined as

$$\bar{g}_i^k = \frac{\sum_{\lambda=1}^{\lambda_i} \nabla f_i\left(w_i^{k,\lambda}\right)}{\lambda_i}, \tag{14}$$

is used to obtain the local model parameters,

$$w_i^k = w_i^{k-1} - \eta \bar{g}_i^k, \tag{15}$$

where w_i^{k-1} is the local model parameters in the $(k-1)$-th episode, $w_i^{k,\lambda}$ is the local model parameters obtained in the λ-th iteration, and η is the learning rate.

Furthermore, to mitigate the difference between the local model parameters and the global ones caused by the local model obsolescence and lag-effect, with the staleness, the local model parameters are also modified as

$$w_{i,new}^k = s\left(\theta_i\right) \cdot w^k + \left(1 - s(\theta_i)\right) \cdot w_i^k, \tag{16}$$

where w^k is the global model parameters in the $(k-1)$-th episode, and $s\left(\theta_i\right)$ is the inconsistency between the local model and the global model, which is defined as

$$s\left(\theta_i\right) = \frac{1}{1+e^{-\theta_i}}, \ s\left(\theta_i\right) \in [0,1]. \tag{17}$$

With the modified local model parameters, in the stated semi-asynchronous global update mode, the weighted method is employed to aggregate the global model parameters as

$$w^{k+1} = \sum_{i=1}^{N} \left(\frac{D_i}{\sum_{i=1}^{N} D_i} \cdot w_{i,new}^k \right). \tag{18}$$

Through mapping the aggregated global model parameters in the DT layer, the global DT model would be formed as

$$DT^{k+1} = \mathbb{Q}\left(w^{(k+1)}, D^{(k+1)}, S^{(k+1)}, DT^k, CN^{(k+1)}\right), \tag{19}$$

where $D^{(k+1)}$ is the operating data from the physical entity, e.g., the operating status of the equipment in the production line, $S^{(k+1)}$ is the operating state of the virtual entity, DT^k is the DT model data in the $(k-1)$-th episode, and $CN^{(k+1)}$ indicates the association between the physical entity and the virtual entity.

With the solution, to ensure the accurate DT construction/update, it is worth noting that the unselected terminals also must transmit the operating state into the DT layer to maintain the data consistency of virtual-real space. Additionally, for the convenience of implementation and maintenance of EDT-SaFL addressing the real-world IIoT service requirements, we can design the three important operations in a modular way and conduct well-defined interfaces to achieve the coordination.

V. Evaluation

To demonstrate the efficiency of EDT-SaFL, the convergence, complexity and communication overhead are theoretically analyzed.

A. Convergence

For Non-IID data in the considered scenarios, the defined global loss function $F\left(w^k\right)$ might not strictly satisfy L-smooth and μ-strongly convex [36]. Therefore, we assume that $F\left(w^*\right)$ is the optimum of $F\left(w^k\right)$, on r_i, with the weakened L-smooth, there is

$$\begin{aligned} F(w_i^k) \leq F(w_i^{k-1}) + \left\langle \nabla F(w_i^{k-1}), w_i^k - w_i^{k-1} \right\rangle \\ + \frac{L}{2} ||w_i^k - w_i^{k-1}||^2 + \epsilon, \end{aligned} \tag{20}$$

where ϵ is an error term accounting for local variations in gradients. With the stated $w_i^k = w_i^{k-1} - \eta \bar{g}_i^k$, the difference between the local training loss $F(w_i^k)$ and the optimal global loss $F\left(w^*\right)$ is represented as

$$\begin{aligned} &L_i(k) = F(w_i^k) - F(w^*) \\ &\leq F(w_i^{k-1}) + \eta \cdot \left\langle \nabla F(w_i^{k-1}), -\bar{g}_i^k \right\rangle \\ &+ \frac{L \cdot \eta^2}{2} \left\| \bar{g}_i^k \right\|^2 + \epsilon - F(w^*) \\ &\leq F(w_i^{k-1}) - \frac{1}{2 \cdot L} \left\| \nabla F(w_i^{k-1}) \right\|^2 \\ &+ \frac{L \cdot \eta^2}{2} \left\| \bar{g}_i^k - \frac{1}{L \cdot \eta} \nabla F(w_i^{k-1}) \right\|^2 + \epsilon - F(w^*). \end{aligned} \tag{21}$$

To weak the assumption of μ-strongly convex and hold for a broader class of functions, the Polyak-Lojasiewicz (PL) condition is introduced. With the PL, there is

$$\left\| \nabla F(w_i^{k-1}) \right\|^2 \geq 2\mu \cdot \left(F(w_i^{k-1}) - F(w^*) \right). \tag{22}$$

Due to the critical heterogeneity in computing and communication capacities on the distinct industrial terminals participating in the DT model training, it is difficult to obtain the complete optimal global loss while using the FL. Therefore, an upper bound of permitted error denoted as Γ is introduced to achieve the approximate optimal global loss of $F(w^*)$, and the difference can be further defined as

$$L_i(k) \leq \left(1 - \frac{\mu}{L}\right) \cdot \left(F(w_i^{k-1}) - F(w^*) \right) + \frac{L\eta^2 C}{2} + \epsilon + \Gamma, \tag{23}$$

where $||\bar{g}_i^k - \frac{1}{L\eta} \nabla F\left(w_i^{k-1}\right)||^2 \leq C$, and C is the upper bound of the norm.

With the definition in Eq. 16 and the PL condition, there is

$$F(w_{i,new}^{k-1}) \leq s(\theta_i) \cdot F(w^{k-1}) + (1 - s(\theta_i)) \cdot F(w_i^{k-1}), \tag{24}$$

and with the definition in Eq. 18, the difference between the global loss and the optimal global loss is defined as

$$\begin{aligned} G(k) &= F\left(w^k\right) - F\left(w^*\right) \\ &= F\left(\sum_{i=1}^{N}\left(\frac{D_i}{\sum_{i=1}^{N} D_i}\cdot w_{i,new}^{k-1}\right)\right) - F\left(w^*\right) \\ &\le \sum_{i=1}^{N}\left(\frac{D_i}{\sum_{i=1}^{N} D_i}\cdot\left(\begin{array}{l} s(\theta_i)\cdot F(w^{k-1}) \\ +(1-s(\theta_i))\cdot F(w_i^{k-1}) \end{array}\right)\right) - F(w^*). \end{aligned} \tag{25}$$

With $F(w^*) = \sum_{i=1}^{N}\left(\frac{D_i}{\sum_{i=1}^{N} D_i}\cdot F(w^*)\right)$, there is

$$\begin{aligned} &\sum_{i=1}^{N}\left(\frac{D_i}{\sum_{i=1}^{N} D_i}\cdot\left(\begin{array}{l} s(\theta_i)\cdot F(w^{k-1}) \\ +(1-s(\theta_i))\cdot F(w_i^{k-1}) \end{array}\right)\right) - F(w^*) \\ &= \sum_{i=1}^{N}\left(\frac{D_i}{\sum_{i=1}^{N} D_i}\cdot\left(\begin{array}{l} s(\theta_i)\cdot F(w^{k-1}) - F(w^*) \\ +(1-s(\theta_i))\cdot F(w_i^{k-1}) \end{array}\right)\right) \\ &= \sum_{i=1}^{N}\left(\frac{D_i}{\sum_{i=1}^{N} D_i}\left(\begin{array}{l} s(\theta_i)\left(F(w^{k-1}) - F(w^*)\right) \\ +(1-s(\theta_i))\left(F(w_i^{k-1}) - F(w^*)\right) \end{array}\right)\right). \end{aligned} \tag{26}$$

Iteratively using Eq. 21 to deduce $F(w_i^{k-1})$ until $k=1$, and with Eq. 23, $G(1)\cdot\left(1-\frac{\mu}{L}\right)^{k-1} + \frac{L^2\eta^2 C}{2(L-\mu)} + \epsilon + \Gamma$ can be obtained. Assuming that $\phi_i = \frac{D_i}{\sum_{i=1}^{N} D_i}$, the difference $G(k)$ can be further less than or equal to

$$\begin{aligned} &\sum_{i=1}^{N}\left(\phi_i\left(\begin{array}{l} \left(1-s(\theta_i)\right)\left(G(1)\left(1-\frac{\mu}{L}\right)^{k-1} + \kappa\right) \\ +s(\theta_i)G(k-1) \end{array}\right)\right) \\ &\le m_k\cdot\left(\ (1-s(\theta))\cdot(G(1)\cdot\zeta+\kappa) + s(\theta)\cdot G(k-1)\ \right), \end{aligned} \tag{27}$$

where $m_k = \sum_{i=1}^{N}\phi_i$, $\zeta = \left(1-\frac{\mu}{L}\right)^{k-1}$, $\kappa = \frac{L^2\eta^2 C}{2(L-\mu)} + \epsilon + \Gamma$, and there is

$$\begin{aligned} &\left(1-s\left(\theta\right)\right)\cdot\left(G\left(1\right)\cdot\zeta+\kappa\right) + s\left(\theta\right)\cdot G\left(k-1\right) \\ &= \max_{r_i\in\Re_N}\left\{\left(1-s\left(\theta_i\right)\right)\cdot\left(G\left(1\right)\cdot\zeta+\kappa\right) + s\left(\theta_i\right)\cdot G\left(k-1\right)\right\} \end{aligned} \tag{28}$$

Since $s\left(\theta_i\right)\in[0,1]$, the aforementioned analysis has been shown to demonstrate the convergence of this solution.

B. Complexity and Communication Overhead

Considering the parallelism in a training episode, with the stated $\mathcal{T}_k = \mathcal{T}_k^{LT} + \mathcal{T}_k^{TR} + \mathcal{T}_k^{AGG}$, the global time complexity is $O(\mathcal{T}_k) = O(\mathcal{T}_k^{LT} + \mathcal{T}_k^{TR} + \mathcal{T}_k^{AGG})$. With K as the total training episodes, the total global time complexity can be defined as $O(\mathcal{T}_{total}) = O\left(\sum_{k=1}^{K}\mathcal{T}_k\right)$. Due to the introduced constraint $\sum_{k=1}^{K}\mathcal{T}_k \le \mathcal{T}_{max}$, $O(\mathcal{T}_{total}) \le \mathcal{T}_{max}$ is acceptable.

As aforementioned, in a training episode, the total communication overhead of EDT-SaFL consists of the communication overhead of distributing initial global model parameters $\mathcal{O}_{comm}^{g}$, that of conducting asynchronous updates $\mathcal{O}_{comm}^{a}$, and that of conducting synchronous updates $\mathcal{O}_{comm}^{s}$. Assuming that the communication overhead of transmitting one unit of data is o_t, with the data volume of model parameters $|w|$, there is $\mathcal{O}_{comm}^{g} = o_t\cdot|w|$.

For an asynchronous update, the communication overhead is caused by uploading the trained local model parameters, denoted as $o_t\cdot|w|$. Based on ***Theorem* 1**, for the $N - \lfloor N/(\psi+1)\rfloor$ asynchronous updates in a training episode, there is

$$\mathcal{O}_{comm}^{a} = \Big(N - \lfloor N/(\psi+1)\rfloor\Big)\cdot o_t\cdot|w|. \tag{29}$$

For a synchronous update, e.g., the m-th synchronous update, the communication overhead should consist of the $(N - m\psi - (m-1))\ \cdot\ o_t\ \cdot\ |w|$ caused by uploading the trained local model parameters, and the $(N - m(\psi+1))\cdot o_t\cdot|w|$ caused by distributing updated global model parameters to the suspended terminals, where, based on ***Theorem* 1**, $(N - m\psi - (m-1))$ is the number of terminals participating in this synchronous update, and $(N - m(\psi+1))$ is the number of suspended terminals. Therefore, for the $\lfloor N/(\psi+1)\rfloor$ synchronous updates in a training episode, there is

$$\begin{aligned} \mathcal{O}_{comm}^{s} = &\sum_{m=1}^{\lfloor N/(\psi+1)\rfloor}\left(\Big(N - m\psi - (m-1)\Big)\cdot o_t\cdot|w|\right) \\ &+ \sum_{m=1}^{\lfloor N/(\psi+1)\rfloor}\left(\Big(N - m(\psi+1)\Big)\cdot o_t\cdot|w|\right). \end{aligned} \tag{30}$$

To sum up, with K as the total training episodes, the total communication overhead of EDT-SaFL can be defined as $\mathcal{O}_{total} = K\cdot(\mathcal{O}_{comm}^{g} + \mathcal{O}_{comm}^{a} + \mathcal{O}_{comm}^{s})$.

VI. Experiments

A. Experiment Setting

To comprehensively evaluate the performance of EDT-SaFL, on the server: Intel(R) Xeon(R) Silver 4210R CPU at 2.40GHz, and 64 GB of RAM, the experiments were programmed using Python 3.9.0 executed on a Ubuntu 18.04.6 LTS operating system. The two datasets used in the experiments, CIFAR-10 and Industrial-Equipment from Kaggle, have distinct data characteristics. Specifically, the CIFAR-10 dataset consists of 60,000 images of size $32\times32\times3$, divided into 10 classes. The Industrial-Equipment dataset contains 6,890 images of size $100\times100\times3$, classified into 2 categories. In the experimental prototype of the DITEN for IIoT, there are an edge server and $|\Re| = 20$ heterogeneous industrial terminals. The major experimental parameters are configured in Table II. To address the heterogeneity of industrial terminals, and the complex communication conditions in the real industrial scenarios, the ranges of computing and communication capacities of distinct industrial terminals are configured with the performance of common industrial devices, and the maximal acceptable global training time is configured as $\mathcal{T}_{max} = 1800s$. Since the local data on terminals is Non-IID, all the data samples are randomly partitioned into subsets with a Dirichlet distribution, each terminal is assigned a different number of training samples, and the learning rate η defined as $0.3/(\mathrm{k}+1)$ is dynamically decreasing with the training episode k for achieving the stable convergence [37].

TABLE II
CONFIGURATION OF EXPERIMENTAL PARAMETERS.

Param.	Val.	Param.	Val.
K	40	$buffer\_size$	32
η	0.3/(k+1)	λ_0	5
$\mathcal{T}_{max}$	1800 s	N_0	-174 dBm/Hz
ρ_{gnv}	0.6	ρ_{com}, ρ_{comm}	0.2
B	2 MHz	h_i	2 dBi
P_i	[0.1,0.5] W	$\mathcal{C}_i$	[2,5] GHz

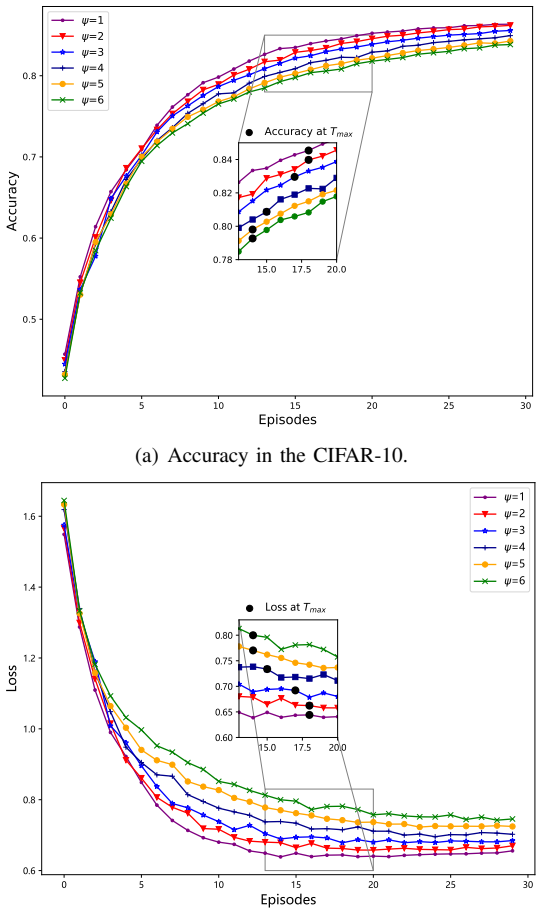
B. Performance of EDT-SaFL

To mitigate the impact of outdated models and ensure the effective training, it is necessary to determine appropriate ψ to enhance the adaptability of EDT-SaFL to the environment.

In the CIFAR-10, with the degree of Non-IID is 0.5, the performance verification in accuracy and loss with different ψ is shown in Fig. 4. In terms of the stated semi-asynchronous mode, if the value of ψ is low, the staleness would be smaller, that is, the local trained model parameters could more accurately indicate the evolutionary direction of the current global model parameters, and the performance of updated global model could be improved better. Therefore, the achieved performance of low ψ is higher while the global training time reaching the defined $\mathcal{T}_{max}$, e.g., $\psi = 1$ and $\psi = 2$. However, in the condition of low ψ, the synchronous update would be performed more frequently, causing more resources consumption. Therefore, with the constraint of $\mathcal{T}_{max}$, to balance the achieved performance and the resources consumption, $\psi = 2$ is more reasonable to satisfy the requirements of low-latency and high-reliability in the industrial scenarios.

In the Industrial-Equipment containing less data samples, with the degree of Non-IID is 0.5, the performance verification in accuracy and loss with different ψ is shown in Fig. 5. Due to the relatively small data volume, the performance shows obvious fluctuation with different ψ. With low ψ, similarly, the achieved performance is relatively higher, e.g., $\psi = 1$ and $\psi = 2$. However, if $\psi = 1$, in terms of the stated semi-asynchronous mode, the synchronous update would be performed more frequently, and the local training on the other training terminals would be frequently interrupted while conducting the global updates, that is, the other training terminals have relatively limited training effectiveness, frequently synchronizing the insufficient local model parameters would negatively affect the performance of the global model, and the performance of $\psi = 1$ shows obvious instability in the process of convergence. Therefore, with the constraint of $\mathcal{T}_{max}$, to achieve the stable performance, $\psi = 2$ is more reasonable to satisfy the requirements of low-latency and high-reliability.

To effectively verify the performance of EDT-SaFL, the six algorithms, FedAsync [38], FedAvg [39], Synchronous FL(SA), Asynchronous FL(SA), HSA_FL(adaptive) [33] and HSA_FL(fixed) [33], are selected for comparisons. With the native FL framework, both FedAsync and FedAvg consist of the two operations, e.g., randomly selecting terminals for model training and fixing the local training iterations. With distinct global aggregation methods, as variations of the EDT-SaFL,



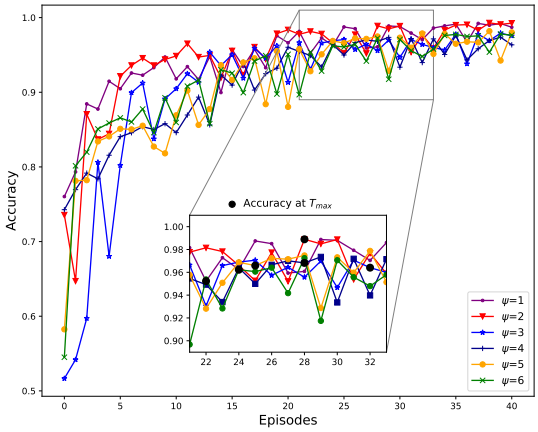
(a) Accuracy in the CIFAR-10.

(b) Loss in the CIFAR-10.

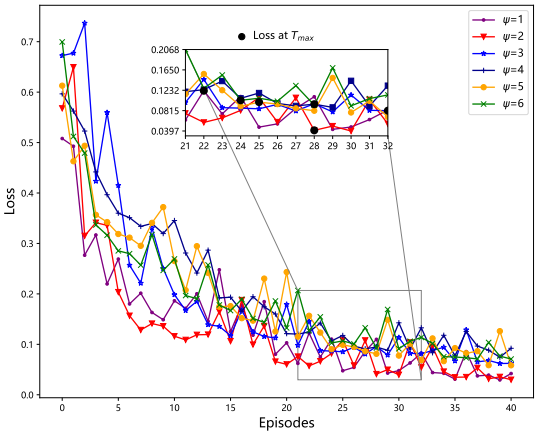
Fig. 4. Performance with different ψ in the CIFAR-10.

both Synchronous FL(SA) and Asynchronous FL(SA) consist of the two operations, the terminal Selection for model training and self-Adaptation of local training iterations designed in the EDT-SaFL. It is worth noting that, to achieve the optimal performance in both HSA_FL(adaptive) and HSA_FL(fixed), being different from the parameter configurations for the EDT-SaFL, the received proportion of the local model while initiating the global update is set to 0.5 in the HSA_FL(adaptive), and the training time in each episode is fixed to 100s in the HSA_FL(fixed).

In the CIFAR-10, with $\psi = 2$, the performance verification in accuracy and loss along with the variation of the configured degree of Non-IID is shown in Fig. 6. With the Non-IID, the data imbalance across terminals significantly impacts the model performance, leading to slow convergence and low accuracy. As shown in Fig. 6(a), with the increase of the degree of Non-IID, the accuracies show downward trends. Through comprehensively incorporating computing power, communication capability, data volume, and data quality in the three key operations of EDT-SaFL, the negative effects of data heterogeneity could be mitigated to some extent. However, the inherent data imbalance in Non-IID remains a significant challenge to achieve higher accuracy. Due to the weak capability in tackling the data heterogeneity, the achieved accuracies in the comparative methods are lower than that of

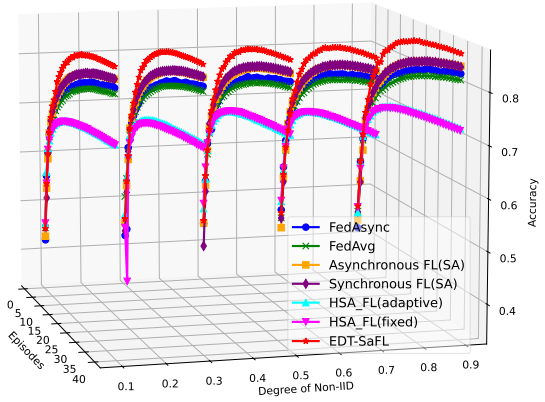


(a) Accuracy in the Industrial-Equipment.

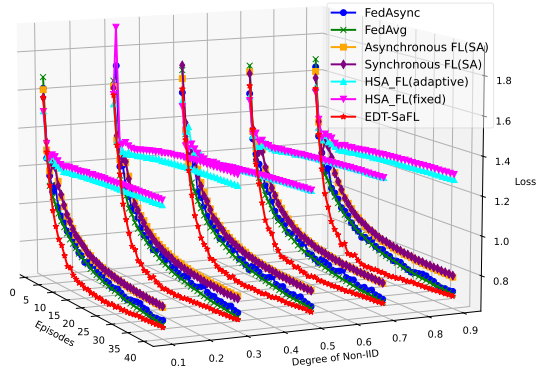


(b) Loss in the Industrial-Equipment.

Fig. 5. Performance with different ψ in the Industrial-Equipment.



(a) Accuracy in the CIFAR-10.



(b) Loss in the CIFAR-10.

Fig. 6. Performance with different degrees of Non-IID in the CIFAR-10.

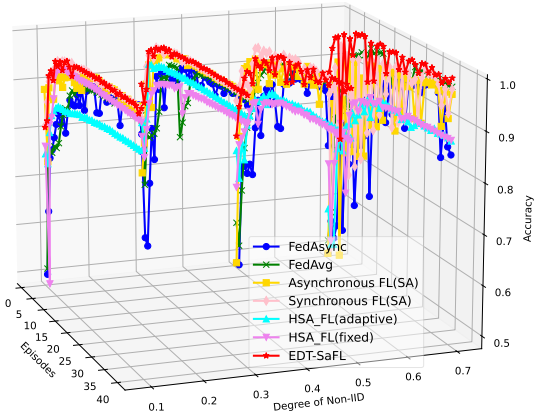
EDT-SaFL.

On the other hand, in the considered DITEN for IIoT, while using both HSA_FL(adaptive) and HSA_FL(fixed), due to no consideration of the GNV in the terminal selection and the sole consideration of the heterogeneous computing powers of industrial terminals for adaptively adjusting the local training iterations, the achieved performance is generally inferior to the other comparison algorithms. While using the classical FedAvg and FedAsync, although both random selection and fixed local training iterations are used, the two algorithms still have better adaptability to the considered scenarios than both HSA_FL (adaptive) and HSA_FL (fixed). However, both random selection and fixed local training iterations might result in the negative affect on the FL flexibility, obtaining the sub-optimal local training results on the selected terminals. Additionally, since the adopted method of average aggregation fails to fully consider the positive contributions of selected terminals on the global update, some ineffective local updates might be aggregated into the global model.

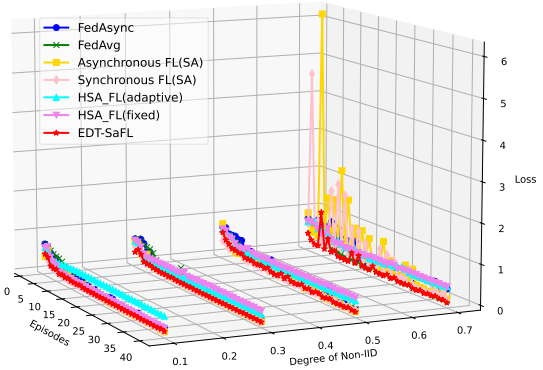
In contrast, through prioritizing the ***Terminal Selection for Model Training*** and the ***Self-Adaptation of Local Training Iterations*** to address the unbalanced distribution of resources and data, there is a significant improvement on the performance achieved by EDT-SaFL and the two algorithms using SA. However, with the inherent global update mode in the native FL framework, due to the noise introduced in the asynchronous updates, the accuracy of the Asynchronous FL(SA) is overall lower than that of the Synchronous FL(SA), and due to the lower update frequency, the accuracy of the Synchronous FL(SA) is also not the best among the comparison algorithms. With the proposed semi-asynchronous mode, the EDT-SaFL leverages the partial asynchrony to reduce the waiting delay of global update, the moderate increase of update frequency to achieve faster convergence, and the partial synchronization to reduce the instability caused by the asynchronous updates, enabling the accuracy to be stabilized faster. Therefore, the accuracy achieved by EDT-SaFL is overall superior to that of the other comparison algorithms. Similarly, the achieved loss also shows a faster reduction and the stability retention with fewer training episodes.

In the Industrial-Equipment, with $\psi = 2$, the performance verification in accuracy and loss along with the variation of the configured degree of Non-IID is shown in Fig. 7. Due to the relatively small data volume, while using both HSA_FL(adaptive) and HSA_FL(fixed), the adopted schemes would significantly decrease the achieved accuracy with the high degree of Non-IID. Similarly, the noise introduced in the asynchronous updates is relatively more serious, resulting in the relatively sharp fluctuations in the performance of FedAsync and Asynchronous FL(SA). In the synchronous FL, e.g., FedAvg and Synchronous FL(SA), since it is necessary to wait for all terminals completing the local training while

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3595117



(a) Accuracy in the Industrial-Equipment.

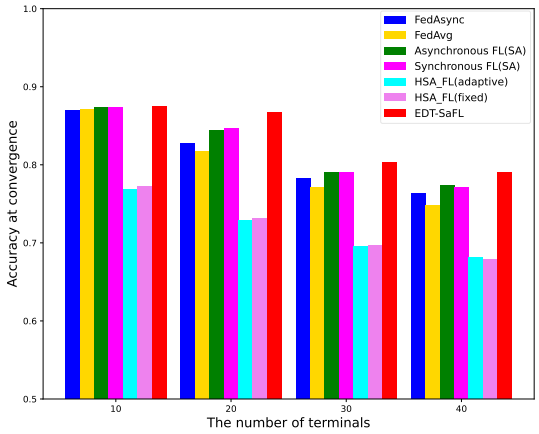


(b) Loss in the Industrial-Equipment.

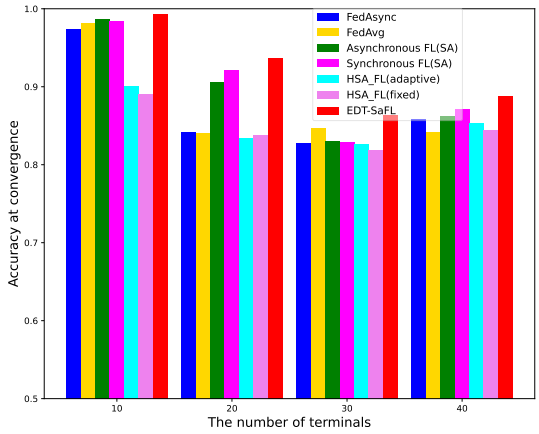
Fig. 7. Performance with different degrees of Non-IID in the Industrial-Equipment.

conducting the global update, the achieved performance is relatively steady with the low degree of Non-IID, but, along with the increase of the degree of Non-IID, the performance shows the dramatic fluctuation. With the ***Terminal Selection for Model Training***, the ***Self-Adaptation of Local Training Iterations*** and the ***Semi-asynchronous Global Aggregation*** integrating the strengths of both synchronization and asynchronism, the performance of EDT-SaFL is still the superior.

With the degree of Non-IID is 0.5 and $\psi = 2$, to evaluate the performance of EDT-SaFL along with the varying number of terminals, the accuracies at convergence in the two datasets are shown in Fig. 8. In both HSA_FL(adaptive) and HSA_FL(fixed), the terminal selection mainly depends on the distinct performance of heterogeneous terminals, which might focus on choosing the terminals with strong computing power and communication capability, ignoring the terminals with high-quality data. Meanwhile, determining the number of local training iterations only by the computing power would also lead to the excessive local training on the terminals with strong computing power, which might cause the global model overly dependents on the data from powerful terminals, ignoring the valuable data of other terminals. In both FedAvg and FedAsync adopting the random terminal selection, different terminals all have a certain probability of being selected in each training episode, the training data contributing to the global model would be more widely distributed. Therefore, the achieved accuracies at convergence of FedAvg and FedAsync are better than that of HSA_FL(adaptive) and HSA_FL(fixed). Due to the adopted ***Terminal Selection for Model Training*** and ***Self-Adaptation of Local Training Iterations*** in Synchronous FL(SA), Asynchronous FL(SA) and EDT-SaFL, the terminal capacities, and the data quantity and quality are fully considered, addressing the unbalanced distribution of resources and data, the achieved accuracies at convergence are better than that of the former four algorithms. Especially, with the proposed semi-asynchronous mode leveraging the advantages of both synchronization and asynchronism to enhance the efficiency of global update while maintaining the consistency, the accuracy at convergence achieved by EDT-SaFL is improved more evidently.



(a) Accuracy in the CIFAR-10.



(b) Accuracy in the Industrial-Equipment.

Fig. 8. Accuracy at convergence with different number of terminals.

To evaluate the convergence of EDT-SaFL, with the degree of Non-IID is 0.5 and $\psi = 2$, training episodes of achieving the predefined accuracy thresholds in CIFAR-10 and Industrial-Equipment are shown in Fig. 9. It is worth noting that, with low thresholds, e.g., 0.64 for CIFAR-10 and 0.79 for Industrial-Equipment, there is relatively little difference in convergence among the algorithms. With higher thresholds, since some algorithms exhibit clear benefits while handling challenging high-precision requirements, the difference in convergence is relatively significant. Due to the lack of consideration for the heterogeneity of terminal-

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3595117

s and local data, and the failure of fully leveraging the advantages of both synchronous and asynchronous updates, FedAsync, FedAvg, HSA_FL(adaptive) and HSA_FL(fixed) have slower convergence, requiring more training episodes to achieve the thresholds. In contrast, through prioritizing terminals with high-quality data, strong computing power and communication capability to avoid ineffective updates, dynamically adjusting the number of local training iterations with terminal capabilities and data quality, and combining the benefits of synchronous and asynchronous updates, the EDT-SaFL could mitigate the lag-effect in synchronous updates and the staleness in asynchronous updates to achieve the superior convergence, which needs less training episodes while achieving the predefined accuracy thresholds.

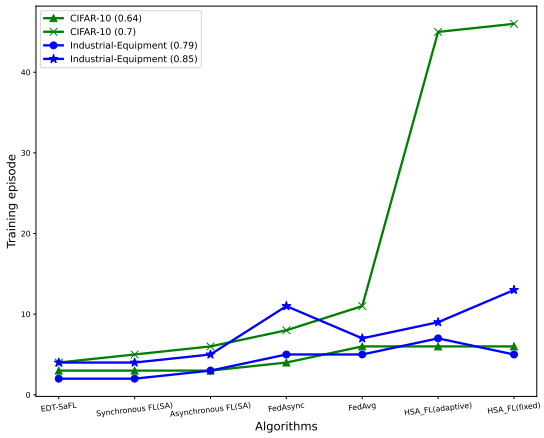
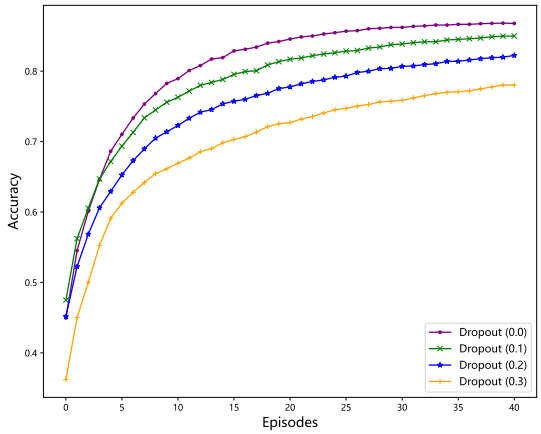
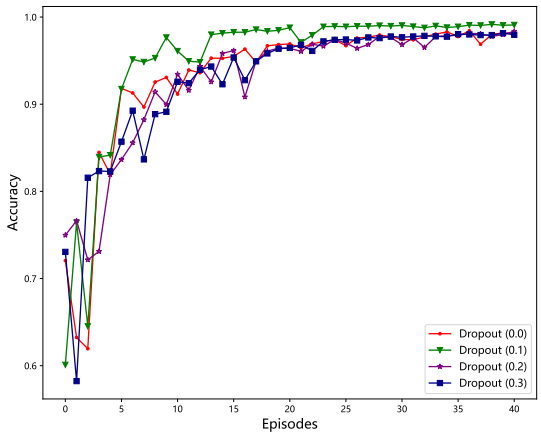


Fig. 9. Training episodes of achieving the predefined accuracy thresholds.



(a) Accuracy in the CIFAR-10.



(b) Accuracy in the Industrial-Equipment.

Fig. 10. Accuracies against different dropping ratios.

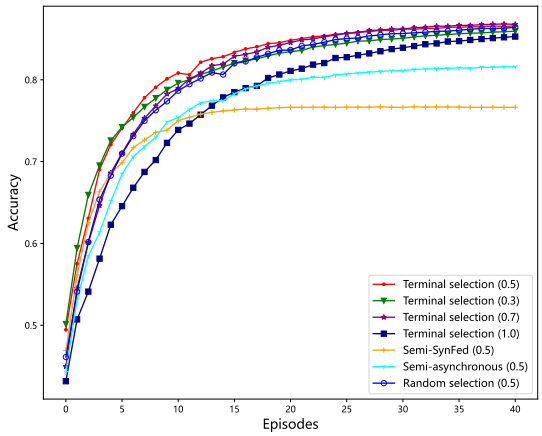
In time-varying IIoT scenarios, terminal dropout or communication failures might occur due to some unexpected factors. Therefore, through randomly selecting a proportion of terminals participated in the local training as dropped terminals in each episode, the potential impact on global model updates has been discussed. With the degree of Non-IID is 0.5 and $\psi = 2$, accuracies against different dropping ratios in both datasets are shown in Fig. 10. With the CIFAR-10, the accuracy of EDT-SaFL decreases with the increasing dropping ratio, which indicates that the dropped terminals have noticeable effect on the efficiency of EDT-SaFL, and from the indirect perspective, the superiority of terminal selection strategy in the EDT-SaFL can be demonstrated. Despite the decrease in accuracy, the overall convergence still can remain stable, demonstrating the adaptability of EDT-SaFL to real IIoT environments. With the Industrial-Equipment, the EDT-SaFL also can effectively tackle the influence of dropout on accuracy through ***Terminal Selection for Model Training*** and the ***Semi-asynchronous Global Aggregation***, achieving the competitive performance in different dropping ratios. Notably, given the smaller size and Non-IID data nature in the Industrial-Equipment, when the dropping ratio is 0.1, the high-quality data and efficient training from the remaining terminals can effectively compensate for the absence of dropped terminals, which enhances the performance of EDT-SaFL on the contrary, further demonstrating the adaptability to dynamic IIoT environments.

C. Evaluation of Terminal Selection for Model Training

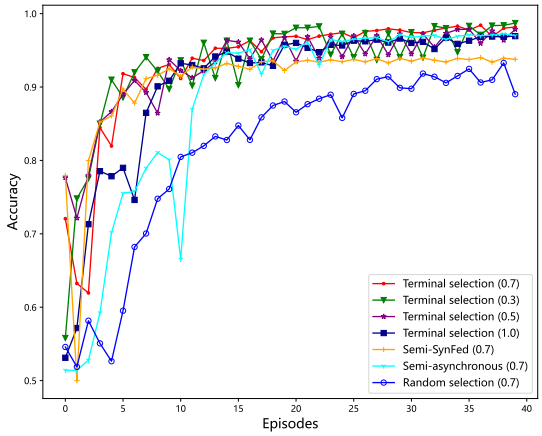
In the two datasets, with the degree of Non-IID is 0.5 and $\psi = 2$, the variation of accuracy achieved by EDT-SaFL through adopting different terminal selection ratios is shown in Fig. 11, in which, the accuracies achieved with different terminal selection ratios are compared to select the optimal terminal selection ratio.

With the CIFAR-10 in Fig. 11(a), the accuracy achieved with the terminal selection ratio of 0.3 is relatively poor, which indicates that a smaller number of selected terminals would result in insufficient data for the global model update, adversely affecting the model accuracy. If the selection ratio is 1.0, with the introduced significant computation and communication overheads, the terminals with low-quality data or sub-optimal training results would introduce unnecessary noise into the global model update, also adversely affecting the model accuracy with the decreased convergence and stability. Comparatively, if the selection ratio is 0.5 or 0.7, the achieved accuracy is the superior due to the better balance on computation/communication overheads and synchronous/asynchronous influences. Especially, the accuracy of selection ratio 0.5 is always ahead in the convergence process. Therefore, the selection ratio of 0.5 is taken as the optimal one in the CIFAR-10. With the Industrial-Equipment having less data samples in

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3595117



(a) Accuracy in the CIFAR-10.



(b) Accuracy in the Industrial-Equipment.

Fig. 11. Accuracy with different terminal selection strategies.

Fig. 11(b), if the low selection ratio is adopted, e.g., 0.3 and 0.5, that is, only a small number of terminals are selected in each training episode, which is difficult to fully learn the diversity of features in the dataset with less data samples, resulting in the significant fluctuation in the accuracy. If the selection ratio is 1.0, the low-quality data and training on some terminals would adversely affect the model accuracy. Therefore, from Fig. 11(b), the selection ratio of 0.7 would be taken as the optimal one through comprehensively considering the performance, stability and training cost.

With the aforementioned experimental analysis, a multi-criteria quantification method for determining the optimal terminal selection ratio can be adopted to enhance the robustness of EDT-SaFL. Through comprehensively considering the four critical metrics, training accuracy and loss, convergence, and stability, a discrete set of candidate ratios, e.g., {0.3, 0.5, 0.7, 1.0}, can be systematically evaluated. Specifically, each evaluation metric can be normalized to [0, 1] to ensure the scale consistency, and with a weighted sum, a composite score for each candidate ratio will be generated. The one achieving the highest score will be selected as the optimal terminal selection ratio for the EDT-SaFL, to enable the optimal balance between performance and cost.

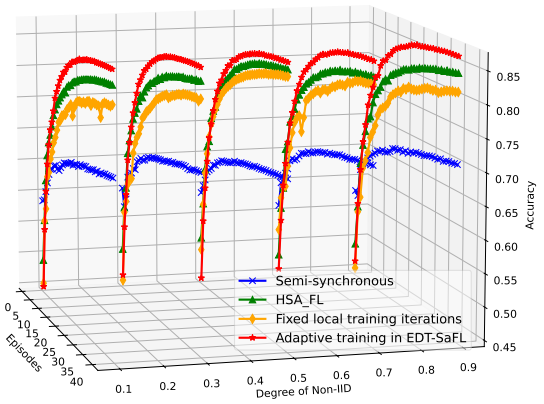
Furthermore, with the optimal terminal selection ratio, the variations of accuracy achieved by the three comparative strategies, e.g., Semi-SynFed [35], Semi-asynchronous [40] and Random selection [41], are also depicted in Fig. 11. In the Semi-SynFed, the terminal selection is conducted on the terminal capacities and the GNV in the last layer of the network, failing to fully consider the impact of local data on the model effectiveness. In the Semi-asynchronous, the terminal selection is conducted on the terminal capacities and the local data volume, ignoring the impact of local data quality on the model effectiveness. Therefore, the achieved accuracies of the two algorithms are relatively poor. With the CIFAR-10 providing sufficient data samples, the Random selection is superior to a certain extent. However, with the Industrial-Equipment providing less data samples, the Random selection might ignore those terminals with more and higher quality data, negatively impacting the effectiveness of global model update. For the ***Terminal Selection for Model Training*** in the EDT-SaFL, through introducing the UCB-based optimistic estimation strategy to determine the distinct terminal capability and the GNV indicating the importance of local data for global model update and convergence, it could select more suitable terminals for the local model training, resulting in better training accuracy and stability.

D. Evaluation of Self-Adaptation of Local Training Iterations

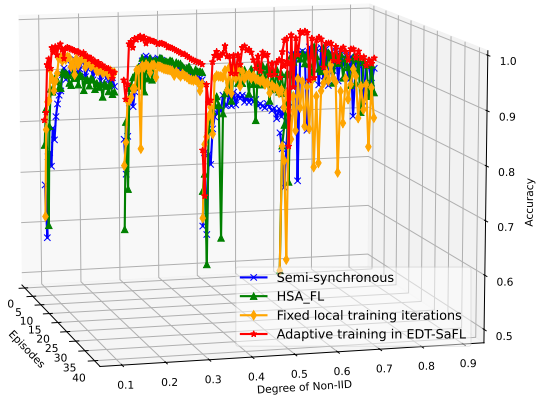
In the two datasets, the performance of ***Self-Adaptation of Local Training Iterations*** with $\psi = 2$ under different degrees of Non-IID is verified in Fig. 12. Integrated the proposed ***Terminal Selection for Model Training*** and ***Semi-asynchronous Global Aggregation*** in the EDT-SaFL, the three strategies, Fixed local training iterations with λ_0, Semi-synchronous [42] and HSA_FL [33], are selected for comparisons.

Along with the variation of degree of Non-IID, with the CIFAR-10 providing sufficient data samples, the achieved accuracies of four algorithms are relatively stable. However, with the Industrial-Equipment providing less data samples, the achieved accuracies show obvious fluctuations. Due to the lack of flexibility, the Fixed local training iterations might result in the negative affect, obtaining the sub-optimal local training results on the selected terminals. Moreover, in both Semi-synchronous and HSA_FL, the corresponding methods are adopted to adaptively adjust the local training iterations. For the Semi-synchronous, the global training time in each episode is configured with the product of the local training time consumed by the selected terminal with the weakest computing power and λ_0. With the configured global training time, the local training iterations on the selected terminals would be unbalanced due to the heterogeneity of computing power, negatively impacting the training accuracy. For the HSA_FL, although the local training iterations on the selected terminal are determined by the distinct computing power, the local data quantity and quality are not comprehensively considered. Therefore, under the change of Non-IID, the selected comparative algorithms are limited by the defect of adopted local training iterations strategies.

In contrast, for the EDT-SaFL, through defining the self-adaptation rules in terms of the relationship between the



(a) Accuracy in the CIFAR-10.



(b) Accuracy in the Industrial-Equipment.

Fig. 12. Accuracy with different local training iterations under the change of non-IID.

training capacity and the inherent computing power, and considering the local data quantity and quality on the training efficiency, the local training iterations could be adaptively adjusted on the selected terminal, improving the efficiency of resource utilization and further enhancing the local training efficiency. Due to less data samples in the Industrial-Equipment and the significant impact of local data that is Non-IID, the achieved accuracy also shows a fluctuation. Nevertheless, the EDT-SaFL also significantly outperforms the comparative strategies, fully demonstrating the adaptability of ***Self-Adaptation of Local Training Iterations*** in EDT-SaFL under different degrees of Non-IID.

VII. Conclusion

To address the DT construction/update in the DITEN for IIoT, through considering the heterogeneity of industrial terminals in the DT model training, an optimization problem of global loss minimization with constraint has been formulated to balance the training cost and loss in the whole life cycle of DT model. With this problem, a new approach of semi-asynchronous FL, named EDT-SaFL, has been proposed to offer a promising solution. To qualitatively and quantitatively evaluate the effectiveness of EDT-SaFL, the convergence has been demonstrated, the complexity and communication overhead have been analyzed, and the superior performance also has been verified based on the experimental results conducted on the datasets of CIFAR-10 and Industrial-Equipment.

With the constructed DT models, through comprehensively considering the uncertainty of resource demand, the finiteness of multi-domain edge resources, and the time variability of decision execution, we will further explore the mapping association between DT models and edge nodes, and the elastic adaptation between multi-domain edge resources and diversified DT services, to further improve the overall service performance of the DITEN for IIoT.

References

[1] H. Xu, J. Wu, Q. Pan, and et al., "A survey on digital twin for industrial internet of things: Applications, technologies and tools," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2569–2598, 2023.

[2] Y. Li and Y. Zhang, "Digital twin for industrial internet," *Fundamental Research*, vol. 4, no. 1, pp. 21–24, 2024.

[3] M. H. Hung, Y. C. Lin, H. C. Hsiao, and et al., "A novel implementation framework of digital twins for intelligent manufacturing based on container technology and cloud manufacturing services," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1614–1630, 2022.

[4] X. Huang, W. Wu, S. Hu, and et al., "Digital twin based user-centric resource management for multicast short video streaming," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 1, pp. 50–65, 2024.

[5] H. Yu, D. Yu, C. Wang, and et al., "Edge intelligence-driven digital twin of cnc system: Architecture and deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102418, 2023.

[6] D. Xia, J. Shi, K. Wan, and et al., "Digital Twin and Artificial Intelligence for Intelligent Planning and Energy-Efficient Deployment of 6G Networks in Smart Factories," *IEEE Wireless Communications*, vol. 30, no. 3, pp. 171–179, 2023.

[7] M. Picone, M. Mamei, F. Zambonelli, and et al., "A flexible and modular architecture for edge digital twin: Implementation and evaluation," *ACM Transactions on Internet of Things*, vol. 4, no. 1, pp. 1–32, 2023.

[8] M. Tao, L. Liao, R. Xie, and et al., "Saei-DT: Semi-asynchronous Edge Intelligence for Industrial Digital Twin Networks in 6G," *IEEE Network*, 2024, DOI: 10.1109/MNET.2024.3517532.

[9] P. Bellavista, C. Giannelli, M. Mamei, and et al., "Application-driven network-aware digital twin management in industrial edge environments," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7791–7801, 2021.

[10] G. N. Schroeder, C. Steinmetz, R. N. Rodrigues, and et al., "A methodology for digital twin modeling and deployment for industry 4.0," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 556–567, 2021.

[11] J. Zhang, H. Cui, A. L. Yang, and et al., "An intelligent digital twin system for paper manufacturing in the paper industry," *Expert Systems with Applications*, vol. 230, p. 120614, 2023.

[12] P. Jia, X. Wang, and X. Shen, "Accurate and efficient digital twin construction using concurrent end-to-end synchronization and multi-attribute data resampling," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 4857–4870, 2023.

[13] J. Li, S. Guo, W. Liang, and et al., "Digital Twin-Enabled Service Provisioning in Edge Computing via Continual Learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 7335–7350, 2024.

[14] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[16] R. Y. Zhong, X. Xu, E. Klotz, and et al., "Intelligent manufacturing in the context of industry 4.0: A review," *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[17] Z. Lv, "Digital twins in industry 5.0," *Research*, vol. 6, p. 71, 2023.

[18] X. Shen, J. Gao, W. Wu, and et al., "Holistic network virtualization and pervasive network intelligence for 6g," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 1–30, 2022.

[19] H. Yang, K. Y. Lam, L. Xiao, and et al., "Lead federated neuromorphic learning for wireless edge artificial intelligence," *Nature communications*, vol. 13, no. 1, p. 4269, 2022.

[20] W. Wu, M. Li, K. Qu, and et al., "Split learning over wireless networks: Parallel design and resource management," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 1051–1066, 2023.

[21] L. Jiang, H. Zheng, H. Tian, and et al., "Cooperative federated learning and model update verification in blockchain-empowered digital twin edge networks," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11154–11167, 2022.

[22] S. AbdulRahman, S. Otoum, O. Bouachir, and et al., "Management of digital twin-driven iot using federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3636–3649, 2023.

[23] W. Xiang, J. Li, Y. Zhou, and et al., "Digital twin empowered industrial iot based on credibility-weighted swarm learning," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, pp. 775–784, 2024.

[24] S. Wang, T. Tuor, T. Salonidis, and et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[25] Y. Lu, X. Huang, K. Zhang, and et al., "Communication-efficient federated learning for digital twin edge networks in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5709–5718, 2021.

[26] W. Yang, W. Xiang, Y. Yang, and et al., "Optimizing federated learning with deep reinforcement learning for digital twin empowered industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1884–1893, 2023.

[27] L. Tang, M. Wen, Z. Shan, and et al., "Digital twin-enabled efficient federated learning for collision warning in intelligent driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 3, pp. 2573–2585, 2024.

[28] C. Xu, Y. Qu, Y. Xiang, and et al., "Asynchronous federated learning on heterogeneous devices: A survey," *Computer Science Review*, vol. 50, p. 100595, 2023.

[29] C. H. Hu, Z. Chen, and E. G. Larsson, "Scheduling and aggregation design for asynchronous federated learning over wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 874–886, 2023.

[30] D. Van Huynh, V. D. Nguyen, S. R. Khosravirad, and et al., "Distributed communication and computation resource management for digital twin-aided edge computing with short-packet communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3008–3021, 2023.

[31] S. Han, H. Ding, S. Zhao, and et al., "Practical and robust federated learning with highly scalable regression training," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 10, pp. 13801–13815, 2024.

[32] D. Russo and B. Van Roy, "Eluder dimension and the sample complexity of optimistic exploration," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pp. 2256–2264, 2013.

[33] S. Chen, X. Wang, P. Zhou, and et al., "Heterogeneous semi-asynchronous federated learning in internet of things: A multi-armed bandit approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 5, pp. 1113–1124, 2022.

[34] L. Qian, P. Yang, M. Xiao, and et al., "Distributed learning for wireless communications: Methods, applications and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 326–342, 2022.

[35] F. Liang, Q. Yang, R. Liu, and et al., "Semi-synchronous federated learning protocol with dynamic aggregation in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4677–4691, 2022.

[36] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, and et al., "Fast-convergent federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 201–218, 2021.

[37] C. Xu, S. Liu, Z. Yang, and et al., "Learning rate optimization for federated learning exploiting over-the-air computation," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3742–3756, 2021.

[38] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous Federated Optimization," in *Proceedings of the 12th Annual Workshop on Optimization for Machine Learning*, pp. 1-11, 2020.

[39] B. McMahan, E. Moore, D. Ramage, and et al., "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

[40] J. Hao, Y. Zhao, and J. Zhang, "Time efficient federated learning with semi-asynchronous communication," in *Proceedings of the IEEE 26th International Conference*

on Parallel and Distributed Systems, pp. 156–163, 2020.

[41] L. Fu, H. Zhang, G. Gao, and et al., "Client selection in federated learning: Principles, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21811–21819, 2023.

[42] D. Stripelis, P. M. Thompson, and J. L. Ambite, "Semi-synchronous federated learning for energy-efficient training and accelerated convergence in cross-silo settings," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 5, pp. 1–29, 2022.

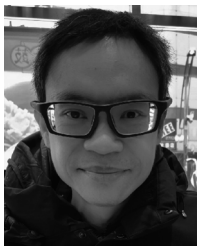
Ming Tao (Member, IEEE) received the B.S. degree in computer science and technology from Anhui University, Hefei, China, in 2007 and the M.S. and the Ph.D. degrees in computer application technology from South China University of Technology, Guangzhou, China, in 2009 and 2012, respectively. Currently, he is a Professor with the School of Computer Science and Technology at the Dongguan University of Technology, Dongguan, China. His primary research interests include AI+edge computing/cloud computing.

Lingling Liao (Student Member, IEEE) received the B.S. degree in electronic commerce from Guangdong Polytechnic Normal University, Guangzhou, China, in 2023. She is currently working toward the Master degree in School of Computer Science and Technology with Dongguan University of Technology, Dongguan, China. Her primary research interests include AI+edge computing/cloud computing.

Yin Zhang (Senior Member, IEEE) received the Ph.D. degree in system analysis and integration from Yunnan University, Kunming, China, in 2013. He is currently a Full Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. He was a Distinguished Scholar of Hubei, China. He has authored or coauthored more than 100 prestigious conference and journal papers, including 14 ESI Highly Cited Papers. His Google Scholar Citations reached 4700+ with an h-index of 33 and i10-index of 62. His research interests include mobile computing, edge intelligence, cognitive wireless communications, etc. He is a Co-Chair of IEEE Computer Society Big Data STC. He was an Editor or Associate Editor for IEEE Network, IEEE Systems Journal, Information Fusion, Journal of Circuits Systems and Computers, etc. He got the Systems Journal Best Paper Award of the IEEE Systems Council in 2018. He was named in Clarivate Analytics Highly Cited Researchers List in 2019.

Lei Liu (Member, IEEE) received the B.Eng. degree in communication engineering from Zhengzhou University, Zhengzhou, China, in 2010, and the M.Sc. and Ph.D. degrees in communication engineering from Xidian University, Xi'an, China, in 2013 and 2019, respectively. He is currently an Associate Professor with the Guangzhou Institute of Technology, Xidian University. His research interests include vehicular ad hoc networks, intelligent transportation, mobile edge computing, and the Internet of Things.

Geyong Min (Member, IEEE) received the BSc degree in computer science from the Huazhong University of Science and Technology, China, in 1995 and the PhD degree in computing science from the University of Glasgow, United Kingdom, in 2003. He is a professor of High Performance Computing and Networking with the Department of Computer Science, the University of Exeter, United Kingdom. His research interests include computer networks, wireless communications, parallel and distributed computing, ubiquitous computing, multimedia systems, modeling, and performance engineering.



Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Canada, in 2008. He is a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of sustainability, edge intelligence, decentralized machine learning, and incentive mechanism design.

Schahram Dustdar (Fellow, IEEE) received the PhD degree in business informatics from the University of Linz, Austria, in 1992. He is currently a full professor of computer science (informatics) with a focus on internet technologies heading the Distributed Systems Group, TU Wien, Wein, Austria. He has been the Chairman of the Informatics Section of the Academia Europaea, since 2016. Since 2013, he has been the member of Academia Europaea: The Academy of Europe, Informatics Section, Section Committee of Informatics of the Academia Europaea, since 2015, and IEEE Conference Activities Committee (CAC), since 2016. He was a recipient of the ACM Distinguished Scientist Award, in 2009 and the IBM Faculty Award, in 2012. He is an associate editor of the IEEE Transactions on Services Computing, ACM Transactions on the Web, and ACM Transactions on Internet Technology. He is on the editorial board of IEEE.