Edge AI Enabled PMU Event Classification with Knowledge Distillation and Federated Learning

Zhirui Tian, *Student Member, IEEE*, Hamed Mohsenian-Rad, *Fellow, IEEE*, and Chenye Wu, *Senior Member, IEEE*

***Abstract*—Phasor Measurement Units (PMUs) enable analysis and classification of events and abnormalities in modern power systems. However, in large and geographically dispersed networks, centralizing PMU data streams for sub-second decisions is hindered by cross-regional data-sharing restrictions, multi-layer aggregation, as well as communication delays and jitter. Purely local training at the network edge on the PMU devices themselves can avoid these bottlenecks but lacks exposure to diverse operating conditions, leading to suboptimal models. This tension calls for methods that *retain data locality and privacy* while *sharing knowledge efficiently*. To this end, this paper proposes an edge AI framework that integrates *knowledge distillation* and *federated learning*, driven by the core requirement of different data-sharing constraints. First, we assume a trusted data center, where we develop a large language model-based teacher model that is fine-tuned on global data and efficiently transfers knowledge into student models trained locally via knowledge distillation. Next, we relax such an assumption and design a fair federated learning framework that exchanges gradient information among edge models while enforcing Pareto-optimal fairness to ensure stable and balanced performance. We also customize a lightweight neural network tailored for on-device deployment. With fewer than 5,000 parameters, the model adopts a compact end-to-end architecture to effectively characterize temporal dynamics and frequency-domain information in PMU time series, thereby achieving both high efficiency and strong representational capability. Experiments conducted on real-world PMU measurements demonstrate the superior performance of our framework, and ablation studies further confirm the unique contributions of each design component.**

***Index Terms*—Phasor Measurement Unit (PMU), Event Classification, Edge AI, Knowledge Distillation, Federated Learning**

I. Introduction

PHASOR Measurement Units (PMUs) are high-precision measurement devices that provide time-synchronized phasor data acquisition for electrical power systems [1], [2]. By providing time-synchronized voltage and current phasor measurements (magnitude and phase angle) from several locations on power systems, PMUs enable direct comparison of electrical quantities across geographically dispersed substations. Consequently, PMU data play a crucial role in system-wide monitoring, analysis of dynamic events, and the identification of anomalies within the power network, thereby supporting the reliable and stable operation of the electric grid [3], [4].

However, power systems [5] are large-scale networks that traverse multiple utilities and jurisdictions, or even span multiple countries. This complexity introduces several practical challenges, including difficulties in sharing proprietary data among different entities, communication delays, and the computational burden associated with collecting and aggregating all PMU data streams into a single trusted location.

Although it is currently common to deploy *cloud servers* to aggregate PMU data in one location for accurate analysis and event classification, one cannot use such an approach in challenging wide-area or cross-national power systems, due to legal data sharing obstacles, non-negligible network delays, jitter, and the layered phasor-data aggregation pipeline.

To systematically address these challenges, we introduce a *hybrid* paradigm that adopts *different data-sharing constraints* as the driving factor. It flexibly integrates knowledge distillation and federated learning to enable efficient Edge AI training and deployment at PMU devices themselves. Although the model relies solely on local edge data, the proposed mechanism can effectively *perceive and leverage external knowledge*, thereby improving the model performance. This approach achieves a *balanced trade-off* among performance, communication overhead, and privacy protection, offering a practical pathway for online state awareness in large-scale power systems.

A. Literature Review

PMU measurements are a specialized form of distributed multivariate time-series data. Recent advances in general time-series learning, such as multi-scale modeling (Time Mixer [6]) and patch-based representations (PatchTST [7]), have shown strong capability in capturing long-range temporal dependencies and improving representation efficiency. Likewise, federated learning has become an important paradigm for distributed time-series analytics, with recent studies exploring fairness-aware optimization [8] and personalization under heterogeneous local data. However, unlike standard benchmark time series, PMU data exhibit strong spatial coupling, strict real-time requirements, limited edge-side resources, and privacy/communication constraints across geographically distributed utilities. Consequently, directly applying general time-series techniques is often insufficient, necessitating a focused examination of literature tailored to PMU-specific event and fault classification.

Received 9 November 2025; revised 13 April 2026; accepted 17 June 2026. This work was supported by the National Natural Science Foundation of China under Grant 72271213, the Shenzhen Science and Technology Program under Grant RCYX20221008092927070, the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515240024, and by the University of California, Riverside's Winston Chung Endowed Chair Professorship on Energy Innovation. *(Corresponding author: Chenye Wu.)*

Z. Tian and C. Wu are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China.

H. Mohsenian-Rad is with the Department of Electrical and Computer Engineering, University of California, Riverside, CA, USA.

TABLE I
SUMMARY OF THE RELATED WORKS

References		Edge AI	Privacy Constraints	Lightweight Model	Knowledge Distillation	Large Language Model	Frequency Learning
Signal Processing-Based	[9]–[12]	✗	✗	✓	✗	✗	✓
AI-Based	[13], [14]	✗	✗	✓	✗	✗	✗
	[15]	✗	✗	✓	✗	✗	✓
	[16]–[18]	✗	✗	✗	✗	✗	✗
	[19]–[22]	✗	✗	✓	✗	✗	✓
	[23]	✓	✗	✗	✗	✗	✓
	Our Work	✓	✓	✓	✓	✓	✓

Techniques for event classification based on PMU measurements can be broadly categorized into two main approaches: signal processing-based methods and machine learning-based methods. Traditional signal processing techniques contain wavelet transform [9], locally linear embedding [10], swinging door trending [11], and Fourier transform [12]. For example, Kim *et al.* proposed a wavelet-based algorithm to detect and classify nonstationary PMU measurements, achieving satisfactory performance in real-world cases [9]. Jiang *et al.* developed an enhanced Fourier Transform-based method that effectively suppresses measurement noise, enabling precise extraction of fundamental frequency components for reliable event detection and location identification [12]. While effective in many scenarios, these methods often rely on extensive manual feature engineering and parameter tuning, resulting in performance degradation when they are mismatched with the grid topology or operating conditions. Signal processing methods also often neglect spatial correlations unless explicitly modeled, thereby constraining scalability.

With the rapid development of AI technology, machine learning methods have been widely applied in event classification in power systems. Such technologies include Support Vector Machine (SVM) [13], Extreme Learning Machine (ELM) [15], and Decision Tree (DT) [14], among many others. More recently, deep learning models have also been widely used in event classification, including methods based on Multilayer Perceptron (MLP) [16], Long Short-Term Memory (LSTM) [17], and Convolutional Neural Network (CNN) [18], among others. For instance, Yuan *et al.* introduced a spatial pyramid pooling–enhanced CNN to achieve efficient and accurate PMU event identification, demonstrating high recognition accuracy and strong robustness in real-world conditions [19]. Lal *et al.* proposed an ensemble learning framework combining an auto-encoder with a Gated Recurrent Unit (GRU), and adopted a Grey Wolf Optimizer for deep learning parameter tuning, which maintained effective PMU event classification performance even under limited data conditions [24]. Wang *et al.* transformed conventional PMU time-series sequences into image representations and leveraged a multi-layer CNN to extract discriminative features, achieving superior performance compared with traditional frequency-based methods, with an improvement of more than 48%, while also enabling faster decision-making [20].

Although these AI-based methods have substantially improved classification accuracy, many of them emphasize predictive performance without explicitly accounting for deployment efficiency, communication burden, or edge-side resource constraints. With the continued expansion of PMU deployment, the need for lightweight PMU-based classification models has attracted increasing attention in the literature. Rafferty *et al.* highlighted that the rapid growth of synchrophasor data makes manual analysis increasingly impractical for system operators, and accordingly proposed a dimensionality reduction strategy to improve computational efficiency for real-time decision support [21]. Liu *et al.* developed a Nyström principal component analysis-based method to accelerate PMU event classification, thereby improving both computational efficiency and classification accuracy [22]. Yang *et al.* proposed a PMU data compression framework to mitigate the communication burden imposed by the frequent transmission of measurements from geographically distributed PMUs [23].

Despite the above advancements in event classification, a significant gap remains in this field. In a large network, due to data-sharing restrictions across regions, as well as data communications challenges, it may *not* be realistic to assume that all PMU measurements can be aggregated from all locations in real-time to conduct accurate event classification. To address this shortcoming, one may attempt to train a model solely on the local data from a single PMU. However, in practice, phasor changes can be subtle when the event occurs far from that PMU, leading to inaccurate classifications.

Against this background, this paper addresses the following question: *Can we design an AI model at the edge of the sensor network that captures cross-location dependencies from local PMU data, while respecting data-sharing restrictions on proprietary measurements and maintaining high predictive accuracy?* We demonstrate that the answer to the above question is affirmative. This is achieved through several key technical contributions, as detailed in the following subsection. **Table I** provides a comparison between our work and existing literature.

B. Our Contributions

This paper presents a novel edge AI framework for implementation at individual devices, such as on PMUs themselves, to achieve high-accuracy locally deployed event classification models under varying data availability conditions. The framework comprises two complementary information sharing mechanisms: one leverages a data-sharing knowledge distillation framework to transfer knowledge from the global model to edge AI, and the other employs a federated learning paradigm to enable efficient collaboration across distributed

data sources. The contribution lies not merely in combining knowledge distillation and federated learning, but in providing a coordinated solution for the same PMU edge-classification task under different data availability assumptions. To the best of our knowledge, *we are the first to* combine knowledge distillation with federated learning for advancing edge AI for event classification in power systems, and the techniques we use have also advanced significantly compared to previous methods. The ***major contributions*** of this paper can be summarized as follows:

- *Flexible Information Sharing in Different Data Availability Scenarios*: To enable cross-location information exchange based on edge deployment, we propose two approaches under different data-sharing regimes within a unified framework: (i) when raw data sharing of the previous period *is* allowed, we employ knowledge distillation to transfer knowledge from a teacher model trained on global data to a student model trained on local data; (ii) when privacy constraints prohibit raw-data sharing across clients, we use the same lightweight edge model within a fairness-aware federated learning framework, enabling collaborative edge training through model-update sharing.
- *Customized Lightweight Deep Learning Suitable for Edge Deployment*: An innovative lightweight architecture is proposed that seamlessly integrates Patch with Frequency Embedding to capture salient *time-domain* features in PMU measurements, while mitigating noise. Building on this, a learnable filter is achieved that enables end-to-end *frequency-domain* modeling. With fewer than 5,000 parameters, it maintains excellent performance under extremely low computational overhead and rapid inference, thereby alleviating resource constraints in edge deployment.
- *Large Language Model (LLM) to Effectively Enhance the Teacher Model's Performance*: To better model system-wide PMU data and provide stronger supervision for local edge models, a pretrained GPT-2 module is incorporated into the teacher model together with task-adaptive embedding heads and partial fine-tuning. In this way, richer sequence representation capability can be exploited at the teacher side and transferred to lightweight student models through distillation, leading to improved alignment with the downstream event-classification task.
- *Pareto Frontier to Boost Fairness in Federated Learning*: To reduce performance disparity among geographically distributed PMU clients, a Pareto-based multi-objective optimization strategy is introduced into federated training. Compared with conventional gradient averaging (i.e., FedAvg), this design improves fairness across local models while avoiding direct raw-data sharing and incurring only limited additional communication overhead.
- *Enhanced Practical Value for Power System Monitoring*: Validated using real PMU data, the edge AI framework can capture system-wide patterns into local models, allowing the edge AI on each PMU to attain near-global situational awareness from its own stream, thereby enhancing real-time monitoring and decision-making without the need for centralized data sharing.

The remainder of the paper proceeds as follows. Section II details the data-sharing knowledge distillation pipeline, including the network architecture design and the distillation process. Section III presents the privacy-preserving federated learning framework and the proposed Pareto Frontier-based fairness guidance. Section IV describes the dataset selection, experimental setup, hyperparameters, and evaluation metrics. Section V reports comprehensive comparisons against baselines and ablation studies for each component. Section VI concludes with key findings.

II. Knowledge distillation with trusted data center

Knowledge distillation typically relies on the trusted data center, as the teacher model must be trained on a large-parameter architecture using full-scale global data in order to effectively transfer knowledge to the student model, which is trained solely on local data. Under conditions that permit previous data sharing, knowledge distillation generally achieves superior model performance compared to gradient-sharing approaches such as federated learning.

A. Design Principles of the Student Model

As a model intended for edge deployment, the student model compresses and inherits knowledge from the teacher model. Its architecture must be carefully designed to address two critical aspects required for real-world operation.

1) Keep the Model Parameters as Few as Possible: Due to constrained hardware budgets and ongoing maintenance costs in practice, models with large parameter sizes are generally unsuitable for local deployment. Their substantial memory and computing requirements not only demand expensive edge hardware upgrades but also increase power usage and latency, resulting in prohibitive overall deployment costs.

2) Learn Only from Historical Local Data: Edge deployment inherently limits access to diverse data sources that are typically available in centralized infrastructures such as the cloud or data centers. Under such settings, the student model must be trained solely from historical PMU measurements collected at the local site. To better match practical edge scenarios, the proposed framework does not rely on any manual preprocessing pipeline before inference or training. Specifically, the multivariate PMU features are directly concatenated and fed into the model as the input sequence, without performing handcrafted feature selection, denoising, or other external preprocessing operations. Instead, all data processing is embedded into the learnable components of the network and carried out in an end-to-end manner.

B. Lightweight Student Model Architecture

In accordance with the above principles, we designed a lightweight neural network for the student model. Although the total number of parameters is fewer than 5,000, the model achieves highly effective representational capacity through an innovative embedding strategy and a joint time-frequency domain learning mechanism, as shown in **Figure 1**.

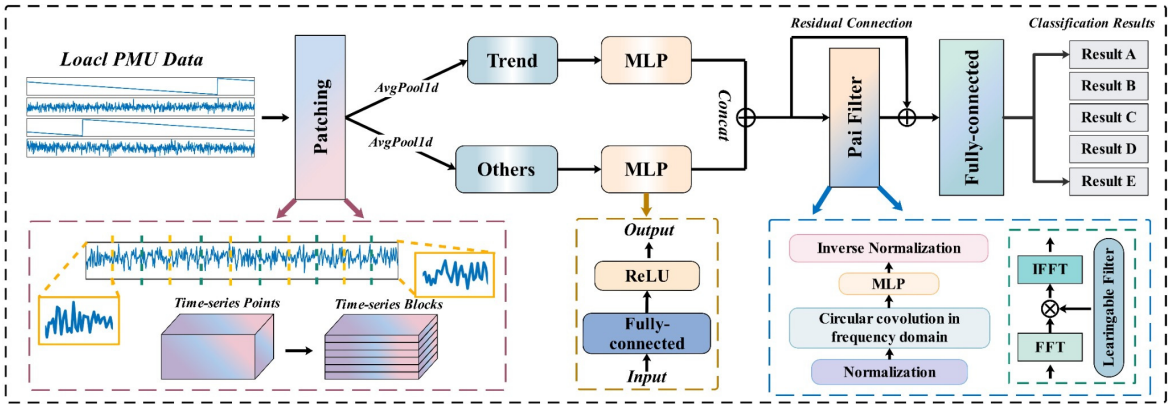

Fig. 1. The proposed architecture of the student model at each PMU to build an Edge AI with low parameter size and high representation performance.

The following provides a detailed description of the student model architecture, which is specifically tailored based on the common characteristics of real-world PMU data.

1) Patch Embedding: Upon receiving the historical time series, the model first performs patch-based processing. Unlike conventional point-by-point learning [25], this approach reduces computational overhead and mitigates measurement noise in PMU data. By aggregating time points into temporal patches, patch-wise learning enhances data efficiency, which is particularly critical for models with limited parameter capacity. Let the original time series be defined as:

$$X = [x_1, x_2, \ldots, x_T], \quad x_t \in \mathbb{R}^d \tag{1}$$

where T is the total number of look-back window, and d is the feature dimension at each time step.

We divide the sequence into N non-overlapping patches, each of length L, such that:

$$N = \left\lfloor \frac{T}{L} \right\rfloor \tag{2}$$

The i-th patch is defined as:

$$P_i = [x_{(i-1)L+1}, x_{(i-1)L+2}, \ldots, x_{iL}], \quad \forall 1 \le i \le N \tag{3}$$

The resulting patch sequence is:

$$\mathcal{P} = [P_1, P_2, \ldots, P_N], \quad \mathcal{P} \in \mathbb{R}^{N \times L \times d} \tag{4}$$

After applying the patch operation, the change of input matrix, [batch size (BS), look-back window (LB), feature dimension (FD)], can be expressed as:

$$[BS, LB, FD] \overset{\text{Unfold+Reshape}}{\rightarrow} [BS \times FD, PN, PL], \tag{5}$$

where PN is the patch number and PL is the patch size. This transformation—from using the *look-back window* (i.e., the number of time points) to using the *number of patches* (i.e., the number of time blocks)—significantly reduces the model's learning burden along the temporal dimension, thereby enhancing both training efficiency and stability.

2) Frequency Embedding: After applying patch embedding, we further enhance feature extraction and improve learning efficiency by introducing a frequency-based embedding. This method decomposes the input data to capture underlying trend information. Because PMU data typically exhibit pronounced trend variations during events, this design enables the model to effectively capture such dynamic characteristics. We employ a channel-independent average-pooling strategy, which allows for efficient summarization of frequency components without introducing inter-channel dependencies.

Let $p = \lfloor k/2 \rfloor$. For any sample, channel $c \in [1, C]$, and time index $t \in [1, T]$, define the local window as:

$$W_t = \{\, j \in \{1, \ldots, T\} \mid |j - t| \le p \,\} \cap \{1, \ldots, T\}. \tag{6}$$

The number of valid samples is:

$$s_t = |W_t| = \sum\nolimits_{j=1}^{T} \mathbf{1}(|j - t| \le p). \tag{7}$$

Then the "trend" sequence (moving average without counting padded zeros) is:

$$\text{trend}_{c,t} = \frac{1}{s_t} \sum_{j \in W_t} x_{c,j} = \frac{\sum_{j=1}^{T} x_{c,j}\, \mathbf{1}(|j - t| \le p)}{\sum_{j=1}^{T} \mathbf{1}(|j - t| \le p)}. \tag{8}$$

The residual (detrended part) is:

$$\text{others}_{c,t} = x_{c,t} - \text{trend}_{c,t}. \tag{9}$$

After applying patch embedding and frequency embedding, two independent multilayer perceptrons (MLPs [26]) are employed to project the extracted trend component and the residual component into higher-dimensional feature spaces. Each MLP consists of a fully connected layer followed by a nonlinear activation function (ReLU). The resulting representations are then concatenated along the feature dimension, enabling effective information fusion. Given an input vector $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$, the output of a single-layer MLP block can be written as:

$$\mathbf{h} = \text{ReLU}(W\mathbf{x} + \mathbf{b}), \tag{10}$$

where $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the weight matrix and $\mathbf{b} \in \mathbb{R}^{d_{\text{out}}}$ is the bias term.

For each element $z_i = (W\mathbf{x} + \mathbf{b})_i$, the ReLU activation is:

$$\text{ReLU}(z_i) = \max(0, z_i). \tag{11}$$

3) Learnable Filter: Given the relatively small number of parameters, learning solely in the time domain may still be insufficient to effectively capture all discriminative features. To address this limitation, we introduce the **Pai Filter** [27], a learnable filter with high performance and low parameter complexity. Synchro-phasors inherently exhibit periodicity, phase consistency, and distinctive spectral patterns. Leveraging these characteristics through joint time–frequency representations can complement time-domain learning and enhance the ability to capture more discriminative information. The Pai Filter is composed of four key components: (i) instance normalization to mitigate distributional shifts in time series data, (ii) a learnable filtering module, (iii) an MLP for further representation learning, and (iv) a final de-normalization step to restore the original scale.

4) Instance Normalization: Time series often exhibit non-stationarity, which causes distribution shifts and performance degradation in forecasting tasks. To mitigate this issue, we adopt instance normalization on input $\mathbf{X}$, which dynamically normalizes each instance to eliminate sample-wise mean and variance variations, thus improving robustness to non-stationary fluctuations, which is denoted as Norm:

$$\mathrm{Norm}(\mathbf{X}) = \left[\frac{X_i^{1:L} - \mathrm{Mean}_L(X_i^{1:L})}{\mathrm{Std}_L(X_i^{1:L})}\right]_{i=1}^{N}, \tag{12}$$

where Mean_L and Std_L denote the mean and standard deviation along the temporal dimension.

The corresponding inverse operation is:

$$\begin{aligned}\mathrm{InverseNorm}(\mathbf{P}) = \big[P_i^{L+1:L+\tau} \times \mathrm{Std}_L(X_i^{1:L}) \\ +\mathrm{Mean}_L(X_i^{1:L})\big]_{i=1}^{N},\end{aligned} \tag{13}$$

with

$$\mathbf{P} = \left[P_1^{L+1:L+\tau},\ \ldots,\ P_N^{L+1:L+\tau}\right] \in \mathbb{R}^{N\times\tau} \tag{14}$$

representing the predicted values.

5) Frequency Filter: After normalization, the data are transformed into the frequency domain using the Fast Fourier transform (FFT) and learned by instantiating a frequency filter. This filter is parameterized by randomly initialized learnable weights that are multiplied by the transformed data. It transforms the multivariate PMU sequence into the frequency domain, adaptively reshapes its spectral components through learnable weights, and then maps the filtered representation back to the time domain, so that informative oscillatory/trend-related components can be preserved while noisy or less relevant fluctuations are attenuated. For multivariate time series, prior studies have shown that channel-independence in modeling is generally more effective than channel-mixing. Following this principle, we adopt a channel-independent design for the frequency filter, in which parameters are shared across channels.

Given the time series input $\mathbf{Z} \in \mathbb{R}^{N\times L}$ and the filter $\mathcal{H}_\phi$, Pai Filter is applied as:

$$\begin{aligned}&\mathcal{Z} = \mathcal{F}(\mathbf{Z}),\\ &\mathcal{S} = \mathcal{Z} \odot_L \mathcal{H}_\phi, \quad \mathcal{H}_\phi \in \{\mathcal{H}_\phi^{(Uni)},\ \mathcal{H}_\phi^{(Ind)}\},\\ &\mathbf{S} = \mathcal{F}^{-1}(\mathcal{S}),\end{aligned} \tag{15}$$

where $\mathcal{F}$ is the Fourier transform (FFT), $\mathcal{F}^{-1}$ is the inverse Fourier transform (iFFT), and $\odot_L$ denotes the element-wise product along the L dimension. Here $\mathcal{H}_\phi^{(Uni)} \in \mathbb{C}^{1\times L}$ is the learnable frequency filter, $\mathcal{H}_\phi^{(Ind)} \in \mathbb{C}^{N\times L}$ is the individual plain shaping filter, and $\mathbf{S} \in \mathbb{R}^{N\times L}$ is the output of the filter.

6) Residual Connection: We adopt a residual connection to ensure that the frequency learning module contributes positively to the optimization process:

$$\mathbf{Y} = \mathbf{X} + \mathcal{O}_{\text{Pai Filter}}(\mathbf{X}), \tag{16}$$

where $\mathbf{X}$ denotes the input, $\mathcal{O}_{\text{Pai Filter}}(\mathbf{X})$ represents the output of the frequency learning filter, and $\mathbf{Y}$ is the output after applying the residual connection.

After obtaining the feature representations, the final classification output is directly produced through a fully connected layer. Despite its compact size (fewer than 5,000 parameters), the proposed model is capable of efficiently extracting informative features from time series and supporting fast inference, thereby fully achieving the intended design objectives.

C. Design Principles of the Teacher Model

As the core carrier of knowledge, the teacher model serves as a fundamental guarantee for the deployment performance of the student model. In designing the teacher model, this work follows two key principles:

1) Train on Global Data: Without original data sharing restrictions, global PMU data can be sourced from trusted data centers across the power system. Compared to local data, global data offers a more comprehensive view of system characteristics, improving the teacher model's accuracy and adaptability.

2) Employ Large-scale Parameter Models: The primary objective of the teacher model lies in achieving high performance and strong generalization. Large-scale parameter models are capable of capturing richer feature relationships, enabling more robust performance under complex conditions and unseen scenarios.

D. LLM-enhanced Teacher Model Architecture

Based on the aforementioned design principles, the architecture of the teacher model is illustrated in **Figure 2**. The teacher retains the overall framework of the student model, but replaces its learnable filters with a large-scale pretrained LLM, namely GPT-2, whose parameter scale exceeds 100 million. The motivation for introducing a pretrained LLM is not to exploit its text-domain modeling ability itself, but to leverage the more general sequence representation capability developed through large-scale pretraining [28]. Compared with conventional sequence models trained only on the target dataset, a pretrained LLM typically provides stronger long-range dependency modeling, richer contextual representation, and more effective pattern abstraction, which are beneficial for characterizing the temporal evolution of PMU time-series data. In addition, such a pretrained representation prior may offer improved robustness and generalization, particularly under limited labeled data and complex event patterns.

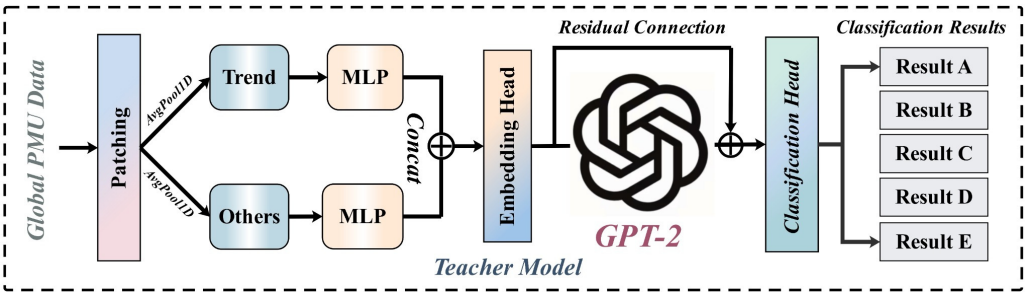

Fig. 2. The proposed architecture of the teacher model based on LLM.

Among available pretrained LLM models, GPT-2 was adopted in this work as a representative and practically suitable choice. Specifically, GPT-2 provides a mature and stable decoder-only architecture with publicly available implementations, well-established pretrained weights, and a moderate parameter scale that is sufficient to serve as a high-capacity teacher model while remaining computationally manageable for our distillation setting. Therefore, GPT-2 offers a suitable balance among representation capability, implementation stability, and computational cost for the present study. In addition to the Patch Embedding and Frequency Embedding modules, which are the same as those in the student model, we introduce an embedding head and a classification head to interface with GPT-2. The embedding head is designed to further project the original encoded representations into a higher-dimensional space, thereby aligning them with the embedding dimension of GPT-2 [29] (768). The matrix change of the embedding layer can be formally expressed as:

$$[BS \times FD, PN, FE] \overset{\text{Embedding Head}}{\longrightarrow} [BS \times FD, PN, 768],$$

where FE is the dimension after frequency embedding. The classification head compresses the high-dimensional hidden states of GPT-2 (768) into a low-dimensional task label space, producing the final classification results. Both the embedding and classification heads are implemented as single fully connected layers that perform linear projections.

$$[BS \times FD, PN, 768] \overset{\text{Classification Head}}{\longrightarrow} [BS \times FD, PN, DC],$$

where DC denotes the dimension after compression. To further improve the adaptability of the LLM to downstream tasks, we unfreeze the last two layers of the pretrained LLM while keeping all other parameters fixed to strike a balance between performance and efficiency. This design is motivated by the common observation [30] that lower layers of pretrained Transformer-based models tend to capture more general and transferable sequence representations, whereas higher layers are more task-specific and therefore more suitable for downstream adaptation. Under this view, fine-tuning only the top layers allows the model to preserve most of the pretrained representation capability while efficiently adapting the high-level features to the target PMU event-classification task. In this way, the proposed strategy can reduce training cost and mitigate overfitting risk, while still maintaining strong task-specific modeling ability, which can be expressed as follows:

$$\theta_{t+1}^{(k)} = \begin{cases} \theta_t^{(k)} - \eta \nabla_{\theta^{(k)}} \mathcal{L}, & k = L-1, L, \\ \theta_t^{(k)}, & k = 1, \ldots, L-2. \end{cases} \tag{17}$$

where $\theta^{(k)}$ is the parameter of the k-th layer, t is the training step, $\theta_t^{(k)}$ and $\theta_{t+1}^{(k)}$ denote the parameter values at step t and $t+1$, respectively, η is the learning rate, $\mathcal{L}$ is the loss function, and $\nabla_{\theta^{(k)}} \mathcal{L}$ is the gradient of the loss with respect to $\theta^{(k)}$.

E. Knowledge Distillation Training Pipeline

In the design of the distillation process, we follow the standard logit-based knowledge distillation paradigm, in which the student model learns from the softened probability distribution produced by the teacher model. This design is particularly beneficial in our setting because the student model, as an edge AI model, can only access local PMU data and therefore lacks direct exposure to system-wide measurements. As a result, a locally trained student may have difficulty capturing the broader discrimination patterns that are available to a teacher trained on global data, especially when local observations are ambiguous or only weakly informative. By learning from the teacher's soft outputs, the student is guided not only by the target label itself, but also by the relative relationships among event classes reflected in the teacher's logits. Since these soft targets are generated by a teacher trained on global PMU data, they implicitly convey a more global decision tendency and richer inter-class structure learned from system-wide observations. In this way, the lightweight student model, although restricted to local input, can partially inherit the teacher's global-view classification behavior and thus make more informed predictions under purely local measurements, as visualized in **Figure 3**. Accordingly, the overall loss function is formulated as a weighted combination of the *hard target loss* and the *soft target loss*.

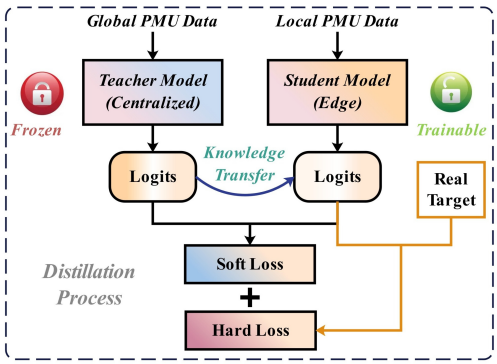

Fig. 3. Transfer global knowledge from teacher to student.

Algorithm 1 Knowledge Distillation with Trusted Data Center

Input: Dataset $\mathcal{D} = \{(X, y)\}$ with $X = [X^{(1)}, \ldots, X^{(M)}]$; learning rates η_t, η_s; epochs E_t, E_s; temperature $T > 1$; balance $\alpha \in [0, 1]$; PMU index policy $k \in \mathcal{P}$.

Stage A: Teacher Pre-training (all PMUs)
for epoch $= 1, \ldots, E_t$ **do**
  for all $(X, y) \in \mathcal{D}$ **do**
    $z_t \leftarrow f_t(X; \theta_t)$, $p_t \leftarrow \mathrm{softmax}(z_t)$
    $\mathcal{L}_t \leftarrow -\sum_{i=1}^{C} \mathbf{1}[y = i] \log(p_{t,i})$
    $\theta_t \leftarrow \theta_t - \eta_t \nabla_{\theta_t} \mathcal{L}_t$
  end for
end for
Freeze $\hat{\theta}_t \leftarrow \theta_t$
Stage B: Distillation (single PMU input)
for epoch $= 1, \ldots, E_s$ **do**
  for all $(X, y) \in \mathcal{D}$ **do**
    Choose PMU index $k \in \mathcal{P}$
    Compute hard loss $\mathcal{L}_{\text{hard}}$ (Eq. 18)
    Compute soft loss $\mathcal{L}_{\text{soft}}$ (Eq. 20)
    $\mathcal{L}_{\text{KD}} \leftarrow \alpha \mathcal{L}_{\text{hard}} + (1 - \alpha) T^2 \mathcal{L}_{\text{soft}}$
    $\theta_s \leftarrow \theta_s - \eta_s \nabla_{\theta_s} \mathcal{L}_{\text{KD}}$
  end for
end for
return $\hat{\theta}_s \leftarrow \theta_s$

The hard target loss preserves the discriminative ability of the student model under the direct supervision of ground-truth labels. It is typically defined as the standard cross-entropy loss:

$$\mathcal{L}_{\text{hard}} = \mathcal{H}(y, p_s) = -\sum_{i=1}^{C} y_i \log\left(p_{s,i}\right), \tag{18}$$

where C denotes the number of classes, $y \in \{0, 1\}^C$ is the one-hot encoded ground-truth label, and $p_s = \mathrm{softmax}(z_s)$ is the predicted probability distribution with logits z_s.

The soft target loss guides the student model to approximate the softened probability distribution of the teacher model. To this end, a temperature parameter $T > 1$ is introduced to smooth the logits, thereby amplifying the relative relationships among non-maximum classes. The softened distributions of the teacher and student are defined as:

$$p_t^{(T)} = \mathrm{softmax}\left(\frac{z_t}{T}\right), \quad p_s^{(T)} = \mathrm{softmax}\left(\frac{z_s}{T}\right), \tag{19}$$

where z_t and z_s are the logits of the teacher and student models, respectively. The soft loss is typically measured by the Kullback–Leibler divergence (KL divergence):

$$\mathcal{L}_{\text{soft}} = \mathrm{KL}\big(p_t^{(T)} \,\|\, p_s^{(T)}\big) = \sum_{i=1}^{C} p_{t,i}^{(T)} \log \frac{p_{t,i}^{(T)}}{p_{s,i}^{(T)}}. \tag{20}$$

By combining the above two components, the overall KD loss function can be formulated as:

$$\mathcal{L}_{\text{KD}} = \alpha\, \mathcal{L}_{\text{hard}} + (1 - \alpha)\, T^2\, \mathcal{L}_{\text{soft}}, \tag{21}$$

where $\alpha \in [0, 1]$ is a balancing hyperparameter, and the factor of T^2 is commonly introduced into the loss to maintain gradient consistency. The proposed edge AI knowledge distillation mechanism can be expressed by the following pseudo code (**Algorithm 1**).

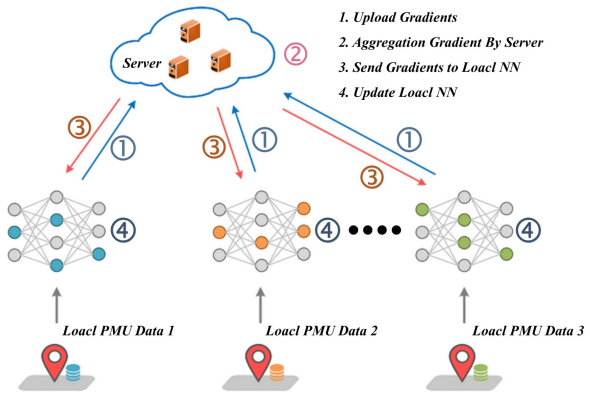

Fig. 4. Basic process of federated learning on event classification.

III. Federated Learning without Original Data-sharing

While knowledge distillation has been widely recognized for its ability to transfer global knowledge and achieve strong performance in edge deployment, it typically relies on global datasets (trusted data center) to train the teacher model, which is often infeasible when the sharing of previous data is not allowed. To address this issue, we introduce federated learning (FL) [31], which replaces data sharing with gradient sharing. Without exposing raw data, FL enables collaborative knowledge transfer across distributed nodes, thereby ensuring strong performance in edge deployment under privacy constraints.

The basic workflow of federated learning for PMU event classification is as follows: The central server initializes and distributes a global model to all participating nodes. Each node trains the model locally using its own PMU data and computes parameter updates. Instead of sharing the raw data, nodes send the gradient updates to the server. The server then aggregates these gradients to produce a new global model, which is redistributed to the nodes. Through iterative rounds, the global model gradually converges, achieving accurate event classification without original data-sharing, which is shown in **Figure 4**. In this framework, the locally trained model is the same as the student model (**Figure 1**), meeting the core requirements of a small parameter size and fast inference.

Standard federated learning, such as FedAvg, aggregates local model parameters or updates as follows:

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k, \tag{22}$$

where $\frac{n_k}{n}$ denotes the weight of client k in the averaging. However, in FedAvg, each client computes gradient updates based on its local data distribution. Under non-*i.i.d.* settings, these gradients often point toward different local optima [32]. When the server averages them, the global update direction may diverge from some clients' objectives, forcing those clients away from their optimal solutions. This misalignment leads to slower convergence and uneven performance across clients, thus causing unfairness. To ensure that each local model achieves good performance, we introduce FedMDFG to address this issue through multi-objective optimization.

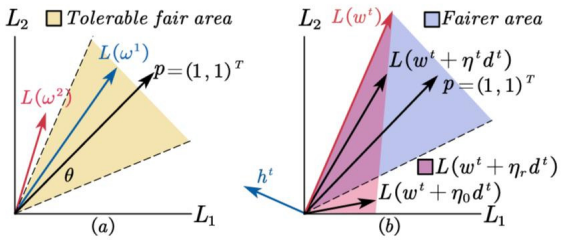

Fig. 5. Fairness Measurement based on inverse cosine value.

1) Fairness Measurement: The global model parameter is denoted as ω, and the local objective for client i is $L_i(\omega)$. Collecting all client objectives:

$$L(\omega) = \big(L_1(\omega), \ldots, L_m(\omega)\big) \in \mathbb{R}^m, \tag{23}$$

where ω is the global model parameter, $L_i(\omega)$ is the loss of client i, m is the number of clients, and $L(\omega)$ means loss vector across all clients.

The fairness indicator is defined as the angle with respect to:

$$\phi\big(L(\omega), \mathbf{1}\big) \;=\; \arccos\!\left(\frac{L(\omega)^\top \mathbf{1}}{\|L(\omega)\|\,\|\mathbf{1}\|}\right), \tag{24}$$

where $\mathbf{1}$ is an all-ones vector, serving as the fairness reference, and $\phi(\cdot)$ is the angle function that measures fairness. $\|\cdot\|$ is the euclidean norm. We fix a tolerable-fair threshold $\theta \in (0, \pi/2]$. The model is *tolerable-fair* if $\phi(L(\omega), \mathbf{1}) \le \theta$, and under the condition that the above situation is not met, it will trigger the fairness enhancement mechanism. **Figure 5** visualizes the fairness measurement.

2) Pareto-Based Multi-Objective Formulation: Under distributed and heterogeneous local data, federated training can be formulated as the following multi-objective optimization problem:

$$\min_{\omega} \; \big(L_1(\omega), \ldots, L_m(\omega)\big). \quad (MOP) \tag{25}$$

Since different client objectives may conflict with each other, the goal is to seek a Pareto-optimal solution rather than to minimize all objectives through a single scalar surrogate. Specifically, a point ω^* is Pareto optimal if there does not exist another ω such that:

$$L_i(\omega) \le L_i(\omega^*), \quad \forall i = 1, \ldots, m, \tag{26}$$

and

$$L_j(\omega) < L_j(\omega^*), \quad \text{for at least one } j. \tag{27}$$

The set of all such non-dominated solutions forms the Pareto frontier.

Accordingly, a first-order Pareto-stationary point satisfies:

$$\sum\nolimits_{i=1}^{m} \xi_i \nabla L_i(\omega^*) = \mathbf{0}, \qquad \xi_i \ge 0, \quad \sum\nolimits_{i=1}^{m} \xi_i = 1, \tag{28}$$

which implies that there exists no common descent direction $d \in \mathbb{R}^n$ such that:

$$\big(\nabla L_i(\omega^*)\big)^\top d < 0, \qquad \forall i = 1, \ldots, m. \tag{29}$$

Therefore, the global update should be determined by jointly considering all client objectives, rather than by directly averaging local gradients.

3) Fairness-Enhanced Direction: When the model is not tolerable-fair, a fairness guidance direction is introduced as:

$$h_t = \mathrm{normalize}\!\left(\frac{\mathbf{1}^\top L(\omega_t)}{\|L(\omega_t)\|^2} L(\omega_t) - \mathbf{1}\right), \tag{30}$$

where $\mathrm{normalize}(\cdot)$ denotes vector normalization. Based on this guidance, the original multi-objective problem is extended to:

$$\min_{\omega} \; \Big(L_1(\omega), \ldots, L_m(\omega),\, L(\omega)^\top h_t\Big), \quad (MOP + Fair) \tag{31}$$

where the first m objectives correspond to the client losses and the additional term $L(\omega)^\top h_t$ serves as a fairness-driven objective. To ensure that the fairness objective is also improved, the descent direction d is required to satisfy:

$$\big(\nabla L(\omega_t) h_t\big)^\top d \le \alpha_t, \tag{32}$$

where α_t is a tolerance parameter.

Based on the KKT conditions, the fairness-enhanced descent direction is obtained as:

$$d^t = -Q\lambda, \tag{33}$$

where λ is the optimal solution of the following dual quadratic program:

$$\begin{aligned} \max_{\lambda} \quad & -\frac{1}{2}\lambda^\top (Q^\top Q)\lambda \\ \text{s.t.} \quad & \sum\nolimits_{i=1}^{|\lambda|} \lambda_i = 1, \\ & \lambda_i \ge 0, \qquad i = 1, 2, \ldots, |\lambda|. \end{aligned} \tag{34}$$

In practice, when the current model is tolerable-fair, we set:

$$Q = \nabla L(\omega^t) \in \mathbb{R}^{n \times m}, \tag{35}$$

so that the descent direction is computed only from the original client objectives. Otherwise, the fairness objective is appended and

$$Q = \mathrm{concat}\Big(\nabla L(\omega^t),\, \nabla L(\omega^t) h^t\Big) \in \mathbb{R}^{n \times (m+1)}, \tag{36}$$

where $\mathrm{concat}(\cdot)$ denotes column-wise concatenation. In this way, the resulting direction jointly accounts for client-wise optimization and fairness enhancement during federated training. This is to yield d^t by solving an $(m+1)$-dimensional problem. Through this guarantee, the fairness-aware FL strategy can effectively improve and balance the classification performance of each local model under raw-data sharing restrictions. This mechanism is also helpful when a small number of local updates are affected by data corruption or gradient inconsistency. Since the global descent direction is obtained by jointly considering all client objectives, rather than directly averaging local gradients, the influence of a few deviated local updates can be partially moderated by the remaining normal clients. Overall, the proposed edge AI training mechanism can be expressed by the following pseudo code (**Algorithm 2**).

Algorithm 2 Training Strategy Under Different Data-Sharing Conditions

Input: Local datasets $\{D_i\}$, raw-data sharing flag $Trusted\ Data - Center$

Output: Final model M

if $Trusted\ Data - Center =$ True **then**
  $D_{all} \leftarrow \bigcup_i D_i$
  Train $Teacher$ on D_{all}
  Distill $Teacher \rightarrow Student_i$ based on D_i
  $M \leftarrow$ Aggregate $\{Student_i\}_{i=1}^{m}$
else
  Initialize M
  Federated training with $\{D_i\}$ and fairness constraint
end if
return M

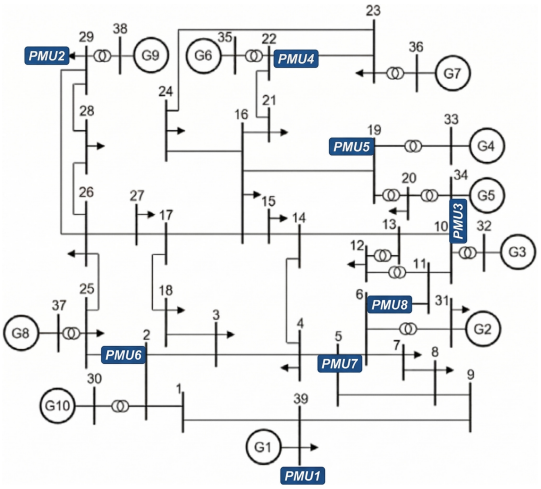

Fig. 6. IEEE 39-Bus system with 8 PMUs.

IV. Experimental Settings

In this section, we introduce the data selection process, the hyperparameters and internal parameters of the proposed models, and the evaluation indices used for comprehensive evaluation.

A. Hardware-in-the-Loop Setting and Task Definition

Instead of relying on simulated datasets, this work employs measurements from a Hardware-in-the-Loop (HIL) Synchrophasor Testbed [33], which integrates several advanced components, including a Real-Time Digital Simulator (RTDS), hardware PMUs, a Real-Time Automation Controller (RTAC), the PingThings cloud platform, and the NS-3 network simulator. In this IEEE 39-Bus power system, 8 PMUs are installed at different locations according to two criteria: (a) buses directly coupled to generator buses, and (b) buses with the strongest connectivity to the rest of the network, following the guideline in [34], which is shown in **Figure 6 & Figure 7**.

To demonstrate the HIL interconnection, the testbed uses a hybrid PMU implementation: two PMUs (Bus 29 and Bus 39) are realized by hardware PMUs, while the remaining PMUs are software-based. The two hardware PMUs are connected to the RTDS at Bus 29 and Bus 39. To accomplish the overall objective of generating high-fidelity, time-synchronized data that mirrors real-world conditions, the testbed physically interconnects multi-vendor hardware devices with real-time simulation tools. For Bus 29, the RTDS sends analog measurements directly via its GTAO output card to an SEL Intelligent Electronic Device (IED). Conversely, for Bus 39, the RTDS streams measurements to another SEL IED via an Ethernet connection using the IEC 61850-9-2 Sample Values (SV) protocol. Both IEDs then convert these measurements to act as hardware PMUs. These PMUs stream the synchrophasor data upstream as C37.118 servers, utilizing Transmission Control Protocol (TCP) as the underlying communication layer protocol. To ensure strict sub-microsecond time synchronization across this pipeline, the setup utilizes the IEC 61850-9-3:2016 Precision Time Protocol (PTP) profile, which provides robustness against network delays and jitter. The streaming data is first received by local Phasor Data Concentrators (PDCs), represented by SEL RTACs. These RTACs function as C37.118 clients to receive the local PMU measurements and simultaneously act as C37.118 servers to forward the data to a software SuperPDC (OpenPDC).

The exchange of information across these nodes is managed by the NS-3 network simulator, which is employed to establish the communication network infrastructure by simulating the Wide Area Network (WAN). NS-3 is used to simulate the WAN and stream data from all local RTAC PDC instances to OpenPDC. Furthermore, this simulated communication layer allows for the realistic emulation of network faults, such as dropped SV packets in the Ethernet communication, effectively replicating variable network conditions and cyber events like data drops. Finally, real-time data streaming and online analysis are achieved through integration with the PingThings cloud platform. The OpenPDC leverages its concentrator output stream capability to transmit all aggregated PMU measurements over the network in a single continuous frame directly to the PingThings cloud database. This cloud platform displays the incoming data streams in real time and provides access via the Berkeley Tree Database (BTrDB) API, allowing for comprehensive online analysis of the streaming PMU measurements.

Each PMU samples three-phase electrical quantities at 30 Hz, corresponding to one measurement every 0.033 s, and records the voltage angle, voltage magnitude, current angle, and current magnitude for each phase. This study focuses on a supervised multi-class classification problem. Specifically, the label space covers five operating conditions and disturbance categories, as summarized in **Table II**. These labels correspond to physically distinct system states and events, whose timely identification is important for real-time situational awareness and subsequent control or protection response. It is worth noting that the class imbalance shown in **Table II** is consistent with the realistic construction of the dataset, rather than resulting from an artificially balanced data-generation process. Since the underlying HIL testbed was designed to emulate practical grid operating scenarios, the relatively small numbers of *Fault* and *Line Outage* samples are consistent with the fact that such events are inherently rare and sporadic in real-world operation, whereas other operating disturbances, such

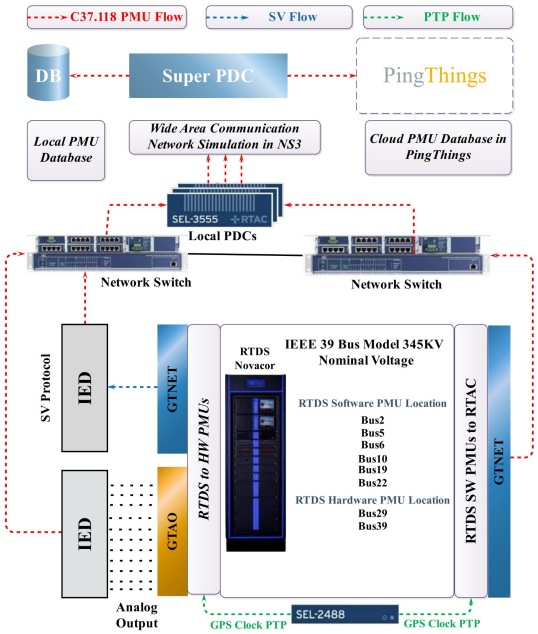

Fig. 7. Hardware-in-the-Loop Connection.

as generation and load changes, can be represented more frequently in scenario-based data generation.

- **Normal Operation:** This state denotes the nominal operating condition of the power system without significant disturbance. It provides the reference baseline for distinguishing abnormal events from normal dynamic variations in PMU measurements.
- **Fault:** This category refers to physical fault disturbances, which typically induce abrupt responses in voltage, current, frequency, and phase angle trajectories. Accurate fault identification is particularly important because such events may pose immediate threats to system security and require rapid protection actions.
- **Line Outage:** This event corresponds to the loss of a transmission line, resulting in network topology changes and power flow redistribution. In contrast to fault events, its impact is primarily associated with structural reconfiguration rather than fault-induced transient behavior.
- **Generation Change/Outage:** This category describes disturbances originating from the generation side, including generation variation or generator outage. Its main significance lies in its direct influence on power balance and frequency dynamics, which distinguishes it from topology-related and fault-driven events.
- **Load Change/Drop:** This state represents disturbances caused by substantial load variation or load shedding. Compared with generation-side events, it arises from the demand side, while still producing observable changes in system operating trajectories.

B. Data Processing and Parameter Design

To enable real-time analysis, we adopt a sliding window of length 300 as the look-back window for learning models. The event corresponding to the first sample within each window is assigned as the label, which ensures that potential events can be identified at their onset rather than after the entire window has passed. In this way, the detection framework maintains real-time responsiveness while still leveraging sufficient historical information for reliable decision-making. During dataset partitioning, we divided the data into training, validation, and test sets with a ratio of 7:1:2, which is sequential and can effectively avoid information leakage. Given the significant imbalance in sample sizes across different classes, we employed a noise-based data augmentation to expand the samples of underrepresented categories, thereby mitigating class imbalance and enhancing the model's generalization capability.

TABLE II
LABEL AND EVENT CORRESPONDENCE

Label	Event	Sample Size
0	Normal Operation	142145
1	Fault	149
2	Line Outage	126
3	Generation Change/Outage	9528
4	Load Change/Drop	9132

To alleviate the severe class imbalance in the training set, we applied a noise-based augmentation strategy only to the underrepresented classes (Label 1 and Label 2). Let $\mathcal{D}_c = \{x_i^{(c)}\}_{i=1}^{N_c}$ denote the set of training samples belonging to class c, where N_c is the number of samples in that class. For each minority-class sample $x_i^{(c)} \in \mathbb{R}^{T\times d}$, an augmented sample $\tilde{x}_i^{(c)}$ is generated as:

$$\tilde{x}_i^{(c)} = x_i^{(c)} + \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, \sigma_c^2 I), \tag{37}$$

where ϵ_i denotes additive Gaussian noise with zero mean and variance σ_c^2, σ_c controls the augmentation strength for class c, and I is the identity matrix. In this way, the generated samples preserve the original class semantics while introducing small local perturbations around the observed trajectories. Such a design is reasonable because PMU measurements are inherently affected by measurement uncertainty and small stochastic fluctuations in practical operation. Therefore, injecting low-amplitude noise can enhance the local diversity of minority-class samples without altering their physical event labels, thereby helping mitigate class imbalance and improve model generalization.

To improve training stability and representation quality, two normalization stages were adopted. First, a standard min–max normalization was applied in preprocessing to rescale each feature into a comparable range, which reduces inter-feature scale discrepancy and stabilizes optimization. Specifically,

$$x^{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{38}$$

Second, Instance Normalization (IN) was embedded in the model to perform sample-wise normalization on intermediate features (**Details in Section II.B (4)**). Unlike min–max normalization, which acts at the dataset level, IN adaptively normalizes each instance and thus alleviates the influence of local distribution shifts across different operating conditions and event scenarios. As a result, the model can focus more

TABLE III
MAIN PARAMETER SETTING

	Model	Parameter	Values
Student Model & FL Model	MLP 1 (For Trend)	Hidden Dim	10
		Activation Function	ReLU
	MLP 2 (For De-Trend)	Hidden Dim	10
		Activation Function	ReLU
	MLP 3 (For Pai Filter)	Hidden Dim	[64, 30]
		Activation Function	ReLU
		FC Layers	2
Teacher Model	Embedding Head	Hidden Dim	768
	GPT2-Small	Decoder Layer	12
	Classification Head	Hidden Dim	128
Hyper-Parameters	KD	Look-back Window	300
		Max Epoch	100
		Patch Size	10
		Learning Rate	0.01
		Balancing Coefficient	0.1
		Optimizer	AdamW
	FL	Communication Round	10 Epoch
		Optimizer	SGD

on informative temporal patterns rather than scale variations, leading to improved robustness and training stability.

Table III shows the core parameters involved in the whole prediction framework. The hyperparameters in this paper are optimized by grid search.

C. Evaluation Metrics

We adopt **Precision**, **Recall**, and **F1-score** as the metrics for evaluation. Their definitions are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{39}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{40}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{41}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

D. Operation Environment

All experiments were conducted on a workstation equipped with an Intel i7-13700KF CPU and two NVIDIA RTX 4090D GPUs. The models were implemented in PyTorch, with PyCharm used as the development environment.

V. CASE STUDIES

In this section, we demonstrate the remarkable performance of the proposed algorithm through seven comprehensive case studies and further elucidate the unique contribution of each component via ablation studies.

A. Performance Comparison Across Models

In this experiment, we systematically evaluated the overall event classification performance of the proposed framework. For baseline models, we included traditional machine learning approaches (MLP, LSTM, and CNN), as well as recent SOTA models—Informer [35] and PatchTST [7]. All baseline models were independently trained on local data. All hyper-parameters were tuned via grid search, and an early-stopping strategy was adopted during training to mitigate overfitting. **Tables IV & V** report the parameter settings of the comparative models and the corresponding experimental results. Due to space limitations, some hyper-parameters that are identical to those in **Table III** are omitted from **Table IV**.

The results demonstrate that information-sharing strategies, whether through knowledge distillation or federated learning, significantly outperform all baseline models. Although fed-

TABLE IV
PARAMETER SETTINGS OF THE COMPARATIVE MODELS

Model	Block	Hyperparameter	Value
MLP	MLP	Layers	3
		Hidden Size	256
Bi-LSTM	Bi-LSTM	Layers	1
		Hidden Size	128
CNN	Conv1D	Out Channels	[64, 128, 256]
		Kernel Size	3
		Padding	1
	MaxPooling	Kernel Size	3
		Stride	2
Informer	Encoder	Dimension	512
		Heads	8
		Layers	2
		FFN Dim	2048
		ProbSparse Factor	5
	Decoder	Dimension	512
		Heads	8
		Layers	1
		FFN Dim	2048
		Distilling	True
PatchTST	Patching	Patch Length	16
		Patch Stride	8
	Encoder	Dimension	512
		Heads	8
		Layers	3
		FFN Dim	2048

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT MODELS

Label	Model	Precision	Recall	F1-Score
Label 0	MLP	0.990	0.951	0.970
	LSTM	0.997	0.984	0.990
	CNN	0.997	0.997	0.997
	Informer	0.996	0.961	0.978
	PatchTST	0.997	0.999	0.998
	Our Work (Teacher Model)	**1.000**	**1.000**	**1.000**
	Our Work (Student Model)	**1.000**	**1.000**	**1.000**
	Our Work (FL)	**1.000**	**1.000**	**1.000**
Label 1	MLP	0.018	0.931	0.036
	LSTM	0.097	1.000	0.177
	CNN	0.361	0.897	0.515
	Informer	0.019	0.793	0.037
	PatchTST	0.730	0.931	0.818
	Our Work (Teacher Model)	**1.000**	**1.000**	**1.000**
	Our Work (Student Model)	**0.935**	**1.000**	**0.967**
	Our Work (FL)	**0.906**	**1.000**	**0.951**
Label 2	MLP	0.197	1.000	0.329
	LSTM	0.102	0.920	0.183
	CNN	0.202	0.800	0.323
	Informer	1.000	0.200	0.333
	PatchTST	0.266	0.680	0.382
	Our Work (Teacher Model)	**0.962**	**1.000**	**0.980**
	Our Work (Student Model)	**1.000**	**1.000**	**1.000**
	Our Work (FL)	**0.962**	**1.000**	**0.980**
Label 3	MLP	1.000	0.946	0.972
	LSTM	0.999	0.948	0.973
	CNN	1.000	0.947	0.973
	Informer	0.999	0.944	0.971
	PatchTST	1.000	0.955	0.977
	Our Work (Teacher Model)	**0.998**	**0.999**	**0.999**
	Our Work (Student Model)	**0.999**	**0.997**	**0.998**
	Our Work (FL)	**1.000**	**0.994**	**0.997**
Label 4	MLP	0.989	0.821	0.897
	LSTM	0.993	0.989	0.991
	CNN	0.995	0.993	0.994
	Informer	0.993	0.954	0.973
	PatchTST	0.990	0.994	0.992
	Our Work (Teacher Model)	**1.000**	**0.998**	**0.999**
	Our Work (Student Model)	**0.998**	**0.999**	**0.999**
	Our Work (FL)	**0.996**	**0.999**	**0.998**

TABLE VI
COMPARISON OF DEPLOYMENT POTENTIAL

Model	Parameters (K)	Inference Time (MS)	Memory Usage (MB)
MLP	68.357	5.380	1.51
LSTM	266.117	16.016	1.99
CNN	31.973	3.987	0.36
Informer	5790.213	12.958	23.18
PatchTST	2102.891	7.273	10.27
Our Work (Teacher Model)	**124471.581**	**357.132**	**442.52**
Our Work (Student Model & FL)	**4.269**	**1.993**	**0.14**

Fig. 8. Confusion matrices of LLM and PatchTST-based teacher model.

erated learning exhibits slightly lower performance than the distilled student model due to its limitation of sharing only gradient information, it still demonstrates strong detection capability and robustness.

Furthermore, we evaluated the edge deployment potential from multiple perspectives, including Parameter size (K), Inference time (MS), and Memory usage (MB), as shown in **Table VI**. The results indicate that our edge-deployable model is highly lightweight in terms of resource requirements, while achieving extremely fast inference speed, thereby making it highly suitable for edge deployment scenarios.

Remark. To further examine the scalability of the proposed framework, we conducted a preliminary study on a publicly available simulation-based IEEE 118-bus dataset [36]. The preliminary results were encouraging and suggested that the proposed method remains effective on a larger-scale power system. However, since this dataset is purely simulation-based, it is not fully aligned with the core experimental setting of this work, which focuses on HIL-based real-world synchrophasor measurements. Therefore, to maintain the consistency and focus of the main experimental section, we do not include this study as a formal experiment in our paper. A more comprehensive HIL-based validation on larger-scale systems will be pursued in future work.

B. Remarkable Performance of LLM in Teacher Model

In this experiment, we verify the necessity of embedding the LLM into the teacher model and fine-tuning it. To this end, we set up several comparative baselines, including embedding the LLM without enabling fine-tuning (w/o Fine-tuning), as well as directly adopting PatchTST and Informer as teacher models. To further verify the contribution of the pretrained LLM's prior knowledge to improving the teacher model's performance, we additionally introduce two comparative experiments. The first is an ablation study: we directly remove the LLM component (w/o LLM) and replace the LLM with a standard Transformer structure while keeping the remaining modules and training settings unchanged, in order to examine whether incorporating the LLM leads to consistent performance gains. The second is a prior-knowledge removal study: we eliminate the knowledge encoded by the LLM's pretraining (w/o Pretrained Knowledge), degenerating it into a stacked standard Transformer Decoder architecture, thereby assessing the impact of pretrained prior knowledge itself on the final performance.

The experimental results, as shown in **Figure 8** and **Table VII**, demonstrate that incorporating the LLM via embedding and fine-tuning consistently improves the teacher model's performance; moreover, the prior knowledge encoded in the pretrained LLM provides an additional and significant boost. Although the LLM has a relatively large number of parameters, its superior performance and generalization indicate that this design choice is both necessary and justified for teacher models.

TABLE VII
COMPARISON OF DIFFERENT TEACHER MODELS

Label	Model	Precision	Recall	F1-Score
Label 0	Informer	1.000	0.999	0.999
	PatchTST	0.998	0.999	0.999
	w/o LLM	0.998	0.999	0.999
	w/o Pretrained Knowledge	0.999	0.998	0.998
	w/o Fine-tuning	0.999	1.000	1.000
	Our Work	**1.000**	**1.000**	**1.000**
Label 1	Informer	0.829	1.000	0.906
	PatchTST	0.763	1.000	0.866
	w/o LLM	0.337	1.000	0.504
	w/o Pretrained Knowledge	0.967	1.000	0.983
	w/o Fine-tuning	1.000	0.931	0.964
	Our Work	**1.000**	**1.000**	**1.000**
Label 2	Informer	0.625	1.000	0.769
	PatchTST	0.558	0.960	0.706
	w/o LLM	0.511	0.960	0.667
	w/o Pretrained Knowledge	0.294	1.000	0.455
	w/o Fine-tuning	0.926	1.000	0.962
	Our Work	**0.962**	**1.000**	**0.980**
Label 3	Informer	0.997	0.988	0.992
	PatchTST	0.998	0.989	0.994
	w/o LLM	1.000	0.958	0.979
	w/o Pretrained Knowledge	0.995	0.994	0.994
	w/o Fine-tuning	1.000	0.995	0.997
	Our Work	**0.998**	**0.999**	**0.999**
Label 4	Informer	0.989	0.991	0.990
	PatchTST	0.995	0.969	0.982
	w/o LLM	0.999	0.990	0.995
	w/o Pretrained Knowledge	0.991	0.978	0.984
	w/o Fine-tuning	0.998	0.995	0.996
	Our Work	**1.000**	**0.998**	**0.999**

C. Analysis of LLM's Fine-tuning Strategy

To improve the adaptability of the pretrained LLM to downstream tasks, we fine-tune its last two layers and perform a set of comparative experiments to justify this design choice. As GPT-2 small comprises 12 stacked Transformer decoder blocks, we conduct a layer-wise sweep with a granularity of two layers, covering configurations from fully frozen training (0 fine-tuned layers) to full-parameter fine-tuning (all 12 layers). Performance is evaluated using the macro-average F1 score across all labels. Training efficiency is assessed via training throughput in iterations per second (it/s), i.e., the number of optimization iterations completed per second, which we use as a proxy for computational overhead. The results are summarized in **Figure 9** and **Table VIII**.

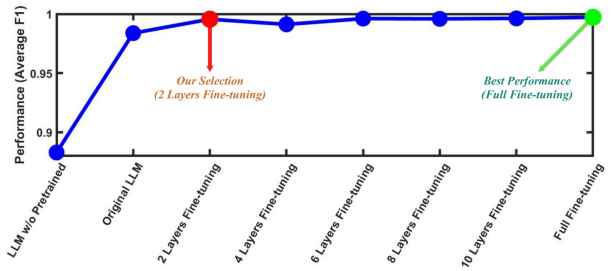

Fig. 9. Performance evolution under different fine-tuning strategies.

TABLE VIII
COMPARISON OF DIFFERENT LLM FINE-TUNING STRATEGIES

Fine-tuning Strategy	Performance (Average F1)	Training Efficiency (it/s)
LLM w/o Pretrained	0.8828	0.73
Original LLM	0.9838	1.04
2 Layers Fine-tuning	0.9956	0.98
4 Layers Fine-tuning	0.9912	0.91
6 Layers Fine-tuning	0.9961	0.87
8 Layers Fine-tuning	0.9959	0.82
10 Layers Fine-tuning	0.9963	0.75
Full Fine-tuning	0.9972	0.70

From the experimental results, fine-tuning the last two layers already delivers near-saturated performance (0.9956 Average F1) while maintaining high training efficiency (0.98 it/s). In contrast, full fine-tuning only yields a marginal gain but noticeably reduces throughput (0.70 it/s). Therefore, we select two-layer fine-tuning as a better accuracy–efficiency trade-off.

D. Flexibility for PMUs with Varying Sampling Rates

In practical engineering applications, power systems typically span vast geographical areas and may even extend across national borders. As a result, the PMUs deployed at different locations may have inconsistent sampling rates. However, it is worth noting that the proposed framework demonstrates strong scalability under such conditions: for knowledge distillation, it only requires introducing an additional fully connected layer in the embedding module of the teacher model to achieve unified mapping of different sampling rates; while federated learning itself can adaptively handle data with different frequencies without further adjustment. To simulate this scenario of device heterogeneity in a large power grid, we downsampled the sampling frequency of PMU1 from the original 30 Hz to 15 Hz and 10 Hz, while keeping the other PMUs unchanged, and compared the performance differences (F1-score) of the proposed method and models trained solely on local data at the PMU1 location, which is shown in **Table IX**.

From the experimental results, it can be observed that the lower the sampling rate, the greater the performance improvement compared to models trained solely on local data. This indicates that the knowledge embedded in high-sampling-rate PMUs can be effectively transferred to low-sampling-rate PMUs, thereby significantly enhancing their performance. Furthermore, this result highlights the tremendous potential of the proposed framework in engineering applications, as this mechanism reduces the need for frequent upgrades of legacy devices to some extent and also alleviates the pressure of standardizing equipment across different regions.

TABLE IX
PERFORMANCE IMPROVEMENT ON DIFFERENT SAMPLING RATES

Label	Model	30 Hz	15 Hz	10 Hz
Label 0	KD	+1.4%	+3.7%	+8.1%
	FL	+1.3%	+3.5%	+8.0%
Label 1	KD	+17.4%	+26.1%	+46.2%
	FL	+15.2%	+23.4%	+46.4%
Label 2	KD	+54.3%	+72.7%	+78.3%
	FL	+51.0%	+68.3%	+72.9%
Label 3	KD	+2.5%	+6.1%	+6.7%
	FL	+1.8%	+5.9%	+6.7%
Label 4	KD	+2.6%	+4.0%	+5.2%
	FL	+2.0%	+3.2%	+4.8%

TABLE X
COMPARISON OF DIFFERENT FL MODELS

Label	Model	Precision	Recall	F1-Score
Label 0	FedAvg	0.997	0.993	0.995
	qFedAvg	0.998	0.999	0.999
	Our Work (FL)	**1.000**	**1.000**	**1.000**
Label 1	FedAvg	0.630	1.000	0.773
	qFedAvg	1.000	0.897	0.945
	Our Work (FL)	**0.906**	**1.000**	**0.951**
Label 2	FedAvg	0.379	1.000	0.549
	qFedAvg	0.781	1.000	0.877
	Our Work (FL)	**0.962**	**1.000**	**0.980**
Label 3	FedAvg	0.991	0.960	0.975
	qFedAvg	0.995	0.994	0.994
	Our Work (FL)	**1.000**	**0.994**	**0.997**
Label 4	FedAvg	0.909	0.967	0.937
	qFedAvg	0.994	0.980	0.987
	Our Work (FL)	**0.996**	**0.999**	**0.998**

E. The Necessity of Fairness Federated Learning

In this experiment, we validated the necessity of incorporating fairness into federated learning. We adopted the standard FedAvg as a baseline method and further introduced a simple fairness-aware weighted approach, qFedAvg [37], for comparison. We additionally evaluated the training overhead of various federated learning algorithms under the same hyperparameter setting (one communication round per 10 local epochs). The reported overhead metrics include the number of communication rounds, the per-round payload size (uplink + downlink) in Mbit, and the corresponding communication cost measured in MB.

The experimental results (**Table X and Table XI**) demonstrate that the proposed fair federated learning not only substantially improves the overall performance of federated learning but also effectively mitigates the large performance discrepancies across different deployment sites caused by data heterogeneity. Meanwhile, we observe that the payload size is directly proportional to the number of model parameters. Since the edge model is extremely lightweight, the communication burden remains modest across all algorithms. Compared with the conventional FedAvg baseline, our method introduces only a negligible increase in payload per round, while substantially reducing the required number of communication rounds and achieving markedly better performance. These results further highlight the necessity and practical value of fairness-aware federated learning.

TABLE XI
TRAINING OVERHEAD OF DIFFERENT FL MODELS

	Model Parameters	Communication Rounds	Payload (Mbit)	Communication Cost (MB)
FedAvg	4269	28	2.186	0.273
qFedAvg		36	2.186	0.273
Our Work		17	2.191	0.274

TABLE XII
PERFORMANCE UNDER VARIOUS COMPROMISED-PMU CONDITIONS

	Label 0	Label 1	Label 2	Label 3	Label 4
Dropout (1 PMU)	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%
Dropout (2 PMUs)	-0.0%	-1.2%	-3.6%	-0.2%	-0.2%
Dropout (3 PMUs)	-1.1%	-13.4%	-21.8%	-3.5%	-4.0%
Noise (1 PMU)	-0.0%	-1.4%	-2.3%	-0.0%	-0.3%
Noise (2 PMUs)	-0.3%	-1.8%	-2.5%	-0.4%	-0.5%
Noise (3 PMUs)	-0.6%	-5.2%	-8.7%	-1.1%	-0.7%
Timestamp offset (1 PMU)	-0.0%	-0.0%	-1.2%	-0.0%	-0.1%
Timestamp offset (2 PMUs)	-0.2%	-1.6%	-4.9%	-0.5%	-0.4%
Timestamp offset (3 PMUs)	-3.0%	-21.4%	-41.3%	-5.0%	-7.1%

F. PMU Compromised Analysis in Federated Learning

In practical power systems, PMU data may become compromised (i.e., degraded or less reliable) due to engineering uncertainties or even malicious attacks in both the measurement and time-synchronization chains. Typical causes include transient communication outages leading to missing samples, electromagnetic interference or sensor aging that elevates measurement noise, and degraded GPS spoofing attacks that induce timestamp offsets.

Under the federated learning paradigm, such impaired measurements are propagated into the global model through local updates, potentially perturbing aggregation and degrading generalization. To emulate these non-adversarial yet realistic impairments, we consider three representative anomaly types: Dropout, where intermittent reporting failures or packet losses yield contiguous or sparse missing data; Noise, where random perturbations are injected into phasors and derived quantities to reflect measurement noise and front-end drift (5% Gaussian noise in this experiment); and Timestamp offset, where fixed or randomly varying time shifts desynchronize samples and disrupt cross-PMU temporal alignment (random time-step lags in this experiment), typically exerting a stronger impact than purely additive noise.

Without altering the FL training protocol, we inject these anomalies into participating PMUs and vary the number of compromised PMUs as 1/2/3, reporting the relative performance changes on different Labels (percentages; negative values indicate degradation based on F1 score) to systematically assess robustness and sensitivity with respect to impairment type and scale, as shown in **Table XII**. The results show that when only one or two PMUs are compromised, the performance across all labels remains almost unchanged; as the number increases to 3, degradation becomes more evident. Among the three anomaly types, timestamp offset is the most detrimental and exhibits a steeper degradation trend as more PMUs are affected, consistent with the strong dependence of synchrophasor analytics on precise time synchronization. These results suggest that the proposed framework has meaningful practical robustness under mild corruption conditions, while stronger compromised scenarios remain challenging and warrant further investigation.

TABLE XIII
ABLATION EXPERIMENT OF EDGE AI MODEL

Label	Model		Precision	Recall	F1-Score
Label 0	w/o Patching	KD	0.998	0.998	0.998
		FL	0.999	0.995	0.997
	w/o Decomposition	KD	1.000	0.999	1.000
		FL	0.999	0.999	0.999
	w/o Pai Filter	KD	0.999	0.997	0.998
		FL	0.998	0.999	0.999
	Our Work	**KD**	**1.000**	**1.000**	**1.000**
		FL	**1.000**	**1.000**	**1.000**
Label 1	w/o Patching	KD	0.382	0.897	0.536
		FL	0.311	0.966	0.471
	w/o Decomposition	KD	0.966	0.966	0.966
		FL	0.967	1.000	0.983
	w/o Pai Filter	KD	0.467	0.966	0.629
		FL	0.397	0.931	0.557
	Our Work	**KD**	**0.935**	**1.000**	**0.967**
		FL	**0.906**	**1.000**	**0.951**
Label 2	w/o Patching	KD	0.431	1.000	0.602
		FL	0.173	0.880	0.289
	w/o Decomposition	KD	0.806	1.000	0.893
		FL	0.781	1.000	0.877
	w/o Pai Filter	KD	0.375	0.960	0.539
		FL	0.219	1.000	0.360
	Our Work	**KD**	**1.000**	**1.000**	**1.000**
		FL	**0.962**	**1.000**	**0.980**
Label 3	w/o Patching	KD	0.999	0.957	0.978
		FL	0.999	0.960	0.979
	w/o Decomposition	KD	0.995	0.999	0.997
		FL	0.996	0.995	0.995
	w/o Pai Filter	KD	1.000	0.959	0.979
		FL	0.999	0.959	0.979
	Our Work	**KD**	**0.999**	**0.997**	**0.998**
		FL	**1.000**	**0.994**	**0.997**
Label 4	w/o Patching	KD	0.996	0.996	0.996
		FL	0.992	0.993	0.992
	w/o Decomposition	KD	0.998	0.998	0.998
		FL	0.998	0.991	0.994
	w/o Pai Filter	KD	1.000	0.991	0.996
		FL	1.000	0.991	0.995
	Our Work	**KD**	**0.998**	**0.999**	**0.999**
		FL	**0.996**	**0.999**	**0.998**

G. Ablation Experiment of Edge AI Model

In this work, we carefully designed a lightweight AI model tailored for edge deployment. Despite its small parameter size, the model achieves substantial performance improvements through diverse information extraction mechanisms. To validate the contribution of each component, we conducted ablation studies by sequentially removing the Patch module, the decomposition module, and the learnable filter module to evaluate their roles, which is shown in **Table XIII**. The experimental results clearly demonstrate that both the Patch module and the learnable filter significantly enhance model performance, while the decomposition module provides a relatively modest improvement, yet still proves to be an indispensable part of the overall design.

VI. CONCLUSIONS

This paper introduced an edge-centric event classification framework for PMU data that jointly addresses data-sharing constraints, latency, and accuracy through two complementary information-sharing strategies: knowledge distillation from a global LLM-enhanced teacher to lightweight on-device student models when limited data exchange is permissible, and fairness-aware federated learning that shares gradients instead of raw data when data sharing is fully restricted. The proposed edge model ($\approx$4.3K parameters) incorporates patch embedding, frequency-trend decomposition, and a learnable

frequency filter, enabling efficient and robust inference on PMU hardware. Hardware-in-the-loop evaluations on an IEEE 39-bus testbed (8 PMUs) demonstrate that both the distilled students and the federated model outperform local-only and recent deep baselines, maintain deployability (2 ms inference, 0.14 MB memory), and remain resilient to heterogeneous sampling rates, with ablation studies validating each component's contribution.

For the power industry and system operators, the framework enables near-real-time, situational awareness at the grid edge under different privacy scenarios, reducing dependence on wide-area data aggregation, lowering communication overhead, and delivering more uniform performance across substations under non-IID conditions. This supports faster, more reliable detection of faults, line/generation changes, and load events—directly where decisions are made.

Although the proposed edge AI framework demonstrates strong performance, several limitations remain to be addressed. Accordingly, we highlight the following avenues for future enhancement:

1) Sensitivity to rare and previously unseen event patterns: The experimental results show that the proposed lightweight student model can achieve satisfactory classification performance under the studied settings. Nevertheless, as with many compact edge-oriented models, further improvement may still be possible when dealing with extremely rare or previously unseen complex event patterns. This issue is of practical interest in edge AI applications, where model compactness and expressive capability need to be carefully balanced. Future work will therefore explore ways to enhance the sensitivity and recognition ability of the lightweight model for such challenging cases, for example, through improved teacher-student transfer, rare-event-aware training strategies, and other mechanisms for strengthening representation under limited model capacity.

2) Generalizability to heterogeneous systems and sensing modalities: The proposed framework has shown promising performance in the PMU-based setting considered in this work. Meanwhile, its extension to power systems with different topologies, operating characteristics, or sensing modalities deserves further investigation. In particular, applying the current framework beyond PMU-based measurements to other types of sensor data, such as WMU (Waveform Measurement Units), may involve additional adaptation in feature representation and model design. Future research will therefore investigate transfer learning and cross-domain adaptation strategies to improve the portability of the proposed framework across heterogeneous power-system settings and to facilitate its application to a broader range of sensing data.

REFERENCES

[1] H. Mohsenian-Rad, *Smart Grid Sensors: Principles and Applications*. Cambridge, U.K.: Cambridge University Press, 2022.

[2] J. Zhao, G. Zhang, K. Das, G. N. Korres, N. M. Manousakis, A. K. Sinha, and Z. He, "Power system real-time monitoring by using pmu-based robust state estimation method," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 300–309, 2016.

[3] A. G. Phadke and T. Bi, "Phasor measurement units, wams, and their applications in protection and control of power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 4, pp. 619–629, 2018.

[4] F. L. Grando, A. E. Lazzaretti, and M. Moreto, "The impact of pmu data precision and accuracy on event classification in distribution systems," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1372–1382, 2022.

[5] S. Jia and Q. Guo, "Functional safety analysis and mitigation in power systems: A cyber-physical perspective," *Energy Internet*, vol. 2, no. 3, pp. 198–217, 2025.

[6] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, "Timemixer: Decomposable multiscale mixing for time series forecasting," in *Proc. Int. Conf. Learn. Represent.*, Vienna, Austria, May 2024.

[7] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. Int. Conf. Learn. Represent.*, Kigali, Rwanda, May 2023.

[8] Z. Pan, C. Li, F. Yu, S. Wang, H. Wang, X. Tang, and J. Zhao, "FedLF: Layer-wise fair federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 13. Vancouver, BC, Canada: AAAI Press, Mar. 2024, pp. 14 527–14 535.

[9] D.-I. Kim, T. Y. Chun, S.-H. Yoon, G. Lee, and Y.-J. Shin, "Wavelet-based event detection method using pmu data," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1154–1162, 2017.

[10] G. Liu, X. Li, C. Wang, Z. Chen, R. Chen, and R. C. Qiu, "Hessian locally linear embedding of pmu data for efficient fault detection in power systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–4, 2022.

[11] M. Cui, J. Wang, J. Tan, A. R. Florita, and Y. Zhang, "A novel event detection method using pmu data with high precision," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 454–466, 2019.

[12] J.-A. Jiang, J.-Z. Yang, Y.-H. Lin, C.-W. Liu, and J.-C. Ma, "An adaptive pmu based fault detection/location technique for transmission lines. i. theory and algorithms," *IEEE Transactions on Power Delivery*, vol. 15, no. 2, pp. 486–493, 2000.

[13] A. Shahsavari, M. Farajollahi, E. M. Stewart, E. Cortez, and H. Mohsenian-Rad, "Situational awareness in distribution grid using micro-pmu data: A machine learning approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6167–6177, 2019.

[14] Y. Liu, L. Yang, A. Ghasemkhani, H. Livani, V. A. Centeno, P.-Y. Chen, and J. Zhang, "Robust event classification using imperfect real-world pmu data," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7429–7438, 2023.

[15] M. Biswal, S. M. Brahma, and H. Cao, "Supervisory protection and automated event diagnosis using pmu data," *IEEE Transactions on Power Delivery*, vol. 31, no. 4, pp. 1855–1863, 2016.

[16] P. Khaledian and H. Mohsenian-Rad, "Automated event region identification and its data-driven applications in behind-the-meter solar farms based on micro-pmu measurements," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 2094–2106, 2022.

[17] B. Wang, Y. Li, and J. Yang, "LSTM-based quick event detection in power systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting*. Montr'eal, QC, Canada: IEEE, Aug. 2020, pp. 1–5.

[18] M. Pavlovski, M. Alqudah, T. Dokic, A. A. Hai, M. Kezunovic, and Z. Obradovic, "Hierarchical convolutional neural networks for event classification on pmu measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[19] Y. Yuan, Y. Guo, K. Dehghanpour, Z. Wang, and Y. Wang, "Learning-based real-time event identification using rich real pmu data," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5044–5055, 2021.

[20] W. Wang, H. Yin, C. Chen, A. Till, W. Yao, X. Deng, and Y. Liu, "Frequency disturbance event detection based on synchrophasors and deep learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3593–3605, 2020.

[21] M. Rafferty, X. Liu, J. Rafferty, L. Xie, D. Laverty, and S. McLoone, "Sequential feature selection for power system event classification utilizing wide-area pmu data," *Frontiers in Energy Research*, vol. Volume 10 - 2022, 2022.

[22] G. Liu, H. Chen, X. Sun, N. Quan, L. Wan, and R. Chen, "Low-complexity nonlinear analysis of synchrophasor measurements for events detection and localization," *IEEE Access*, vol. 6, pp. 4982–4993, 2018.

[23] J. Yang, H. Yu, P. Li, H. Ji, W. Xi, J. Wu, and C. Wang, "Real-time d-pmu data compression for edge computing devices in digital distribution networks," *IEEE Transactions on Power Systems*, vol. 39, no. 4, pp. 5712–5725, 2024.

[24] D. L. M and R. Varadarajan, "Optimised autoencoder-based ensemble deep learning approaches for cyber-physical event classification utilizing

synchrophasor pmu data," *Results in Engineering*, vol. 27, p. 105884, 2025.
[25] Z. Tian and C. Wu, "Enhancing long-sequence photovoltaic power forecasting accuracy through multi-modal learning: Integrating satellite cloud images and time–frequency domain fusion," *Advanced Engineering Informatics*, vol. 74, p. 104622, 2026.
[26] Y. Cheng, S. Zhang, and H. Li, "Optimising time series forecasting transformers to linear complexity," *Energy Internet*, vol. 2, no. 4, pp. 275–286, 2025.
[27] K. Yi, J. Fei, Q. Zhang, H. He, S. Hao, D. Lian, and W. Fan, "Filternet: Harnessing frequency filters for time series forecasting," 2024. [Online]. Available: https://arxiv.org/abs/2411.01623
[28] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *Proc. Int. Conf. Learn. Represent.*, Vienna, Austria, May 2024.
[29] A. Jena, F. Ding, J. Wang, Y. Yao, and L. Xie, "Llm-based adaptive distribution voltage regulation under frequent topology changes: An in-context mpc framework," *IEEE Transactions on Smart Grid*, vol. 16, no. 5, pp. 4297–4300, 2025.
[30] J. Lee, R. Tang, and J. Lin, "What would elsa do? freezing layers during transformer fine-tuning," 2019. [Online]. Available: https://arxiv.org/abs/1911.03090
[31] M. Cui, Y. Han, G. Luo, S. Wang, B. Kang, and J. Zhang, "Privacy-preserving distribution system state estimation considering dynamic impacts of transmission network," *IEEE Transactions on Power Systems*, vol. 41, no. 3, pp. 1841–1854, 2026.
[32] Z. Pan, S. Wang, C. Li, H. Wang, X. Tang, and J. Zhao, "FedMDFG: Federated learning with multi-gradient descent and fair guidance," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 8. Washington, DC, USA: AAAI Press, Feb. 2023, pp. 9364–9371.
[33] H. M. Mustafa, V. Sivaramakrishnan, V. V. G. Krishnan, and A. Srivastava, "Realistic synchrophasor data generation for anomaly detection using cyber-power testbed," in *Proc. 56th North Amer. Power Symp.* El Paso, TX, USA: IEEE, Oct. 2024, pp. 1–6.
[34] B. Xu and A. Abur, "Optimal placement of phasor measurement units for state estimation," Power Systems Engineering Research Center (PSERC), Texas A&M University, College Station, TX, USA, Final Project Report PSERC Publication 05-58, Oct. 2005. [Online]. Available: https://pserc.wisc.edu/wp-content/uploads/sites/755/2018/08/S-23G_Final-Report_Oct-2005.pdf
[35] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12. Virtual Event: AAAI Press, Feb. 2021, pp. 11 106–11 115.
[36] H. Hassani, E. Hallaji, R. Razavi-Far *et al.*, "Learning from high-dimensional cyber-physical data streams: a case of large-scale smart grid," *International Journal of Machine Learning and Cybernetics*, vol. 16, pp. 1819–1831, 2025.
[37] R. Wang, H. Qiu, H. Gao, C. Li, Z. Y. Dong, and J. Liu, "Adaptive horizontal federated learning-based demand response baseline load estimation," *IEEE Transactions on Smart Grid*, vol. 15, no. 2, pp. 1659–1669, 2024.

Zhirui Tian (Student Member, IEEE) received the B.S. degree from Dongbei University of Finance and Economics, Dalian, China, in 2023, where he was recognized as an Outstanding Graduate of Dongbei University of Finance and Economics. He is currently pursuing the Ph.D. degree in Computer and Information Engineering at the School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), under the supervision of Prof. Chenye Wu. His research interests include smart grid, deep learning, and edge AI systems.

Hamed Mohsenian-Rad (M'09-SM'14-F'20) received the Ph.D. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2008. He is currently a Professor of electrical engineering and a Winston Chung Endowed Chair Professor in Energy Innovation at the University of California, Riverside, CA, USA. His research is on monitoring, data analysis, and optimization of power systems and smart grids. He is the author of the textbook *Smart Grid Sensors: Principles and Applications* published by Cambridge University Press in 2022. He was the recipient of the National Science Foundation (NSF) CAREER Award, the Best Paper Award from the IEEE Power and Energy Society General Meeting, the Best Paper Award from the IEEE Conference on Smart Grid Communications, and a Technical Achievement Award from the IEEE Communications Society. He has been the PI or co-PI on close to twenty million dollars research grants in the area of smart grid. He has served as Editor for the IEEE TRANSACTIONS ON POWER SYSTEMS, the IEEE TRANSACTIONS ON SMART GRID and the IEEE POWER ENGINEERING LETTERS.

Chenye Wu (Senior Member, IEEE) is an Assistant Professor at the School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen). Before joining CUHK Shenzhen, he was an Assistant Professor at the Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University. Dr. Wu received his Ph.D. from IIIS, Tsinghua University, in July 2013. His Ph.D. advisor was Professor Andrew Yao, the laureate of the A.M. Turing Award in the year of 2000. Dr. Wu was a co-recipient of the Best Paper Award at IEEE SmartGridComm 2012, IEEE PES General Meeting 2013, and IEEE PES General Meeting 2020. Currently, he is working on economic analysis, optimal control, and the operation of power systems.