An Effective Federated Learning Verification Strategy and Its Applications for Fault Diagnosis in Industrial IoT Systems

Yuanjiang Li, Yunfeng Chen, Kai Zhu, Cong Bai, *Member, IEEE*, and Jinglin Zhang

***Abstract*—Due to the diverse equipment and uneven load distribution in industrial environments, data regarding faults are often unbalanced. Moreover, data and models from clients may become contaminated or damaged, affecting diagnostic performance. To overcome these problems, this study proposes a stacking model for diagnosing interturn short circuit (ITSC) faults in permanent magnet synchronous motors (PMSMs). Federated learning (FL) is used to train the model to increase data security and overcome data islanding in distributed scenarios. Moreover, an improved verification strategy was adopted to select appropriate client models in each round to update the FL global model. We created a secondary server-side data set to validate the client weightings. The data set contains clean sample data for all ITSC fault categories. By calculating the fault diagnosis accuracy of the global model on the auxiliary data set, the model eliminates low-quality clients with uneven fault distributions. The improved particle swarm optimization (PSO) is used to optimize the weight coefficients of clients involved in aggregation, improving the robustness of the aggregation strategy under a joint learning system. In evaluation experiments, compared with the federated average (FedAvg) model, the proposed dynamic verification model exhibited the better diagnostic accuracy in situations of data imbalance, incurred lower communication costs, and prevented local oscillations in the model.**

***Index Terms*—Federated learning (FL), improved particle swarm optimization (PSO), permanent magnet synchronous motor (PMSM), verification strategy.**

I. Introduction

INTELLIGENT manufacturing [1], [2] is the integration of manufacturing and information technologies, such as artificial intelligence. It includes more than a dozen manufacturing paradigms, such as flexible manufacturing, cloud manufacturing, and distributed manufacturing. These paradigms reflect the continual integration and development of information technology and the manufacturing industry and reflect the digital, networked, and intelligent characteristics of manufacturing industry to varying degrees. With the rapid development of artificial intelligence algorithms, data-driven mechanical fault diagnosis methods have been successfully developed. In complex smart manufacturing scenarios, collecting a large amount of high-quality and accurately labeled fault data for training deep learning models is difficult. In particular, obtaining data regarding moderate and severe faults is generally impossible in real-world production lines. Therefore, the lack of and imbalances in fault samples substantially affects the diagnostic performance of data-driven models. Intelligent manufacturing systems include multiple combined tasks such as resource allocation, production, failure analysis, and health management. Central computing resources are often limited, and a reliance on traditional centralized training methods can burden the entire intelligent manufacturing system; thus, achieving iterative training and updating of the fault diagnosis model is challenging.

Permanent magnet synchronous motors (PMSMs) are a core component of numerous types of intelligent manufacturing equipment. Many of these motors are scattered throughout manufacturing systems, and their state directly affects the performance of these intelligent systems. If a fault occurs, it affects the operation of the entire intelligent factory. Therefore, the continuous monitoring of PMSM status is critical in intelligent manufacturing. An interturn short circuit (ITSC) [3] is a common and extremely destructive PMSM fault. If a minor ITSC fault goes unnoticed, the short-circuited current continues to rise and the resulting high temperatures cause demagnetization and the eventual failure or destruction of the motor. By arranging edge computing nodes on multiple PMSM sides and using federated learning (FL), one can fully utilize the unbalanced sample resources at each edge to train an ITSC fault diagnosis model with excellent accuracy, generalizibility, and efficiency (i.e., it achieves good performance without placing a high computational burden on the intelligent manufacturing system) [4].

Fig. 1 presents the principles of FL in an Industrial Internet of Things (IIoT) system. Internet of Things (IoT) [5], [6] devices with edge computing functions can be deployed in

Manuscript received 28 December 2021; accepted 7 February 2022. Date of publication 23 February 2022; date of current version 7 September 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0126100, and in part by the Key Research and Development Program of Jiangsu Province under Grant BE2021093. *(Corresponding author: Jinglin Zhang.)*

Yuanjiang Li and Yunfeng Chen are with the Ocean College, Jiangsu University of Science and Technology, Zhenjiang 212013, China (e-mail: liyuanjiang@just.edu.cn; 199030019@stu.just.edu.cn).

Kai Zhu is with the School of Automobile and Traffic Engineering, Jiangsu University of Technology, Changzhou 213163, China (e-mail: fatkyo@jsut.edu.cn).

Cong Bai is with the College of Computer Science and Technology and the Key Laboratory of Visual Media Intelligent Processing Technology of Zhejiang Province, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: congbai@zjut.edu.cn).

Jinglin Zhang is with the School of Control Science and Engineering, Shandong University, Jinan 250061, China, and also with the School of Computer Science and Engineering, Linyi University, Linyi 276012, China (e-mail: jinglin.zhang37@gmail.com).

Digital Object Identifier 10.1109/JIOT.2022.3153343

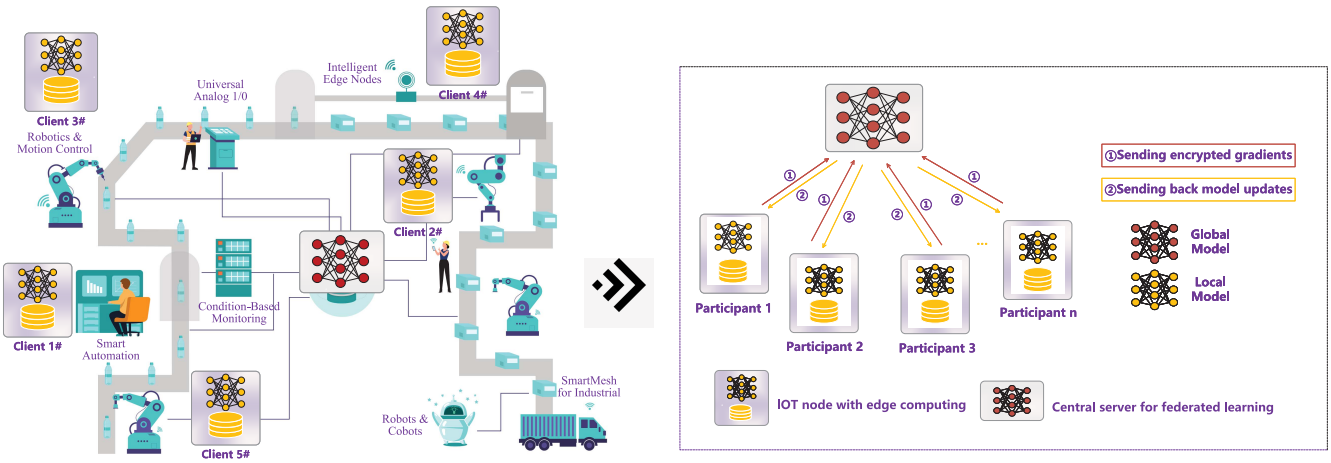

Fig. 1. FL in an industrial IoT system where IoT devices with edge computing functionality are deployed in a smart factory. FL can be used to train the model and fully utilize data from various devices without compromising security.

smart factories, and FL can be used for model training. FL features the adoption of a distributed learning architecture that fully utilizes the computing power of IIoT devices and combines global model updating with privacy protections for local data [7]. Instead of using all the edge device data, the system can use the weights of each local model to aggregate parameters and to update both global and local models.

In this article, a fault diagnosis model for PMSM turn-to-turn short circuits based on a stacked self-coding network is proposed, and the model is trained using FL to overcome the problem of data islands in a distributed scenario. In addition, an improved dynamic verification strategy is used to adaptively adapt the model aggregation process and decrease the negative effects of scattered low-quality data. The following are our major contributions.

1) We address the problem of insufficient feature learning ability in shallow neural network models. Specifically, we propose a stacked diagnosis network based on a normalized sparse autoencoder (NSAE) and denoised autoencoder (DAE) that is trained using FL to fully utilize the fault data from different PMSM clients and increase the sparsity and robustness of the fault diagnosis model.
2) Our method reduces the influence of low-quality customers when the FL global model is updated; it does so through our proposed improved weight dynamic verification method to select weights uploaded by high-quality clients prior to model aggregation. The weights submitted by the client are verified using the dynamically updated verification set to eliminate the influence of abnormal customers on the global model. An improved particle swarm optimization (PSO) algorithm is also used to optimize the coefficients of the weights of participating clients to improve the robustness of the aggregation strategy under the FL system.
3) We conducted experiments to demonstrate that the proposed fault diagnosis model: a) Performs excellently in identifying different degrees of PMSM ITSC faults and b) can fully acquire unbalanced sample knowledge—by using a dynamic verification strategy and FL—from multiple PMSM clients in an intelligent manufacturing system to obtain a global fault diagnosis model with excellent performance under nonindependent and identically distributed (Non-IID) conditions.

The remainder of this article is organized as follows. The principle underlying ITSC faults is presented in Section II. The stacked model and the improved validation strategy are described in Section III. The experimental results and a performance evaluation are discussed in Section IV. Finally, the study is concluded in Section V.

A. Related Works

In recent years, many smart data-driven fault diagnosis approaches have been developed and used successfully in a variety of industrial applications [8], [9]. A variety of methods for diagnosising PMSM ITSC faults has also been formulated. Zhao *et al.* [3] analyzed the causes of ITSC and documented their hazards. They also adopted mathematical modeling and simulations to uncover the factors affecting the amplitude of short-circuit current, such as load status and speed. Qi *et al.* [10] proposed an equivalent model to describe the relationship between fault severity and short circuit turns, and they described the effects of ITSC faults on the electrical parameters of a PMSM. Traditional signal processing methods use shallow models, and extracting the implicit relationship between different data features in the input data set is difficult, leading to unsatisfactory diagnostic results [11]. Deep learning [12] enables the establishment of a multilevel representation, transforms big data features into more abstract modules, and converts high-dimensional nonlinear features into low-dimensional features by using hidden layers. This process effectively captures hidden information in big data and greatly improves the learning ability of the algorithm; deep learning methods have achieved great success in data-driven fault diagnosis. For example, Wang *et al.* [13] proposed a mechanical fault diagnosis method based on a convolutional neural network (CNN) and achieved good diagnostic results. Liu *et al.* [14] used a stacked sparse denoising self-encoder with a softmax classifier for robust fault identification in

a complex industrial process. In addition, Yao *et al.* [15] proposed ResNet-LSTM, which used neural networks and long short-term memory networks (LSTMs) to adaptively extract fault features by using deep residual convolutional networks and LSTM, effectively reducing the difficulty of deep neural network training.

Motors are distributed across various devices in smart factories, and these devices play an important role in the manufacturing process. PMSMs differ in their working conditions; thus, a centralized training method must first collect data from mobile devices and then send these data to a cloud-based server or data center for processing and training. This approach not only increases the communication burden and calculation loss of the system but also fails to improve the robustness and generalization of the model. In some scenarios involving user data, these methods may even lead to serious privacy concerns and data leakage. Therefore, FL was developed to protect the privacy of user data in model learning in a big data environment.

FL [16], [17] is a machine learning framework designed for privacy and data security; devices can collaborate without exchanging data, greatly increasing the privacy of user data. Edge computing [18], [19] can also alleviate the computing pressure of the cloud center. Data calculations, storage, and applications can be completed at the network edge (i.e., at data source) without transmission to the cloud. The federated average (FedAvg) algorithm [20] was the first practically proposed communication-efficient algorithm. The theoretical analysis and algorithmic improvement of FedAvg is an important research topic in federated optimization. Hao *et al.* [21] formalized the basic principles of FL and discussed its potential applications in industrial scenarios. Inspired by edge computing, Ye *et al.* [22] proposed edge FL (EdgeFed), which separates the process of updating local models to improve learning efficiency and decrease the frequency of global communications. Zhang *et al.* [23] proposed an FL method for machine fault diagnosis where a dynamic validation scheme is used to update weights before model aggregation is performed, guaranteeing data privacy among various clients. Moreover, Chen *et al.* [24] proposed an asynchronous model updating strategy and a temporally weighted aggregation method, reducing the communication costs and improving the learning performance in FL.

FL is a feasible solution that can solve the problem of data islands and can achieve privacy protection and data security. Similar to that in existing machine learning methods, the first problem in FL is data collection. In practice, local data from various devices are generally Non-IID. The extent and type of the Non-IID scenario greatly affect the training efficiency of machine learning models under FL and may even greatly decrease model performance; thus, a theoretical analysis of the convergence of the federated optimization algorithm is extremely difficult [25]. To overcome this challenge, Wang *et al.* [26] proposed an FL-based AMC (FedeAMC), which introduces balanced cross-entropy as a loss function to solve the class imbalance problem for each local client. Zhang *et al.* [27] proposed a new FL framework FedSens, which is specifically designed to overcome the of class imbalance problem, improve the accuracy of the model, and reduce the energy cost of edge devices. In addition, Duan *et al.* [28] constructed the Astraea self-balancing FL framework, which uses adaptive data enhancement and downsampling to alleviate the global imbalance. Compared with the ordinary FedAvg algorithm, Astraea improves diagnostic accuracy and further reduces communication costs.

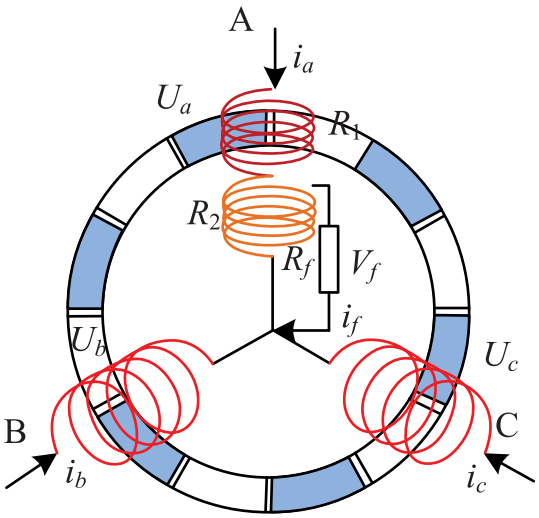

Fig. 2. Schematic of a PMSM ITSC. It shows the mathematical model of single-phase ITSC fault of a PMSM. When a number of short-circuit coils increase, it will irreversibly demagnetize and burn the PMSM.

II. ITSC Faults of PMSMs

Fig. 2 presents the fault model of the PMSM if an ITSC fault occurs in the phase A winding. Here, n is the total number of coils in the phase A winding and each coil has N turns. N_f is the number of short-circuited turns. Thus, the turn ratio can be defined as $\mu = N_f/(nN)$; this ratio directly affects the severity of the ITSC fault of the PMSM. As illustrated in the model in Fig. 2, if the generated fault current is i_f when an ITSC occurs, the fault resistance under the damaged insulation layer is R_f.

In the stationary coordinate system abc, when an ITSC fault occurs, the phase voltage equation of the PMSM can be described as follows:

$$u_{abc} = R_{sh} \cdot i_{abc} + \frac{d}{dt}\left(L_{sh} \cdot i_{abc} + \lambda_{PM,abc}\right) - R_{sh} \cdot FP\mu i_f - \frac{d}{dt}\left(L_{sh} \cdot FP\mu i_f\right) \tag{1}$$

where

$$u_{abc} = \begin{bmatrix} u_a & u_b & u_c \end{bmatrix}^T \tag{2}$$

$$i_{abc} = [i_a \quad i_b \quad i_c]^T \tag{3}$$

$$R_{sh} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \tag{4}$$

$$L_{sh} = \begin{bmatrix} L_{ms} + L_{ls} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{ms} + L_{ls} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{ms} + L_{ls} \end{bmatrix} \tag{5}$$

$$\lambda_{PM,abc} = \lambda_{PM}\left[\cos(\theta) \quad \cos\left(\theta - \frac{2\pi}{3}\right) \quad \cos\left(\theta + \frac{2\pi}{3}\right)\right]^T \tag{6}$$

$$FP = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T. \tag{7}$$

The terms in the formulas are defined as follows.

u_{abc}: Stator winding three-phase voltage matrix.

u_a, u_b, u_c: Phase voltage of the three stator winding phases.

i_{abc}: Stator winding three-phase current matrix.

i_a, i_b, i_c: Phase current of the three stator winding phases.

R_{sh}: Stator resistance matrix.

R_s: Stator resistance.

L_{sh}: Stator inductance matrix.

L_{ms}: Self induction.

L_{ls}: Transformer coupling coefficient.

$\lambda_{PM,abc}$: Three-phase flux matrix.

λ_{PM}: Flux linkage.

θ: Three-phase alternating current electric potential angle.

FP: Fault phase vector matrix.

If an ITSC fault occurs in phase b or c of the PMSM, FP can be expressed as $\mathrm{FP} = [0 \quad 1 \quad 0]^T$ or $\mathrm{FP} = [0 \quad 0 \quad 1]^T$.

The corresponding fault voltage can be expressed as

$$\begin{aligned} u_f &= R_f i_f \\ &= \mu \mathrm{FP}^T \cdot u_{abc} - \mu(1-\mu)R_s i_f \\ &\quad - \mu^2\left[\left(\frac{n}{1-\gamma} - 1\right)L_{ms} + (n-1)L_{ls}\right]\frac{di_f}{dt} \end{aligned} \tag{8}$$

where u_f indicates the fault voltage and γ represents the coupling coefficient of the flux linkage.

Negative sequence current [29] is a typical criterion for judging the status of a PMSM because an ITSC fault causes a negative sequence current, leading to an imbalance of three-phase current. In a three-phase circuit, any current component can be decomposed into three phase components, such as the positive sequence, negative sequence, and zero sequence. Under ideal conditions, the motor is in a three-phase symmetric state, and only the positive sequence component exists. If an ITSC fault occurs, the short-circuit loop current forms a pulsating magnetic field, which generates a reverse magnetomotive force in the corresponding short-circuit winding. Then, negative sequence components are produced, causing a three-phase imbalance. In this section, the negative sequence current under three-phase asymmetry is calculated based on vector control.

First, high-order harmonics are ignored, and the current calculation is performed for only the fundamental wave. In general, the neutral point of the motor is not grounded and zero sequence current is unavailable. In this situation, the three-phase current is expressed as follows:

$$\begin{aligned} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} &= I^+ \begin{bmatrix} \cos(\omega t + \varphi_1) \\ \cos\left(\omega t + \varphi_1 - \frac{2}{3}\pi\right) \\ \cos\left(\omega t + \varphi_1 + \frac{2}{3}\pi\right) \end{bmatrix} \\ &\quad + I^- \begin{bmatrix} \cos(\omega t + \varphi_2) \\ \cos\left(\omega t + \varphi_2 + \frac{2}{3}\pi\right) \\ \cos\left(\omega t + \varphi_2 - \frac{2}{3}\pi\right) \end{bmatrix} \end{aligned} \tag{9}$$

where i_A, i_B, and i_C are the three-phase currents, I^+ and I^- are the amplitudes of the positive and negative sequence currents, ω is the fundamental frequency of the current, and φ_1 and φ_2 are the phase angles of positive and negative sequence currents, respectively.

The short-circuit current is the sum of the positive and negative sequence currents

$$I = I^+ e^{j(\omega t + \varphi_1)} + I^- e^{j(-\omega t + \varphi_2)}. \tag{10}$$

Equation (9) can be converted into an amplitude relationship between the positive sequence and negative sequence current

$$\begin{bmatrix} I^+ \\ I^- \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \end{bmatrix} \cdot \begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix} \tag{11}$$

where α is the operator and $\alpha = e^{j(2/3)\pi}(120°)$ and I_A, I_B, and I_C are the current amplitudes of phases A, B, and C, respectively.

The phase angles φ_1 and φ_2 can be obtained using the law of cosines

$$\begin{cases} \cos\varphi_1 = \frac{I_A^2 + I_B^2 - I_C^2}{2I_A I_B} \\ \cos\varphi_2 = \frac{I_A^2 + I_C^2 - I_B^2}{2I_A I_C} \end{cases} \tag{12}$$

The negative sequence current $I^- e^{j(-\omega t + \varphi_2)}$ of the short-circuit winding can be determined using (10)–(12).

The PMSM has a three-phase asymmetry due to several reasons, such as short circuits between turns, load fluctuations, and an unbalanced power supply voltage. This asymmetry is reflected in the negative sequence component. However, adopting these negative sequence current characteristics results in substantial error that affects the training results. As a result, the negative sequence current is tuned to improve the data set's robustness. The negative sequence current obtained in a balanced state equals the required negative sequence current value $I_{(\mathrm{fault})}$ caused by the ITSC fault. This voltage imbalance results in an asymmetry reflected in the negative sequence current component, which is the error negative sequence current $I_{(\mathrm{load})}$. In severe cases, $I_{(\mathrm{load})}$ may be higher than $I_{(\mathrm{fault})}$.

The total negative sequence current is as follows:

$$I^- = I^-_{(\mathrm{fault})} + I^-_{(\mathrm{load})}. \tag{13}$$

To remove the error negative sequence current, we first consider a situation without an ITSC fault. The negative sequence current and voltage can be obtained by using the three-phase current and voltage

$$V^- = \frac{1}{3}\left(V_A + \alpha^2 V_B + \alpha V_C\right) \tag{14}$$

$$I^- = \frac{1}{3}\left(I_A + \alpha^2 I_B + \alpha I_C\right). \tag{15}$$

At this time, the negative sequence current is $I^- = I_{(\mathrm{load})}$; the negative sequence impedance Z^- can then be obtained as follows:

$$Z^- = \frac{V^-}{I^-}. \tag{16}$$

For a case with an ITSC, $I_{(\mathrm{fault})}$ can be calculated after the known negative sequence impedance is subtracted from

Algorithm 1 Optimizing Negative Sequence Current Flow

1: Calculate (V^-, I^-) according to the three-phase voltage
2: **if** u_a, u_b, u_c are unbalanced
3: **if** there is no ITSC
4: $I_{(load)} = I^-$, and calculate negative sequence resistance $Z^- = \frac{V^-}{I_{(load)}}$
5: **else**
6: $I_{(load)} = \frac{V^-}{I^-}$, $I_{(fault)} = I^- - I_{(load)}$
7: **else** $I_{(load)} = 0$, $I_{(fault)} = I^-$
8: Obtain the optimized negative sequence current $I_{(fault)}$

the total negative sequence current (phase subtraction). The method is presented in Algorithm 1.

As an indicator of motor health, electromagnetic torque is closely related to safety, work efficiency, and motor life. This torque is mainly influenced by the magnetic field strength and winding current. We assume that the magnetic core is saturated and that the counter electromotive force is sinusoidal. Consequently, the electromagnetic torque of a PMSM under normal conditions is as follows:

$$T_e = \sum_{i \in S} K_i I_i \tag{17}$$

where S is the phase winding; K_i is the electromotive force coefficient of the ith phase, and I_i is the phase current of the ith phase.

When a PMSM fails, the corresponding electromagnetic torque is as follows:

$$T_e = \sum_{i \in S_n} K_i I_i + \sum_{j \in S_f} K_j I_j = T_u + T_r \tag{18}$$

where S_n is the normal phase winding, S_f is the faulty phase winding, T_u is the controllable electromagnetic torque that guarantees the operation of the motor following the fault, and T_r is the uncontrollable electromagnetic torque that depends on the motor fault type.

In the case of a fault, the instantaneous phase current can be expressed as follows:

$$I_{jn} = (T_e - T_r)) \frac{K_{jn}}{5K_m^2 - \sum_{jf \in S_f} K_{jf}^2} \tag{19}$$

where K_m is the peak value of the phase electromotive force coefficient.

Equation (19) reveals that high-order harmonic components exist in the instantaneous phase current in the case of a fault. Under the short circuit condition, the short-circuit current is limited to the rated current; thus, the torque caused by the ITSC can be expressed as follows:

$$\begin{aligned} T_r &= \sum_{i \in S_f} -K_m \sin\left(\theta_i - \frac{\pi}{2}\right) \\ &= \sum_{i \in S_f} K_m I_m \sin\theta_i \cos\theta_i \\ &= \sum_{i \in S_f} \frac{K_m I_m}{2} \sin(2\theta_i) \end{aligned} \tag{20}$$

where I_m is the peak value of the phase current and θ_i is the phase angle.

Equation (20) demonstrates that the peak value of the phase current directly causes changes in the electromagnetic torque. Moreover, if an ITSC occurs in a phase winding of a PMSM, the overall amplitude of the phase current increases. If the number of short-circuit turns increases and the fault becomes more severe, the phase current peak value increases at an increasing rate, driving a rapid increase in the electromagnetic torque. This analysis proves that electromagnetic torque in a PMSM can be used to identify an ITSC.

III. Proposed Method

A. NSAE and DAE

Traditional autoencoders require for the number of hidden layer neurons to be fewer than the number of input layer neurons. A sparse autoencoder (SAE) imposes sparsity constraints on hidden layer neurons, inhibiting most neurons. Only a small number of random neurons remain active; this increases the sparsity of the network, improves the sparse representation of the input features, and facilitates data classification. Thus, a sparseness penalty unit is added to the loss function as follows:

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^{m} a_j(x_i) \tag{21}$$

$$\sum_{j=1}^{h} \mathrm{KL}\left(\rho \| \hat{\rho}_j\right) = \sum_{j=1}^{h} \left(\rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \right). \tag{22}$$

Here, $\mathrm{KL}(\cdot)$ is the KL divergence, $\hat{\rho}_j$ is the average activation of the training data on the jth neuron, a_j is the activation degree of the jth neuron, and ρ is the sparse rate of the sparse regular term. Thus, the loss function of an SAE is defined as follows:

$$J_{\mathrm{SAE}} = \sum L(x, y) + \beta \sum_{j=1}^{h} \mathrm{KL}\left(\rho \| \hat{\rho}_j\right). \tag{23}$$

In contrast to traditional SAE models, the rectified linear unit (ReLU) and $L1$ norm are employed to replace the sigmoid function and the KL divergence function in an NSAE [30], respectively. The sparse activation of ReLU is beneficial for training deep networks, whereas the $L1$ norm produces a sparse feature representation of the input data with fewer parameters, facilitating its implementation in IIoT devices. Due to the invisibility of ITSC characteristics and the diversity of triggering factors, access to high-quality fault samples with higher severity is challenging. Moreover, an unbalanced training set is a critical factor that affects the performance of fault model diagnoses; therefore, a soft orthonormality constraint is introduced into the cost function of the NSAE to improve its learning capability, to identify different features, and to guarantee improvement in feature learning.

Given a set of unlabeled data $\{X_m, K_n\}_{m=1}^{M}$, where K_n denotes the severity of an ITSC and M is the number of samples, in order to learn various features from the original data, orthogonal constraints are added to the weight matrix of NSAE. The cost function of NSAE with hard orthogonal

constraints is shown as follows:

$$\min_{W} \frac{1}{2M}\sum_{m=1}^{M}||\hat{x_m}-x_m||_2^2+\lambda\sum_{m=1}^{M}||h_{\mathrm{NSAE}}^m||_1$$
$$\text{s.t. } W_{\mathrm{NSAE1}}W_{\mathrm{NSAE2}}=E \tag{24}$$

where λ is the regularization coefficient of the NSAE, after W_{NSAE1} and W_{NSAE2} are replaced by W and W^T, respectively, the optimization problem in the NSAE network boils down to minimizing the following cost function:

$$J_{\mathrm{NSAE}}=\frac{1}{2M}\sum_{m=1}^{M}||W^T\sigma_r(Wx_m)-x_m||_2^2+\lambda\sum_{m=1}^{M}||\sigma_r(Wx_m)||_1$$
$$\text{s.t. } \sigma_r(z)=\begin{cases}0, \text{ if } z<0\\ z, \text{ if } z\geq 0\end{cases} \tag{25}$$

where σ_r represents the ReLU function. This strategy effectively reduces the number of parameters and accelerates the training process. The monitoring data of the PMSMs is greater than 0; thus, $\sigma_r(Wx_m)$ does not cause a severe loss in the ITSC information. In addition, the calculation of the gradient of J relative to W can be expressed as

$$D=W^T\sigma_r(Wx)-x \tag{26}$$
$$\nabla J=\frac{\partial J_{\mathrm{NSAE}}}{\partial W}=\left[\left(\frac{1}{M}WD+\lambda\cdot\mathrm{sgn}\right)\cdot\sigma_r'(Wx)\right]x^T+\frac{1}{M}\sigma_r(Wx)D^T \tag{27}$$

where sgn is the result of the sign function of $\sigma_r(W_x)$, $\sigma_r^{'}$ is the derivative function of ReLU, and x is the matrix form of x_m.

After NSAE weight optimization is completed, its weight matrix is actually normalized by orthogonal constraints; in addition, NSAE needs sufficient training to accurately calculate the probability density of ITSC, and then enter the softmax classification network, efficiently predict short-circuit faults between turns.

DAE and autoencoder have the same network structure and the same learning goals. The DAE network structure also includes an encoder and decoder. The difference is that a DAE actively adds random noise to the sample data during training. Learning to eliminate noise interference results in the obtaining of a more robust feature expression and an improvement in the model's generalization ability for input data. Given a training set, the DAE model parameters $\{W_1, W_2, b_1, b_2\}$ can be learned by minimizing the following objective function:

$$\left\{\hat{W}_1,\hat{W}_2,\hat{b_1},\hat{b_2}\right\}=\underset{W_1,W_2,b_1,b_2}{\mathrm{argmin}}\{L_{DAE}(W,B)\}=\underset{W_1,W_2,b_1,b_2}{\mathrm{argmin}}\left(\frac{1}{2n}\sum_{i=1}^{n}L\big(g\big(f\big(\tilde{X}_i\big)\big),X_i\big)+\frac{\lambda}{2}\sum_{r=1}^{2}||W_r||_F^2\right) \tag{28}$$

where $\tilde{X}_i$ is the corrupted version of X_i by a stochastic mapping; that is, $\tilde{X}_i\sim q_m(\tilde{X}_i|X_i)$.

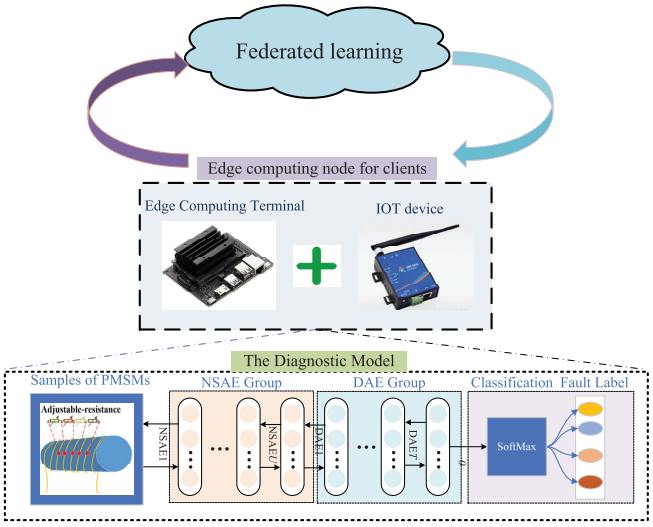

Fig. 3. Stacked model based on NSAE and DAE. This model is deployed on edge computing nodes that can perform deep learning operations. IoT devices are also deployed on the edge nodes to facilitate FL.

B. Stacked Diagnosis Model

An ITSC in a PMSM is a progressive fault generated by physical, electrical, vibrational, and other factors; thus, it is difficult to establish a precise mathematical model to describe the fault state for all conditions. Due to the complexity of the PMSM, only limited high-quality data are available to construct a defective fault model. To improve the value of mined data, a stacked structure based on NSAE and DAE is proposed in this article (Fig. 3). This model is deployed on edge computing nodes that can perform deep learning operations. IoT devices are also deployed on the edge nodes to facilitate FL.

In this design, the samples are collected from simulated experimental devices to simulate various states of a PMSM. These samples are then adopted to train the proposed stacked network model based on NSAE and DAE. Finally, the SoftMax layer is applied to discriminate between actual and labeled samples. In accordance with the number of local clients and the limited computing resources of IIoT devices, the stacked structure is determined in the first experiment described in Section IV.

The training process of the stacked network includes pretraining and fine-tuning. Pretraining is used for weight initialization, and each layer of the network is trained in turn by using the greedy strategy. All NSAEs and DAEs are individually pretrained. Similarly, the softmax classifier is also pretrained based on the error between the actual output and the input labels. Weights are then initialized, the appropriate optimization algorithms are selected to fine-tune the network, and the best parameters are obtained. Specifically, the fault diagnosis algorithm based on the stacked network comprises the following steps.

1) Data Preprocessing: First, a linear function transformation is used to normalize the original data. The linear function conversion formula is as follows:

$$\hat{D}_{\mathrm{train}}=\frac{D_{\mathrm{train-MinValue}}}{\mathrm{MaxValue}-\mathrm{MinValue}} \tag{29}$$

where D_{train} is the original sample data, $\hat{D}_{\text{train}}$ is the sample data after conversion, and MaxValue and MinValue are the maximum and minimum values in the sample, respectively.

2) Determine the Network Structure: Subsequently, the stacked network structure is determined; that is, the number of network layers and number of neuron nodes in each layer are determined, and the weight w and bias b of each layer of the encoder are randomly initialized.

3) Pretraining: A stacked autoencoder network is formed by stacking M NSAEs and N DAEs with n input layer nodes, $M + N$ hidden layers, and $s = [s_1, s_2, \ldots, s_l]$ hidden layer nodes; the weight matrix is $w = [w_1, w_2, \ldots, w_l, w_{l+1}]$, and the bias matrix is $b = [b_1, b_2, \ldots, b_l, b_{l+1}]$. The encoding process of the stacked autoencoding network is as follows:

$$h_t = f(z_t) \tag{30}$$

$$z_t = w_{(t,1)} h_{t-1} + b_{(t,1)} \tag{31}$$

where $t \in \{1, 2, \ldots, l\}$, and the decoding process is as follows:

$$h_{(t+l+2)} = f\big(z_{(t+l+2)}\big) \tag{32}$$

$$z_{(t+l+3)} = w_{(l+2-t,2)} h_{(l+2+t)} + b_{(l+2-t,2)}. \tag{33}$$

For the layer-by-layer training of the stacked autoencoder network, the first M NSAEs are trained followed by training the N DAEs. The outputs of the first trained encoder are the first-order features of the original input; these features are used as the input of the second autoencoder, and the second-order features of the original input are obtained through training. Due to these iterations, stacked networks with deeper layers have better feature extraction compared with shallow neural networks, and they also have clear advantages for handling high-dimensional data.

4) Fine-Tuning: By using a small amount of labeled data as input, the system uses the network weights w and biases b after unsupervised layer-by-layer training as the initial parameters of the overall network. The parameters, including the training epochs, sparsity rate, learning rate, and dropout rate, are first set. A Softmax classifier is added at the end of the network for supervised classification; finally, a suitable optimization algorithm is used to update the weights and biases in each iteration to fine-tune the parameters of the entire deep network. For N input samples, the cross-entropy cost function is used to calculate the error of the Softmax layer as follows:

$$\begin{aligned} L(w) &= \frac{1}{N} \sum_{n=1}^{N} H(p_n, q_n) \\ &= \frac{1}{N} \sum_{n=1}^{N} \big[y_n \log \hat{y_n} + (1 - y_n) \log (1 - \hat{y_n}) \big] \end{aligned} \tag{34}$$

where p_n is the actual label, q_n is the predicted value, and $H(p_n, q_n)$ is a measure of the difference between p_n and q_n.

C. FL With Dynamic Validation

The complexity of the diagnostic algorithm, the performance of the edge devices, and the communication time still substantially affect the performance of the global models. FL enables the use of different gradient or weight information acquired from local scenarios for parameter aggregation. Due to the presence of such biased information, these model parameters obtained by weighting on the server side are not suitable for improving the performance of the client model. If FedAvg is adopted to train deep learning networks, it is difficult to achieve a global-optimization model in the absence of any consideration of the inherent characteristics of the local data. In a smart factory supported by 5G and IIoT, an ITSC fault is unlikely to occur in each PMSM; thus, the spatial distribution of samples in the client is unbalanced. Moreover, client conditions are time varying and asynchronous, resulting in poor convergence of the global model after the gradient information has been weighted by the server.

To balance the generality and locality of each training round, an improved validation strategy was developed to diminish the negative influence of local information in the model aggregation process. On the server, we constructed a validation set with definite labels to evaluate the models uploaded by the clients in turn. Accuracy indices, such as the mean and variance, were used to set the threshold, select clients with high quality for weighting, and promote the convergence of the global model.

Suppose $\{\beta_i^{\text{global}}\}_{i=1}^{N_{\text{round}}}$ represents the parameter set of the global model in the FL framework, where N_{round} denotes the total training rounds and β_0^{global} stands for the initialization parameters of the global model. Before the ith round of model aggregation, the server sends $\beta_{i-1}^{\text{global}}$ to the clients; the clients then use the parameters to train local models independently. Then, the related parameters of these locally trained models are uploaded again to the server. Suppose V_{elu} denotes a validation set that is extracted from the training set. With communication loss and the number of clients accounted for, the sample size of V_{elu} is 200 with a ratio of $1 : 1 : 1 : 1$ for completed classes. After the server receives all parameters $\{\beta_i^j\}_{j=1}^{N_{\text{client}}}$ from all the clients, the refined validation strategy is performed to evaluate the accuracy $\{\gamma_i^j\}_{j=1}^{N_{\text{client}}}$ of the different local models.

Here, larger scores indicate that the parameters of some clients are capable of enhancing the performance of the global model; smaller scores indicate low-quality clients. To reduce the influence of low-quality models during aggregation, the mean μ_{mean} and variance σ_{acc} of the accuracy are calculated to define the selected local models. Local models with lower accuracy tend to cause abnormal convergence of the global model and are thus neglected. The optimal number of clients N_{better} participating in the model aggregation is obtained from the threshold $|\mu_{\text{mean}} - \sigma_{\text{acc}}|$ at each round, where $N_{\text{better}} \leq N_{\text{client}}$. Filter local models with scores higher than threshold $|\mu_{\text{mean}} - \sigma_{\text{acc}}|$ to participate in aggregation. Specifically, the updated global models are defined as follows:

$$\beta_i^{\text{global}} = \sum_{j=1}^{N_{\text{better}}} \vartheta_{i-1}^j \beta_{i-1}^j. \tag{35}$$

Improved PSO is used to find the optimal value of the coefficient of the weight θ_{i-1}^j of the client participating in the aggregation to improve the robustness of the aggregation

strategy in FL. The inertia weight factor ϖ is introduced into the PSO algorithm to adjust the particle flight speed and to improve the algorithm's convergence speed and global search ability, enabling it to quickly obtain the globally optimal result. Let the position of the ith particle be represented as a vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{\mathrm{id}})$. The current optimal position found for the ith particle is then $p_{\mathrm{best}} = (p_{i1}, p_{i2}, \ldots, p_{\mathrm{id}})$, the current optimal position found by the entire particle swarm is $g_{\mathrm{best}} = (g_1, g_2, \ldots, g_d)$, and the velocity of the ith particle is $v_i = (v_{i1}, v_{i2}, \ldots, v_{\mathrm{id}})$. The adjustment principle of the inertia weight factor ϖ to speed is presented in (36)

$$v_{id}(t+1) = \varpi v_{\mathrm{id}}(t) + c_1 r_1 (p_{\mathrm{id}}(t) - x_{\mathrm{id}}(t)) + \mathrm{c}_2 r_2 (g_d(t) - x_{\mathrm{id}}(t)) \tag{36}$$

where c_1 and c_2 are acceleration factors and r_1 and r_2 are random numbers in [0, 1]. ϖ primarily indicates the degree of influence of the state of the previous generation of particles on the current particle state. Larger values of ϖ indicate a greater influence. The global search capability of the algorithm also increases with ϖ. Therefore, modifying the value of ϖ can improve the global search capability of the particle swarm algorithm.

Similar manner to what was done in [31], to further improve the optimization performance and efficiency of the particle swarm algorithm, the optimized distance control factor is used to dynamically modify the inertia weight ϖ. The underlying principle is presented in (37)

$$\begin{cases} \mathrm{dist}(t) = \dfrac{\sqrt{\sum_{i=1}^{N} (x_i - p\mathrm{best}_i(t))}}{N} \\ \mathrm{cont}(t) = \dfrac{\mathrm{dist}(t)}{\max(\mathrm{dist}\, t)} \\ \varpi = \varpi_{\max} - (\varpi_{\max} - \varpi_{\min})\, \mathrm{cont}(t)^2 \end{cases} \tag{37}$$

where dist_i is the average distance of each particle from the global optimal value; max(dist) is the maximum average distance thus far; cont is the distance control factor (i.e., the ratio of dist_i and max(dist)); and $\varpi_{\max}$ and $\varpi_{\min}$ are the maximum and minimum inertia weights, respectively. After multiple training rounds, the final global model is $\beta_{N_{\mathrm{round}}}^{\mathrm{global}}$. The proposed FL algorithm is presented in Algorithm 2.

Unbalanced data distribution and category imbalance are the two main manifestations of the Non-IID distribution of client data. In particular, the total data volume of the client combination generated by uniform sampling also has a category imbalance. Therefore, the test performance curve of the global model often has substantial local oscillation. The verification strategy proposed in this article features a client weight selection method to give full play to the value of client local data in FL. After a corresponding device combination is selected in each round of communication, the test performance of each category of the global model can be balanced. Under the Non-IID condition, in [25], the batch size b_k is dynamically set through reinforcement learning. When the value of b_k for a client tends to 0, the client does not participate in the weighted average on the server. We can assume that K devices are uniformly sampled from N devices. For a nonconvex objective function F at a fixed learning rate η and a nonfixed batch size b_k, for all $T \in \mathbb{N}$, the lower bound of the expected mean square gradient norm of F is as follows:

Algorithm 2 Improved FL Algorithm With Dynamic Validation

1: Input: $\beta_0^{global}, N_{round}, N_{client}$
2: **Server Side:**
3: **for** each training round $i = 1 \rightarrow N_{round}$
4: Send model β_{i-1}^{global} to all the clients
5: Wait for uploaded models $\left\{\beta_i^j\right\}_{j=1}^{N_{client}}$ from all clients
6: Calculate the accuracy of each model, defined as $\left\{\gamma_i^j\right\}_{j=1}^{N_{client}}$; use V_{elu}
7: Calculate the μ_{mean} and the std σ_{acc} of $\left\{\gamma_i^j\right\}_{j=1}^{N_{client}}$
8: Filter these local models; delete $\left\{\beta_i^j\right\}_{j=N_{client}-N_{better}}^{N_{client}}$ if $\gamma_i^j \leq |\mu_{mean} - \sigma_{acc}|$
9: $\beta_i^{\mathrm{global}} = \sum_{j=1}^{N_{\mathrm{better}}} \vartheta_{i-1}^j \beta_{i-1}^j$, find the optimal ϑ_{i-1}^j through the improved PSO algorithm
10: **end for**
11: **Client Side:**
12: **for** each training round $i = 1 \rightarrow N_{round}$
13: **for** each client $j = 1 \rightarrow N_{client}$
14: Download global model β_{i-1}^{global} as local models $\left\{\beta_i^j\right\}_{j=1}^{N_{client}}$
15: **for** each local training epoch
16: Update $\left\{\beta_i^j\right\}_{j=1}^{N_{client}}$, to minimize the loss function
17: **end for**
18: Upload $\left\{\beta_i^j\right\}_{j=1}^{N_{client}}$ to the server
19: **end for**
20: **end for**
21: Output: $\beta_{N_{round}}^{global}$

$$\begin{aligned} &\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}||\nabla F(\mathrm{w}_t^k)||_2^2 \\ &\leq \frac{F(\mathrm{w}_1) - F^*}{T(G-A-C)} + \frac{L\eta^2}{2K(G-A-C)} \sum_{k=1}^{K} p_k^2 \beta_k m_k \\ &+ \frac{L^2\eta^3}{12K(G-A-C)} \sum_{k=1}^{K} p_k^2 \beta_k (m_k - 1) m_k (2m_k - 1) \end{aligned} \tag{38}$$

$$p_k = D_k/D,\ m_k = D_k E/b_k,\ C = \frac{L\eta^2}{2K} \sum_{k=1}^{K} m_k \tag{39}$$

$$\beta_k = \sigma_k^2/b_k,\ A = \frac{L^2\eta^3}{4K} \sum_{k=1}^{K} m_k(m_k - 1) \tag{40}$$

$$G = \frac{\eta}{2K} \sum_{k=1}^{K} (m_k - 1). \tag{41}$$

Here, $k \in K$, F is a nonconvex objective function, ∇F is a Lipschitz-continuous objective function, F^* is a scalar bound of the sequence of iterations $F_{(W_t)}$ in an open set, $L > 0$ is a Lipschitz constant, m_k is the number of local updates in each

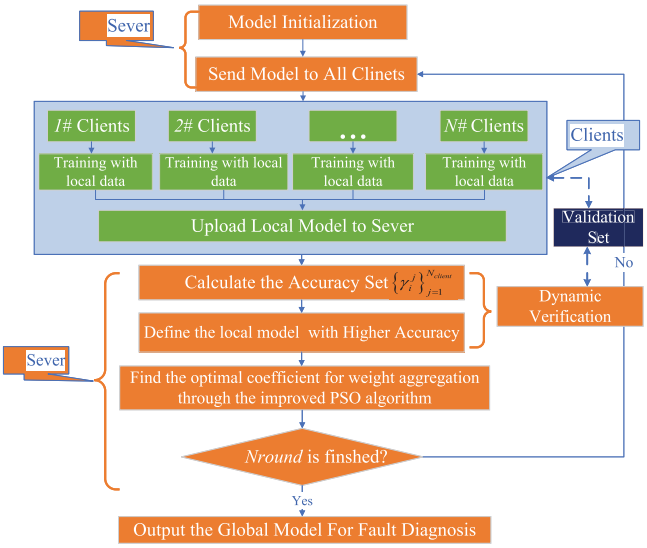

Fig. 4. Flow chart of the proposed method in the FL framework. It presents the whole process of training and interaction between the server and all clients.

communication cycle, $\mathbb{E}$ represents each client's amount of epochs, σ_k^2 indicates the bound of the variance of the stochastic gradient in each client, and D_k is the local data set of client k.

In the fault diagnosis model, the clients who will participate in the weighting will be chosen using a fixed learning rate and batch size. In the present study's method, the batch size is set as a constant. The expected average squared gradient norms of F still converge to a nonzero constant as $T \to \infty$, which is also applicable to the federation mechanism proposed in this article. In actual tasks, the appropriate reduction of the accuracy rate facilitates the choice of the fixed learning rate, batch size, and number of rounds for reducing the computational complexity of the model and improving the model's robustness.

A flowchart of the proposed method is presented in Fig. 4, and it illustrates the entire training process and interactions between the server and all clients. A validation strategy is used to upgrade the robustness of model aggregation. Due to the differences in the data distribution between the local models, if a client's model leads to a decrease in the fault diagnosis rate, the verification strategy based on diagnostic accuracy is beneficial for increasing the convergence of the global model. The validation set in Fig. 4 is extracted from the training set. the sample size of validation set is 200 with a ratio of 1 : 1 : 1 : 1 for completed classes. The validation set is crucial for the generalization of the global model. In an industrial scenario, some unknown failure samples can be dynamically added to the server's verification set. If the label category, rather than the number of samples, is controlled, then the validation set can drive the global model to train and optimize to complete the target task.

IV. Experimental Study

A. Data Descriptions

The experimental platform comprises a server and four sets of motor fault monitoring systems. Each fault monitoring system includes a PMSM, motor control system, wireless sensor system, and embedded deep learning module. By adjusting the digital potentiometer connected across the PMSM coil, we can simulate the various ITSC states of the motor. The ITSC status classification is presented in Table I. The health status indicates that the motor has not failed. The fault levels I and II, respectively, indicate that the PMSM is within the allowable range of unbalanced voltage, and there may be a slight and severe ITSC. fault level III indicates that the PMSM is damaged. In the simulation of short-circuit faults between turns, each motor fault monitoring device is at the same rated state and has the same number of short-circuit turns. In accordance with the experimental methods in [4], 2500 sets of data, including all states, were obtained from the motor fault monitoring system at the same state. The ratio of each state was 52 : 25 : 15 : 8. Therefore, the total data set contains 10 000 groups of data with various ITSC states. The test set is composed of 1100 groups of data, with a ratio of 25 : 15 : 9 : 6. The entire experiment was designed on the TensorFlow platform. First, intensive training was used to determine the model structure and hyperparameters of the stacked network based on NSAE and DAE. The model is compared against other traditional methods to verify its effectiveness in ITSC fault diagnosis. Under Non-IID conditions, the feasibility of using FL for stacking model training is analyzed and compared with the traditional FedAvg method, and the superiority of the verification strategy proposed in this article in client selection and fault diagnosis model training is verified. The FL parameters are presented in Table II. We suppose there are 6 units of clients participating in the FL training process. In Table II, ξ represents the number of rounds of local training prior to the aggregation for each local unit. If ξ is small, communication costs increase and the local data set cannot be fully learned, causing a fitting failure in the network. However, an overly large ξ results in greater differences between the local models and increased losses during aggregation. B is the minimum batch size updated by the client. If the value of B value is overly large, calculation costs and memory usage are

TABLE I
Label Classification of the PMSM Defined by the ITSC. It Defines Various States of the ITSC Fault. We Can Distinguish the Severity of the Fault by the Change of Inter Turn Resistance and the Number of Short-Circuit Turns

Status of the PMSM	Number of short circuit turns	Turn-to-turn resistance	Label value
Health status	0	$> 1M\Omega$	0
Fault Level I	1	$< 1M\Omega$	1
Fault Level II	2	$< 1M\Omega$	2
Fault Level III	5	$< 1M\Omega$	3

TABLE II
Hyperparameter Selection Under the FL Framework, ξ Represents the Number of Rounds of Local Training Prior to the Aggregation for Each Local Unit. B Is the Minimum Batch Size Updated by the Client

Parameter	Value
ζ	10
B	100
Number of clients	6

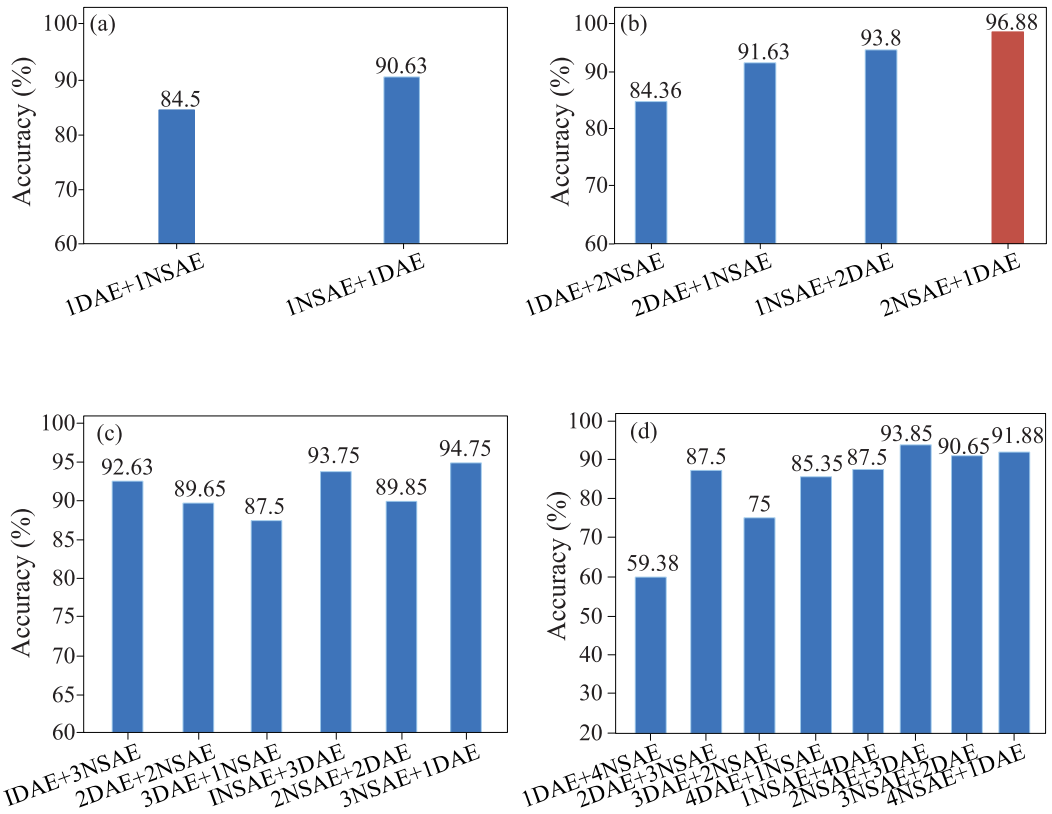

Fig. 5. Shows the diagnostic accuracy of the data set for different combinations: (a) two hidden layers, (b) three hidden layers, (c) four hidden layers, and (d) five hidden layers. The number of layers of the model is not the more the better. On the one hand, it will cause difficulties in training and convergence, on the other hand, it will consume computing resources and can not be deployed on the IoT devices.

TABLE III
HIGHEST DIAGNOSTIC ACCURACY UNDER DIFFERENT HIDDEN LAYERS, "1DAE + 1NSAE" INDICATES THAT THE STACKED NETWORK COMPRISES A DAE AND A NSAE. THE NSAE IS STACKED AFTER THE DAE, AND SIMILAR INDICATORS ARE USED FOR THE OTHER REPRESENTATIONS

Hidden Layers	Learning Rate	Sparsity Rate	Optimization algorithm	Accuracy
1NSAE+1DAE	0.02	0.6	Adam	90.63%
2NSAE+1DAE	0.025	0.6	Adam	96.88%
3NSAE+1DAE	0.005	0.6	Adam	94.75%
2NSAE+3DAE	0.025	0.6	Adam	93.85%

TABLE IV
MODEL PARAMETER OPTIMIZATION. IT LISTS THE HIGHEST DIAGNOSTIC ACCURACY FOR VARIOUS COMBINATIONS OF HYPERPARAMETERS

stacked structure	learning rate	sparsity rate	optimization algorithm	accuracy
2NSAE+DAE	0.001	0.6	Adam	87.75%
2NSAE+DAE	0.025	0.6	Adam	96.88%
2NSAE+DAE	0.01	0.6	Adam	93.15%
2NSAE+DAE	0.05	0.6	Adam	81.25%
2NSAE+DAE	0.025	0.6	SGD	87.25%
2NSAE+DAE	0.025	0.6	Adam	96.88%
2NSAE+DAE	0.025	0.6	Adagrad	83.74%
2NSAE+DAE	0.025	0.6	Adadelta	79.36%
2NSAE+DAE	0.025	0.2	Adam	90.63%
2NSAE+DAE	0.025	0.4	Adam	93.5%
2NSAE+DAE	0.025	0.6	Adam	96.88%
2NSAE+DAE	0.025	0.8	Adam	87.5%

large and the gradient direction for different batches can easily cause trapping in a local minimum. If B is too small, the loss oscillates and results in a relatively large loss of computing power on the edge devices. The cost requirements for the edge devices also increase, which is not conducive to lightweight deployment.

B. Network Architecture and Parameter Determination

To obtain a lightweight model suitable for deployment in IIoT devices, the main parameters (stacked structure, learning rate, sparsity rate, and optimization algorithm) of the stacked network are optimized by centralized training. Fig. 5, Table III, and Table IV present the experimental results for centralized training, where "1DAE + 1NSAE" indicates that the stacked network comprises a DAE and a NSAE. The NSAE is stacked after the DAE, the robustness of the input features is enhanced by the DAE, and the sparsity is then reinforced by the NSAE. By contrast, "1NSAE + 1DAE" indicates that the DAE is stacked after the NSAE for feature extraction. Similar indicators are used for the other representations.

The structure of the stacked network is a key factor affecting its feature learning and fault diagnosis performance, including the combination order of NSAE and DAE and the number of their layers. Table III lists the highest diagnostic accuracy for various combinations of hidden layers. As the number of hidden layers increases, the highest diagnostic accuracy first increases and then decreases. Therefore, more hidden layers

TABLE V
PARAMETER SETTINGS OF THE METHODS. IT SHOWS SEVERAL HYPERPARAMETERS OF VARIOUS CLASSICAL MODELS IN THE ITSC FAULT DIAGNOSIS

	RNN	CNN	BP	3NSAE	3DAE	2NSAE+DAE
Input size	7	7	7	7	7	7
Hidden layers	3	5	2	3	3	12
Output size	4	4	4	5	5	5
Learning rate	0.01	0.01	0.01	0.005	0.002	0.025
Sparsity rate	-	-	-	0.5	0.5	0.6
Classifier	Softmax	Softmax	Softmax	Softmax	Softmax	Softmax
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy

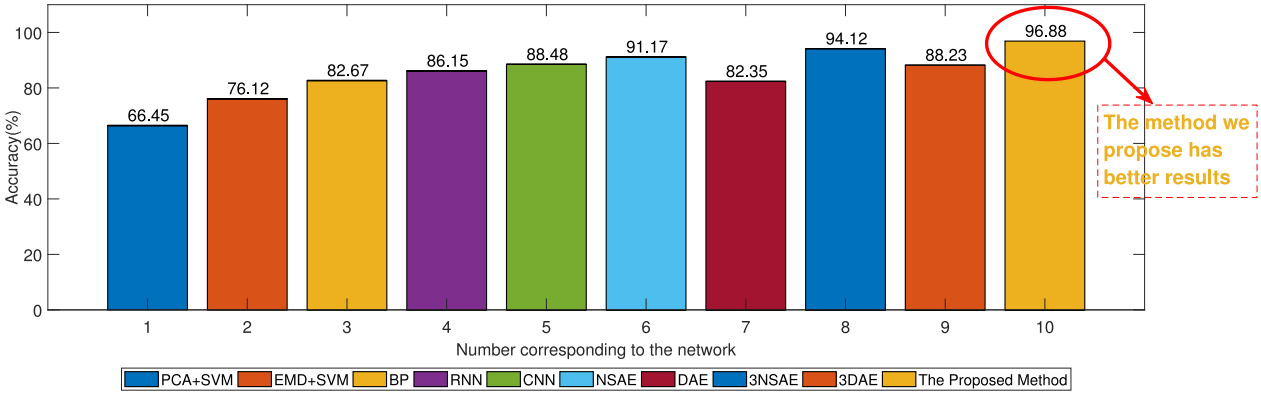

Fig. 6. Diagnosis results of different models in the same set. It shows that the efficiency of fault diagnosis is related to the design of the model, the number of hidden layers and various parameters. Simply increasing the number of hidden layers in the stack model can improve the feature extraction and generalization ability of the model to a certain extent, but more than a certain number will sharply affect the convergence performance of the network.

do not always result in higher diagnostic accuracy. Although in theory, an increase in the number of hidden layers can enhance the description of fault characteristics, it also increases the training difficulty, training time, and the risk of overfitting; thus, increasing the layers is unsuitable for a lightweight deployment. NSAE and DAE stacked in three layers yielded the optimal fault diagnosis results.

As presented in Fig. 5, for the same number of hidden layers, the highest diagnostic accuracy for "NSAE + DAE" is higher than that of "DAE + NSAE." Placing NSAE at the front of the stack model is helpful for effectively identifying a small number of fault samples in the unbalanced data set. Placing DAE after NSAE not only has no effect on the feature distribution of the input data but also effectively improves the robustness of the model. Therefore, the fault diagnosis accuracy of the model with NSAE placed in the front layer of the stacked network is generally higher than that of the model with DAE placed in the front. Considering the highest diagnosis accuracy results, we chose the "2NSAE + DAE" architecture.

In addition to the three-layer stack structure, the learning rate, sparsity rate, and optimization algorithm are also essential factors affecting feature learning and fault diagnosis. As indicated in Table IV, if the learning rate is too large for a given model structure, the weight range of the model is also overly large, preventing convergence. By contrast, if the learning rate is too small and the updating speed of the parameters is too slow, a large number of iterations are required for model convergence. According to the results, the highest accuracy is obtained at a learning rate of 0.025. The sparsity rate also greatly influences fault diagnosis ability because high sparsity rates cause data overcompression, and low sparsity rates prevent the network from achieving optimal feature descriptions, affecting diagnostic accuracy. A sparsity rate of 0.6 yielded higher precision than the other sparsity rates of 0.2, 0.4 and 0.8. Due to the long training time of the stochastic gradient descent (SGD) method, the Adagrad method still requires the global learning rate to be manually set and AdaDelta method always fluctuates about the local minimum. At the same learning rate, Adam is more suitable for solving large-scale data and parameter optimization problems. Therefore, Adam adopted as the network solving algorithm in this study.

The effectiveness of traditional and proposed algorithms in ITSC fault diagnosis is evaluated in this experiment. The parameter settings for different networks are presented in Table V. Fig. 6 shows the diagnostic results.

As indicated in Fig. 6, neural networks yielded more accurate fault diagnosis relative to traditional methods. The efficiency of fault diagnosis is related to the design of the model, the number of hidden layers, and various other parameters. Simply increasing the number of hidden layers in the stack model increases the feature extraction and generalization capabilities of the model to a certain extent but it also dramatically affects the convergence performance of the network if a threshold is exceeded. Compared with the CNN network, RNN network, single-layer AE network, and other forms of stacked networks, the present study's fault diagnosis method yielded no less than 5% improvement in accuracy, indicating its superior feature extraction and robustness and, by consequence, accuracy in ITSC diagnosis.

C. Results for Different Non-IID Data Settings

Two types of Non-IID data settings were employed to demonstrate the effectiveness of our proposed algorithm. First, each of the 6 clients contains samples of all categories with

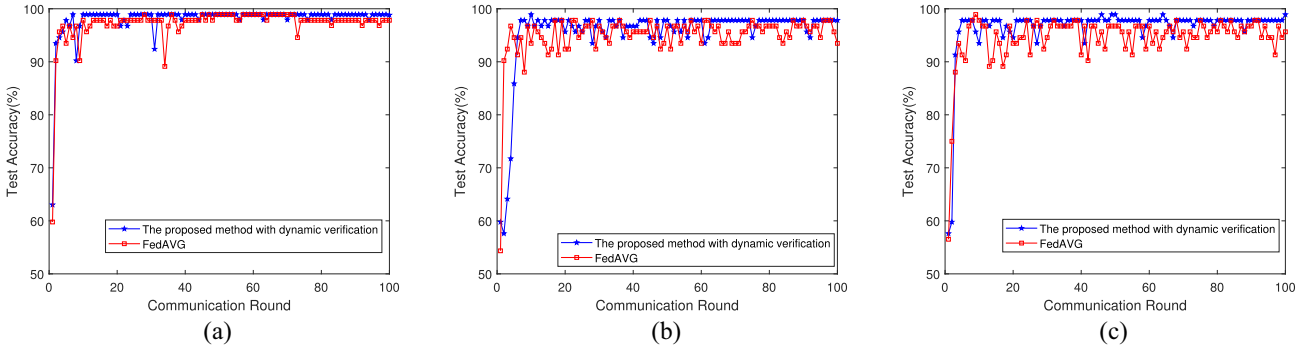

Fig. 7. Accuracy for various degrees of Non-IID (first type: σ was used to represent the proportion of each samples. σ=0.65 means that 65% of the samples on each client belong to one category and 35% belong to the other three categories). (a) $\sigma = 0.5$. (b) $\sigma = 0.65$. (c) $\sigma = 0.8$.

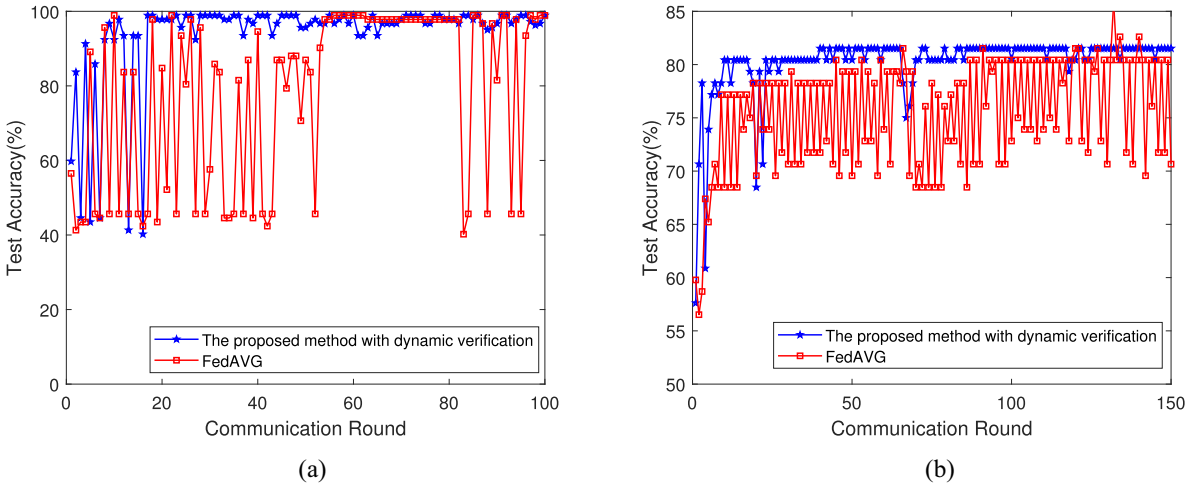

Fig. 8. Accuracy for various degrees of Non-IID (second type: $H = 2$ or $H = 3$ indicate that each client only contains 2 or 3, respectively, categories of samples). (a) $H = 3$. (b) $H = 2$.

TABLE VI
NETWORK PARAMETERS OF THE STACKED MODEL. IT SHOWS THE TECHNICAL DETAILS OF THE STACKED MODEL WHEN IT IS USED FOR DETECTING THE ITSC FAULT

Parameters	Value
Stacked structure	2NSAE+1DAE
Input size	7
Output size	5
Hidden layer nodes	12
Learning rate	0.025
Sparsity rate	0.6
Dropout	0.5
Classifier	Softmax

different proportions. σ was used to represent the proportion of each samples. For example, $\sigma = 0.65$ means that 65% of the samples on each client belong to one category and 35% belong to the other three categories. Second, $H = 2$ or $H = 3$ indicate that each client only contains 2 or 3, respectively, categories of samples. For each Non-IID data setting, we train the stacked diagnosis model described in Table VI and use the same test set to evaluate the accuracy of fault diagnosis in the task to evaluate the performance of the algorithm. Figs. 7 and 8 present the experimental results for two Non-IID data settings.

Both an increase in σ and a decrease in H indicate a larger imbalance of Non-IID data. Figs. 7 and 8 reveal that as the data imbalance increases, greater communication time is required to achieve convergence. The FedAvg algorithm does not consider the effects of the local model on the performance of the global model, and clients with Non-IID samples interfere with the convergence direction during model aggregation in the FL process. As presented in Fig. 8, during the training process of FedAvg, the Non-IID problem causes drastic fluctuations in the global model. However, due to the elimination of low-quality models, the method in this article can still maintain good stability during the model training process, and its performance is significantly better than that of the FedAvg algorithm. Moreover, our dynamic verification FL algorithm can always achieve higher diagnostic accuracy under fixed communication wheels, revealing that the dynamic verification FL algorithm is superior to the FedAvg algorithm in test accuracy.

To more clearly observe the changes in diagnostic accuracy during the training process under different Non-IID data settings, we divide the training process into five stages according to the total number of communications, and the average diagnostic accuracy rate of each stage is calculated. The experimental results under the two types of Non-IID data setting are

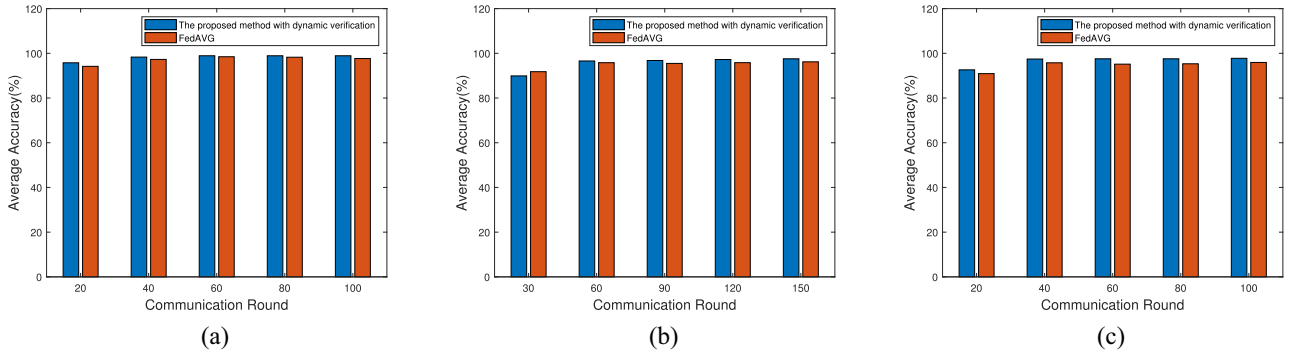

Fig. 9. Average Accuracy for various degrees of Non-IID (first type: σ was used to represent the proportion of each samples. $\sigma = 0.65$ means that 65% of the samples on each client belong to one category and 35% belong to the other three categories). (a) $\sigma = 0.5$. (b) $\sigma = 0.65$. (c) $\sigma = 0.8$.

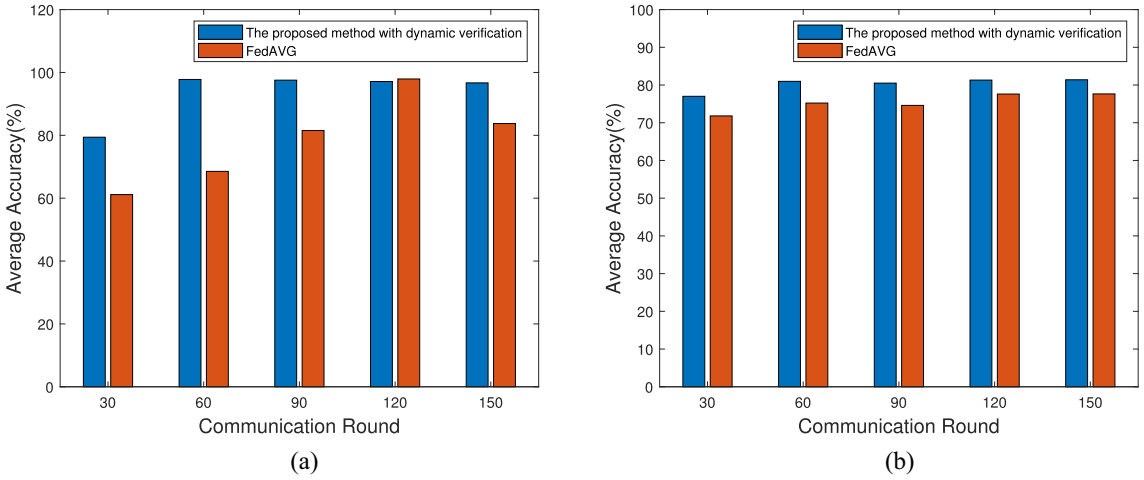

Fig. 10. Average Accuracy for various degrees of Non-IID (second type: $H = 2$ or $H = 3$ indicate that each client only contains 2 or 3, respectively, categories of samples). (a) $H = 3$. (b) $H = 2$.

shown in Figs. 9 and 10. The results reveal that the average diagnostic accuracy of our dynamically verified FL algorithm is always better than that of the FedAvg algorithm. The gap is most obvious for the Non-IID data settings of $H = 2$ and $H = 3$. The average accuracy of the algorithm in this article is approximately 5% and 12% higher than that of the FedAvg algorithm, respectively, in these two situations. Therefore, for Non-IID data, the FL training method with dynamic verification has better fault diagnosis performance than the FedAvg method.

Finally, we compare the number of communication rounds necessary to reach convergence with the final test accuracy to demonstrate the training efficiency of our proposed approach. After reaching convergence, we record the needed communication rounds and final test accuracy, the relevant results are shown in Figs. 11 and 12. The results reveal that the number of rounds of communication required by the dynamically verified FL algorithm is always lower than those required by the FedAvg algorithm under all Non-IID cases, indicating that the FL algorithm has higher communication efficiency. Moreover, the FL algorithm has a greater final test accuracy than FedAvg. All results indicate the superiority of the proposed algorithm in handling Non-IID data.

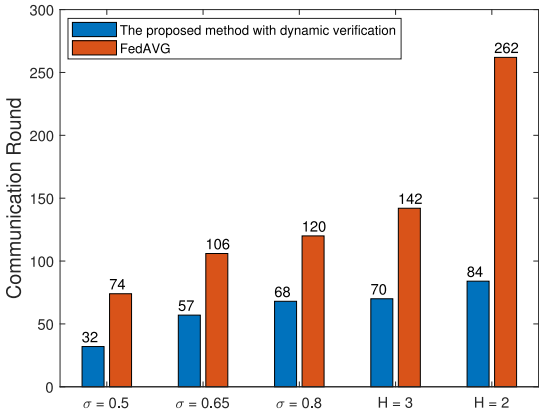

Fig. 11. Required number of rounds of communication to achieve convergence, represents the communication efficiency of the model.

V. Conclusion

In this article, we presented a stacked network model based on NSAE and DAE for ITSC fault diagnosis. NSAE can expand the recognition ability of the network when features are unbalanced and the sample is small, and DAE can increase the robustness of the network. In contrast with state-of-the-art

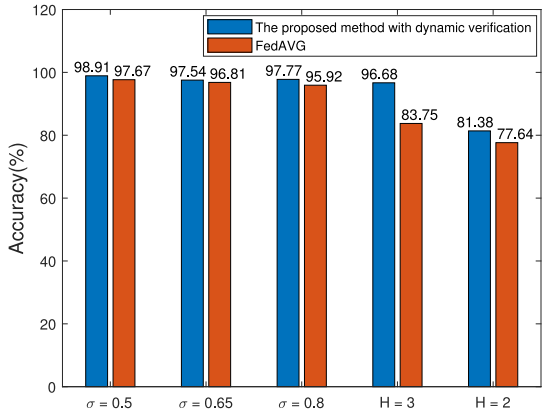

Fig. 12. Final test accuracy after convergence is achieved, indicates the diagnostic performance of the model.

methods using centralized training styles, the proposed method is trained using FL. During model aggregation, an improved validation scheme is adopted to reduce the influence of low-quality models in each round, and the improved PSO is used to optimize the client weighting coefficients, increasing the robustness of the FL models against data poisoning. The experimental results demonstrated that the strategy proposed in this article has better training ability than the FedAvg and that the stacked model's performance is superior for cases with the Non-IID problem. Furthermore, high-quality models are selected from all the clients, improving the stability of the model in the FL process and increasing the generalizability of the global model.

REFERENCES

[1] D. Guo, R. Y. Zhong, Y. Rong, and G. G. Q. Huang, "Synchronization of shop-floor logistics and manufacturing under IIoT and digital twin-enabled graduation intelligent manufacturing system," *IEEE Trans. Cybern.*, early access, Sep. 13, 2021, doi: 10.1109/TCYB.2021.3108546.

[2] L. Li *et al.*, "Sustainability assessment of intelligent manufacturing supported by digital twin," *IEEE Access*, vol. 8, pp. 174988–175008, 2020.

[3] J. Zhao, X. Guan, C. Li, Q. Mou, and Z. Chen, "Comprehensive evaluation of inter-turn short circuit faults in PMSM used for electric vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 611–621, Jan. 2021.

[4] J. Zhang, Y. Wang, K. Zhu, Y. Zhang, and Y. Li, "Diagnosis of interturn short-circuit faults in permanent magnet synchronous motors based on few-shot learning under a federated learning framework," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8495–8504, Dec. 2021.

[5] C. Wang, G. Yang, G. Papanastasiou, H. Zhang, J. J. P. C. Rodrigues, and V. H. C. de Albuquerque, "Industrial cyber-physical systems-based cloud IoT edge for federated heterogeneous distillation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5511–5521, Aug. 2021.

[6] C. Wang, S. Dong, X. Zhao, G. Papanastasiou, H. Zhang, and G. Yang, "SaliencyGAN: Deep learning semisupervised salient object detection in the fog of IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2667–2676, Apr. 2020.

[7] M. H. U. Rehman, A. M. Dirir, K. Salah, E. Damiani, and D. Svetinovic, "TrustFed: A framework for fair and trustworthy cross-device federated learning in IIoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8485–8494, Dec. 2021.

[8] T. Han, Y.-F. Li, and M. Qian, "A hybrid generalization network for intelligent fault diagnosis of rotating machinery under unseen working conditions," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021, doi: 10.1109/TIM.2021.3088489.

[9] Y. Wang, J. Yan, Q. Sun, Q. Jiang, and Y. Zhou, "Bearing intelligent fault diagnosis in the Industrial Internet of Things context: A lightweight convolutional neural network," *IEEE Access*, vol. 8, pp. 87329–87340, 2020.

[10] Y. Qi, E. Bostanci, M. Zafarani, and B. Akin, "Severity estimation of interturn short circuit fault for PMSM," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7260–7269, Sep. 2019.

[11] Y. Li, Y. Wang, Y. Zhang, and J. Zhang, "The diagnosis of inter-turn short circuit of permanent magnet synchronous motor based on deep learning and small fault samples," *Neurocomputing*, vol. 442, pp. 348–358, Jun. 2021.

[12] J. Shen, S. Li, F. Jia, H. Zuo, and J. Ma, "A deep multi-label learning framework for the intelligent fault diagnosis of machines," *IEEE Access*, vol. 8, pp. 113557–113566, 2020.

[13] X. Wang, X. Lin, K. Zhou, and Y. Lu, "CNN based mechanical fault diagnosis of high voltage circuit breaker using sound and current signal," in *Proc. IEEE Int. Conf. High Voltage Eng. Appl. (ICHVE)*, Beijing, China, 2020, pp. 1–4.

[14] J. Liu *et al.*, "Toward robust fault identification of complex industrial processes using stacked sparse-denoising autoencoder with softmax classifier," *IEEE Trans. Cybern.*, early access, Sep. 22, 2021, doi: 10.1109/TCYB.2021.3109618.

[15] P. Yao, S. Yang, and P. Li, "Fault diagnosis based on RseNet-LSTM for industrial process," in *Proc. IEEE 5th Adv. Inf. Technol. Electron. Autom. Control Conf. (IAEAC)*, Chongqing, China, 2021, pp. 728–732.

[16] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.

[17] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[18] T. Alfakih, M. M. Hassan, A. Gumaei, C. Savaglio, and G. Fortino, "Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA," *IEEE Access*, vol. 8, pp. 54074–54084, 2020.

[19] N. Cha, C. Wu, T. Yoshinaga, Y. Ji, and K.-L. A. Yau, "Virtual edge: Exploring computation offloading in collaborative vehicular edge computing," *IEEE Access*, vol. 9, pp. 37739–37751, 2021.

[20] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[21] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6532–6542, Oct. 2020.

[22] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu, "EdgeFed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209191–209198, 2020.

[23] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Federated learning for machinery fault diagnosis with dynamic validation and self-supervision," *Knowl. Based Syst.*, vol. 213, Feb. 2021, Art. no. 106679.

[24] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.

[25] J. Zhang *et al.*, "Adaptive federated learning on non-IID data with resource constraint," *IEEE Trans. Comput.*, early access, Jul. 26, 2021, doi: 10.1109/TC.2021.3099723.

[26] Y. Wang, G. Gui, H. Gacanin, B. Adebisi, H. Sari, and F. Adachi, "Federated learning for automatic modulation classification under class imbalance and varying noise condition," *IEEE Trans. Cogn. Commun. Netw.*, early access, Jun. 16, 2021, doi: 10.1109/TCCN.2021.3089738.

[27] D. Y. Zhang, Z. Kou, and D. Wang, "FedSens: A federated learning approach for smart health sensing with class imbalance in resource constrained edge computing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Vancouver, BC, Canada, 2021, pp. 1–10.

[28] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, and L. Liang, "Self-balancing federated learning with global imbalanced data in mobile systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 59–71, Jan. 2021.

[29] Y.-J. Li, Z.-L. Zhang, M.-H. Li, H.-F. Wei, and Y. Zhang, "Fault diagnosis of inter-turn short circuit of permanent magnet synchronous motor based on deep learning," *Electr. Mach. Control*, vol. 24, no. 9, pp. 173–180, 2020.

[30] F. Jia, Y. G. Lei, L. Guo, J. Lin, and S. B. Xing, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," *Neurocomputing*, vol. 272, pp. 6772–6786, Jan. 2018.

[31] S. Guo *et al.*, "A fault diagnosis method with application of HMM based on adaptive particle swarm optimization," *J. Vib. Shock*, vol. 40, no. 20, pp. 264–270, 2021.

Yuanjiang Li received the M.S.E. degree in signal and information processing from Jiangsu University of Science and Technology, Zhenjiang, China, in 2006, and the Ph.D. degree in information and communication engineering from Nanjing University of Science and Technology, Nanjing, China, in 2013.

Since 2006, he has been with the Faculty of School of Electronic Information, Jiangsu University of Science and Technology. His current research interests include big data, computer vision, high-performance computing, interdisciplinary research with pattern recognition, and fault diagnosis.

Yunfeng Chen received the bachelor's degree in electronics and information engineering from Jingjiang College of Jiangsu University, Zhenjiang, China, in 2019. She is currently pursuing the master's degree in electronics and communication engineering with Jiangsu University of Science and Technology, Zhenjiang.

Her research interests include fault diagnosis and federated learning.

Kai Zhu received the Ph.D. degree in control science and control engineering from Nanjing University of Science and Technology, Nanjing, China, in 2015.

He is currently a Teacher with the School of Automobile and Traffic Engineering, Jiangsu University of Technology, Changzhou, China. His research interests include the application of big data and intelligent transportation systems.

Cong Bai (Member, IEEE) received the B.E. degree from Shandong University, Jinan, China, in 2003, the M.E. degree from Shanghai University, Shanghai, China, in 2009, and the Ph.D. degree from the National Institute of Applied Sciences, Rennes, France, in 2013.

He is an Associate Professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His research interests include computer vision and multimedia processing.

Jinglin Zhang received the M.S. degree in circuits and systems from Shanghai University, Shanghai, China, in 2010, and the Ph.D. degree in electronics and telecommunications from the National Institute of Applied Sciences, Rennes, France, in 2013.

From 2014 to 2020, he was with the Faculty of the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. Since 2021, has been with the Faculty of the School of Artificial Intelligence, Hebei University of Technology, Tianjin, China. In 2022, he also become an Adjunct Professor with Linyi University, Linyi, China. His current research interests include computer vision, high-performance computing, interdisciplinary research with pattern recognition, and atmospheric science.