Detection of False Data Injection Attacks in Distribution Networks: A Vertical Federated Learning Approach

Mert Kesici, *Graduate Student Member, IEEE*, Bikash Pal, *Fellow, IEEE*, and Guangya Yang, *Senior Member, IEEE*

***Abstract*—This paper proposes a collaborative learning framework based on vertical federated learning for detecting false data injection attacks in distribution networks. The proposed framework empowers entities that are responsible for a sub-network to collaboratively construct an FDIA detection model, effectively addressing issues associated with data sharing and enabling the utilization of various measurements from each sub-network. The proposed framework enables real-time collaboration between the server and the grid edge-side by allocating the two models created through the split learning approach applied to the proposed attention-based hybrid deep learning model. The grid edge-side is tasked with extracting spatial features, while the server is responsible for extracting temporal features from the data processed by the grid edge-side. The edge-side model is designed by adopting an attention module integrated into a deep learning model while the server-side model is designed based on the Bi-LSTM model. The effectiveness of the proposed framework is demonstrated on the IEEE 123 and IEEE 37 node test systems.**

***Index Terms*—False data injection attack, vertical federated learning, cyber attack detection, state estimation, split learning.**

I. Introduction

THE integration of information and communication technologies has transformed distribution networks into cyber-physical systems. This increased adoption of such technologies in distribution networks has brought about a new concern, the emergence of cyber-attacks [1]. These cyber-attacks pose threats to the availability, integrity or security of the distribution network. Numerous cyber attacks in the energy business have been documented over recent years [2]. These incidents have highlighted the need for a thorough understanding and proper approach to address the cyber-attack risks on the electric power grid. While the form and impacts of cyber attacks exhibit considerable diversity, false data injection attack (FDIA) is a category that presents a formidable challenge since they have widespread consequences and are challenging to detect [3].

The detection techniques for FDIA can primarily be categorized into two groups: model-based [4] and data-driven methods. The former requires precise information about the system model and parameters and slight parameter changes or uncertainties may cause inaccurate detection performance. As opposed to model-based algorithms, data-driven algorithms are model-free and neither the system model nor the system parameters are utilized [2]. The primary focus of this paper lies in data-driven methods.

In [5], derived coefficients from principal component analysis (PCA) and canonical correlation are fed into machine learning classifiers for FDIA detection. In [6], estimated state variables are analysed through wavelet transformation and a deep neural network. In [7], a voting-based ensemble classifier is employed. In [8], a two-level learning framework based on Kalman filter and recurrent neural network is employed. In [9], a PCA-Density-based data-driven framework is proposed. However, all of these studies focused on FDIA detection at the transmission level while research focusing on FDIAs within distribution networks is notably scarce [10]. In [11], a set of machine learning models such as multi-layer perceptions, support vector machine and decision tree, are used as benchmark models for FDIA detection in balanced distribution networks. In [12], a semi-supervised deep learning algorithm utilizing an autoencoder integrated generative adversarial network model is proposed for the detection of FDIA targeting distribution system state estimation (DSSE) function considering an unbalanced three-phase network. In [13], an unsupervised LSTM-based autoencoder model is proposed to detect the FDIA targeting DSSE, leveraging time series measurements to capture spatio-temporal relationships in unbalanced three-phase distribution networks. In [14], a generalized graph Laplacian matrix is used to capture the spatio-temporal features which are then fed to a Bayes classifier for FDIA detection in unbalanced three-phase DSSE.

In the above studies, it is assumed that the measurements are collected and processed in a central control server without any concern regarding the measurement collection from the field. Recently, due to deregulation policies and the growing

Manuscript received 24 October 2023; revised 8 March 2024; accepted 5 May 2024. Date of publication 10 May 2024; date of current version 23 October 2024. This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant under Grant 956433, and in part by the Resilient Operation of Sustainable Energy Systems (ROSES) U.K.-China (EPSRC-NSFC) Programme on Sustainable Energy Supply under Grant EP/T021713/1. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising. Paper no. TSG-01741-2023. *(Corresponding author: Mert Kesici.)*

Mert Kesici and Bikash Pal are with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ London, U.K. (e-mail: m.kesici21@imperial.ac.uk).

Guangya Yang is with the Department of Wind and Energy Systems, Technical University of Denmark, 2800 Kongens Lyngby, Denmark.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2024.3399396.

Digital Object Identifier 10.1109/TSG.2024.3399396

presence of distributed generations, a set of measurements distributed throughout the network might be owned by various entities (i.e., utilities, electricity providers, and system operators). This emerging scenario poses data-sharing challenges, as entities might be unwilling to share their data due to its sensitive, confidential and private nature [15], [16], [17], [18]. Concerns about data-sharing further complicate the development of effective and collaborative FDIA detection models for smart grids.

Horizontal federated learning [19] has been utilized to address the aforementioned concerns by few studies [15], [16], [17], [18] for FDIA detection. Federated learning can be classified into two main categories: horizontal federated learning (HFL) and vertical federated learning (VFL), based on the data partition scheme between participants. HFL necessitates uniformity in data features (sets of measurements) among all participants while the number of samples may vary for each participant. Unlike HFL, VFL allows participants to adopt various features complementary to the same sample. In [15], a cross-device HFL-based framework utilizing a transformer neural network is proposed, where each bus separately processes its own local measurements (injected powers and transmitted powers to an adjacent bus). A generative adversarial model-based cross-silo HFL framework is proposed in [16] where each participant is responsible for a sub-network. The assumption is that the system is divided into sub-networks of equal size to have uniformity in measurements as required by their HFL-based framework, and each sub-network is equipped with an edge processor dedicated to decentralized FDIA detection based on HFL. In [18], a cross-silo HFL framework utilizing an autoencoder model is proposed. In [17], a cross-silo HFL framework with an incentive mechanism is proposed. There are several drawbacks to the aforementioned studies;

- First of all, the HFL-based studies [15], [16], [17], [18] consider FDIA targeting state estimation at the transmission network where abundant measurements ensure the fulfilment of feature uniformity requirements.
- These studies employing an HFL framework assume uniformity of measurement types at each bus [15] and additionally the equity in the number of buses within each sub-network [16], [17], [18]. Uniformity in measurements across all buses in the network cannot be guaranteed. This implies that these proposed methods may become dysfunctional or inefficient if the measurements vary between buses or sub-networks. Furthermore, the assumptions made in these studies are not applicable to distribution networks due to the observability challenges (measurements are not abundant) and their inherent unbalanced multi-phase nature.
- Lastly, it is important to note that these studies [15], [16], [17], [18] solely process measurements at a single time step, overlooking the crucial temporal relationships between consecutive measurement data points. Given that distribution networks exhibit higher fluctuations compared to transmission systems, exploiting this temporal relationship significantly enhances the detection rate of attack samples within distribution networks.

To address the aforementioned drawbacks and mitigate the data-sharing concerns, this paper proposes a novel collaborative learning framework utilizing split learning-based vertical federated learning for FDIA detection in distribution networks. This paper introduces, for the first time, a vertical federated learning framework based on the split learning approach for FDIA detection in distribution networks. Distribution networks are considered three-phase unbalanced networks, consisting of one-, two-, and three-phase nodes with unbalanced loading conditions.

HFL-based frameworks operate in a completely decentralized manner, where each grid edge computing unit independently processes its local data without the need for communication with others, subsequent to a joint training stage. In VFL, unlike HFL, the grid edge computing units continue to cooperate in real-time operation after the training stage, thereby offering deeper insights of the current network status. In VFL, the intermediate representations obtained from the grid-edge computing units and the gradients are exchanged in tandem, enabling the collaborative processing of participants' measurements in real-time while mitigating data-sharing concerns by meticulously refraining from divulging sensitive/private raw measurement data. Real-time collaborative data processing in VFL substantially enhances the detection model's performance. This is achieved by collaboratively processing the diverse and complementary features contributed by the participants.

As a detection model, an attention-based hybrid deep learning model is constructed, facilitating the extraction of spatial-temporal features inherent in the measurement set collected from the distribution network. An attention module namely, convolutional block attention module (CBAM), is adopted within the proposed detection model to enable the model to focus on the important features. The integration of the attention mechanism with a deep learning model, a convolutional neural network, automates the focus on the most significant aspects of the input data. Then, the bidirectional long short-term memory (Bi-LSTM) model which is able to extract temporal features is adopted after the attention-based deep learning model. To integrate the proposed attention-based hybrid deep learning model into the proposed VFL-based framework, the splitting scheme is designed, resulting in two distinct models designated for deployment at the grid edge and server: the grid edge-side model and the server-side model, respectively. The designed splitting scheme empowers the grid edge-side models in extracting spatial features from their respective sub-networks while the server-side model focuses on extracting temporal features from aggregated data derived from the grid edge-side models. The main contributions of this paper are outlined as follows:

1) To the best of the authors' knowledge, this is the first paper proposing a novel collaborative learning framework utilizing split learning-based vertical federated learning for FDIA detection in distribution networks. The proposed framework allows entities to employ various measurements, removing the necessity for uniformity in measurements across entities as required in [15], [16], [17], [18]. This feature is of

particular significance for distribution networks, characterized by their unbalanced multi-phase nature and challenges related to observability (limited number of measurements).

2) The proposed VFL-based framework empowers real-time collaboration between the grid edge-side models and the server-side model, emphasizing the crucial role of collective real-time contributions from all entities. This collaborative effort significantly enhances the effectiveness of the detection mechanism while addressing data-sharing concerns by mitigating the necessity to share raw data. This sets it apart from other frameworks [15], [16], [17], [18] which are only jointly trained and can only process local data using their trained model without further communication with other entities in real-time.
3) An attention-based hybrid deep learning model is proposed for spatial-temporal feature extraction, which is strategically split into the grid edge-side model specializing in spatial feature extraction with the incorporated attention module focusing on the important features and the server-side model dedicated to capturing temporal features, distinguishing from the existing studies [15], [16], [17], [18] which solely process measurements at a single time step.

The following outlines the structure of the remaining paper: Section II presents a preliminary overview. In Section III, a comprehensive explanation of the proposed framework is presented. Section IV entails a thorough analysis of the outcomes and various performance assessments. Section V provides a conclusion and offers final insights.

II. Background

A. State Estimation and Bad Data Detection

The link between state variables and the set of measurements is expressed as $\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e}$ where $\boldsymbol{z}$ represents an M-dimensional measurement vector, and $\boldsymbol{x}$ represents an N-dimensional state vector in conventional state estimation [20]. The function $\boldsymbol{h}(\boldsymbol{x})$ represents the measurement function associated with the state vector $\boldsymbol{x}$. The noise in the measurements, represented by vector $\boldsymbol{e}$, follows a Gaussian distribution characterized by a mean of zero and a covariance matrix $\boldsymbol{R} = \text{diag}\,[\sigma_1^2, \sigma_2^2, \ldots, \sigma_M^2]$. The diagonal elements of the matrix $\boldsymbol{R}$ correspond to the variances of individual measurement errors, symbolized as σ_i^2, where M denotes the overall number of measurements.

The state variables are derived using a Weighted Least Squares (WLS) method [21], aiming to minimize the sum of weighted measurement residuals denoted as J:

$$\hat{\boldsymbol{x}} = \arg\min J = \arg\min[z - \boldsymbol{h}(\boldsymbol{x})]^T \boldsymbol{W}[z - \boldsymbol{h}(\boldsymbol{x})] \tag{1}$$

where $\boldsymbol{W}$ represents a weight matrix assigned to measurements, gauging the confidence levels of various measurements and $\boldsymbol{W} = \boldsymbol{R}^{-1}$.

The Gauss-Newton method is employed for iterative solution of the optimal estimated states until each element of $\Delta\boldsymbol{x}$ in every iteration is suitably minimized:

$$\partial J/\partial \boldsymbol{x} = \boldsymbol{H}(\boldsymbol{x})^T \boldsymbol{W}[z - \boldsymbol{h}(\boldsymbol{x})] = 0 \tag{2}$$

$$\boldsymbol{H}(\boldsymbol{x})^T \boldsymbol{W} \boldsymbol{H}(\boldsymbol{x}) \Delta\boldsymbol{x} = \boldsymbol{H}(\boldsymbol{x})^T \boldsymbol{W}[z - \boldsymbol{h}(\boldsymbol{x})] \tag{3}$$

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \Delta\boldsymbol{x} \tag{4}$$

where $\boldsymbol{H}(\boldsymbol{x}) = \partial \boldsymbol{h}(\boldsymbol{x})/\partial \boldsymbol{x}$ stands for the Jacobian matrix.

Measurements may experience random noise due to communication interference or human error. Such errors can lead to discrepancies between estimated state variables and their actual values. These deviated measurements are referred to as bad data and they can have a detrimental effect on system operation. Hence, it is essential to detect and remove bad data. To identify whether a measurement is erroneous, the ℓ_2-norm detector is widely used by verifying if it satisfies the following criterion:

$$\|\boldsymbol{r}\| = \left\|z - \boldsymbol{h}\left(\hat{\boldsymbol{x}}\right)\right\| \geq \tau, \tag{5}$$

where τ is a pre-defined threshold. Once the threshold is violated, the bad data is detected.

B. DSSE

In an unbalanced three-phase distribution network, the system states consist of the voltage phasors at all nodes are denoted as $\boldsymbol{x} = [v_1^a, v_1^b, \ldots, v_n^c]$. Here, v_k^φ represents the voltage phasor at node k for phase $\varphi = \{a, b, c\}$, where k ranges from 1 to n, with n being the total number of nodes in the system. Denote the voltage at node k as $\boldsymbol{V}_k = [V_k^a, V_k^b, V_k^c]^T = \boldsymbol{v_k}$ and the current at branch $k - l$ as $\boldsymbol{I}_{kl}$. The relationship between the current measurement at branch $k - l$ and the states are represented as:

$$\boldsymbol{I}_{kl} = \left[I_{kl}^a, I_{kl}^b, I_{kl}^c\right]^T = \boldsymbol{Y}_{kl}(\boldsymbol{v}_k - \boldsymbol{v}_l) \tag{6}$$

where the line admittance at branch $k - l$ is denoted as $\boldsymbol{Y}_{kl}$. $\boldsymbol{S}_k$, $\boldsymbol{S}_{kl}$ representing complex power measurements at node k and at branch $k - l$ respectively, are represented through a nonlinear relation involving the states as

$$\begin{aligned} \boldsymbol{S}_k &= \boldsymbol{v}_k(\boldsymbol{I}_k)^* \\ \boldsymbol{S}_{kl} &= \boldsymbol{v}_i(\boldsymbol{I}_{kl})^* \end{aligned} \tag{7}$$

$\boldsymbol{I}_{kl}$ can be calculated using (6), and the current injection at node k can be denoted as $\boldsymbol{I}_k = [I_k^a, I_k^b, I_k^c]^T = \boldsymbol{Y}_k \boldsymbol{v}_k$ where the nodal admittance is denoted as $\boldsymbol{Y}_k$. Then, the DSSE model is represented as

$$z = [\boldsymbol{V}_k, \boldsymbol{I}_{br}, \boldsymbol{S}_{bus}, \boldsymbol{S}_{br}]^T = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e} \tag{8}$$

The DSSE is iteratively solved due to the nonlinear relationship between power measurements and voltages.

Due to the inherent limitations in knowledge of state variables and measurements within distribution networks, linearization of DSSE is considered a necessity [22], which augments computational efficacy throughout the iterative DSSE process. Recently, several methods have been introduced to address this need [20]. In the context of three-phase unbalanced distribution networks discussed in this paper, the DSSE method incorporating a linear approximation, as delineated in [23] is employed as it is found to be closer to

the nonlinear solution derived from (8) in comparison to a simplistic linear solution obtained through representing AC distribution network as a DC model [23]. Moreover, it aligns with the recent studies on FDIA detection in distribution networks [12], [22], [24]. Through this approximate solution, the complex power measurements are converted to the equivalent currents as

$$\begin{aligned} \boldsymbol{I}_{k\_eq} &= \left(\boldsymbol{S}_k/\hat{\boldsymbol{V}}_k\right)^* \\ \boldsymbol{I}_{kl\_eq} &= \left(\boldsymbol{S}_{kl}/\hat{\boldsymbol{V}}_k\right)^* \end{aligned} \tag{9}$$

where $\hat{\boldsymbol{V}}_k$ denotes the estimated voltage. The adopted approach allows for a significant simplification of the DSSE process as the Jacobian matrix, representing the partial derivatives of the estimated variables, remains constant and comprises elements solely from the admittance matrix. Hence, it is computed only once with relatively low computational effort and utilized for all subsequent iterations.

C. FDIA in Distribution Networks

This section presents a method for constructing stealthy (unobservable) FDIA in distribution networks. FDIA construction, as presented in [23] which is based on the adopted DSSE, is achieved without incurring significant computational expenses.

Given the presence of the nearly linear relationship in the utilized DSSE, it is possible to represent the state estimator in a linear form as

$$\tilde{z} = \boldsymbol{H}\tilde{x} + \boldsymbol{e} \tag{10}$$

where $\tilde{z}$ and $\tilde{\boldsymbol{x}}$ denote the measurement and the closed-form estimated vector. With the injection of attack vector $\boldsymbol{a}$, the residual of the compromised measurements $\boldsymbol{r}_a$ can be denoted as [23]

$$\begin{aligned} \boldsymbol{r}_a &= \tilde{z}_a - \boldsymbol{H}\tilde{\boldsymbol{x}} \\ &= \tilde{z} + \boldsymbol{a} - \boldsymbol{H}\left[\hat{\boldsymbol{x}} + \left(\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{a}\right] \end{aligned} \tag{11}$$

where $\tilde{z}_a$ represents the compromised measurements as $\tilde{z}_a = \tilde{z} + \boldsymbol{a}$. If the attack vector is designed as

$$\boldsymbol{a} = \boldsymbol{H}\boldsymbol{c} \tag{12}$$

where $\boldsymbol{c}$ refers to the error injected into the system state by an attacker. Subsequently, following the attack, the compromised measurement residual $\boldsymbol{r}_a$ is identical to the measurement residual $\boldsymbol{r}$ prior to the attack, depicted as follows

$$\boldsymbol{r}_a = \tilde{z} - \boldsymbol{H}\hat{\boldsymbol{x}} + \boldsymbol{H}\boldsymbol{c} - \boldsymbol{H}\left(\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{H}\boldsymbol{c} \tag{13}$$

$$= \tilde{z} - \boldsymbol{H}\hat{\boldsymbol{x}} = \boldsymbol{r} \tag{14}$$

If the residual $\boldsymbol{r}$ manages to elude the residual test, the compromised residual $\boldsymbol{r}_a$ containing malicious data can similarly elude the residual test. Attackers can effectively execute unobservable attacks by the attack model (12). Hence, developing an effective solution to detect unobservable FDIA is vital.

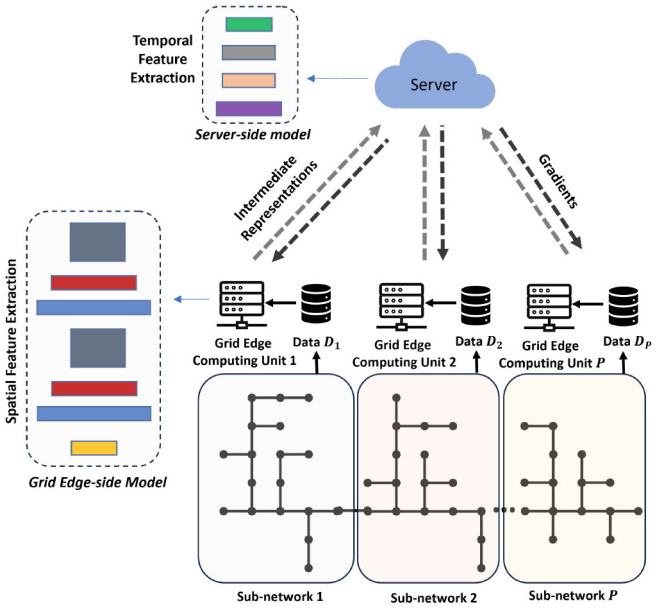

Fig. 1. Proposed Collaborative FDIA Detection Framework based on VFL.

III. Proposed Framework

In this section, the proposed collaborative FDIA detection framework based on vertical federated learning, taking into account of the data-sharing concerns, is presented. Initially, the structure of the proposed framework is outlined, followed by the introduction of the detection mechanism that relies on vertical federated learning. Lastly, the proposed detection model is presented in detail.

A. Architecture of the Proposed FDIA Detection Framework

The proposed framework is based on a cross-silo vertical federated learning framework consisting of two main components: the grid edge computing units and the server as depicted in Fig. 1. In the proposed framework, $p = \{1, \ldots, P\}$ entities are considered to collaboratively train an FDIA detection model $\mathcal{G}_f$ using entities' local datasets $\mathbf{D}_p$ and a server. The model $\mathcal{G}_f$ is split based on the split learning approach [25] into two models: the grid edge-side model and the server-side model, which are assigned to the grid edge computing units and the server respectively as seen in Fig. 1.

Each sub-network (a portion of a distribution network) is assumed to be operated by a different entity [26]. The grid edge-side models are allocated to different sub-networks where the local data is processed by these models. The server-side model orchestrates the conclusive processing stage, leveraging the output from the grid edge-side models to obtain the final output. In VFL, a sample-wise consensus is necessary since complementary features from different participants, belonging to the same sample, need to be processed together.

Each entity that is responsible for a sub-network possesses its respective grid edge-side model $\mathcal{G}_p$ with its respective model parameters θ_p while the server has the ownership of the server-side model $\mathcal{G}_0$ with its respective model parameters θ_0. These two models, the grid edge-side model $\mathcal{G}_p$ and the server-side model $\mathcal{G}_0$ form the complete FDIA detection model

$\mathcal{G}_f = \{\mathcal{G}_p; \mathcal{G}_0\}$. As shown in Fig. 1, the proposed framework consists of the following elements:

1) Grid Edge Computing Units: Each grid edge computing unit holding its respective grid edge-side model $\mathcal{G}_k$ is responsible for a certain part of the distribution network for collaborative FDIA detection. The grid edge computing units collect a set of measurements from local sensors within their respective sub-networks and process them using their own grid edge-side models through forward propagation. The grid edge-side models actively participate in the model update process by exchanging their model outputs and gradients derived from the server-side model. Thus, the edge-side models are active participants in the learning process. To fulfil this, it is assumed that each edge-side model has adequate storage, capability, and computational power.

2) The Server: The server holds the server-side model $\mathcal{G}_0$ and has multiple responsibilities during training and the FDIA detection process in real-time. First, it aggregates the intermediate representations (outputs of the edge-side models) collected from the grid edge computing units. Secondly, it continues forward propagation with the aggregated intermediate representations. Then, it updates its own model (the server-side model) and sends the gradients calculated up to the split layer to each edge-side model to update the client-side models. These operations are performed sequentially during the training stage by the server. After the training stage, the server is also actively involved in real-time to generate the final output, distinguishing itself from HFL. The details about the training stage are given in the next section.

B. VFL Based Detection Mechanism

This paper proposes a novel collaborative FDIA detection framework based on a vertical federated learning framework. The core concept of the proposed framework is that multiple entities collaborate to train an FDIA detection model while addressing the entities' data-sharing concerns without sharing their raw data. A general overview of the proposed framework is given in Fig. 1. The general training procedure of the proposed framework is outlined in nine steps, as illustrated below and additionally, its mechanism is detailed in pseudo-code within Algorithm 1.

1. Initialization: The proposed framework begins by initializing the parameters of the server-side model $\mathcal{G}_0$ at the server and the client-side models $\mathcal{G}_p$ in each grid edge computing unit. The number of epochs E, learning rates η_0 and η_p for the grid edge-side models and the server-side model, loss function L are also initialized. Each iteration commences with the involved parties reaching a consensus on a randomly selected subset $\mathcal{B}^t$ where t represents the iteration number.

2. Local data processing at the edge: After the initialization is carried out, each grid edge computing unit performs forward propagation using its own grid edge-side model $\mathcal{G}_p$ and its own local dataset $\mathbf{D}_p^{\mathcal{B}^t}$ representing the selected mini-batch data at communication round t for entity p. Then, as a result of the forward propagations, the intermediate representations $\boldsymbol{H}_p^t$ are obtained from each grid edge-side model at communication round t which is calculated as follows

$$\boldsymbol{H}_p^t = \mathcal{G}_p\left(\theta_p^t; \mathbf{D}_p^{\mathcal{B}^t}\right) \tag{15}$$

4. Collection of the intermediate representations: After performing forward propagation in each grid edge computing unit, the intermediate representations $\boldsymbol{H}_p^t$ are collected and concatenated at the server as follows

$$\boldsymbol{H}_{all}^t \leftarrow \text{Concat}\left(\boldsymbol{H}_1^t, \ldots, \boldsymbol{H}_p^t\right) \tag{16}$$

6. The server-side model execution: The server continues performing forward propagation with the server-side model $\mathcal{G}_0$ with concatenated intermediate representations $\boldsymbol{H}_{all}^t$ to obtain the server-side model output $\hat{\boldsymbol{y}}^t = \mathcal{G}_0(\theta_0^t; \boldsymbol{H}_{all}^t)$ which is the predicted output. Then, the loss is calculated based on the predicted output $\hat{\boldsymbol{y}}^t$ and the ground truth labels $\boldsymbol{y}^t$ with $\mathcal{L}$ which is chosen as the cross-entropy loss function.

7. The server-side model update: The server-side model parameters are updated as follows

$$g_{\mathcal{G}_0}^t \leftarrow \nabla_{\mathcal{G}_0}\mathcal{L}\left(\hat{\boldsymbol{y}}^t, \boldsymbol{y}^t\right) \tag{17}$$

$$\theta_0^{t+1} = \theta_0^t - \eta_0 g_{\mathcal{G}_0}^t \tag{18}$$

where $g_{\mathcal{G}_0}^t$ is the gradient of the loss function with respect to the server-side model at communication round t and η_s is the learning rate of the server-side model $\mathcal{G}_0$.

Algorithm 1 VFL-Based Collaborative FDIA Detection

Require: Number of communication round T, local data from each grid edge computing unit $\mathbf{D}_p | p \in P$
Ensure: Collaborative FDIA Detection Framework

Initialization:
The server-side model $\mathcal{G}_0$
The edge-side models of P entities $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_p$
Learning rates η_0, η_p for $\mathcal{G}_0$, $\mathcal{G}_p$ respectively
Procedure:
for $t = 1$ to T **do**
  for each mini-batch $\mathcal{B}^t$ **do**
    Grid Edge Computing Units:
    for $p = 1$ to P **in parallel do**
      $\boldsymbol{H}_p^t = \mathcal{G}_p\left(\theta_p^t; \mathbf{D}_p^{\mathcal{B}^t}\right)$
      ▷ Intermediate representations from $\mathcal{G}_p$
    end for
    Send each $\boldsymbol{H}_p^t$ to the server
    The server:
    $\boldsymbol{H}_{all}^t \leftarrow \text{Concatenate}\left(\boldsymbol{H}_1^t, \ldots, \boldsymbol{H}_p^t\right)$
    $\hat{\boldsymbol{y}}^t = \mathcal{G}_0\left(\theta_0^t; \boldsymbol{H}_{all}^t\right)$ ▷ Output of the server-model
    Calculate the loss function $\mathcal{L}(\hat{\boldsymbol{y}}^t, \boldsymbol{y})$
    $g_{\mathcal{G}_0}^t \leftarrow \nabla_{\mathcal{G}_0}\mathcal{L}(\hat{\boldsymbol{y}}^t, \boldsymbol{y}^t)$ ▷ Gradients w.r.t. $\mathcal{G}_0$
    $\theta_0^{t+1} = \theta_0^t - \eta_0 g_{\mathcal{G}_0}^t$ ▷ the server-side model update
    for $p = 1$ to P **do**
      $g_p^t \leftarrow \nabla_{\boldsymbol{H}_p^t}\mathcal{L}(\hat{\boldsymbol{y}}^t, \boldsymbol{y}^t)$ ▷ Gradients w.r.t. $\boldsymbol{H}_p^t$
    end for
    Send g_p^t to each entity p
    Grid Edge Computing Units:
    for $p = 1$ to P **in parallel do**
      $g_{\mathcal{G}_p}^t \leftarrow g_p^t \nabla_{\mathcal{G}_p^t} \boldsymbol{H}_p^t$
      $\theta_p^{t+1} = \theta_p^t - \eta_p g_{\mathcal{G}_p}^t$ the edge-side model update
    end for
  end for
end for

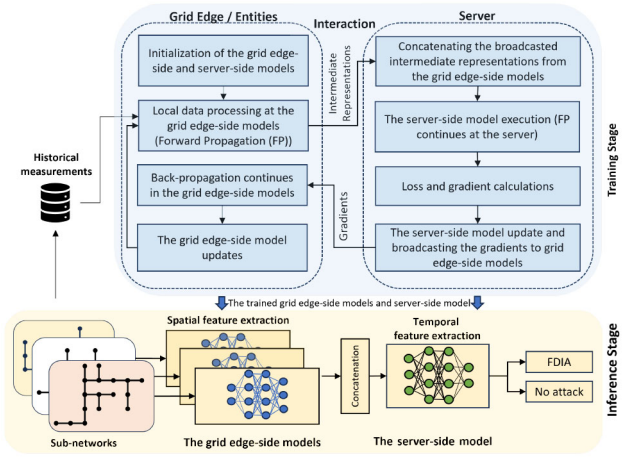

Fig. 2. Implementation stages of the proposed framework.

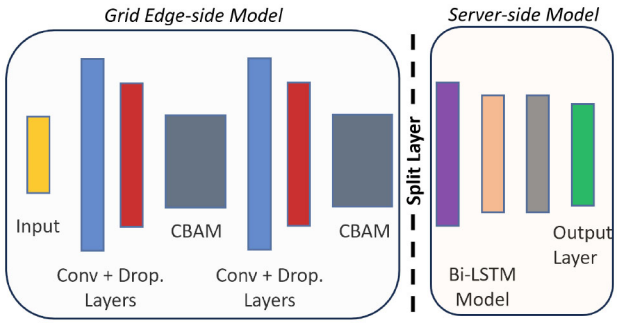

Fig. 3. The proposed attention-based hybrid deep learning model.

8. Broadcasting the parameters: After updating the server model, it proceeds to update each grid edge-side model in parallel. For each grid edge computing unit, the gradient of the loss function with respect to each intermediate representation $g_p^t \leftarrow \nabla_{\boldsymbol{H}_p^t} \mathcal{L}(\hat{\boldsymbol{y}}^t, \boldsymbol{y})$ is calculated and then sent to respective grid edge computing unit to complete the backpropagation.

9. The edge-side model update: Each grid edge computing unit computes the gradients for its local model parameters using its own data and the gradients derived from the intermediate representations provided by the server. Subsequently, the gradients are applied to update their client-side model as follows

$$g_{\mathcal{G}_p}^t \leftarrow g_p^t \nabla_{\mathcal{G}_p} \boldsymbol{H}_p^t \tag{19}$$

$$\theta_p^{t+1} = \theta_p^t - \eta_p g_{\mathcal{G}_p}^t \tag{20}$$

where $g_{\mathcal{G}_p}^t$ is the gradient of the loss function with respect to the grid edge-side model $\mathcal{G}_p$. Following this step, one communication round concludes. The process iterates through step two until achieving a satisfactory performance. Subsequently, training for the collaborative FDIA detection model is accomplished.

In summary, each step contributes to collaborative knowledge sharing while addressing data-sharing concerns, leading to the iterative improvement of the full model's performance over successive rounds of training. After the training, in real-time, the grid-edge models process their local measurements and transmit the intermediate representations to the server-side model, which generates the final output.

C. Implementation of the Split Learning-Based VFL Framework for FDIA Detection

The implementation process of the proposed framework is depicted in Fig. 2 which includes two main stages: the collaborative training and inference stages.

1) Training Stage: The collaborative training that is explained in detail in Section III is performed between the grid edge-side models and the server-side model based on the generated data. The steps to be taken by both the grid edge-side and the server-side models and information exchange during the training stage are shown in Fig. 2. Then, comprehensive grid edge-side models and the server-side model can be obtained after the training, which are utilized in real-time.

2) Inferece Stage: The comprehensive grid edge-side models and the server-side model, deployed at the grid edge-side and the server respectively, perform FDIA detection in collaboration during the inference stage based on the collected measurements from the sub-networks, as depicted in Fig. 2.

For both the training and inference stages, all entities involved in the proposed framework must reach a consensus on a sample-by-sample basis, as complementary features from different entities are utilized for FDIA detection in the proposed framework. This requirement is specific to the split learning-based VFL framework and not required by the HFL-based studies [15], [16], [17], [18]. This requirement is an additional processing stage and a potential challenge compared to the HFL-based studies which conduct training locally and only exchange parameters such as model weights with a server after certain iterations without depending on a sample-wise consensus between participants. It is assumed that the computational capability of each entity is similar which is also a requirement for the HFL-based studies [15].

The intermediate representations and gradients are exchanged between the grid edge-side models and the server-side model in the proposed framework as shown in Fig. 2. As the raw data is not shared, the data-sharing concerns are addressed without violating their privacy.

D. The Proposed Detection Model

In this section, the complete FDIA detection model $\mathcal{G}_f$ is introduced, which is split and deployed on both the grid edge and server sides, as depicted in Fig. 3. The model is designed to facilitate the extraction of both spatial and temporal relationships. The proposed model mainly consists of two convolution layers followed by Convolutional Block Attention Modules (CBAM) [27] for spatial feature extraction. Then, the last CBAM is followed by a Bi-LSTM model to extract the temporal features. In the proposed framework, the proposed model is split after the second attention module, thereby designating the edge-side model $\mathcal{G}_p$ for spatial feature extraction from a set of measurements collected from their respective sub-network and the server-side model $\mathcal{G}_0$ for temporal feature extraction from the aggregated intermediate representations. It is important to note that each intermediate

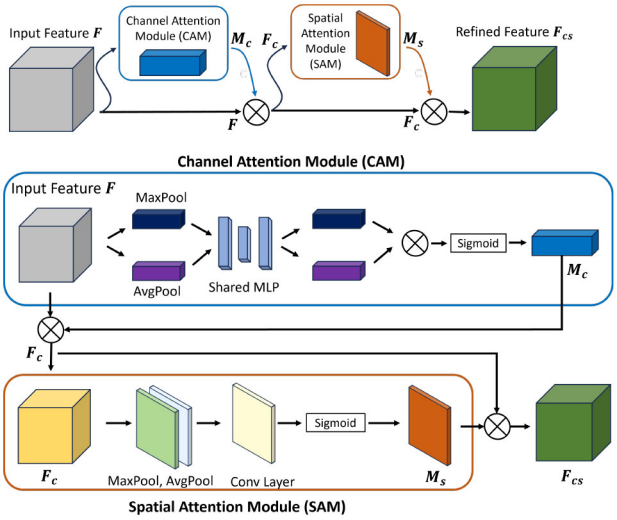

Fig. 4. Convolutional Block Attention Module.

representation $\boldsymbol{H}_p$ represents the spatial features extracted by the grid edge-side model $\mathcal{G}_p$.

1) The Edge-Side Model Extracting the Spatial Features: To extract the hidden spatial features from the input measurements, the proposed attention-based hybrid deep learning model is split after the second attention module, enabling the grid edge-side model $\mathcal{G}_p$ to extract the important features from the channel and spatial dimensions. As depicted in Fig. 3, the proposed edge-side model comprises two convolutional layers, each followed by a pooling layer and CBAM, respectively. Within the convolutional neural networks, all channels are assigned equal importance, potentially resulting in the loss of critical information. Hence, the attention module is adopted after the convolutional layers to enhance the model's ability to prioritize important features within the data acquired by the grid edge-side models. CBAM combines channel and spatial attention mechanisms, allowing the model to highlight important channel and spatial features. This leads to enhanced feature representation and more efficient learning.

The edge-side model $\mathcal{G}_p$ first applies a convolutional layer to its input data which are power injection, power flow and voltage measurements collected over T data points from the sub-network p, resulting in a feature map $\mathbf{F} \in \mathbb{R}^{H\times W\times C}$ where C stands for the number of filters, also called channels which are specified by the convolutional layers while H and W represents the height and width of the input data. CBAM consists of two modules namely the Channel Attention Module (CBAM) and Spatial Attentional Module. Attention weighting operations on the channel and spatial dimensions are performed by CAM and SAM sequentially as depicted in Fig. 4.

In CAM, the spatial dimension is compressed to calculate the channel attention. First, average pooling and maximum pooling operations are carried out simultaneously, resulting in average pooled and maximum pooled features respectively. These pooled features are then utilized by a shared network which is a multi-layer perceptron (MLP) to produce channel attention map $\mathbf{M_c} \in \mathbb{R}^{C\times 1\times 1}$ as follows

$$\mathbf{M_c}(\mathbf{F}) = \sigma\left(MLP(\text{Avg P.}(\mathbf{F})) + MLP(\text{Max P.}(\mathbf{F}))\right) \quad (21)$$

where σ represents the sigmoid function. Subsequently, the resulting channel attention map is multiplied with the input feature map $\mathbf{F}$ as follows

$$\mathbf{F}_c = \mathbf{M_c}(\mathbf{F}) \otimes \mathbf{F} \quad (22)$$

where $\otimes$ represents element-wise multiplication and $\mathbf{F}_c$ represents the feature map processed by CAM, which is used as input to SAM.

In SAM, a spatial attention map is created by leveraging the inter-spatial relationships among features. Contrary to CAM, SAM emphasizes the significance of various parts within the feature maps, acting as a complement to CAM. To calculate spatial attention map $\mathbf{M_s}(\mathbf{F}_c)$, average-pooling and max-pooling operations are employed along the channel axis, and their results are concatenated to create efficient feature descriptors. A convolution layer is then applied to the concatenated feature descriptors as follows

$$\mathbf{M_s}(\mathbf{F}_c) = \sigma\left(f\left(\left[\text{Avg Pool}(\mathbf{F}_c); \text{Max Pool}(\mathbf{F}_c)\right]\right)\right) \quad (23)$$

where f represents a convolution operation. Lastly, the resulting spatial attention map $\mathbf{M}_s$ is multiplied by $\mathbf{F}_c$ to obtained the final refined output by CBAM as follows

$$\mathbf{F}_{cs} = \mathbf{M_s}(\mathbf{F}_c) \otimes \mathbf{F}_c \quad (24)$$

where $\mathbf{F}_{cs}$ represents the final refined output processed by CBAM. Note that in the proposed framework, given the second CBAM module serving as the final layer of the grid edge-side model $\mathcal{G}_p$, $\mathbf{F}_{cs}$ denotes the intermediate representation (defined as $\boldsymbol{H}_k$ in the previous section) transmitted to the server by $\mathcal{G}_p$. The concatenated intermediate representations $\boldsymbol{H}_{all}$ representing the extracted spatial features from each grid edge-side model are processed by the server model $\mathcal{G}_0$, as introduced in the subsequent section.

2) The Server-Side Model Extracting Temporal Information: By splitting the proposed model after the second CBAM module, as depicted in Fig. 3, the server-side model is tasked with extracting temporal features from its input. This input comprises the concatenated intermediate representations $\boldsymbol{H}_{all}$ representing the extracted spatial features. The Bi-LSTM model is adopted for the server-side model to extract temporal relationships due to its proficiency in capturing both forward and backward dependencies thereby enhancing model performance. Additionally, it addresses the common challenges of vanishing gradients during back-propagation and insufficient modelling of backward dependencies, which are prevalent in most recurrent models. Additionally, it resolves the limitation of unidirectional LSTMs, which can only handle dependencies in one direction. The concept of Bi-LSTM is derived from bidirectional RNN [28] which processes sequence data in both forward and backward directions by utilizing two distinct hidden layers.

In the Bi-LSTM architecture, every LSTM block comprises a memory cell, a forget gate, an input gate, and an output gate. Representing the activation vectors of these gates and

the memory cell as $\boldsymbol{f}, \boldsymbol{i}, \boldsymbol{o}$, and $\boldsymbol{c}$, respectively, the nodal connections and tensor operations in a forward LSTM hidden layer are defined as:

$$\boldsymbol{f}_t = \text{Sigmoid}\big(\mathbf{W}_f\big[\boldsymbol{h}_{t-1}, \mathbf{X}_t\big] + \mathbf{b}_f\big) \tag{25}$$

$$\boldsymbol{i}_t = \text{Sigmoid}\big(\mathbf{W}_i\big[\boldsymbol{h}_{t-1}, \mathbf{X}_t\big] + \mathbf{b}_i\big) \tag{26}$$

$$\tilde{\boldsymbol{c}}_t = \tanh\big(\mathbf{W}_c\big[\boldsymbol{h}_{t-1}, \mathbf{X}_t\big] + \mathbf{b}_c\big) \tag{27}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t * \boldsymbol{c}_{t-1} + \boldsymbol{i}_t * \tilde{\boldsymbol{c}}_t \tag{28}$$

$$\boldsymbol{o}_t = \text{Sigmoid}\big(\mathbf{W}_o\big[\boldsymbol{h}_{t-1}, \mathbf{X}_t\big] + \mathbf{b}_o\big) \tag{29}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{c}_t) \tag{30}$$

where $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_c$, and $\mathbf{W}_o$ represent the input weight matrices, while $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_c$, and $\mathbf{b}_o$ correspond to the bias vectors. The functions Sigmoid and tanh denote the logistic sigmoid and hyperbolic tangent activation functions, respectively. The hidden state, denoted as $\boldsymbol{h}$, serves as the output of the LSTM hidden layer, while $\tilde{\mathbf{c}}$ represents the new state candidate vector. The concatenation operator is symbolized by brackets.

It is crucial to consider both forward and backward dependencies in the analysis of time series data. Backward dependencies, obtained from reverse-chronologically ordered data, offer distinctive and valuable insights that cannot be obtained from forward dependencies alone. Consequently, to comprehensively capture both types of dependencies present in the data, a bidirectional LSTM layer is incorporated into the LSTM configuration. The nodal connections and tensor calculations in a BiLSTM closely resemble those of a unidirectional LSTM, differing primarily in processing directions. In particular, operations within a BiLSTM are bidirectional, involving two directions of computation:

$$\overrightarrow{\boldsymbol{f}_t} = \text{Sigmoid}\Big(\overrightarrow{\mathbf{W}}_f\Big[\overrightarrow{\boldsymbol{h}}_{t-1}, \overrightarrow{\mathbf{X}}_t\Big] + \overrightarrow{\mathbf{b}}_f\Big) \tag{31}$$

$$\overleftarrow{\boldsymbol{f}}_t = \text{Sigmoid}\Big(\overleftarrow{\mathbf{W}}_f\Big[\overleftarrow{\boldsymbol{h}}_{t+1}, \overleftarrow{\mathbf{X}}_t\Big] + \overleftarrow{\mathbf{b}}_f\Big) \tag{32}$$

where represent forward and backward operations are represented by symbols $\rightarrow$ and $\leftarrow$, respectively. Two distinct hidden state vectors, $\overrightarrow{\boldsymbol{h}}_t$ and $\overleftarrow{\boldsymbol{h}}_t$, are computed independently and then combined through concatenation to form the final hidden state vector within the BiLSTM layer as follows

$$\boldsymbol{h}_t = \Big[\overrightarrow{\boldsymbol{h}_t}, \overleftarrow{\boldsymbol{h}_t}\Big] \tag{33}$$

The server-side model consists of dense layers following the Bi-LSTM model, as shown in Fig. 3. Consequently, the decision for FDIA detection is obtained within the server-side model, in collaboration with the grid-edge side models.

IV. Results and Discussion

In this section, a comprehensive set of experiments is conducted to evaluate the effectiveness of the proposed framework on the IEEE 123 node and IEEE 37 node test systems [29] which are depicted with locations of voltage and power measurements which are based on [22] in Fig. 5. Power injections are also monitored for each load and distributed generation nodes. In the IEEE 123 node test system, certain switches along the branches are typically in a closed state, like branches 13-152, 18-135, 60-160, and 97-197. To simplify the representation, each pair of nodes linked by these branches is merged [22], [30]. Moreover, six distributed generations are added as in [31]. The IEEE 123 node test system is partitioned into six sub-networks, taking inspiration from [26] and the IEEE 37 node test system is partitioned into two sub-networks, which are depicted in Fig. 5.

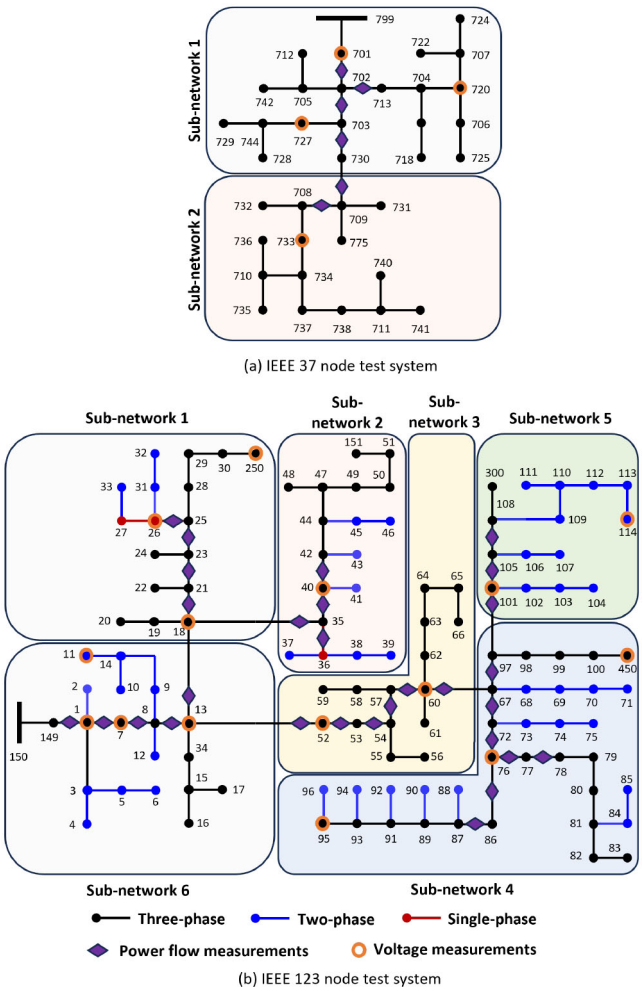

Fig. 5. The distribution networks. (a) IEEE 37 node test system. (b) IEEE 123 node test system.

A. Dataset Generation

Load demand and generation values for a month duration are adopted from PJM interconnection [32] for the generation of normal measurements as [33]. For compromised data generation, stealthy FDIA explained in Section II-C. is considered. Various FDIA strengths are employed, classified into weak, medium, and strong attacks as detailed in [6]. The attack strengths with a set of attack templates ramping-up, ramping-down and random [14] are utilized and implemented over a series of consecutive time steps denoted by T which also serve as the input length to the proposed model. FDIA is applied to either a single or multiple target nodes, wherein one or both of their state variables are modified. Throughout these FDIA scenarios, various attack strengths are considered, both across different types of attacks and for different samples. Compromised samples are generated by applying FDIA to

TABLE I
CONFUSION MATRIX FOR FDIA DETECTION

Samples	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

nodes' state variables within each entity, where each entity is responsible for a sub-network. The input length T is chosen to be 12, indicating that FDIA is applied over 12 consecutive measurement data points. Thereby, 12 consecutive measurement data points from each measurement type are combined to form a sample. Subsequently, these created samples serve as input for the grid edge-side models, consisting of the available measurements collected from the respective sub-networks over 12 consecutive data points.

In total, 16671 samples were generated, consisting of 8475 normal measurements and 8196 compromised measurements for the IEEE 123 node test system and 9984 samples were generated, consisting of 5000 normal measurements and 4984 compromised measurements for the IEEE 37 node test system. The distribution of the generated measurements to respective entities is determined based on the location of the measurement devices within the sub-networks. Consequently, $\mathbf{D}_p$ is generated for each entity p. For the following analyses, a total of 5 entities are taken into account, consolidating sub-network-1 and sub-network-2 into a singular sub-network.

B. Evaluation Metrics

The FDIA detection is designed as a binary classification task. Positive class samples correspond to compromised samples, whilst negative class samples correspond to normal samples. The confusion matrix given in Table I is created to assess the effectiveness of the proposed detection model.

As evaluation metrics, accuracy (ACC), f1-score, precision and recall are calculated based on the confusion matrix as follows:

$$\begin{aligned} ACC &= (TP + TN)/(TP + TN + FP + FN) \\ precision &= TP/(TP + FP) \\ recall &= TP/(TP + FN) \\ F1 &= 2 \cdot (precision \cdot recall)/(precision + recall) \end{aligned} \quad (34)$$

ACC represents the overall detection accuracy of normal and compromised samples while f1-score stands for the harmonic mean of precision and recall.

C. Results

In this section, various robustness tests and comparative studies are performed to evaluate the proposed model's performance under different conditions. All experiments are performed using a Google Colab Pro Notebook with the NVIDIA A100-SXM4-40GB GPU.

1) Comparison With the State-of-the-Art Methods: The proposed attention-based hybrid deep learning model is compared with the state-of-the-art methods which are selected as CNN, LSTM and Bi-LSTM models. The comparison is carried out under the proposed VFL-based framework. The results are given in Fig. 6 for both test systems.

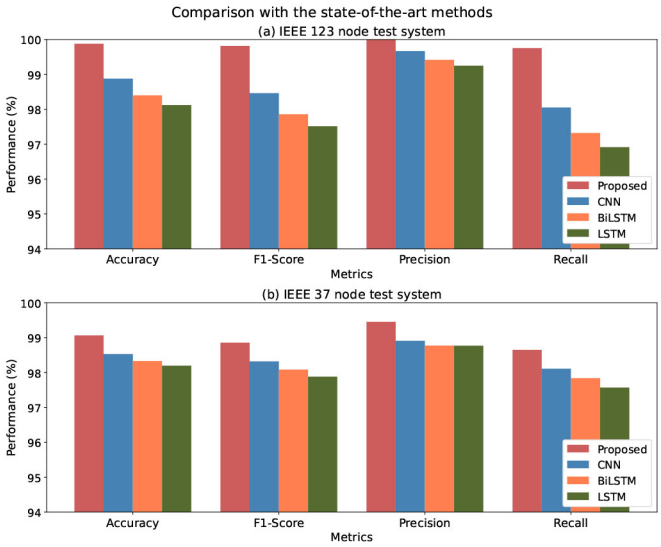

Fig. 6. Comparison with the state-of-the-art methods.

Fig. 6 shows that the proposed approach surpasses the state-of-the-art methods across various evaluation metrics by achieving 99.88% accuracy, 99.82% f1-score, 100% precision and 99.75% recall for the IEEE 123 node test system and 99.06% accuracy, 98.85% f1-score, 99.45% precision and 98.65% recall for the IEEE 37 node test system. In both test systems, LSTM exhibits the poorest performance, achieving accuracy, f1-score, precision, and recall of 98.12%, 97.51%, 99.25%, and 96.91%, respectively for the IEEE 123 node test system and achieving 98.19% accuracy, 97.88% f1-score, 98.77% precision and 97.57% recall for the IEEE 37 node test system. Likewise, the performance of Bi-LSTM and CNN is comparatively inferior to the proposed method. These methods focus solely on extracting either temporal or spatial features from the measurements. In contrast, the proposed attention-based model effectively extracts both spatial and temporal features, resulting in the best detection performance for both test systems.

2) Comparison of Frameworks: The performance of the proposed model is compared in two settings: a centralized framework and the proposed VLF-based framework. The centralized framework represents the case where the measurements from different entities are directly sent to the server, overlooking the data-sharing concerns. The results are given for both test systems in Fig. 7.

The comparison depicted in Fig. 7 illustrates that, for the IEEE 123 node test system, in contrast to the centralized FDIA detection model performing 99.94% accuracy, 99.93% f1-score, 100% precision, and 99.87% recall, the proposed VFL-based framework achieves 99.88% accuracy, 99.81% f1-score, 100% precision, and 99.75% recall. For the IEEE 37 node test system, the proposed framework performs approximately 99% in all metrics while the centralized framework shows a slightly higher performance. This shows a marginal performance trade-off for enhanced privacy preservation in

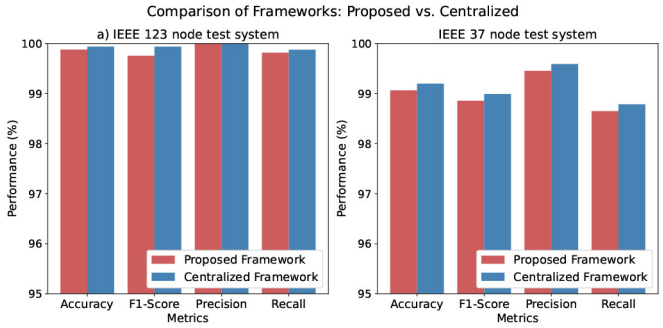

Fig. 7. Performance comparison of different frameworks.

TABLE II
THE NETWORK PARTITION DETAILS

Partition No	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	$\mathcal{E}$	$\mathcal{F}$
2	$1,2,6$	$3,4,5$	-	-	-	-
3	$1,6$	2	$3,4,5$	-	-	-
4	$1,6$	2	3	$4,5$	-	-
5	$1,6$	2	3	4	5	-
6	1	2	3	4	5	6

TABLE III
EFFECT OF DIFFERENT NUMBER OF ENTITIES

Entity No	ACC [%]	F1 [%]	Precision [%]	Recall [%]
2	99.70	99.69	100	99.38
3	99.96	99.95	100	99.91
4	99.84	99.83	100	99.67
5	99.88	99.82	100	99.76
6	99.84	99.83	100	99.67

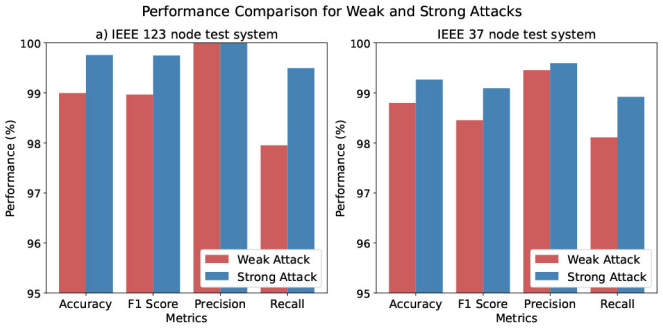

Fig. 8. Performance of the proposed model on weak and strong FDIA.

the proposed framework. As elaborated in Section III, the proposed framework orchestrates collaborative training of grid edge-side models and the server-side model without the necessity to divulge local measurements, achieved through the VFL-based collaborative detection framework by protecting sensitive measurements. Conversely, the centralized method mandates uploading all raw measurements to the cloud centre for FDIA detection model training. While the centralized approach attains slightly better detection performance, it compromises the data privacy of the involved entities. The proposed framework offers comparable performance to the centralized FDIA detection model while addressing the data-sharing concerns for both test systems.

3) Effect of Number of Entities: Six different network partition schemes are utilized to assess the performance of the proposed framework across various numbers of entities, each with a distinct set of features. Network partition schemes are created by combining the sub-networks depicted in Fig. 5. In each network partition configuration, the sub-networks are amalgamated to construct the corresponding partition schemes outlined in Table II.

In Table II, each partition number from two to six signifies the total number of entities taken into account for the collaborative FDIA detection. For instance, in partition scheme number 2, two entities, namely entity-$\mathcal{A}$ and entity-$\mathcal{B}$, emerge through the amalgamation of sub-networks $(1,2,3)$ and $(4,5,6)$ respectively. In partition number scheme number 6, a total of six entities denoted by $\mathcal{A}$ to $\mathcal{F}$ are established, with each entity managing a dedicated sub-network numbered from one to six. For each network partition scheme, varying numbers of entities ranging from two to six are taken into consideration and integrated into the proposed collaborative FDIA detection framework. The corresponding results are given in Table III.

From Table III, it is observed that the proposed framework exhibits the best performance when the number of entities is three, achieving 99.70% accuracy, 99.95% f1-score, 100% precision and 99.91% recall values. Nonetheless, a marginal decrease in accuracy, f1-score, and recall is observed with variations in the number of entities. Therefore, it is concluded that the proposed framework consistently achieves high performance across all evaluation metrics as the number of entities varies from two to six while effectively addressing the data-sharing concerns of varying numbers of entities.

4) Performance Comparison With Different Attack Strengths: The performance of the proposed framework is assessed across various attack strengths. To achieve this, two distinct datasets are created: one comprising solely weak attack samples and the other containing strong attack samples. These datasets are exclusively utilized for evaluating the model's performance and are not used during the training stage. The results are given for both test systems in Fig. 8.

Under weak attack conditions, the proposed framework achieves an accuracy of 99.75%, precision of 100%, recall of 99.49%, and f1-score of 99.74% for the IEE 123 node test system and accuracy of 98.79%, precision of 99.45%, recall of 98.11%, and f1-score of 98.45% for the IEE 37 node test system. Conversely, under strong attack conditions, the proposed framework exhibits 98.99% accuracy, 100% precision, 97.95% recall, and 98.96% f1-score for the IEEE 123 node test system and 99.26% accuracy, 99.59% precision, 98.92% recall, and 99.09% f1-score for the IEEE 37 node test system. While the proposed framework may not detect weak FDIA as accurately as strong FDIA for both test systems, it is important to note that weak FDIA typically doesn't significantly impact network operation. On the other hand, the proposed method demonstrates effective detection of strong FDIA due to its ability to capture substantial changes in measurements using the proposed attention-based hybrid deep learning model. Furthermore, it successfully addresses data-sharing concerns, reinforcing its efficacy.

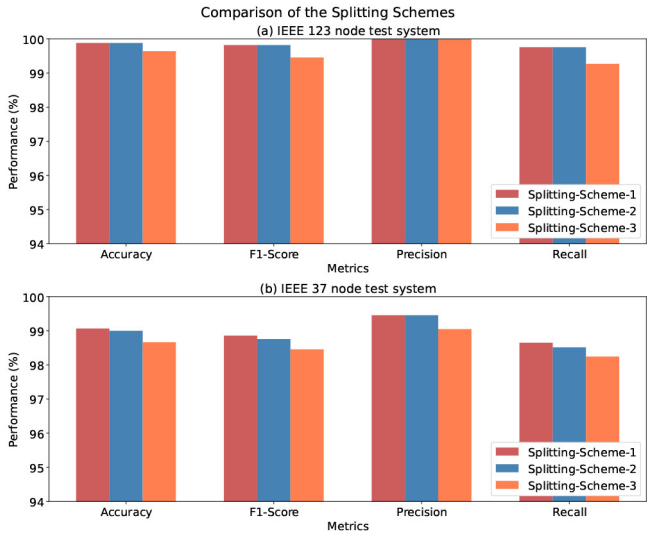

Fig. 9. Performance comparison of different splitting schemes.

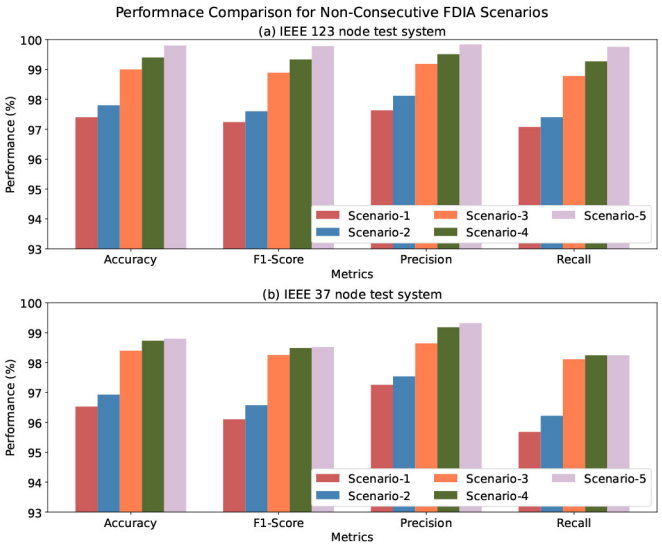

Fig. 10. Performance comparison of non-consecutive FDIA scenarios.

5) Effect of Split Layer Location: The impact of split layer location on the proposed framework's performance is assessed by evaluating three distinct split schemes. These splitting schemes are created by varying the location of the split layer within the attention-based hybrid deep learning model. Split scheme-1 represents the proposed model which is used throughout the paper, where the model is split after the second CBAM layer. In split scheme-2, the proposed model is split after the first CBAM layer, allocating the first convolutional and CBAM layers to the grid edge-side model, and the remaining portion to the server-side model. In split scheme-3, the proposed model is split after the Bi-LSTM model, with only the flatten and dense layers retained in the server-side model. The results are given for both test systems in Fig. 9.

From Fig. 9, it is seen that for IEEE 123 node test system, splitting scheme-1 and splitting scheme-2 show the same performance, achieving 99.88% accuracy, 99.82% f1-score, 100% precision and 99.75% recall. Conversely, splitting scheme-3 demonstrates a slightly lower performance, achieving 99.64% accuracy, 99.45% f1-score, 100% precision and 99.26% recall. For the IEEE 37 node test system, splitting scheme-1 and splitting scheme-2 demonstrate closely comparable performance, achieving approximately 99% in all metrics while splitting scheme-3 performs a slightly lower performance. The performance loss in splitting scheme-3 can be attributed to the fact that the aggregated data is not further processed by the server-side model using any spatial or temporal feature extractors compared to splitting scheme-1 and 2.

6) Effect of Non-Consecutive FDIA: In this section, the performance of the proposed model against non-consecutive FDIA scenarios where FDIA is applied randomly at non-consecutive intervals is tested. Five different scenarios are considered representing different number of time steps that FDIA is implemented intermittently. The number of time steps is chosen as $\{2, 4, 6, 8, 10\}$ for each scenario from 1 to 5 respectively, meaning that for the first scenario, 2 random data points are selected in a non-consecutive way. The results are given in Fig. 10 for both test systems.

From Fig. 10, it is observed that the model's performance decreases by approximately 2.5% for both test systems for the scenario-1 where FDIA is only applied to two non-consecutive data points. The model demonstrates performance of approximately 97% and 96% in all metrics even under scenario-1 and scenario-2 for the IEEE 123 node and IEEE 37 node test systems respectively. However, the impact of these worst-case scenarios (scenario-1 and scenario-2), where only a few samples are under attack in a non-consecutive manner, on the system's performance is comparatively less than that of other scenarios, as their effects last for a shorter total duration. The performance of the model increases to over 99% and 98% for the IEEE 123 node and IEEE 37 node test systems respectively when the non-consecutive data points are more than half the size of the input length as the variation throughout the observation window increases.

TABLE IV
TRAINING TIME OF THE MODEL FOR DIFFERENT INPUT LENGTHS

Test System/Input Length	4	8	12	16
IEEE 123 node	2210s	4584s	6156s	9762s
IEEE 37 node	838s	1650s	2314s	3972s

7) Effect of Input Length: The performance of the proposed framework is evaluated against different input (observation) lengths (T) which are selected as $\{4, 8, 12, 16\}$. The results are given in Fig. 11 for both test systems. The training time of the model for each input length is also given in Table IV for both test systems.

From Fig. 11, it is observed that the model's performance decreases by approximately 1% when the input length is set to 4 in both test systems. This decrease can be attributed to the loss of temporal information compared to longer input lengths. There is only a marginal performance increase when the input length is extended from 8 to 12 which is the input length utilized throughout the paper. Furthermore, there

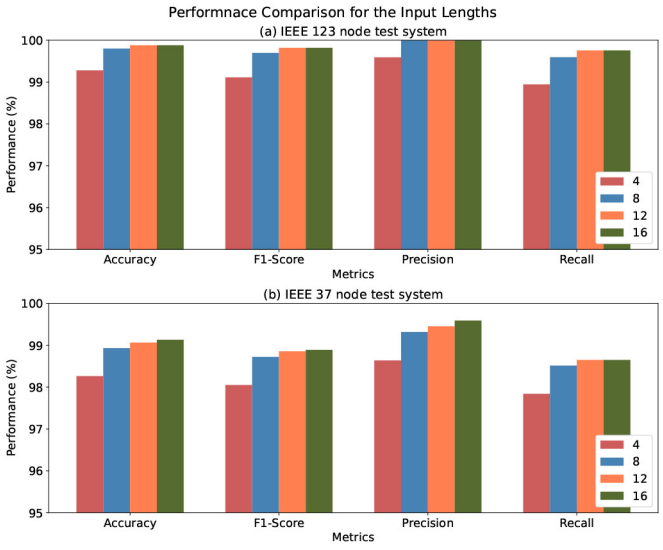

Fig. 11. Performance comparison of the different input lengths.

is no significant difference in performance when the input length is set as 12 or 16 for both test systems. On the other hand, extending the input length leads to increased training time due to increasing computational requirements as seen in Table IV. However, selecting the input length of 12 could be advantageous, as it requires less training time compared to 16, while still offering comparable performance.

V. Conclusion

In this paper, for the purpose of FDIA detection in distribution networks, a vertical federated learning based framework is proposed. The proposed framework effectively addresses data-sharing concerns among entities (system operators, data owners etc.) who may hesitate to share their raw measurements. To the best of the authors' knowledge, this work stands as the first study to employ vertical federated learning in the detection of FDIA within smart grids. Moreover, an attention-based hybrid deep learning model is developed to extract spatial and temporal relationships from the measurements. The proposed model is split into two models: the grid edge-side and the server-side models, strategically allocated to the grid edge computing units and the central server, respectively. The results on the IEEE 123 and IEEE 37 node test systems validate the superior performance of the proposed model when compared to several benchmark models and under various robustness tests. For future studies, adversarial attacks, FDIA requiring incomplete information and asynchronous aggregation techniques will be investigated.

References

[1] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 645–658, Dec. 2011.

[2] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2218–2234, May 2020.

[3] V. Y. Pillitteri and T. L. Brewer, "Guidelines for smart grid cybersecurity," U.S. Dept. Commer., Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. 7628, 2014.

[4] H. Long, Z. Wu, C. Fang, W. Gu, X. Wei, and H. Zhan, "Cyber-attack detection strategy based on distribution system state estimation," *J. Modern Power Syst. Clean Energy*, vol. 8, no. 4, pp. 669–678, Jul. 2020.

[5] A. S. Musleh, G. Chen, Z. Y. Dong, C. Wang, and S. Chen, "Online characterization and detection of false data injection attacks in wide-area monitoring systems," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2549–2562, Jul. 2022.

[6] J. J. Q. Yu, Y. Hou, and V. O. K. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3271–3280, Jul. 2018.

[7] H. Goyel and K. S. Swarup, "Data integrity attack detection using ensemble-based learning for cyber–physical power systems," *IEEE Trans. Smart Grid*, vol. 14, no. 2, pp. 1198–1209, Mar. 2023.

[8] Y. Wang, Z. Zhang, J. Ma, and Q. Jin, "KFRNN: An effective false data injection attack detection in smart grid based on Kalman filter and recurrent neural network," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6893–6904, May 2022.

[9] A. Parizad and C. J. Hatziadoniu, "Cyber-attack detection using principal component analysis and noisy clustering algorithms: A collaborative machine learning-based framework," *IEEE Trans. Smart Grid*, vol. 13, no. 6, pp. 4848–4861, Nov. 2022.

[10] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1630–1638, Jul. 2017.

[11] M. Mohammadpourfard, Y. Weng, and M. Tajdinian, "Benchmark of machine learning algorithms on capturing future distribution network anomalies," *IET Gener., Transm. Distrib.*, vol. 13, no. 8, pp. 1441–1455, 2019.

[12] Y. Zhang, J. Wang, and B. Chen, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 623–634, Jan. 2021.

[13] A. S. Musleh, G. Chen, Z. Y. Dong, C. Wang, and S. Chen, "Spatio-temporal data-driven detection of false data injection attacks in power distribution systems," *Int. J. Elect. Power Energy Syst.*, vol. 145, Feb. 2023, Art. no. 108612.

[14] M. Cui, J. Wang, and B. Chen, "Flexible machine learning-based cyberattack detection using spatiotemporal patterns for distribution systems," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1805–1808, Mar. 2020.

[15] Y. Li, X. Wei, Y. Li, Z. Dong, and M. Shahidehpour, "Detection of false data injection attacks in smart grid: A secure federated deep learning approach," *IEEE Trans. Smart Grid*, vol. 13, no. 6, pp. 4862–4872, Nov. 2022.

[16] M. Abdel-Basset, N. Moustafa, and H. Hawash, "Privacy-preserved generative network for trustworthy anomaly detection in smart grids: A federated semisupervised approach," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 995–1005, Jan. 2023.

[17] W.-T. Lin, G. Chen, and Y. Huang, "Incentive edge-based federated learning for false data injection attack detection on power grid state estimation: A novel mechanism design approach," *Appl. Energy*, vol. 314, May 2022, Art. no. 118828.

[18] M. A. Husnoo et al., "FedDiSC: A computation-efficient federated learning framework for power systems disturbance and cyber attack discrimination," *Energy AI*, vol. 14, Oct. 2023, Art. no. 100271.

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

[20] K. Dehghanpour, Z. Wang, J. Wang, Y. Yuan, and F. Bu, "A survey on state estimation techniques and challenges in smart distribution systems," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2312–2322, Mar. 2019.

[21] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. Boca Raton, FL, USA: CRC press, 2004.

[22] C. Ma, H. Liang, and Y. Jing, "A novel ZSV-based detection scheme for FDIAs in multiphase power distribution systems," *IEEE Trans. Smart Grid*, vol. 14, no. 2, pp. 1236–1248, Mar. 2023.

[23] P. Zhuang, R. Deng, and H. Liang, "False data injection attacks against state estimation in multiphase and unbalanced smart distribution systems," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6000–6013, Nov. 2019.

[24] Y. Zhang, J. Wang, and J. Liu, "Attack identification and correction for PMU GPS spoofing in unbalanced distribution systems," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 762–773, Jan. 2020.

[25] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *J. Netw. Comput. Appl.*, vol. 116, pp. 1–8, Aug. 2018.

[26] W. Zheng, W. Wu, B. Zhang, H. Sun, and Y. Liu, "A fully distributed reactive power optimization and control method for active distribution networks," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 1021–1033, Mar. 2016.
[27] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
[28] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[29] K. P. Schneider et al., "Analytic considerations and design basis for the IEEE distribution test feeders," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3181–3188, May 2018.
[30] C. Muscas, S. Sulis, A. Angioni, F. Ponci, and A. Monti, "Impact of different uncertainty sources on a three-phase state estimator for distribution networks," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 9, pp. 2200–2209, Sep. 2014.
[31] Y. Zhang, J. Wang, and Z. Li, "Interval state estimation with uncertainty of distributed generation and line parameters in unbalanced distribution systems," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 762–772, Jan. 2020.
[32] "Data Miner 2." Jul. 2022. [Online]. Available: https://dataminer2.pjm.com/list
[33] S. Wei, J. Xu, Z. Wu, Q. Hu, and X. Yu, "A false data injection attack detection strategy for unbalanced distribution networks state estimation," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 3992–4006, Sep. 2023.

Mert Kesici (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from Kocaeli University, Kocaeli, Turkey, in 2016, and the M.Sc. degree in electrical engineering from Istanbul Technical University, Istanbul, in 2019. He is currently pursuing the Ph.D. degree with Imperial College London, where he is currently a Maria Skłodowska-Curie Early-Stage Researcher. His research interests include the cyber security of smart grids and the application of privacy-preserving machine learning algorithms to smart grids.

Bikash Pal (Fellow, IEEE) received the B.E.E. degree (Hons.) in electrical engineering from Jadavpur University, Kolkata, India, in 1990, the M.E. degree in electrical engineering from the Indian Institute of Science, Bengaluru, India, in 1992, and the Ph.D. degree in electrical engineering from Imperial College London, London, U.K., in 1999, where he is currently a Professor with the Department of Electrical and Electronic Engineering. His current research interests include renewable energy modeling and control, state estimation, and power system dynamics. He was the Editor-in-Chief of IEEE TRANSACTIONS ON SUSTAINABLE ENERGY from 2012 to 2017, and *IET Generation, Transmission and Distribution* from 2005 to 2012. He was the Vice President of Publications of the IEEE Power and Energy Society from 2019 to 2023.

Guangya Yang (Senior Member, IEEE) received the Ph.D. degree from The University of Queensland, Australia, in 2008. He joined the Technical University of Denmark. From 2020 to 2021, he was a Full-Time Specialist in electrical design, control, and protection of large offshore wind farms with Ørsted. He is currently an Associate Professor with the Technical University of Denmark. His current research interests include the stability and protection of converter-based power systems, with an emphasis on offshore wind applications. He is the Convener of IEC61400-21-5 on configuration, functional specification, and validation of hardware-in-the-loop test bench for wind power plants. In addition, he is also the Coordinator of the H2020 Marie Curie Innovative Training Network Project Innovative Tools for Cyber-Physical Energy Systems and the Lead Editor of the Power and Energy Society Section in IEEE ACCESS.