Towards Distributed Learning of PMU Data: A Federated Learning based Event Classification Approach

Seyed Mahmoud Sajjadi Mohammadabadi*, Yunchuan Liu†, Abraham Canafe‡, and Lei Yang*
* Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557
† Division of Science Mathematics and Technology, Governors State University, IL 60484
‡ Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095

***Abstract*—This paper studies distributed learning of real-world phasor measurement unit (PMU) data to enable privacy-preserving data sharing and analytics among different entities (i.e., PMU data owners). As real-world PMU data are collected/owned by different entities, they may not share their data due to privacy/security concerns. To tackle this challenge, this paper develops a federated learning based event classification approach that enables multiple entities to collaboratively train a good event classifier while keeping the PMU data decentralized and private. As each entity may have different numbers of PMUs, the inputs from different entities to the event classification model can be different. To address this challenge, a federated XGBoost model is proposed, in which the same set of event features can be constructed by each entity with different numbers of PMUs. To enhance the privacy of the federated XGBoost model, the communication during the training process is encrypted using additive homomorphic encryption scheme, which eliminates the requirement of a trusted server. Numerical experiments using the real-world PMU dataset show that the proposed federated XGBoost model achieves almost the same performance as the centralized counterparts and is robust against different data distributions of entities.**

***Index Terms*—Phasor Measurement Units (PMUs), Event Classification, Federated Learning**

I. Introduction

A. Motivation

Recent years have witnessed the booming deployment of phasor measurement units (PMUs). More than 2500 PMUs have been deployed in the North America. As PMUs are of much higher sampling rates (e.g., 30 or 60 samples per second in the U.S.), effective use of PMU data can enable a high level of situational awareness (e.g., real-time event detection and classification) to prevent regional or large-scale blackouts. There have been many studies on PMU based event classification using different machine learning (ML) techniques (e.g., [1–4]). One common assumption is that all the PMU data are available when training event classifiers. However, real-world PMU data are collected/owned by different entities (e.g., utility companies). These entities (i.e., PMU data owners) may not share their data due to privacy/security concerns (e.g., Critical Infrastructure Protection (CIP) and Critical Infrastructure Information (CII)). As the efficacy of existing ML techniques generally depends on the amount of training data, the amount of training data (i.e., recorded events) at an individual entity may not be sufficient for training a good model. Thus, it is of critical importance to enable privacy-preserving cross-entity data sharing and analytics to effectively learn PMU data and develop good event classifiers.

Federated learning (FL) that enables multiple entities to analyze sensitive data while providing improved privacy protections has received tremendous attention recently and can improve privacy in comparison with traditional centralized ML paradigm. In this paper, we aim to develop a FL based event classification approach that can train the event classifier without accessing the data from each entity in a distributed manner.

B. Related Works

Existing studies on PMU based event classification (e.g., [5–8]) leverage different ML techniques to train event classifiers, whose performance depends on large amounts of high-quality labeled PMU data. However, real-world PMU data are often insufficiently labeled, not to mention the potential low quality of raw PMU measurements [4]. To address this challenge, different methods have been proposed by either leveraging synthetic data (e.g., from simulation [9] or Generative Adversarial Neural Network [10]) or the large amount of unlabeled PMU data (e.g., using unsupervised learning [11] or semi-supervised learning [12], [13] or weakly supervised learning [4]).

Although these works show promising event classification results, one common assumption of these works is that all the PMU data are available at a central location when training event classifiers, i.e., they all consider a centralized ML scenario. In practice, real-world PMU data are collected/owned by different entities, who may not share their data due to privacy/security concerns. Moreover, with the development of grid edge, there is a pressing need of distributed ML approaches for distributed monitoring and decision-making at the grid edge without the transmission of large amounts of raw PMU data to a central location.

Recently, FL (e.g., [14–18]) has attracted tremendous attention from both academia and industry. FL enables multiple entities to individually train/update a global model using their local data in a privacy-preserving manner. Each entity's data do not need to be transmitted to a central location during the FL process. This provides a promising way towards distributed learning of PMU data for event classification. To the best of our knowledge, FL techniques have not been previously used for event classification using PMU data.

C. Main Contributions

This paper develops a FL based event classification approach that enables multiple entities to collaboratively train a good event classifier without the need of sharing any PMU data. The proposed FL based event classification approach develops a federated XGBoost model, as the centralized XGBoost used in our previous study [3] shows not only good classification performance with strong interpretability but also improved robustness of event classification against bad and missing data. Using the proposed FL based event classification approach, multiple entities collaboratively learn a federated XGBoost model without leaking any PMU data to each other. When training the federated XGBoost model, we leverage the simple and efficient FedSGD framework [14], in which each entity updates the current global model using its local data in parallel and then sends the model update to a server for aggregation from time to time. To enhance the privacy of the federated XGBoost model, the communication during the training process is encrypted using additive homomorphic encryption scheme [19], which enables using additive operations on homomorphically encrypted data to reduce decryption overhead. To the best of our knowledge, this paper is the first study of FL based event classification in power systems. The findings in the paper can shed the light on distributed learning of PMU data for event classification.

Using the two-year real-world PMU data from the Western Interconnection of the continental U.S. transmission grid, the performance of the proposed FL based event classification approach is evaluated. The experimental results show that under different scenarios, the proposed federated XGBoost model can achieve almost the same performance as the centralized counterparts. This shows a promising way of privacy-preserving cross-entity data sharing and analytics using the proposed FL based event classification approach.

The rest of this paper is organized as follows. Section II elaborates the key challenges of training an event classifier with decentralized PMU data and provides details of the proposed federated learning framework. Section III describes the real-world PMU data used in this paper and evaluates the performance of proposed event classifiers trained with decentralized PMU data under various settings. Section IV concludes the paper.

II. Federated Learning based Event Classification

A. Problem Statement

This paper studies the federated learning (FL) of PMU data from multiple entities (i.e., data owners) for event classification. Each entity has a collection of raw PMU data and associated event logs. As the raw PMU data are often of low quality (e.g., bad data, dropouts, and timestamp errors), the data quality issues are initially fixed by each entity using the data preprocessing method proposed in our recent work [3].

As each entity may have different numbers of PMUs, the inputs from different entities to the event classification model can be different. This impedes the development of FL based event classification, which requires the dimension of the inputs from different entities to be the same. To address this challenge, we let each entity construct the same set of event features from the PMU data based on [3]. These features capture the patterns of various event types (see [3] for details), and the number of features depends on only the number of measurement types (i.e., the voltage magnitude, the current magnitude, and ROCOF). By using these features, we can not only make the inputs from different entities consistent, but also improve the robustness of the event classifiers and the interpretability of the classification results as shown in [3].

Specifically, assume M entities collaboratively train an event classification model using FL. Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N_m} \sim \mathcal{D}_m$ denote the training dataset of entity m, where $\mathbf{x}_i \in \mathbb{R}^D$ denotes the features of an event and y_i denotes the corresponding event label (i.e., event type from the event logs, e.g., line outage, transformer outage, or frequency event). $\mathcal{D}_m$ denotes the distribution of the training dataset of entity m and N_m denotes the number of training samples (i.e., recorded events) at entity m. D denotes the number of features in each event. The size and the distribution of the training dataset of each entity can be different.

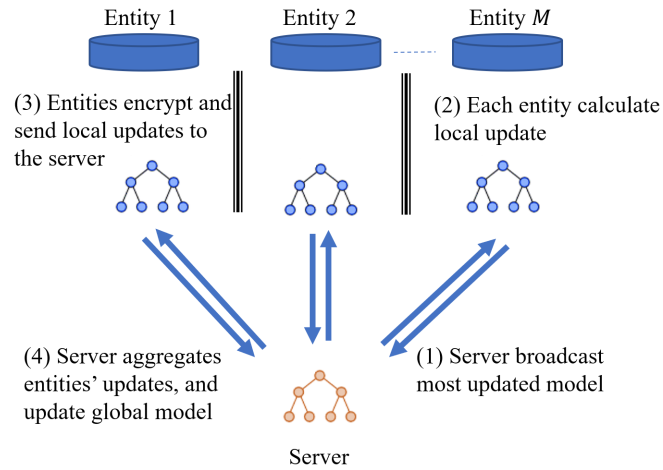

Fig. 1: Illustration of the training process of the federated XGBoost model.

In this paper, we adopt a regression tree-based ensemble model (i.e., XGBoost) from our previous study [3], which shows good classification performance with strong interpretability when it is trained in the centralized case. The objective of FL is to collaboratively learn on PMU data from all the entities while keeping the data decentralized and private, and obtain an event classifier f (i.e., K regression trees with parameters $\mathbf{w}$) that can accurately classify an event type based on the event features. Given the features $\mathbf{x}_i$, the estimated label $\hat{y}_i$ from the federated XGBoost model can be obtained as:

$$\hat{y}_i = f(\mathbf{w}; \mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{w}; \mathbf{x}_i), \tag{1}$$

where $f_k(\cdot)$ denotes the model of the kth regression tree.

We formulate the problem as finding $\mathbf{w}^*$ of a set of regression trees that minimizes the following regularized loss:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) := \tfrac{1}{M} \textstyle\sum_{m=1}^{M} \mathcal{L}_m(\mathbf{w}; \{(\mathbf{x}_i, y_i)\}), \tag{2}$$

where $\mathcal{L}_m(\mathbf{w}; \{(\mathbf{x}_i, y_i)\})$ is the local regularized loss at entity m:

$$\mathcal{L}_m(\mathbf{w}; \{(\mathbf{x}_i, y_i)\}) = \sum_{i=1}^{N_m} l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \tag{3}$$

where $l(\hat{y}_i, y_i)$ denotes a differentiable convex loss function of the classification model f with $\mathbf{w}$ on data point $(\mathbf{x}_i, y_i)$. $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\mathbf{w}\|^2$ is a regularized function that reduces the complexity of a regression tree and prevents overfitting of the model, where γ and λ are regularization coefficients and T denotes the number of leaves in a tree. The training process of the federated XGBoost model is described in Section II-B.

B. Federated XGBoost Model

The training process of the federated XGBoost model is illustrated in Fig. 1 and detailed in Algorithm 1, where we follow the simple and efficient FedSGD framework [14], [20]. In each iteration, the server first broadcasts the current global model, and then each entity updates the current global model using its local data in parallel and sends the encrypted gradient and hessian update to the server for aggregation.

Specifically, the entities collaboratively train the federated XGBoost model by constructing K regression trees. In each iteration, this model trains a regression tree $f_k(\cdot)$ by building it from a single leaf, which enhances the accuracy of the global model (i.e., $f(\cdot)$) by minimizing the loss function (2). The regularized loss considering all K regression trees at the kth iteration can be presented by the following regularized loss function [21]:

$$\mathcal{L}^{(k)}(\mathbf{w};\{(\mathbf{x}_i,y_i)\}) = \sum_{i=1}^{N} l\left(y_i, \hat{y}_i^{(k)} + f_k(\mathbf{x}_i)\right) + \Omega(f_k) \quad (4)$$

where $N = \sum_{m=1}^{M} N_m$. Before reaching the maximum number of trees K or convergence of the loss function, the training process in each iteration performs the following steps:

1) The server broadcasts the current model f to all M entities. The vanilla XGBoost algorithm greedily enumerates all feasible points to find the best split point in each iteration, which is inefficient in our distributed learning case. To efficiently find the best splitting point, we leverage an approximation method [21], which proposes q candidate splitting points $S_d = \left\{s_{d1}, s_{d2}, \dots, s_{dq}\right\}$ based on percentiles of feature d. Using the candidate splitting points, a regression tree is constructed by entities in a distributed manner.
2) Each entity m in parallel calculates local gradient (i.e., $g_i^m = \partial l(y_i,\hat{y}_i^{(k)})/\partial \hat{y}_i^{(k)}$) and hessian (i.e., $h_i^m = \partial^2 l(y_i,\hat{y}_i^{(k)})/\partial \hat{y}_i^{(k)^2}$) of these splitting points S_d for each feature d using its local dataset $\{(\mathbf{x}_i,y_i)\}_{i=1}^{N_m}$.
3) Then each entity m encrypts the local gradient and hessian for each feature d (see lines 17 and 18 in Algorithm 1) by using Paillier cryptosystem (an additive homomorphic cryptosystem method) that for all arbitrary numbers u and v, we have $Enc(u) \cdot Enc(v) = Enc(u+v)$ [19], where $Enc(\cdot)$ denotes encryption operator. $\mathbf{G}_{dv}^m$ and $\mathbf{H}_{dv}^m$ denote the aggregated encrypted gradient and hessian statistics for feature d, respectively. $\mathbf{G}^m$ and $\mathbf{H}^m$ contain the aggregated encrypted gradient and hessian for all the features and are sent back to the server.
4) The server aggregates $\mathbf{G}^m$ and $\mathbf{H}^m$ from the M entities. Thanks to the additive homomorphic cryptosystem, the server can aggregate $\mathbf{G}^m$ and $\mathbf{H}^m$ from all the entities without knowing the local gradient and hessian of each entity, which enhances the privacy of each entity's data. After aggregation of decrypted gradients and hessians (see lines 22 and 27 in Algorithm 1), the server finds the best split points. As a new regression tree is added at each iteration, we need to approximate the new loss function. By using the second-order approximation of $f(\cdot)$ [21], the loss function (4) can be approximated as

$$\mathcal{L}^{(k)}(\mathbf{w};\{(\mathbf{x}_i,y_i)\}) \simeq \sum_{i=1}^{N}\left[l\left(y_i,\hat{y}_i^{(k)}\right) + g_i f_k(\mathbf{x}_i) + \frac{1}{2}h_i f_k^2(\mathbf{x}_i)\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2, \quad (5)$$

Using (5), we can solve the optimal parameter w_j^* of leaf j of the current regression tree as

$$w_j^* = -\frac{\sum_{j\in P_j} g_j}{\sum_{j\in P_j} h_j + \lambda}, \quad (6)$$

Using (6) and (5), we can evaluate the new regression tree under each candidate splitting point by calculating a score (see line 29 in Algorithm 1). The split with the highest score is the optimal split of the current leaf node.

The server repeats steps 1) to 4) to build a regression tree until this regression tree reaches its maximum depth.

Using Algorithm 1, multiple entities can collaboratively learn a federated XGBoost model without leaking any PMU data to each other. Moreover, the proposed method does not need a trusted server, which further enhances the privacy of the federated XGBoost model.

III. Case study of real-world PMU data

A. Experimental Setup

1) Data: This paper uses real-world PMU data from the Western Interconnection of continental U.S. transmission grid. The dataset contains measurements from 23 PMUs of 60 samples per second over a two-year period (2016–2017), as well as the corresponding event logs for the two years, including start timestamp, end timestamp, event type, event cause, and event description for each event. The dataset contains 3,877 events, including 2,507 line events, 921 transformer events, and 449 frequency events. To ensure the events across the entire year are fairly distributed between the training set and the testing set, events after data preprocessing in each month are randomly split into 80% for training and 20% for testing, i.e., the training dataset contains 3098 events and the testing dataset contains 779 events.

To evaluate the performance of the FL based event classification approach, we constructed the dataset of each entity by randomly sampling the dataset of events. As the raw PMU data are often of low quality, the data quality issues are initially fixed by each entity using the data preprocessing method proposed in our recent work [3]. Then, the entities collaboratively train the federated XGBoost model using Algorithm 1, which is implemented under the FATE framework [22]. Then, the trained federated XGBoost model is evaluated using the testing dataset with 779 events.

2) Evaluation Metrics: Four metrics are used to evaluate the classification performance, i.e., accuracy (ACC), precision (PRE), recall (REC), and F1 score, which are defined as follows:

$$\begin{aligned} \text{ACC} &= (\text{TP+TN})/(\text{TP+TN+FP+FN}), \\ \text{PRE} &= \text{TP}/(\text{TP + FP}), \\ \text{REC} &= \text{TP}/(\text{TP + FN}), \\ \text{F1} &= 2\times(\text{PRE}\times\text{REC})/(\text{PRE + REC}), \end{aligned}$$

Algorithm 1 Training Process of Federated XGBoost.

Initialization: The server and entities initialize model configurations (i.e., number of regression trees K, maximum depth of a node in a tree $maxdepth$, number of split points q). $Enc(\cdot)$ denotes encryption operator, and $Dec(\cdot)$ denotes decryption operator.

```
ServerExecute()
for iteration k = 1 to K do
    Broadcasts the current model to all the entities
    //Build a new regression tree k
    while leaf node depth of k is less than maxdepth do
        Add leaf node j
        for m = 1 to M in parallel do
            G^m, H^m ← EntityExecute(m, j)
        end for
        optimal split ← SplitFinding(G, H)
        Broadcast the optimal split.
    end while
end for
EntityExecute(m, j)
Determine instance space P_j of the current leaf node j
for d = 1 to D do
    Propose splitting points S_d on feature d
    Calculate g_j^m and h_j^m of P_j
    G_dv^m = Enc(Σ_{j∈{j|s_{d,v} ≥ x_{j,d} > s_{d,v−1}}} g_j^m)
    H_dv^m = Enc(Σ_{j∈{j|s_{d,v} ≥ x_{j,d} > s_{d,v−1}}} h_j^m)
end for
G^m = {G_dv^m}, H^m = {H_dv^m}
Return G^m and H^m
SplitFinding(G , H) by server
g = Dec(Π_m Π_d Π_v G_dv^m) and Dec(Π_m Π_d Π_v H_dv^m)
score = 0
for d = 1 to D do
    g_l = 0 and h_l = 0
    for v = 1 to q do
        g_l ← g_l + Dec(Π_m G_dv^m), h_l ← h_l + Dec(Π_m H_dv^m)
        g_r ← g − g_l, h_r ← h − h_l
        score ← max( score , g_l^2/(h_l+λ) + g_r^2/(h_r+λ) − g^2/(h+λ) )
    end for
end for
Return the optimal split of the current node feature based
on the maximum score.
```

where TP (i.e., True Positive) and TN (i.e., True Negative) denote the number of positive and negative instances that are correctly classified, respectively. FP (i.e., False Positive) and FN (i.e., False Negative) denote the number of misclassified negative and positive instances, respectively.

3) Benchmark: We compare the federated XGBoost model with centralized benchmarks (the Random Forest (RF) model and the centralized XGBoost model) and the performance of non-collaborative XGBoost training. For the centralized benchmarks, the raw datasets from all the entities are used for training, whereas the federated XGBoost model cannot access the dataset from each entity.

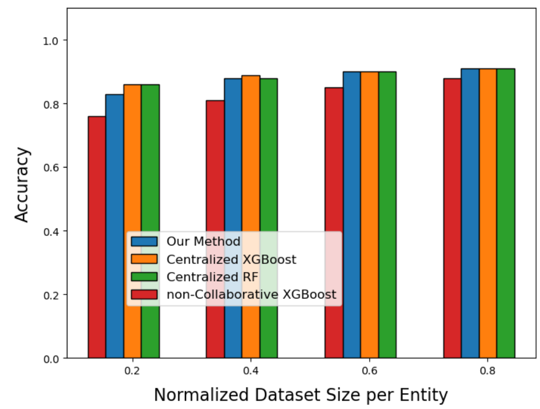

Fig. 2: Performance of the federated XGBoost model under different sizes of entity's dataset. The size of entity's dataset is normalized by the size of the training dataset.

TABLE I
Event classification performance Under different models when each entity has access to 80% of the data.

Model Setting	Accuracy	Precision	Recall	F1-score
Centralized XGBoost	0.91	0.92	0.91	0.91
Centralized RF	0.91	0.91	0.91	0.90
Federated XGBoost	0.91	0.91	0.90	0.90

B. Classification Performance of Federated XGBoost

We evaluate the classification performance of the federated XGBoost model under different settings, in which we fix the number of entities equal to 3 and change the size and distribution of entities' datasets. Each experiment is repeated 10 times, and in each experiment, the datasets of entities are randomly sampled from the training dataset. Then, the trained model is evaluated on the test dataset.

1) IID Data Distribution: We first consider the case that the data distribution of each entity is Independent and Identically Distributed (IID). In this case, when constructing the dataset of each entity, we keep the data distribution of each entity the same as the distribution of the training dataset. We evaluate the performance of the federated XGBoost model under different sizes of entity's dataset, as illustrated in Fig. 2. In Fig. 2, the accuracy of the federated XGBoost model is almost the same as the centralized benchmarks, and the accuracy increases with the size of entity's dataset. The federated XGBoost model outperforms the average performance of non-collaborative individual training where each entity trains the model on its local dataset. Specifically, the accuracy of our method is 83% when each entity has access to only 20% of the training data, and the accuracy increases to 91% when each entity has access to 80% of the data.

Table I compares the event classification performance of federated XGBoost and centralized benchmarks using different metrics, when each entity has access to 80% of data. Fig. 3 shows the confusion matrix of the federated XGBoost model, where the rows represent the estimated event type, the columns represent the true event type, the diagonal and off-diagonal represent the events that are correctly and incorrectly classified, respectively. Clearly, the proposed federated XGBoost model can achieve good performance while keeping the PMU data decentralized and private.

2) non-IID Data Distribution: As the performance of the FL may be impacted by the data distribution of each entity,

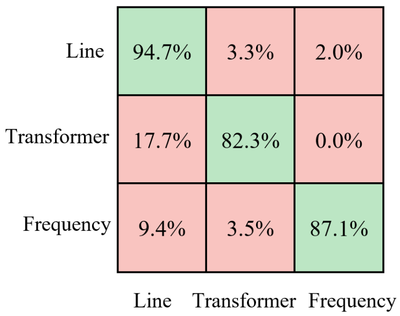

Fig. 3: Confusion matrix of the federated XGBoost model when each entity has access to 80% of the data.

TABLE II
EVENT CLASSIFICATION PERFORMANCE UNDER DIFFERENT DATA DISTRIBUTIONS WHEN EACH ENTITY HAS ACCESS TO 80% OF THE DATA.

Case	#1	#2	#3	IID
KS statistic	0.05	0.15	0.25	0.00
Accuracy	0.90	0.90	0.89	0.91
Precision	0.90	0.89	0.89	0.91
Recall	0.89	0.88	0.89	0.90
F1-score	0.89	0.88	0.89	0.90

we evaluate the performance of the federated XGBoost model when the entities' data distributions are non-IID. Specifically, we fix the size of each entity's dataset and change its distribution when sampling from the training dataset. The data distribution of each entity is different from the distribution of the training dataset, and the distribution difference is quantified using Kolmogorov–Smirnov (KS) statistic [23], which measures the distance between two empirical distributions. The larger the KS statistic is, the larger the distribution difference is. The KS statistic is 0 when the distributions are the same, i.e., the IID case.

Table II shows the performance of the federated XGBoost model under different distributions when each entity has access to 80% of the data. As shown in Table II, the federated XGBoost model can achieve similar performance as the IID case, indicating that it is robust against the non-IID data distribution of entities.

IV. Conclusion

In this paper, we present a FL based event classification approach for building a robust event classifier using decentralized PMU data to improve the awareness of power systems. Our approach has addressed the problems associated with using decentralized real-world PMU data, including different input sizes to the classifier and the privacy of PMU data when collaboratively training the classifier.

One main advantage of the proposed approach is that PMU data owners can train a high-performance event classifier collaboratively while keeping their data decentralized and private. Experiments on real-world data from the Western Interconnection of the U.S. power transmission grid show the proposed federated XGBoost model achieves almost the same performance as its centralized counterparts and is robust against various data distributions of entities. The findings in the paper would shed the light on distributed learning of PMU data and enable privacy-preserving PMU data sharing and analytics among different entities.

Acknowledgment

We are thankful to the Pacific Northwest National Laboratory (PNNL) for providing the data in this study. This work is supported in part by NSF under Grants IIS-1838024, CNS-1950485, and OIA-2148788.

References

[1] 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.

[2] J. Shi, B. Foggo, and N. Yu, "Power system event identification based on deep neural network with information loading," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5622–5632, 2021.

[3] 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*, 2022.

[4] Y. Liu and L. Yang, "Weakly supervised event classification using imperfect real-world pmu data with scarce labels," in *2022 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5, IEEE, 2022.

[5] I. Niazazari, Y. Liu, A. Ghasenikhani, S. Biswas, H. Livani, L. Yang, and V. A. Centeno, "Pmu-data-driven event classification in power transmission grids," in *2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, 2021.

[6] M. MansourLakouraj, M. Gautam, H. Livani, and M. Benidris, "A multi-rate sampling pmu-based event classification in active distribution grids with spectral graph neural network," *Electric Power Systems Research*, vol. 211, p. 108145, 2022.

[7] A. A. Hai, T. Dokic, M. Pavlovski, T. Mohamed, D. Saranovic, M. Alqudah, M. Kezunovic, and Z. Obradovic, "Transfer learning for event detection from pmu measurements with scarce labels," *IEEE Access*, 2021.

[8] A. Thomas, S. Koshy, and R. Sunitha, "Machine learning based detection and classification of power system events," in *2020 International Conference on Power, Instrumentation, Control and Computing (PICC)*, pp. 1–6, IEEE, 2020.

[9] S. Liu, Y. Zhao, Z. Lin, Y. Liu, Y. Ding, L. Yang, and S. Yi, "Data-driven event detection of power systems based on unequal-interval reduction of pmu data and local outlier factor," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1630–1643, 2019.

[10] X. Zheng, B. Wang, D. Kalathil, and L. Xie, "Generative adversarial networks-based synthetic pmu data creation for improved event classification," *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 68–76, 2021.

[11] O. P. Dahal, S. M. Brahma, and H. Cao, "Comprehensive clustering of disturbance events recorded by phasor measurement units," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1390–1397, 2013.

[12] Q. Cui and Y. Weng, "Enhance high impedance fault detection and location accuracy via micro pmus," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 797–809, 2019.

[13] H. Li, Y. Weng, E. Farantatos, and M. Patel, "A hybrid machine learning framework for enhancing pmu-based event identification with limited labels," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, pp. 1–8, IEEE, 2019.

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[15] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *International Conference on Machine Learning*, pp. 12878–12889, PMLR, 2021.

[16] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10174–10183, 2022.

[17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[18] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "Secureboost: A lossless federated learning framework," *IEEE Intelligent Systems*, vol. 36, no. 6, pp. 87–98, 2021.

[19] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*, pp. 223–238, Springer, 1999.

[20] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[21] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[22] "Fate." https://github.com/FederatedAI/FATE.git.

[23] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.