Online Fault Line Detection in Small-Sample and Streaming Data Environments

Online Fault Line Detection in Small-Sample and Streaming Data Environments

Le Zhang, Jizhong Zhu, *Fellow, IEEE*, Shenglin Li, Alberto Borghetti, *Fellow, IEEE,* Di Zhang

***Abstract*—This paper focuses on medium and low voltage networks with neutral grounded through arc-suppression coils. Standard data-driven fault line detection (FLD) approaches assume that the training sample is sufficient, static, and reusable. In practical scenarios, such approaches may be infeasible due to the sporadic and temporary nature of single-phase-to-ground faults, which provide insufficient fault samples, and due to large-scale high-speed dynamic data streams associated with measurements. To tackle these issues, the paper proposes a novel FLD scheme based on personalized federated learning (PFL) and incremental stochastic configuration networks (SCN) for small-sample and streaming data environments. Concretely, the SCN, a concise non-iterative neural network, is exploited as the FLD classifier. To adapt effectively to dynamic and non-reusable environments, an incremental SCN is proposed that can learn fault features without experiencing forgetting when dealing with streaming data. The proposed FLD scheme based on PFL selectively aggregates fault features from multiple substations. This approach addresses the challenge of limited sample sizes while preserving the personalization of each local model. Extensive experiment results using real data show that the proposed method can significantly improve accuracy when dealing with small samples and continuously learn fault features in streaming data.**

***Index Terms*—Fault line detection, personalized federated learning, incremental stochastic configuration networks, small sample, streaming data.**

I. INTRODUCTION

In general, distribution networks with voltage levels of 35 kV and below are commonly referred to as medium and low voltage network. These networks play crucial role in power transmission and distribution, serving as dependable infrastructure for large-scale power supply in cities and industrial areas. The medium and low voltage distribution network systems with neutral grounding via arc-suppression coils, also known as resonant grounded systems, find preference in various contexts and regions [1]. The inductive current generated by the arc-suppression coils neutralizes the capacitive current in the event of a line-to-ground fault, leading to the presence of weak fault currents. This characteristic contributes to an enhancement in power supply reliability. However, due to the reduced fault current, the installed relay protection devices remain inactive, leading to challenges in fault feature extraction and fault line detection (FLD) [2]. Practical industrial experiences indicate that prolonged operation in the presence of a line-to-ground fault can endanger the system. Therefore, embracing advanced technologies for fault feature extraction and improving FLD accuracy is crucial.

Traditional FLD solutions involve identifying fault features based on physical knowledge. For instance, in accordance with Kirchhoff's current law, the zero-sequence current of the fault line equals the sum of the zero-sequence currents of all lines not affected by the fault [3]. Similarly, other steady-state electrical quantities, including but not limited to zero-sequence voltage, negative-sequence current, reactive component, and the fifth harmonic, have served as fault criteria in previous studies [4-6]. Unfortunately, steady-state signals are weak and often mixed with noise, leading to unclear fault features. In light of this, researchers have turned their attention to transient signals for feature extraction, as they possess larger amplitudes compared to steady-state signals. Mathematical tools like Fourier transform, Wavelet transform, Hilbert-Huang transform, and empirical mode decomposition [7, 8] are employed for processing transient signals. However, due to the brief duration and low occurrence rate of transient signals, acquiring sufficient data samples for the transient method can be challenging and expensive.

Recent contributions propose fault detection schemes employing deep neural networks (DNNs) as an effective approach [9]. Convolutional neural networks (CNNs), a classic deep structure, have been proven to adaptively extract line-to-ground fault features and accurately locate fault lines [10]. Considering the temporal correlation in fault data, recurrent neural networks (RNNs), known for capturing temporal features, are applied for fault detection and location using PMU data [11]. Furthermore, to mitigate issues like gradient explosion during training, RNN variants such as long short-term memory (LSTM) and gated recurrent unit (GRU) have gained traction in FLD [12, 13].

Although deep learning methods offer significant advantages in terms of feature self-extraction, high-dimensional nonlinear mapping, and noise resistance, there remain two primary challenges that hinder their practical application in the industrial context:

a) The prevalence of short-term and self-recoverable line-to-ground faults, although accounting for the highest proportion, makes it impractical to amass a comprehensive dataset including diverse operational scenarios for effective

This work has been partly funded by the National Natural Science Foundation of China (52177087) and the High-end Foreign Experts Project (G2022163018L). (*Corresponding authors: Jizhong Zhu, Shenglin Li*).

Le Zhang, Jizhong Zhu, Shenglin Li, and Di Zhang are with the School of Electric Power Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: wjtclwwf@163.com; zhujz@scut.edu.cn; eplishl@mail.scut.edu.cn; deezhang_ee@outlook.com). Alberto Borghetti is with University of Bologna, Italy (e-mail: alberto.borghetti@unibo.it)

DNN model training.

b) Measurement data from power monitoring equipment often takes the form of high-speed data streams characterized by dynamic shifts in data distribution and features [14]. Consequently, the need arises to dynamically adjust parameters to adapt to new models in real-time.

Small-sample learning is a rapidly growing and dynamic research field with theoretical foundations encompassing transfer learning, self-supervised learning, and federated learning (FL) [15]. Among these, transfer learning assumes the existence of a source domain with sufficiently ample samples closely aligned with the target domain for preliminary learning. However, the intricate selection of an appropriate source domain poses a challenge [16]. Self-supervised learning uses sufficient unlabeled data to establish pre-trained models, subsequently transferring the gained fault knowledge to the target task [17]. However, due to the sporadic, short-term, and self-recovery properties of most of line-to-ground faults, the accumulation of sufficient unlabeled data comes with significant time-related costs, making the establishment of pre-trained models often impractical. In contrast, the FL scheme, which augments the sample repository by fusing fault data from multiple substations while preserving privacy, emerges as a competitive alternative. For instance, in [18], FL is applied to exchange model parameters, facilitating joint training of fault diagnosis models for photovoltaic stations to enhance accuracy and generalization. Similarly, a fault diagnosis method rooted in CNN and FL is proposed in [19], training a high-precision global diagnostic model without the need to share data. Moreover, the effectiveness of FL in small-sample learning is affirmed through applications in energy prediction [20] and image classification [21]. However, existing models often aggregate limited data to formulate a global model, inadvertently overemphasizing the global aspect while disregarding individual distinctions. Consequently, when fault data from each substation displays heterogeneity, a commonplace occurrence in FLD scenarios, these global parameters might not optimally suit each local model. In such instances, the concept of personalized federated learning (PFL) [22], striking a balance between the optimal global and local models, could be more suitable. PFL has yet to be explored in the context of fault diagnosis.

To cater to the demands of the streaming data environment, researchers are progressively shifting their focus towards online learning, a methodology aimed at rapid model acquisition and incremental updates [23]. As an alternative to DNNs, randomized neural networks, characterized by a single hidden layer structure without backpropagation and iteration, progressively gain attention in fault diagnosis under extensive streaming data due to their effectiveness and efficiency [24]. Among these, the extreme learning machine (ELM) is proven effective in transmission line fault detection [25]. In [26] and [27], the online incremental learning paradigm OSELM for ELM is used in fault diagnosis and energy prediction with a non-stationary environment, successfully balancing accuracy and efficiency. Further enhancing this, in [28], a random vector functional link network (RVFLN), combined with Wavelet transform and Hilbert transform, is utilized for fault line detection in transmission systems. The incremental RVFLN (IRVFLN) exhibits promising prospects for online fault diagnosis in speed sensors within induction motor drive systems [29]. Additionally, a fault diagnosis method relying on a broad learning system (BLS) for rotors is proposed in [30], offering improved adaptability, swifter computation speed, and better classification accuracy. Notably, incremental learning-based BLS, referred to as IBLS, also exhibits robust performance in continuous fault diagnosis [31]. Despite the strong potential of the abovementioned online learning models, they all necessitate predefining the number of hidden nodes, widely recognized as a pivotal hyperparameter affecting capability. Addressing this, Wang et al. [32] introduced the stochastic configuration network (SCN), a concept that bypasses the need to pre-specify hidden node numbers by incrementally stacking hidden nodes in a supervised manner. In contrast to DNNs, SCN is composed of a single hidden layer, and its parameters are obtained through ridge regression instead of backpropagation, which significantly accelerates the model update, making it suitable for handling large-scale high-speed streaming data. Moreover, unlike other random neural networks, SCN adaptively adjusts the number of hidden nodes, avoiding the suboptimality and blindness of fixed presets. As a result of these advantageous characteristics, SCN has been selected as the FLD classifier in this study. Although SCN-based fault detection approaches have been documented in [33, 34], their incremental learning in the context of streaming data remains unexplored.

To tackle the issues outlined above, this paper proposes an FLD method based on PFL and incremental SCN (ISCN) to address the scarcity of fault samples and to facilitate online learning in streaming data scenarios.

The contributions of the paper are summarized as follows.

1) In recognition of the heterogeneity in data across various substations, covering differences in sample size and feature space, a novel FLD model rooted in PFL is devised. Beyond the conventional weighted aggregation of FL, the proposed approach employs the maximum mean discrepancy (MMD) across high-dimensional fault feature spaces to guide local models in the personalized fusion of fault features from other substations. This improves accuracy in scenarios with limited data sizes.
2) A novel FLD method based on ISCN is introduced to address streaming data issues. SCN, an emerging randomized neural network, serves as an FLD classifier without the need to predefine the number of hidden layer nodes. An incremental learning strategy is outlined for SCN, which detects classification performance changes through a sliding window. This strategy strives to continually and non-forgetfully learn new features, therefore, improving FLD accuracy in streaming data settings.

The subsequent sections of this paper are organized as follows. Section II describes the data collection process and

frames the research problem. Section III provides a brief review of the fundamental concepts of SCN as a precursor to Section IV. The latter section introduces the proposed FLD method based on PFL and ISCN. The experimental and numerical simulation results are presented in Section V. Finally, Section VI concludes this paper.

II. Data Collection and Problem Statement

A. Data Collection

To avoid the expenditure associated with installing dedicated data collection equipment, this paper uses the readily available telemetry from dispatching systems. The schematic diagram of the considered feeders is shown in Fig. 1: Fig. 1(a) refers to the 35 kV side, and Fig. 1(b) pertains to the 10 kV side of an actual county-level system. For the presented analysis, the remote measurements in three feeders from 110 kV/35 kV substations and in three feeders from 110 kV/10 kV substations are selected. As shown in Figure 1, sensors within the operational dispatching system record key quantities, such as three-phase bus voltages V_a, V_b, and V_c, as well as three-phase currents I_a, I_b, I_c, active power P, reactive power Q, and power factor $\cos\varphi$ of the feeders.

It is assumed that these remote measurements possess a sufficiently high resolution to accurately characterize the fault features across the entire cycle. Consequently, the proposed procedure collects the mentioned telemetry data from the feeders during the fault occurrences to create the training sample set. The implementation of this data collection procedure includes the following two steps.

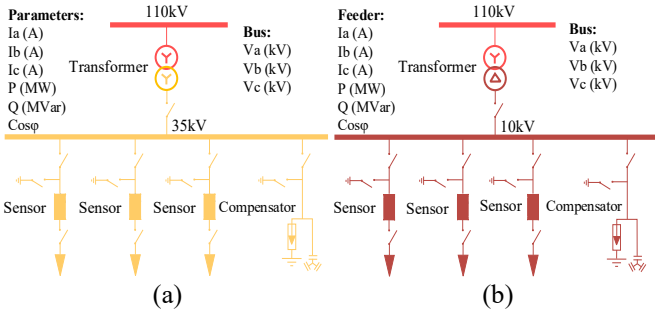

Fig. 1. Scheme of the considered test network: (a) feeders on the 35 kV side; (b) feeders on the 10 kV side.

1) Determination of the fault period. For a line-to-ground fault, the voltage of the grounded phase conductor collapses, while the voltages of the other two conductors tend to increase. This identifies the fault start time t_s. Conversely, when the fault is cleared, all the three-phase voltages return to the rated value range. This moment is recorded as t_e. The interval between these two moments, denoted as $T=[t_s,t_e]$, constitutes the fault window and represents the fault occurrence duration.
2) Collection of the fault data. Telemetry information for each feeder during the fault window T is collected and stored. Additionally, the telemetry captured during each T is transformed into a feature phasor, with the corresponding fault line as a label. This combination of feature phasors and fault line labels forms the training data. The feature phasor of the p-th substation, $\boldsymbol{X}^p$, and its corresponding label, $\boldsymbol{Y}^p$, can be expressed as

$$\boldsymbol{X}^p=\begin{bmatrix}\boldsymbol{x}_1^p\\ \boldsymbol{x}_2^p\\ \vdots\\ \boldsymbol{x}_n^p\end{bmatrix}=\begin{bmatrix}\boldsymbol{x}_{1,1}^p,\boldsymbol{x}_{1,2}^p,\cdots\boldsymbol{x}_{1,m}^p\\ \boldsymbol{x}_{2,1}^p,\boldsymbol{x}_{2,2}^p,\cdots\boldsymbol{x}_{2,m}^p\\ \vdots\\ \boldsymbol{x}_{n,1}^p,\boldsymbol{x}_{n,2}^p,\cdots\boldsymbol{x}_{n,m}^p\end{bmatrix} \tag{1}$$
$$\boldsymbol{Y}^p=[\boldsymbol{y}_1^p,\boldsymbol{y}_2^p,\cdots,\boldsymbol{y}_n^p]^T$$

where $m=6\times l$ is the number of feature parameters of the feeder; l is the number of feeders; n is the number of fault windows; $\boldsymbol{y}_i^p$ corresponds to the fault line of i-th fault window.

B. Problem Statement

In networks characterized by resonant grounding, lines affected by a line-to-ground fault should be disconnected within 1~2 hours to restore the regular power system operation. Line-to-ground faults in diverse operation scenarios are predominantly sporadic, tend to self-recover, and are often influenced by geographical factors and weather conditions. Unfortunately, the application of the arc suppression coil cannot guarantee that all faults are short-term and self-recoverable, although they account for the highest proportion. In the case of short-term and self-recoverable faults, useful data may not be effectively acquired. Consequently, this poses challenges in accumulating adequate data, contributing to the small-sample scenario for detecting non-self-recoverable faults. Therefore, FLD needs to be carried out assuming limited samples.

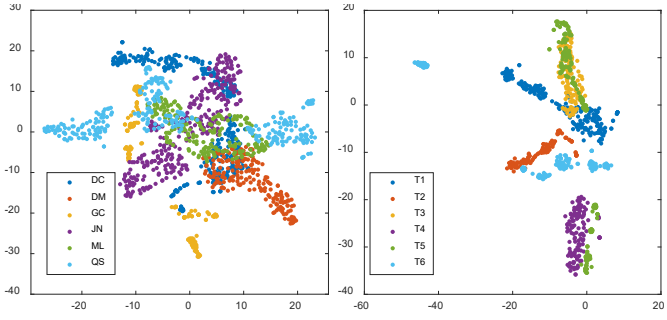

Fig. 2. Training sample data visualization: (a) for different substations; (b) for different fault periods.

To clarify the advantages of PFL as pertinent of this study, t-SNE (t-Distributed Stochastic Neighbor Embedding) is used for the visual representation of the training sample set for each substation, as shown in Figure 2(a), as well as the sample set including six fault periods for the same line, illustrated in Figure 2(b).

The training samples of each substation exhibit heterogeneity in both quantity and spatial distribution due to different operating conditions. The spatial distribution

boundaries of the substation data are indistinct, suggesting the presence of some shared fault features among them. To enhance the feature database while preserving privacy, data from other substations are aggregated through federated learning (FL). However, not all features from other substations are beneficial to the local model due to data heterogeneity. For instance, fault features in substation GC and substation DC differ significantly. Table I presents the Maximum Mean Discrepancy (MMD), a distance metric in the reproduced Hilbert space, between data from various substations. The values in Table I show that the contributions of data from individual substations to the local model are not equal. The MMD between $\boldsymbol{X}^p$ and $\boldsymbol{X}^q$ is calculated using (2). The use of PFL permits models to selectively learn features from other substations, a strategy that improves the accuracy of local models.

$$\begin{aligned} MMD^2(\boldsymbol{X}^p,\boldsymbol{X}^q) &= \left\| \frac{1}{N_p}\sum_{i=1}^{N_p}\phi(x_i^p) - \frac{1}{N_q}\sum_{j=1}^{N_q}\phi(x_j^q) \right\|_{\mathrm{H}}^2 \\ &= \frac{1}{N_p^2}\sum_{i=1}^{N_p}\sum_{j=1}^{N_p}k(x_i^p,x_j^p) - 2\frac{1}{N_p N_q}\sum_{i=1}^{N_p}\sum_{j=1}^{N_q}k(x_i^p,x_j^q) \\ &+ \frac{1}{N_q^2}\sum_{i=1}^{N_q}\sum_{j=1}^{N_q}k(x_i^q,x_j^q) \end{aligned} \tag{2}$$

where N_p and N_q are the number of the p-th and q-th substation sample data; $k(\cdot)$ is the Gaussian kernel function.

TABLE I
MMD BETWEEN DATA FROM DIFFERENT SUBSTATIONS

	DC	DM	GC	JN	ML	QS
DC	0	0.73	1.08	0.78	0.86	0.77
DM	0.73	0	0.84	0.88	0.96	0.87
GC	1.08	1.08	0	0.37	0.52	0.35
JN	0.78	0.88	0.37	0	0.59	0.44
ML	0.86	0.96	0.52	0.59	0	0.58
QS	0.77	0.87	0.35	0.44	0.58	0

In the proposed approach, the FLD is formulated as a supervised classification task $\boldsymbol{Y}^p = f(\boldsymbol{X}^p)$, where f denotes the mapping. Figure 2(b) shows the variability in spatial data distribution in different periods, implying that $\boldsymbol{X}^p$ is susceptible to temporal changes. Consequently, the fixed input-output mapping f may not perform optimally when confronted with streaming data. Additionally, the data from multiple periods exhibit partial space overlap, indicating the presence of shared fault features among them. For example, considerable overlap is observed in the fault features of T3 and T5. This confirms the importance of historical information in guiding future FLD. Therefore, it is effective to integrate online adjustment of f through incremental learning, considering historical information, as a means to sustain the accuracy of FLD.

III. STOCHASTIC CONFIGURATION NETWORK

This section is dedicated to the description of the procedure adopted for the construction of the SCN employed in the FLD procedure.

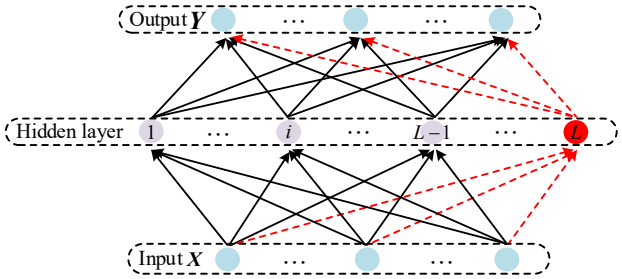

Fig. 3. Structure of the stochastic configuration network.

As shown in Figure 3, SCN consists of an input layer, a single hidden layer, and an output layer, which achieves universal approximation by incrementally adding hidden nodes. The parameters of hidden nodes are randomly assigned within a specific range, while the output layer parameters are determined by ridge regression. The training flow chart of the SCN is summarized in Figure 4, and the detailed explanation follows.

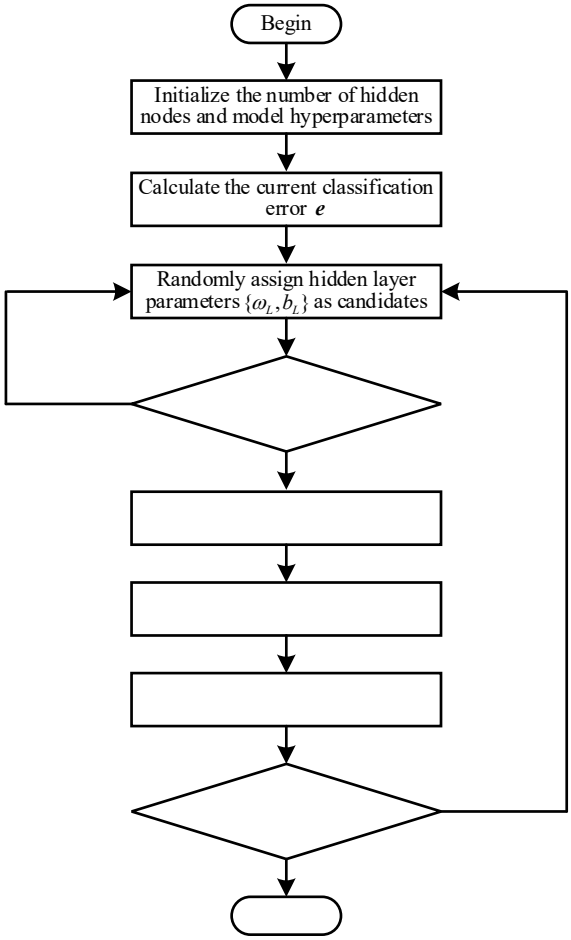

Fig. 4. Training flow chart of SCN algorithm.

Let us represent the dataset as $\{\boldsymbol{X} \in \mathbb{R}^{N\times m}, \boldsymbol{Y} \in \mathbb{R}^{N\times d}\}$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the input and target, respectively. The sample size is denoted as N; m and d are the

dimensions of the input and target, respectively. Assume that the SCN model already includes L-1 hidden nodes, and its mapping is

$$f_{L-1}(\boldsymbol{X}) = \sum_{i=1}^{L-1} \beta_i \square g_i(\omega_i \square \boldsymbol{X} + b_i)(L = 1, 2, \cdots, L_{\max}, f_0 = 0)$$

where $L_{\max}$ is the preset maximum number of hidden nodes, $\{\omega_i, b_i\}$ are the parameters of i-th hidden nodes, β_i is the output layer parameter; $g_i(\square)$ is the sigmoid activation function. The incremental stacking process of L-th node consists of the following steps.

Step 1: Set the expected error tolerance $e_{\max}$, the maximum allowable count of random configurations $T_{\max}$, and the learning parameter r. Choose a set of positive scalars $\Upsilon = \{\lambda_{\min} : \Delta\lambda : \lambda_{\max}\}$ to serve as the range for the hidden node parameters.

Step 2: Calculate the current classification error:

$$\boldsymbol{e} = \boldsymbol{Y} - f_{L-1} = [e_{L-1,1}, e_{L-1,2}, \cdots, e_{L-1,d}] \tag{3}$$

Step 3: Randomly assign parameters $\{\omega_L, b_L\}$ from Υ for a total of $T_{\max}$ times. Screen parameters satisfying the following definitions as candidates:

$$\begin{cases} \xi_{L,i} = \dfrac{(e_{L-1}^T \square h_L)^2}{h_L^T \square h_L} - (1 - r - \mu_L) e_{L-1,i}^T e_{L-1,i} & i = 1, \cdots, d \\ \xi_{L,i} \geq 0 \end{cases} \tag{4}$$

where $h_L = g_L(\omega_L \boldsymbol{X} + b_L)$, $\mu_L = (1 - r)/(L + 1)$.

Step 4: Select parameters $\{\omega_L, b_L\}$ from the candidates that maximize ξ_L given by (5) as the optimal parameters for the L-th node.

$$\xi_L = \sum_{i=1}^{d} \xi_{L,i} \tag{5}$$

Step 5: Calculate the optimal output weights using the least squares method:

$$\boldsymbol{\beta} = \arg\min_{\beta} \left\| \boldsymbol{Y} - \sum_{i=1}^{L} \beta_i \square g_i(\omega_i \square \boldsymbol{X} + b_i) \right\| \tag{6}$$

The solution of (6) is obtained through ridge regression and Moore-Penrose generalized inverse:

$$\begin{aligned} \boldsymbol{\beta} &= \boldsymbol{H}_L^{\dagger} \boldsymbol{Y} \\ \boldsymbol{H}_L &= [h_1, h_2, \cdots, h_L] \end{aligned} \tag{7}$$

where $\boldsymbol{H}_L$ is the output of L hidden nodes; † indicates the pseudo-inverse operator, $\boldsymbol{H}_L^{\dagger} = (\gamma \boldsymbol{I} + \boldsymbol{H}_L^T \boldsymbol{H}_L)^{-1} \boldsymbol{H}_L^T$, with $\boldsymbol{I}$ representing the identity matrix and γ a minimal constant.

Continue iterating through steps 2 to 5 until the classification error $e_{\max}$ reaches the preset threshold or the maximum number of hidden nodes is attained, indicating the conclusion of local model training.

IV. Proposed Online Fault Line Detection Methodology

This section introduces the proposed online FLD method. Firstly, the overall framework and internal logic of the proposed method are described. Subsequently, a detailed description of the implementation process for the adopted PFL and incremental learning techniques is presented.

A. The Overall Framework of the Proposed Method

Figure 5 shows the overall framework of the proposed method, illustrating the flow of information throughout the model training process.

In the initial stage, historical data, collected as described in Section II-A, are normalized and used to train multiple local SCN-based FLD classification models in a separate manner. Subsequently, the obtained hidden layer parameters of each model are uploaded onto the server. The global model is obtained through the weighted aggregation of these uploaded parameters. This strategy effectively addresses the challenges due to the heterogeneity in sample data sizes.

To preserve privacy and address distribution heterogeneity, the MMD, quantifying the distribution distance within high-dimensional feature spaces, guides the personalized update of local parameters. By extending beyond exclusive reliance on local data, this approach allows obtaining personalized local models that combine advantageous features through a weighted aggregation of both global and personalized parameters, effectively tackling the small-sample problem.

Finally, when operating in a streaming data scenario, a sliding window mechanism is used to detect changes in the performance of local models. When performance deteriorates, the model undergoes swift and incremental updates using the unlearned data, thus ensuring the preservation of accuracy.

B. Fault Line Detection Model Based on Personalized Federated Learning

The second module shown in Figure 5 illustrates the proposed personalized learning process. Consistent with Section III, the sample data of the i-th substation are denoted as $\{\boldsymbol{X}^i \in \square^{N_i \times m}, \boldsymbol{Y}^i \in \square^{N_i \times d}\}_{i=1}^{C}$; where $\boldsymbol{X}^i$ and $\boldsymbol{Y}^i$ are the input and target; N_i is the number of the sample data; m and d are the dimensions of the input and target, respectively; C is the number of substations. As in Section III, we consider the incremental stacking of the L-th hidden node. The training steps proceed the follows.

1) ***Upload of local model parameters***: the local parameters $\{\omega_L^i, b_L^i\}_{i=1}^{C}$ of the L-th hidden node, obtained by Step 1-Step 4 described in Section III, are transferred to a central server for weighted aggregation.

2) ***Upload of hidden layer output***: the high-dimensional data features extracted by the hidden layer are transmitted to the central server. The output of the L-th hidden node is

$$h_L^i = g_L^i(\omega_L^i \boldsymbol{X}^i + b_L^i)(i = 1, 2, \cdots, C) \tag{8}$$

where $g_L^i(\square)$ is the sigmoid activation function.

3) ***Calculation of the global model parameters***: in the central server, the primary objective is to merge the fault features. Conventional FL permits the weighted aggregation of the uploaded local parameters to obtain global parameters

$\{\omega_L^*, b_L^*\}$:

$$\omega_L^* = \sum_{i=1}^{C} \frac{N_i}{N} \omega_L^i, \quad b_L^* = \sum_{i=1}^{C} \frac{N_i}{N} b_L^i \tag{9}$$

where $N = \sum_{i=1}^{C} N_i$ represents the total number of samples.

4) ***Calculation of the personalized model parameters***: the central server receives the high-dimensional features $\boldsymbol{h}_L = [h_L^1, h_L^2, \cdots, h_L^C]$ from multiple substations and calculates the distance matrix $\boldsymbol{M}_L^{ij}$ between different feature spaces as:

$$M_L^{ij} = MMD(h_L^i, h_L^j)(i, j = 1, 2 \cdots, C) \tag{10}$$

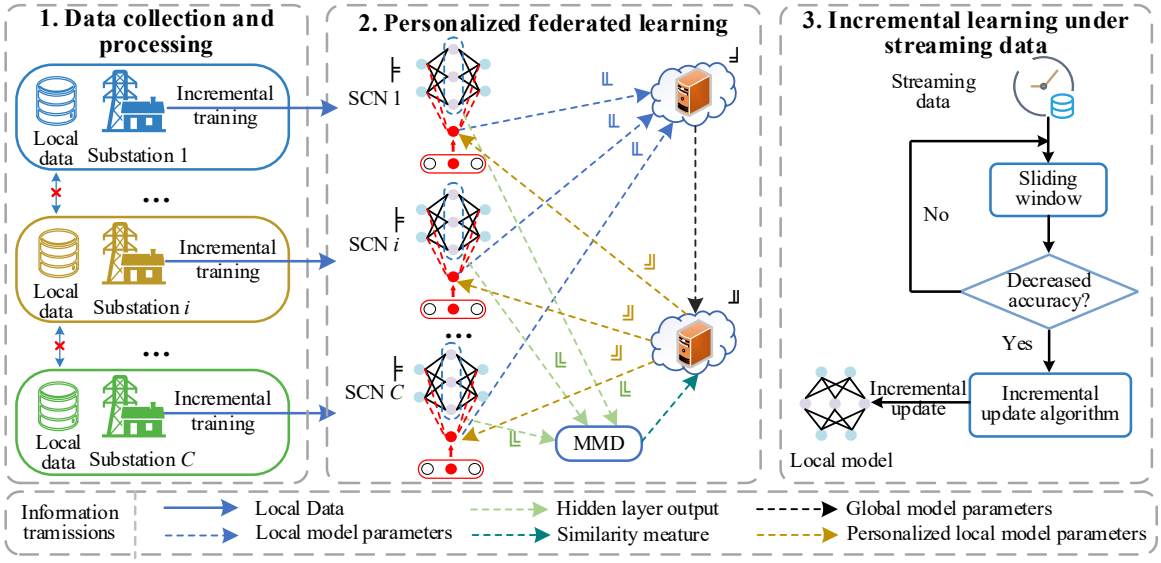

Fig. 5. Framework of FLD method based on personalized federated learning and incremental stochastic configuration network.

The element $\boldsymbol{M}_L^{ij}$ shows a negative correlation with feature similarity between the corresponding substations. The goal of PFL is to aggregate similar parameters. Thus, on the basis of $\boldsymbol{M}_L^{ij}$, a normalized feature coefficient matrix is constructed as:

$$F_L^{ij} = \frac{e^{-M_L^{ij}}}{\sum_{k=1}^{C} e^{-M_L^{ik}}} (i, j = 1, 2, \cdots, C) \tag{11}$$

Subsequently, F_L^{ij} is used to accomplish personalized weighted aggregation of local model parameters:

$$\overline{\omega}_L^i = \sum_{j=1}^{C} F_L^{ij} \omega_L^j, \quad \overline{b}_L^i = \sum_{j=1}^{C} F_L^{ij} b_L^j \tag{12}$$

where $\{\overline{\omega}_L^i, \overline{b}_L^i\}$ are the personalized parameters. In order to coordinate personalized and global parameters, a global factor $0 < \alpha < 1$ is introduced, and the final personalized local model parameters are expressed as

$$\begin{aligned} \tilde{\omega}_L^i &= \alpha\, \omega_L^* + (1-\alpha)\, \overline{\omega}_L^i \\ \tilde{b}_L^i &= \alpha\, b_L^* + (1-\alpha)\, \overline{b}_L^i \end{aligned} \tag{13}$$

5) ***Distribution of personalized model parameters***: the personalized local parameters $\{\tilde{\omega}_L^i, \tilde{b}_L^i\}$ are dispatched from the server to the relevant local client. Subsequently, these parameters replace $\{\omega_L^i, b_L^i\}$ as the L-th hidden node parameters, as described in Step 4 of Section III.

6) ***Calculation of output layer parameters***: following the acquisition of $\{\tilde{\omega}_L^i, \tilde{b}_L^i\}$, the output layer parameter $\boldsymbol{\beta}^i$ can be calculated as

$$\begin{aligned} \{\boldsymbol{\beta}^i\}_{i=1}^{C} &= \{\boldsymbol{H}_L^{i\dagger} \boldsymbol{Y}^i\}_{i=1}^{C} \\ \boldsymbol{H}_L^i &= [h_1^i, h_2^i, \cdots, h_L^i], \quad h_L^i = g_L^i(\tilde{\omega}_L^i \boldsymbol{X}^i + \tilde{b}_L^i) \end{aligned} \tag{14}$$

The above-described training process is summarized in **Algorithm 1**.

Algorithm 1: FLD based on PFL

SCN i:
Input: $\{\boldsymbol{X}^i \in \mathbb{R}^{N_i \times m}, \boldsymbol{Y}^i \in \mathbb{R}^{N_i \times d}\}_{i=1}^{C}$, hidden node index L, error of L hidden nodes $\boldsymbol{e}$, expected error tolerance $\{e_{max}^i\}_{i=1}^{C}$, maximum hidden node number L_{max}, maximum times of random configuration T_{max}, a set of positive scalars $\Upsilon = \{\lambda_{min} : \Delta\lambda : \lambda_{max}\}$

Initialization: $\boldsymbol{e}$=$\boldsymbol{Y}^i$;
while $L \le L_{max}$ *and* $\|\boldsymbol{e}\| \ge e_{max}^i$ **do**
  for $\lambda \in \Upsilon$ **do**
    for $k = 1, 2, \cdots, T^{max}$ **do**
      Randomly assign ω_L^i and b_L^i from $[-\lambda, \lambda]^m$ and $[-\lambda, \lambda]$;
      Calculate $\zeta_{L,i}$ and ζ_L by (4) and (5); Save ζ_L to Ω;
      Save candidates ω_L^i and b_L^i to $\boldsymbol{\omega}$ via Step 3 in Section III;
    end
    if $\boldsymbol{\omega}$ *not empty* **then**
      Select ω_L^i and b_L^i from $\boldsymbol{\omega}$ that maximize ζ_L in $\boldsymbol{\omega}$;
      Calculate hidden layer output $h_L^i = g_L(\omega_L^i \boldsymbol{X} + b_L^i)$;
    end
  end
  Transfer ω_L^i, b_L^i, and h_L^i to the server and go to server;
  Get $\tilde{\omega}_L^*$ and $\tilde{b}_L^*$ from the server; $\tilde{\omega}_L^*$ and $\tilde{b}_L^*$ replace ω_L^i and b_L^i;
  Calculate hidden layer output $h_L^i = g_L(\tilde{\omega}_L^* \boldsymbol{X} + \tilde{b}_L^*)$;
  $\boldsymbol{H}_L^i = [h_1^i, h_2^i, \cdots, h_L^i]$;
  Calculate output layer parameters $\boldsymbol{\beta}^i$ by (14);
  Renew $\boldsymbol{e}$:=$\boldsymbol{H}_L^i \boldsymbol{\beta}^i$-$\boldsymbol{Y}^i$; $L = L + 1$;
end
Server:
Input: $\{\omega_L^i\}_{i=1}^{C}$, $\{b_L^i\}_{i=1}^{C}$, and $\{h_L^i\}_{i=1}^{C}$
Calculate the global parameters ω_L^* and b_L^* by (9);
Calculate the feature coefficient matrix F_L^{ij} by (10) and (11);
Calculate personalized parameters $\overline{\omega}_L^*$ and $\overline{b}_L^*$ by (12);
Calculate local personalized parameters $\tilde{\omega}_L^*$ and $\tilde{b}_L^*$ by (13);
Send $\tilde{\omega}_L^*$ and $\tilde{b}_L^*$ to the corresponding local client and go to client.

C. Fault Line Detection Based on Incremental Stochastic Configuration Networks for Streaming Data

As illustrated in the third module of Figure 5, the incremental SCN-based FLD method using streaming data can be divided into two parts: 1) detection of classification accuracy changes through a sliding window; 2) update of the model through an incremental learning algorithm.

Algorithm 2: FLD based on ISCN

Input: Batch streaming data $\{\boldsymbol{X^a}\in\mathbb{R}^{N_a\times m},\boldsymbol{Y^a}\in\mathbb{R}^{N_a\times d}\}_{i=1}^{C}$, hidden layer parameters $\boldsymbol{\omega^i}=[\omega_1^i,\omega_2^i,\cdots,\omega_L^i]$ and $\boldsymbol{b^i}=[b_1^i,b_2^i,\cdots,b_L^i]$ of i-th SCN model, hidden layer output $\boldsymbol{H_L^i}$ and output layer parameters $\boldsymbol{\beta^i}$ of i-th SCN model

Calculate hidden layer output $\boldsymbol{H_L^a}$ under $\{\boldsymbol{X^a},\boldsymbol{Y^a}\}$, $\boldsymbol{H_L^a}=g_L^a(\boldsymbol{\omega^i X^a}+\boldsymbol{b^i})$;
$\boldsymbol{D_a^T}=\boldsymbol{H_L^a H_L^{i\dagger}}$; $\boldsymbol{A_a}=\boldsymbol{H_L^a}-\boldsymbol{D_a^T H_L^i}$;
if $\boldsymbol{A_a}=0$ **then**
  $\boldsymbol{Q_a}=\boldsymbol{H_L^{i\dagger}}\boldsymbol{D_a}(1+\boldsymbol{D_a^T D_a})^{-1}$;
else
  $\boldsymbol{Q_a}=\boldsymbol{A_a^\dagger}$
end
Updated output layer parameters $\tilde{\boldsymbol{\beta}}^{\boldsymbol{i}}=\boldsymbol{\beta^i}+\boldsymbol{Q_a}(\boldsymbol{Y^a}-\boldsymbol{H_L^a}\boldsymbol{\beta^i})$

1) ***Detection of the classification accuracy changes***. To mitigate single-sample randomness, a sliding window approach is used to detect classification accuracy changes in batch increments. The classification accuracy within the sliding window of historical data serves as the performance evaluation benchmark. A scenario where the accuracy within the sliding window of the streaming data falls below the predefined minimum performance threshold indicates the inadequacy of the present SCN model for the window data, necessitating an update.

2) ***Incremental model update***. Similarly to prior procedures, the window data to be learned is denoted as $\{\boldsymbol{X}^a\in\mathbb{R}^{N_a\times m},\boldsymbol{Y}^a\in\mathbb{R}^{N_a\times d}\}$; where $\boldsymbol{X}^a$ and $\boldsymbol{Y}^a$ are the input and target; N_a is the sample size; m and d are the dimension of input and target, respectively. Within this context, the i-th SCN with L hidden nodes incrementally learns $\{\boldsymbol{X}^a,\boldsymbol{Y}^a\}$. The hidden layer output of the i-th SCN under $\{\boldsymbol{X}^a,\boldsymbol{Y}^a\}$ is denoted as $\boldsymbol{H}_L^a$; where $\boldsymbol{H}_L^a=[h_1^a,h_2^a,\cdots,h_L^a]$, $h_L^a=g_L^a(\tilde{\omega}_L^a\boldsymbol{X}^i+\tilde{b}_L^a)$. Incremental learning promotes the updated output layer parameters $\tilde{\boldsymbol{\beta}}^i$ to satisfy both the mappings $\boldsymbol{X}^i\to\boldsymbol{Y}^i$ and $\boldsymbol{X}^a\to\boldsymbol{Y}^a$ simultaneously, i.e.,

$$\begin{bmatrix}\boldsymbol{H}_L^i\\ \boldsymbol{H}_L^a\end{bmatrix}\tilde{\boldsymbol{\beta}}^i\approx\begin{bmatrix}\boldsymbol{Y}^i\\ \boldsymbol{Y}^a\end{bmatrix} \tag{15}$$

As in (7) and (14), $\tilde{\boldsymbol{\beta}}^i$ is given by:

$$\begin{aligned}\tilde{\boldsymbol{\beta}}^i&=\begin{bmatrix}\boldsymbol{H}_L^i\\ \boldsymbol{H}_L^a\end{bmatrix}^\dagger\begin{bmatrix}\boldsymbol{Y}^i\\ \boldsymbol{Y}^a\end{bmatrix}\\ \begin{bmatrix}\boldsymbol{H}_L^i\\ \boldsymbol{H}_L^a\end{bmatrix}^\dagger&=[\boldsymbol{H}_L^{i\dagger}-\boldsymbol{Q}_a\boldsymbol{D}_a^T\mid\boldsymbol{Q}_a]\end{aligned} \tag{16}$$

where

$$\boldsymbol{Q}_a=\begin{cases}\boldsymbol{A}_a^\dagger & \boldsymbol{A}_a\neq 0\\ \boldsymbol{H}_L^{i\dagger}\boldsymbol{D}_a(1+\boldsymbol{D}_a^T\boldsymbol{D}_a)^{-1} & \boldsymbol{A}_a=0\end{cases}$$
$$\boldsymbol{A}_a=\boldsymbol{H}_L^a-\boldsymbol{D}_a^T\boldsymbol{H}_L^i \tag{17}$$
$$\boldsymbol{D}_a^T=\boldsymbol{H}_L^a\boldsymbol{H}_L^{i\dagger}$$

By combining (15), (16), and (17), $\tilde{\boldsymbol{\beta}}^i$ can be expressed as

$$\begin{aligned}\tilde{\boldsymbol{\beta}}^i&=\boldsymbol{H}_L^{i\dagger}\boldsymbol{Y}^i+\boldsymbol{Q}_a(\boldsymbol{Y}^a-\boldsymbol{H}_L^a\boldsymbol{H}_L^{i\dagger}\boldsymbol{H}_L^i\boldsymbol{\beta}^i)\\ &=\boldsymbol{\beta}^i+\boldsymbol{Q}_a(\boldsymbol{Y}^a-\boldsymbol{H}_L^a\boldsymbol{\beta}^i)\end{aligned} \tag{18}$$

The process adopted to build the proposed ISCN is summarized in **Algorithm 2**.

V. Case Study

This section describes the comparison experiments conducted under small-sample and streaming data scenarios intended to validate the progressiveness and effectiveness of the proposed method. Three state-of-the-art random neural networks, ELM, BLS, and RVFLN, along with their corresponding federated learning and incremental learning variants, are applied as benchmarks for comparison. In addition, the computational efficiency of the proposed method is analyzed.

A. Experimental Data and Environment

As described in Section II-A, a sample set of fault data from six substations of a real county-level power grid company in China, spanning the years 2016 to 2019, is collected. Table II and Table III provide the feeder parameters and specific sample sizes for each substation. The sample set of each substation is evenly divided into two parts, assumed to represent historical and streaming data, respectively. All procedures are implemented by using Matlab 2022b, and the tests are conducted on a PC platform with an Intel (R) Core (TM) i7-11700F CPU and 16GB RAM.

TABLE III
Fault Sample Size

	DC	DM	GC	JN	ML	QS
L1	5406	1875	2242	2839	4290	5723
L2	2357	2184	3042	1687	3427	8526
L3	3562	2265	1698	3421	4721	6782
Total	11325	6324	6982	7947	12438	21031

B. Experimental Comparison Under Small Samples

To evaluate the effectiveness of the proposed method for small sample sizes, a total of nine tests are conducted for each substation, varying the sizes of the historical data. These tests compare the following approaches: 1) ELM; 2) ELM-FL: ELM with federated learning; 3) RVFLN; 4) RVFLN-FL: RVFLN with federated learning; 5) BLS; 6) BLS-FL: BLS with federated learning; 7) SCN; 8) SCN-FL: SCN with federated learning; 9) SCN-PFL (Our method): SCN with personalized federated learning.

The training set includes 80% of the historical samples, divided equally into ten parts for model training with a small-sample data set. The remaining 20% is designated as the test

set, used to evaluate the performance of the trained models.

Figure 6 illustrates the accuracy of each model for the six substations and for different training data sizes. Overall, the proposed method shows the best FLD accuracy, indicating its effectiveness in small-sample environments.

In detail, the comparative analysis shows the following characteristics.

1) SCN vs. other models without FL. As shown in Figure 6, SCN outperforms ELM, RVFLN, and BLS. Unlike other random networks relying on stochastic parameter configuration for hidden layers, SCN is constructed with a supervisory mechanism, which contributes to the model performance. Additionally, SCN dynamically increases hidden layer nodes, thereby enhancing optimization efficiency.

TABLE II
FEEDER PARAMETERS OF EACH SUBSTATION

Substations	DC			JN			ML			DM			GC			QS		
Lines	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3
Level (kV)	35									10								
Length (km)	2.5	3	2.7	3.7	5.6	2.8	2.2	7.7	3.5	1.5	4	2.1	1.6	2.4	2.2	2.3	2.4	1.8
Impedance (Ω/km)	0.072+*j*0.103									0.181+*j*0.095								

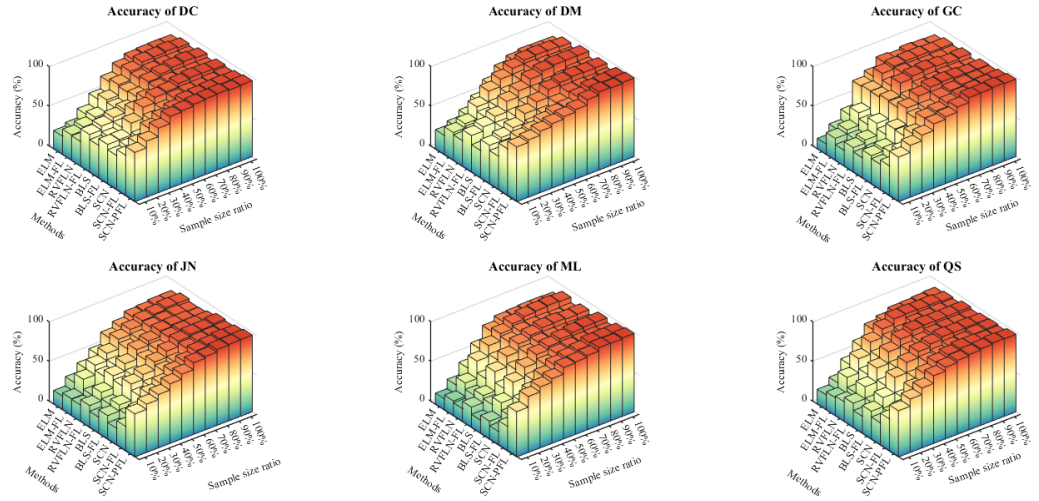

Fig. 6. Accuracy of each model for six different substations and different training data sizes.

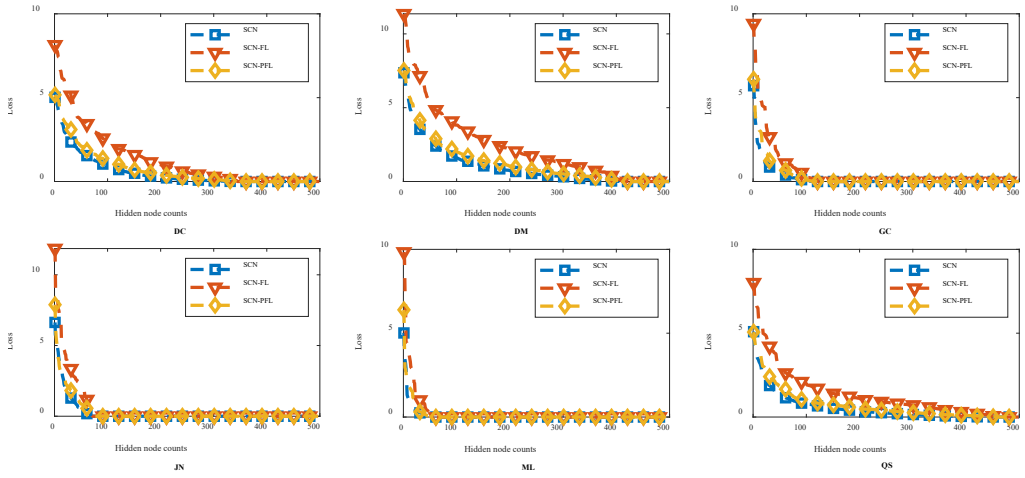

Fig. 7. Performance of SCN, SCN-FL, and SCN-PFL with 500 additive nodes. The loss values measure the alignment of model predictions with true classifications.

2) Models without FL vs. models with FL. Figure 6 shows that the accuracy of models with FL is better than the accuracy of their counterparts without FL, since FL aggregates fault features from other substations, which strengthens the model classification and generalization capabilities. To further demonstrate the advantages of FL for small-sample data sets, Table IV presents the FLD accuracy data relevant to substation DM. When the sample size exceeds 80%, the accuracy of each model tends to stabilize, indicating that the sample is sufficient. In this scenario, FL-based models improve accuracy by up to 11%, exemplified by the improvement from BLS to BLS-FL when using 80% training

samples. In contrast, when sample data is limited, FL-based models improve accuracy by up to 20.3%, as shown in the comparison between ELM and ELM-FL using 40% training samples.

3) SCN-PFL vs. SCN and SCN-FL. SCN-FL outperforms SCN in terms of FLD accuracy, particularly for small-sample data sets, which aligns with expectations. This superiority of the proposed method over SCN-FL can be attributed to the degradation of FL performance due to the heterogeneity of fault data, effectively addressed by the proposed PFL scheme.

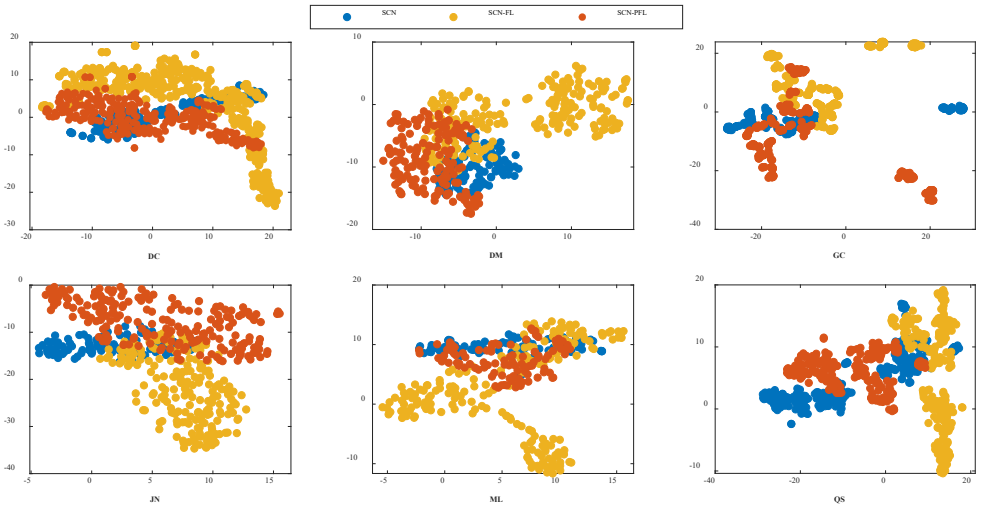

Fig. 8. Visualization of extracted high-dimensional hidden features.

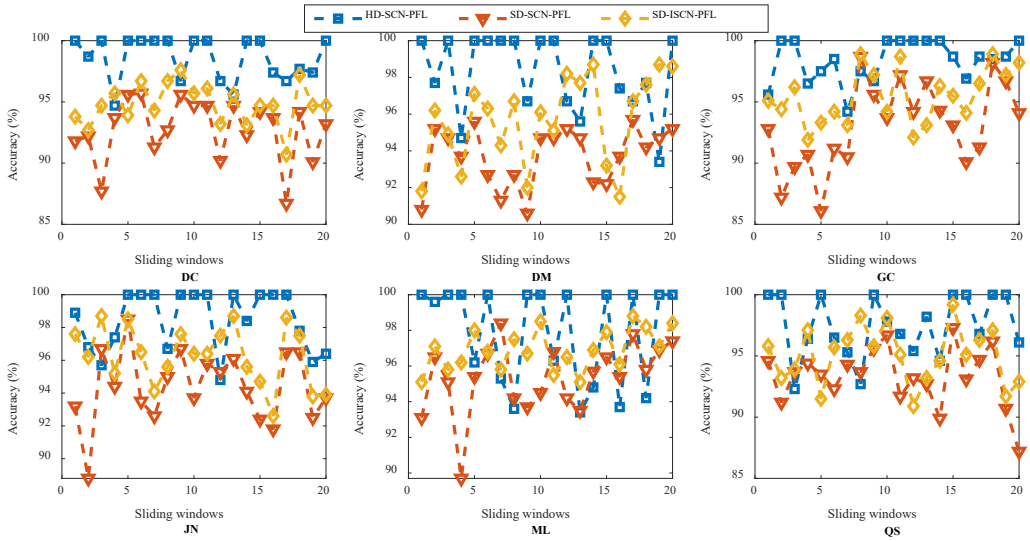

Fig. 9. Performance of SCN-PFL and ISCN-PFL using historical and streaming data.

Figure 7 illustrates the convergence process of the three methods using cross entropy as a metric on the training set, where the cross entropy is used as the loss function. The figure shows that SCN achieves the fastest convergence, followed by SCN-PFL, while SCN-FL exhibits the slowest convergence. This can be explained by the incremental updates in SCN, which yield optimal hidden layer parameters using local fault data. The global parameters weighted aggregation in FL may not be optimal for local data, leading SCN-FL to stack more hidden nodes to reduce errors. Likewise, PFL aggregates other model parameters, causing the weighted aggregation of parameters to deviate from the local optimal parameter space. However, due to the presence of the PFL mechanism, personalized parameters in SCN-PFL tend to be closer to local optimal parameters compared to global parameters. Consequently, SCN-PFL requires fewer hidden nodes than SCN-FL to achieve optimal performance. Despite the slower convergence of the proposed method, it demonstrates improved generalization capability and accuracy on the test set due to the fusion of additional fault features.

Moreover, to illustrate the effectiveness and superiority of the adopted PFL, t-SNE is used to show the high-dimensional hidden features (i.e., outputs of the hidden layer) extracted by the three models: SCN, SCN-FL, and SCN-PFL, as shown in Figure 8. The high-dimensional features extracted by SCN-FL and SCN-PFL are spatially similar to those extracted by SCN, but not entirely coincident. This is due to the incorporation of fault features from other substations by FL and PFL, while acquiring local data features, resulting in the expansion and shift of the feature space. Furthermore, under the data set of each substation, when compared with SCN-FL, the features extracted by SCN-PFL are spatially closer to those extracted by SCN. This proximity implyes that the proposed PFL scheme is more personalized and accurate in formulating the

input-output mapping for the data of each substation.

C. Experimental Comparison for Streaming Data

The aforementioned optimal models are adopted as base models for FLD using streaming data. To illustrate the dynamics of streaming data, Figure 9 presents the FLD accuracy of three scenarios:

1) HD-SCN-PFL: accuracy of SCN-PFL using historical data window;

2) SD-SCN-PFL: accuracy of SCN-PFL using streaming data window;

3) SD-ISCN-PFL: accuracy of SCN-PFL with incremental learning for streaming data window.

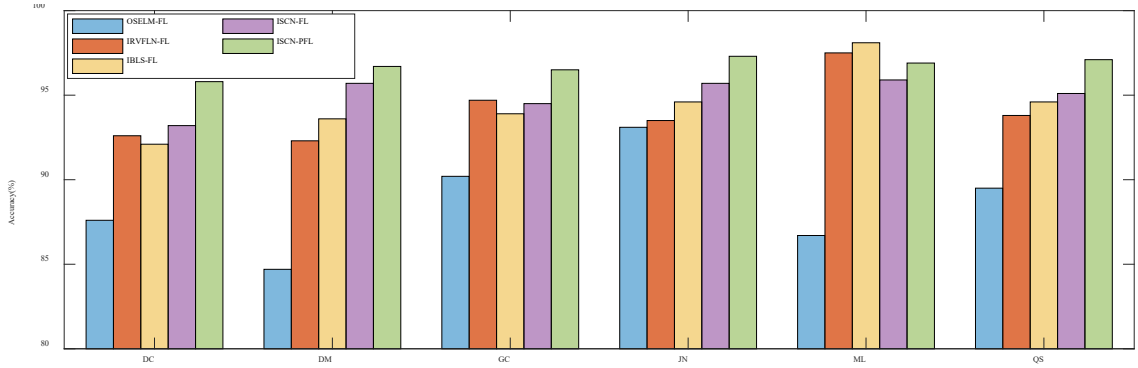

Fig. 10. Online FLD accuracy with incremental learning.

As discussed in Section IV-C, the minimum classification accuracy within the sliding window of historical data serves as a baseline for assessing model degradation in the streaming data set. Figure 9 shows that, for each substation, SD-SCN-PFL can occasionally be lower than the minimum HD-SCN-PFL, indicating changing streaming data and subsequent degradation in model performance over time, as explained in Section II-B. For instance, in the case of substation GC, the minimum HD-SCN-PFL and SD-SCN-PFL are 94.2% and 86.1%, respectively, with a difference of 8.1%. This discrepancy forces incremental model updates. The comparison between SD-SCN-PFL and SD-ISCN-PFL shows that the proposed incremental learning strategy effectively improves FLD accuracy in the streaming data scenario. Indeed, the incremental nature of the proposed method guarantees the retention of acquired information during the learning process.

TABLE IV
ACCURACY FOR THE CASE OF SUBSTATION DM (IN %)

Methods	Sample size ratio (%)									
	10	20	30	40	50	60	70	80	90	100
ELM	24.1	29.5	37.6	**49.9**	64.5	76.8	83.2	85.6	85.1	84.9
ELM -FL	35.2	40.6	56.2	**70.2**	79.6	87.5	92.1	93.1	92.7	91.5
RVFLN	32.4	37.5	50.2	57.8	67.9	80.2	87.9	87.6	86.9	86.7
RVFLN -FL	46.2	51.5	58.9	72.1	79.7	88.9	93.3	95.4	95.6	95.5
BLS	34.8	35.6	51.2	60.8	68.8	79.8	82.6	**83.6**	87.8	88.7
BLS -FL	48.8	52.8	59.6	71.6	80.6	90.6	93.7	**94.6**	94.9	95.1
SCN	42.3	48.6	50.6	63.5	71.3	80.5	85.7	87.3	88.6	90.1
SCN -FL	59.2	67.2	70.3	76.5	80.7	88.1	90.2	94.4	95.7	95.5
SCN -PFL	**69.0**	**72.1**	**77.4**	**81.1**	**84.2**	**92.5**	**94.2**	**97.5**	**97.6**	**97.9**

To validate the superiority of the proposed method, Figure 10 compares FLD accuracy for each optimal model with incremental learning: 1) OSELM-FL; 2) IRVFLN-FL; 3) IBLS-FL; 4) ISCN-FL; 5) ISCN-PFL (i.e., the proposed method). Overall, ISCN-PFL demonstrates the highest performance, indicating that the proposed model outperforms existing state-of-the-art online learning algorithms, both using historical data and streaming data.

D. Computational Efficiency

The computational efficiency of the models is examined with a focus on substation DC. Table V compares the training time, the test time, and the online learning time for the optimal models. To ensure a fair and consistent comparison, the training process for each model is terminated when the loss value falls below the threshold of 0.01. The time spent to reach this point is then considered as the training time. Moreover, all models are configured with optimal parameters, as shown in Table VI. Unlike other models that empirically determine the optimal number of hidden nodes, the proposed method dynamically adjusts the number of hidden nodes. A detailed comparative assessment follows.

TABLE V
COMPARISON OF COMPUTATIONAL TIME

Methods	Time (s)		
	Training	Test	Online learning
ELM-FL	12.5	0.09	5.1
RVFLN-FL	7.9	0.06	3.2
BLS-FL	8.7	0.07	3.6
SCN-FL	**3.5**	0.02	2.3
SCN-PFL	10.2	**0.01**	**1.2**

Training time. The duration of training for ELM-FL, RVFLN-FL, BLS-FL, and SCN-FL is directly influenced by the number of hidden nodes. ELM, which relies on multiple nodes in a single-hidden layer to improve accuracy, requires a longer training time. BLS-FL, which incorporates more hidden nodes and uses an enhancement layer for feature extraction, necessitates more time compared to RVFLN-FL. In contrast, SCN-FL, benefiting from its supervision mechanism, can converge with fewer hidden nodes, resulting in the shortest training time. When PFL replaces FL, training time increases significantly. This can be attributed to the increased complexity introduced by the calculation of the MMD coefficient matrix, which slows down the convergence speed.

Test time. The testing phase solely involves forward

calculations and is proportionate to the number of hidden nodes. Since the proposed method uses the minimum number of hidden nodes, it exhibits the highest efficiency.

Online learning time. Online learning time is typically shorter than training time for two main reasons. Firstly, rather than retraining the entire stream of data, only the window data causing performance degradation necessitates relearning. This improves the efficiency of online learning. Secondly, the proposed incremental learning approach is recursive, as shown by (18), and requires minimal computing resources. Furthermore, due to the reduced number of hidden nodes, the proposed method achieves high efficiency in online learning, thereby enabling fast updates with streaming data.

TABLE VI
PARAMETERS SETTING OF THE MODELS

Methods	Parameter settings
ELM-FL	Number of hidden nodes: 1000
RVFLN-	Number of enhancement nodes: 500
BLS-FL	Number of feature node groups: 50 Number of feature nodes in each group: 10 Number of enhancement nodes: 300
SCN-FL	Number of hidden nodes: 350
SCN-	Number of hidden nodes: 300

VI. CONCLUSION

This paper presents an innovative approach for online fault line detection in power distribution networks with resonant grounding. The proposed approach, which efficiently addresses the challenge posed by small-sample and streaming data scenarios, combines personalized federated learning (PFL) and incremental stochastic networks (ISCN).

PFL facilitates the aggregation of fault features from various substations while maintaining personalization in the presence of heterogeneous data, effectively mitigating the issues associated with limited sample sizes. ISCN ensures continuous model updates in real-time, therefore maintaining accuracy even in scenarios involving high-speed dynamic data streaming.

Extensive tests conducted using real data demonstrate that the proposed method outperforms state-of-the-art online learning algorithms. Given its strong performance in small-sample and streaming data environments, the proposed method holds substantial promise for practical applications in real power systems.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (52177087) and the High-end Foreign Experts Project (G2022163018L).

REFERENCES

[1] X. Wei, X. Wang, J. Gao, D. Yang, K. Wei, and L. Guo, "Faulty Feeder Detection for Single-Phase-to-Ground Fault in Distribution Networks Based on Transient Energy and Cosine Similarity," *IEEE Transactions on Power Delivery,* Article vol. 37, no. 5, pp. 3968-3979, Oct 2022.

[2] X. Wei, D. Yang, X. Wang, B. Wang, J. Gao, and K. Wei, "Faulty Feeder Detection Based on Fundamental Component Shift and Multiple-Transient-Feature Fusion in Distribution Networks," *IEEE Transactions on Smart Grid,* Article vol. 12, no. 2, pp. 1699-1711, Mar 2021.

[3] L. Zichang, L. Yadong, Y. Yingjie, W. Peng, and J. Xiuchen, "An Identification Method for Asymmetric Faults With Line Breaks Based on Low-Voltage Side Data in Distribution Networks," *IEEE Transactions on Power Delivery,* Article vol. 36, no. 6, pp. 3629-3639, Dec 2021.

[4] X. Lin *et al.*, "Zero-sequence compensated admittance based faulty feeder selection algorithm used for distribution network with neutral grounding through Peterson-coil," *International Journal of Electrical Power & Energy Systems,* Article vol. 63, pp. 747-752, Dec 2014.

[5] Z. Zhang, X. Liu, and Z. Piao, "Fault line detection in neutral point ineffectively grounding power system based on phase-locked loop," *IET Generation Transmission & Distribution,* Article vol. 8, no. 2, pp. 273-280, Feb 2014.

[6] L. Jiapei, Z. Guoqing, L. Hongbo, W. Guizhong, H. Yingwei, and L. Wenlong, "A Review of Fault Location Methods for Small Current Grounding Systems," *IOP Conference Series: Materials Science and Engineering,* Conference Paper; vol. 677, pp. 052045 (7 pp.)-052045 (7 pp.), 2019 2019.

[7] H. Gao, Z. Wang, A. Tang, C. Han, F. Guo, and B. Li, "Research on Series Arc Fault Detection and Phase Selection Feature Extraction Method," *IEEE Transactions on Instrumentation and Measurement,* Article vol. 70, 2021 2021, Art. no. 2004508.

[8] T. Jin, F. Zhuo, and M. A. Mohamed, "A Novel Approach Based on CEEMDAN to Select the Faulty Feeder in Neutral Resonant Grounded Distribution Systems," *IEEE Transactions on Instrumentation and Measurement,* Article vol. 69, no. 7, pp. 4712-4721, Jul 2020.

[9] A. Moradzadeh, B. Mohammadi-Ivatloo, K. Pourhossein, and A. Anvari-Moghaddam, "Data Mining Applications to Fault Diagnosis in Power Electronic Systems: A Systematic Review," *IEEE Transactions on Power Electronics,* Review vol. 37, no. 5, pp. 6026-6050, May 2022.

[10] M.-F. Guo, J.-H. Gao, X. Shao, and D.-Y. Chen, "Location of Single-Line-to-Ground Fault Using 1-D Convolutional Neural Network and Waveform Concatenation in Resonant Grounding Distribution Systems," *IEEE Transactions on Instrumentation and Measurement,* Article vol. 70, 2021 2021, Art. no. 3501009.

[11] M. R. Shadi, M.-T. Ameli, and S. Azad, "A real-time hierarchical framework for fault detection, classification, and location in power systems using PMUs data and deep learning," *International Journal of Electrical Power & Energy Systems,* Article vol. 134, Jan 2022, Art. no. 107399.

[12] S. Belagoune, N. Bali, A. Bakdi, B. Baadji, and K. Atif, "Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems," *Measurement,* Article vol. 177, Jun 2021, Art. no. 109330.

[13] Q. Liu, T. Liang, and V. Dinavahi, "Real-Time Hierarchical Neural Network Based Fault Detection and Isolation for High-Speed Railway System Under Hybrid AC/DC Grid," *IEEE Transactions on Power Delivery,* Article vol. 35, no. 6, pp. 2853-2864, Dec 2020.

[14] H. Ye, Z. Zheng, J.-R. C. Cheng, B. Hable, and K. Liu, "Online monitoring of high-dimensional asynchronous and heterogeneous data streams for shifts in location and scale," *International Journal of Production Research,* Article; Early Access 2023.

[15] 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 Transactions on Industrial Informatics,* Article vol. 17, no. 12, pp. 8495-8504, Dec 2021.

[16] W. Sima, H. Zhang, and M. Yang, "Edge-Cloud Collaboration Detection Approach for Small-Sample Imbalanced Faults in Power Lines," *IEEE Transactions on Instrumentation and Measurement,* Article vol. 71, 2022 2022, Art. no. 3513711.

[17] G. Huang, I. Laradji, D. Vazquez, S. Lacoste-Julien, and P. Rodriguez, "A Survey of Self-Supervised and Few-Shot Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Article vol. 45, no. 4, pp. 4071-4089, Apr 1 2023.

[18] Q. Liu *et al.*, "Asynchronous Decentralized Federated Learning for Collaborative Fault Diagnosis of PV Stations," *IEEE Transactions on Network Science and Engineering,* Article vol. 9, no. 3, pp. 1680-1696, May-Jun 2022.

[19] Z. Zhang, X. Xu, W. Gong, Y. Chen, and H. Gao, "Efficient federated convolutional neural network with information fusion for rolling bearing fault diagnosis," *Control Engineering Practice,* Article vol. 116, Nov 2021, Art. no. 104913.

[20] L. Tang, H. Xie, X. Wang, and Z. Bie, "Privacy-preserving knowledge sharing for few-shot building energy prediction: A federated learning

approach," *Applied Energy,* Article vol. 337, May 1 2023, Art. no. 120860.

[21] X. Li, Y. Li, J. Wang, C. Chen, L. Yang, and Z. Zheng, "Decentralized federated meta-learning framework for few-shot multitask learning," *International Journal of Intelligent Systems,* Article vol. 37, no. 11, pp. 8490-8522, Nov 2022.

[22] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare," *IEEE Intelligent Systems,* Article vol. 35, no. 4, pp. 83-93, Jul-Aug 2020.

[23] W. Mao, H. Shi, G. Wang, and X. Liang, "Unsupervised Deep Multitask Anomaly Detection With Robust Alarm Strategy for Online Evaluation of Bearing Early Fault Occurrence," *IEEE Transactions on Instrumentation and Measurement,* Article vol. 71, 2022 2022, Art. no. 3520713.

[24] S. Scardapane and D. Wang, "Randomness in neural networks: an overview," *Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery,* Article vol. 7, no. 2, Mar-Apr 2017, Art. no. e1200.

[25] M. E. Tagluk, M. S. Mamis, M. Arkan, O. F. Ertugrul, and Ieee, "Detecting Fault Type and Fault Location in Power Transmission Lines by Extreme Learning Machines," in *23nd Signal Processing and Communications Applications Conference (SIU)*, Inonu Univ, Malatya, TURKEY, 2015, pp. 1090-1093, 2015.

[26] W. Hao and F. Liu, "Imbalanced Data Fault Diagnosis Based on an Evolutionary Online Sequential Extreme Learning Machine," *Symmetry-Basel,* Article vol. 12, no. 8, Aug 2020, Art. no. 1204.

[27] Z. Yang, S. Al-Dahidi, P. Baraldi, E. Zio, and L. Montelatici, "A Novel Concept Drift Detection Method for Incremental Learning in Nonstationary Environments," *IEEE Transactions on Neural Networks and Learning Systems,* Article vol. 31, no. 1, pp. 309-320, Jan 2020.

[28] M. Sahani and P. K. Dash, "Fault location estimation for series-compensated double-circuit transmission line using EWT and weighted RVFLN," *Engineering Applications of Artificial Intelligence,* Article vol. 88, Feb 2020, Art. no. 103336.

[29] Y. Xia, Y. Xu, B. Gou, and Q. Deng, "A Learning-Based Method for Speed Sensor Fault Diagnosis of Induction Motor Drive Systems," *IEEE Transactions on Instrumentation and Measurement,* Article vol. 71, 2022 2022, Art. no. 3504410.

[30] H. Zhao, J. Zheng, J. Xu, and W. Deng, "Fault Diagnosis Method Based on Principal Component Analysis and Broad Learning System," *IEEE Access,* Article vol. 7, pp. 99263-99272, 2019 2019.

[31] Z. Y. Liu, J. F. Zhang, X. He, Q. H. Zhang, G. X. Sun, and D. H. Zhou, "Fault Diagnosis of Rotating Machinery With Limited Expert Interaction: A Multicriteria Active Learning Approach Based on Broad Learning System," *IEEE Transactions on Control Systems Technology*.

[32] D. Wang and M. Li, "Stochastic Configuration Networks: Fundamentals and Algorithms," *IEEE Transactions on Cybernetics,* Article vol. 47, no. 10, pp. 3466-3479, Oct 2017.

[33] C. Ding and A. Yan, "Fault Detection in the MSW Incineration Process Using Stochastic Configuration Networks and Case-Based Reasoning," *Sensors,* Article vol. 21, no. 21, Nov 2021, Art. no. 7356.

[34] J. Li, G. Zou, W. Wang, N. Shao, B. Han, and L. Wei, "Low-voltage series arc fault detection based on ECMC and VB-SCN," *Electric Power Systems Research,* Article vol. 218, May 2023, Art. no. 109222.