This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

Locational Detection of False Data Injection Attacks in Smart Grids: A Cloud-Edge Framework Based on Split Learning

Mert Kesici, *Graduate Student Member, IEEE*, Malek Abduhaymi, *Graduate Student Member, IEEE*, Anton Ishchenko, Guangya Yang, *Senior Member, IEEE*, Bikash C. Pal, *Fellow, IEEE*

***Abstract*—False data injection attacks (FDIA) pose significant threats to the security of distribution networks, jeopardizing the integrity of measurements and the accuracy of decision-making processes. Furthermore, data-sharing concerns present a substantial obstacle to the development of a comprehensive FDIA localization framework. To address these challenges, this paper proposes a cloud-edge framework based on split learning for the localization of FDIA in distribution networks. The proposed framework consists of two primary components: the grid edge computing units at the edge, which includes the local feature extractor and the feature classifier, and the cloud where the global feature extractor is located. The local feature extractors in the grid edge computing units at the nodes are tasked with extracting local features from node measurements. The cloud is responsible for deriving global/spatial features from the local features extracted by each grid edge computing unit distributed throughout the network. Lastly, each feature classifier orchestrates the conclusive processing stage to make decisions at the grid edge based on the spatial features extracted by the cloud. Attention layers namely the self- and the spatial attentions, are utilized to focus on the most important features for the local feature extractors and the global feature extractor, respectively. Data-sharing concerns are mitigated as both the raw measurements and the labels are kept at the grid edge. Comparative studies indicate that the proposed framework maintains strong resilience to varying attack strengths and effectively handles noisy measurements.**

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

I. Introduction

THE growing demand for reliable energy, coupled with rapid technological advancements, has driven the evolution of smart grids. Smart grids are poised to enhance the existing capabilities of generation, transmission, and distribution systems, providing a robust infrastructure capable of addressing future challenges such as integrating distributed generation, renewable energy sources, electric vehicles, and demand-side management. Technological innovations such as phasor measurement units, wide-area measurement systems, substation automation, and advanced metering infrastructure form the backbone of these advancements, enabling precise monitoring, control, and optimization of grid operations. However, their integration also significantly relies on cyber resources, amplifying the grid's exposure to potential cyber threats [1].

The interaction points between the cyber and physical layers inherently pose potential vulnerabilities to cyber-attacks. Consequently, extensive research has been devoted to developing detection mechanisms to safeguard the various critical tools of smart grids. For instance, tools leveraging wide-area measurements have been identified as potential targets for cyber-attacks in [2]–[5]. Among various cyber-attacks, false data injection attacks are regarded as exceptionally severe. FDIAs have the potential to undermine the integrity of the system states by altering the system measurements in a way that bypasses the conventional bad data detection mechanisms [6]. As a result, the system states diverge from their actual values, leading to misguided decisions by operators and potentially disrupting network operations. Consequently, prompt detection of FDIA is essential to preserving the network's operational integrity.

FDIA detection has been explored through various approaches detailed in [7]. In practical scenarios, beyond merely detecting the presence of an attack, it is crucial to pinpoint its exact location to swiftly implement effective countermeasures. Therefore, this paper primarily focuses on data-driven methods as data-driven methods do not require system models or parameters. Data-driven methods for FDIA localization are primarily explored through training a machine learning model to function as a multi-label classifier, assigning each label to its corresponding sensor in the system [8]–[11]. Consequently, the classifier delivers a decision/label for each sensor. A multi-label classification approach based on a convolutional neural networks (CNN) model in [8], a CNN-Bidirectional Long Short-Term Memory Network (BiLSTM) model in [9], a fast-lightweight CNN model with an early exiting mechanism in [10] and a lightweight ensemble model consisting of CNN, squeeze neural networks and BiLSTM in [11] are proposed for FDIA localization where the system measurements are classified into the labels corresponding to each sensor in the system. In order to utilize the graph structure of the power grid, graph neural networks in [12] and graph convolutional networks in [13] are utilized based on multi-label classification for FDIA localization. In [14], an adaptive and hierarchical FDIA detection and localization framework is proposed for distribution networks.

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant 956433, in part by the Resilient Operation of Sustainable Energy Systems (ROSES) U.K.-China (EPSRC-NSFC) Programme on Sustainable Energy Supply under Grants 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.

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

In the aforementioned studies, the measurements are centrally collected and processed without addressing any challenges regarding data-sharing. However, recent trends such as deregulation policies and the increasing integration of distributed generation have led to a case where measurements are distributed across the network and owned by various data-owners. A data-owner could be a system operator, with the data being measurements related to assets such as transformers [15], and switchgear [16]. Alternatively, a data-owner could be a wind farm operator or owner, with the data being measurements related to turbines [17] or wind power generation [18]. This situation brings forth data-sharing challenges, as data-owners may be hesitant to share their data due to its sensitive, confidential, and private nature. These concerns further complicate the design of efficient and collaborative FDIA detection models for smart grids [19]–[26]. In these studies, data-owners are identified as independent system operators. Accordingly, this terminology is consistently adopted throughout this paper.

A. Related Work

Identifying the exact location of FDIA is essential to ensure timely and effective countermeasures, as failure to localize FDIA can lead to widespread disruptions and system inefficiencies. Furthermore, data-sharing concerns, stemming from privacy issues, hinder the collaborative efforts required to build comprehensive FDIA localization frameworks. These challenges motivate the development of an innovative approach that not only ensures accurate FDIA localization but also addresses the constraints imposed by data-sharing concerns, paving the way for a more secure and collaborative smart grid.

Several studies leverage federated learning (FL) [27] to mitigate data-sharing concerns in FDIA detection, employing both cross-silo [22]–[24] and cross-device [19], [20] approaches. In cross-silo approaches, grid edge computing units are allocated to sub-networks, whereas in cross-device approaches, grid edge computing units are assigned to individual buses. In [22], a generative adversarial network model is assigned to each sub-network where the presence of FDIA is detected for the sub-network level. In [23], a cross-silo approach with an incentive mechanism is proposed, where the local models are assigned to power system data state owners, which are collaboratively trained through the FL framework considering the privacy concerns between the stakeholders. In [24], a cross-silo FL framework is proposed based on a deep autoencoder model detecting the FDIA presence for each sub-network based on the reconstruction error. These studies in [22]–[24] operate under the assumption that the system is divided into sub-networks of equal size, thereby ensuring measurement uniformity across sub-networks or buses as required by the FL framework, which limits the application to the distribution networks due to the observability challenges and their multi-phase nature. Moreover, these studies [22]–[24] are limited to detecting the presence of FDIA for sub-networks and cannot identify the exact locations of the FDIAs.

In [19], a local model predicting the measurement matrix that is used to detect the presence of FDIA in each bus is collaboratively trained using a cross-device FL framework in transmission networks. In [20], a cross-device FL framework is proposed where each bus is equipped with a transformer model classifying the system measurements as normal or malicious for each bus in a transmission network. A similar approach is adopted in [21], where each PMU is equipped with an LSTM-based model, trained within the FL framework, to classify its measurements as either normal or malicious. Even though the FDIA location is identified, the data uniformity requirement (each bus assumed to have the same type of measurements) among buses remains an unsolved challenge in [19], [20].

The unrealistic assumptions about the data uniformity in each bus or sub-network are mitigated by [25], [26]. In [25], the subgrid-level models for extracting the spatial-temporal features are assigned to each sub-grid and a supervising model is assigned to the cloud where the decision is obtained as a result of multi-label classification. The subgrid-level models can deal with data uniformity in each sub-grid as their outputs are concatenated and processed by the supervising model in the cloud. Even though data-sharing concerns are mitigated by separately training the subgrid-level networks and the supervising model, this approach overlooks the benefits of collaborative information-sharing during training. In [26], a collaborative learning framework utilizing vanilla split learning is proposed for FDIA detection to tackle the data-uniformity challenge and data-sharing concerns among sub-networks. Grid edge-side models extracting the spatial features are assigned to sub-networks and a server-side model extracting the temporal features is assigned to the cloud where the presence of FDIA is detected for the entire network. These models are collaboratively trained in contrast to [25]. While the collaborative training process ensures that raw data is not exchanged, the decision-making is centralized, as the final decision is made at the cloud in [26]. This architecture in [26] requires the exchange of label information during the training and inference stages, which can raise potential privacy concerns. Moreover, even though the assumption of data uniformity is addressed by [25] and [26] which detect the presence of FDIA for the sub-networks [25] and the entire network [26], the specific locations of the attacks are not identified. This omission is significant, as pinpointing the exact locations of FDIAs is crucial for implementing effective countermeasures.

B. Research Gaps and Motivation

Based on the aforementioned studies, there are two main gaps in the literature: the exact locations of FDIAs are not adequately investigated and the data-uniformity assumptions are not addressed for FDIA localization in smart grids while addressing the data-sharing concerns. To address these gaps, as outlined in Table I, this paper proposes a cloud-edge framework based on split learning in a U-shape configuration for FDIA localization in distribution networks.

The scope of the paper is to develop a framework for the localization of FDIAs targeting the state estimator, in a collaborative manner, while addressing data-sharing concerns by ensuring that raw data remains at the edge within distribution networks. This includes designing a framework

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

TABLE I
COMPARISON OF FDIA LOCALIZATION STUDIES
A: DECISION LOCATION, B: FRAMEWORK TYPE, C: PRIVACY-PRESERVING, D: NO NEED FOR DATA CONSISTENCY AMONG NODES/BUSES, E: COLLABORATIVE TRAINING, F: COLLABORATIVE INFERENCE (DECISION-MAKING)

Ref.	A	B	C	D	E	F
[8]	Cloud	Centralized Learning	✗	✓	✗	✗
[9]	Cloud	Centralized Learning	✗	✓	✗	✗
[11]	Cloud	Centralized Learning	✗	✓	✗	✗
[12]	Cloud	Centralized Learning	✗	✗	✗	✗
[13]	Cloud	Centralized Learning	✗	✓	✗	✗
[20]	Edge	Federated Learning	✓	✗	✓	✗
[21]	Edge	Federated Learning	✓	✗	✓	✗
This Paper	Edge	U-shape Split Learning	✓	✓	✓	✓

based on split learning in a U-shape configuration, where grid-edge models are responsible for local feature extraction and decision-making, and cloud models handle global feature extraction. Unlike federated learning, which deploys the full model at the edge and restricts the server's role to aggregating model weights, the proposed framework splits the model into distinct parts and assigns unique roles to the grid edge (the local feature extractors and feature classifiers) and the cloud (the global feature extractor). This division of tasks establishes a U-shape communication pattern between the grid edge and the cloud, enabling collaboration during both training and inference stages, unlike federated learning, where collaboration is confined to the training phase.

C. Contributions

The proposed framework is designed to pinpoint the exact locations of FDIAs at the grid edge, unlike the studies based on multi-label classification where the decision is taken at the cloud. Additionally, it eliminates the need for label sharing, as required in [20], thereby enhancing privacy preservation. The main contributions of this paper are outlined as follows:

1) To the authors' best knowledge, this is the first work to propose a cloud-edge framework that utilizes split learning in a U-shape configuration for localizing FDIA in distribution networks. In the proposed framework, the grid edge is tasked with extracting local features and orchestrating the conclusive processing stage to generate the output at the grid edge. Simultaneously, the cloud is responsible for deriving global features. Moreover, the proposed framework addresses the data-uniformity challenge in contrast to the studies based on FL.
2) The proposed framework facilitates continuous collaboration between the grid edge and the cloud throughout both the training and inference stages, setting it apart from [25] supporting collaboration only during the inference stage and the FL-based frameworks in [19], [20] restricting collaboration to the training stage. This continuous collaboration capability underscores the comprehensive integration of functionalities and information between the grid edge and the cloud.
3) Data-sharing concerns are mitigated as raw data is not broadcasted. Moreover, privacy is further safeguarded by retaining the labels at the grid edge, unlike the approaches in [25], [26] necessitating label sharing.
4) The proposed model is designed with distinct components for the grid edge and the cloud. The grid edge and cloud segments employ the self-attention mechanism and spatial-attention module, respectively, to focus on the most critical features locally at the grid edge and to emphasize spatial relationships across the entire network at the cloud. The final component of the model is dedicated to processing the extracted global features, at the grid edge where the decision is obtained for each node.

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. Distribution System State Estimation and FDIA

A three-phase distribution network comprising n nodes and m measurement devices is considered, where $m \geq n$. Let the state variables be represented as $\boldsymbol{x} \in \mathbb{C}^{3n\times 1}$ and the measurements as $\boldsymbol{z} \in \mathbb{C}^{3m\times 1}$. The distribution system state estimation (DSSE) can be described by the following nonlinear model [28]:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{h}(\cdot)$ encompasses the measurement functions, while $\boldsymbol{e} \in \mathbb{C}^{3m\times 1}$ represents the measurement noise following a Gaussian distribution characterized by a mean of zero and a covariance matrix $\boldsymbol{R}$ whose diagonal elements correspond to the variances of individual measurement errors.

Weighted Least Squares (WLS) method is utilized to calculate the state variables, with the objective of minimizing the sum of the weighted measurement residuals, denoted as J:

$$\hat{\boldsymbol{x}} = \arg\min J = \arg\min[\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]^T \boldsymbol{W}[\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] \tag{2}$$

where $\boldsymbol{W} = \boldsymbol{R}^{-1}$ serves as a weight matrix for measurements, evaluating the reliability of measurements. The Gauss-Newton method is utilized to iteratively refine the optimal state estimates, continuing until each component of $\Delta\boldsymbol{x}$ is adequately minimized in every iteration:

$$\partial J/\partial \boldsymbol{x} = \boldsymbol{H}(\boldsymbol{x})^T \boldsymbol{W}[\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] = 0 \tag{3}$$

$$\boldsymbol{H}(\boldsymbol{x})^T \boldsymbol{W} \boldsymbol{H}(\boldsymbol{x}) \Delta\boldsymbol{x} = \boldsymbol{H}(\boldsymbol{x})^T \boldsymbol{W}[\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] \tag{4}$$

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \Delta\boldsymbol{x} \tag{5}$$

where the Jacobian matrix is denoted as $\boldsymbol{H}(\boldsymbol{x})$.

A prevalent approach for identifying erroneous data is the residual-based bad data detector (BDD). The residual vector $\boldsymbol{r}$ is computed as the difference between the actual measurements z and the estimated measurements $\hat{\boldsymbol{z}} \triangleq \boldsymbol{h}(\hat{\boldsymbol{x}})$.

$$r = \boldsymbol{z} - \hat{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}}) \tag{6}$$

While the BDD is effective in identifying erroneous data, it fails to detect stealthy FDIAs [6]. Attackers can inject false data into the system measurements to deceive the system

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

operator. The system measurements under FDIA are denoted as $z_a = z + \boldsymbol{a}$, where $\boldsymbol{a}$ is the attack vector manipulating the original measurement vector z. Stealthy FDIAs are carefully designed leading to manipulated state variables $\hat{\boldsymbol{x}}_a$, which can be expressed as $\hat{\boldsymbol{x}}_a = \hat{\boldsymbol{x}} + \boldsymbol{c}$, where $\hat{\boldsymbol{x}}$ is the estimated state variables in the absence of FDIA, and $\boldsymbol{c}$ represents the deviations in state variables due to the attack vector $\boldsymbol{a}$. Specifically, for the general non-linear DSSE model described in (1), the attack vector $\boldsymbol{a}$ is formulated to meet the condition

$$\boldsymbol{a} = \boldsymbol{h}(\hat{\boldsymbol{x}}_a) - \boldsymbol{h}(\hat{\boldsymbol{x}}) \tag{7}$$

According to this design, and referring to (2), the residual vector during an attack is computed as

$$\boldsymbol{r}_a = \boldsymbol{z}_a - \hat{\boldsymbol{z}}_a = \boldsymbol{z} + \boldsymbol{a} - \boldsymbol{h}(\hat{\boldsymbol{x}}_a) = \boldsymbol{z} - \boldsymbol{h}(\hat{\boldsymbol{x}}) = \boldsymbol{r} \tag{8}$$

This results in the residual vector being identical to the one observed when no attack is present, thus allowing the attack to evade detection by the BDD.

The DSSE approach integrating a linear approximation, as detailed in [29], is preferred due to its closer alignment with the nonlinear solution derived from (1) compared to a basic linear solution based on a DC model [29]. This method also significantly reduces computational demands. The DSSE model incorporating the linear approximation can be expressed as

$$\widetilde{\boldsymbol{z}} = \boldsymbol{H}\widetilde{\boldsymbol{x}} + \boldsymbol{e} \tag{9}$$

where $\widetilde{\boldsymbol{z}}$ and $\tilde{\boldsymbol{x}}$ represent the measurement vector and the estimated vector in closed form, respectively. When the attack vector $\boldsymbol{a}$ is introduced, the residual of the tampered measurements $\boldsymbol{r}_a$ can be described as [29]

$$\begin{aligned}\boldsymbol{r}_a &= \tilde{\boldsymbol{z}}_a - \boldsymbol{H}\tilde{\boldsymbol{x}} \\ &= \widetilde{\boldsymbol{z}} + \boldsymbol{a} - \boldsymbol{H}\left[\widehat{\boldsymbol{x}} + \left(\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{a}\right]\end{aligned} \tag{10}$$

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

$$\boldsymbol{a} = \boldsymbol{H}\boldsymbol{c} \tag{11}$$

consequently, after the attack, the residual of the tampered measurements $\boldsymbol{r}_a$ matches exactly with the measurement residual $\boldsymbol{r}$ observed before the attack.

If the residual $\boldsymbol{r}$ successfully passes the residual test, then the compromised residual $\boldsymbol{r}_a$ can similarly avoid BDD. Attackers can efficiently carry out stealthy attacks using the attack model in (11). Therefore, it is crucial to develop effective methods to detect such undetectable FDIAs.

III. Proposed Framework

In this section, first, the architecture of the proposed framework for the localization of FDIA is presented, followed by the training mechanism and the implementation details. Lastly, the proposed detection model is presented in detail.

A. Architecture of the Proposed Framework

The proposed framework consists of two main parts: the grid edge computing units and the cloud as depicted in Fig. 1. Nodes at the grid edge are equipped with so-called grid edge computing units. The cloud is utilized as a collaborative component of the proposed framework as illustrated in Fig. 1.

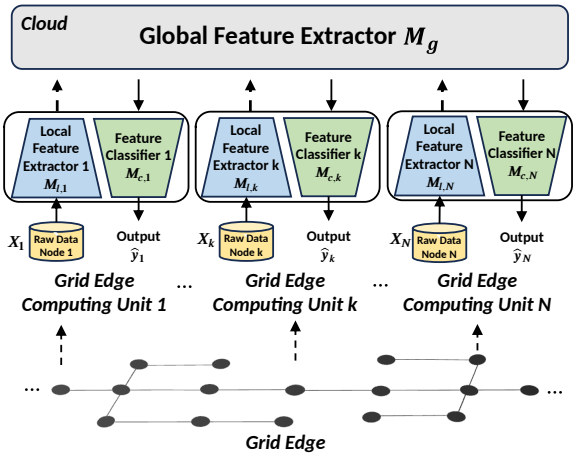


Fig. 1. Architecture of the proposed framework

In the proposed framework, $n = \{1, \ldots, N\}$ nodes are considered to collaboratively train the proposed FDIA localization models $\mathcal{M}_n$ using the nodes' raw measurements $\mathbf{X}_n$ at each node. The proposed framework is built based on the split learning [30] in U-shape configuration. Thus, the FDIA localization models $\mathcal{M}_n$ to be deployed at the grid edge and the cloud is split into three distinct components: the local feature extractor $\mathcal{M}_{l,n}$, the global feature extractor $\mathcal{M}_g$, and the feature classifier $\mathcal{M}_{c,n}$ as depicted in Fig. 1. The local feature extractors and the feature classifiers are allocated to the grid edge computing units while the global feature extractor is allocated to the cloud. Each node possesses its respective local feature extractor $\mathcal{M}_l$ and feature classifier $\mathcal{M}_c$ with their respective model parameters θ_l, θ_c respectively. The proposed framework comprises N nodes, each equipped with a grid edge computing unit that includes one local feature extractor $\mathcal{M}_l$ and one feature classifier $\mathcal{M}_c$. Additionally, there is a single global feature extractor located in the cloud. In this paper, local features represent the information extracted from the node measurements while the global features represent the spatial relationships between the nodes' measurements distributed throughout the network.

During both the training and inference phases, information flows from the grid edge to the cloud and back to the grid edge, forming a U-shaped communication pattern. In the forward propagation stage, the grid edge processes the input data using the first segment of the model, after which intermediate representations are transmitted to the cloud for further computations. The results are then sent back to the grid edge, where the final segment of the model completes the processing. During backward propagation, the gradients follow the reverse path, facilitating collaborative training while ensuring data privacy. This U-shape configuration enables seamless cooperation between the grid edge and the cloud, ensuring that raw data and labels remain securely at the grid edge while leveraging the computational resources and capabilities of the cloud. The U-shape configuration enables the cloud to function as an aggregator, facilitating privacy-preserving data processing while simultaneously leveraging the grid edge as the decision-maker, ensuring localized decision-making,

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

different from [26], where the decision-making is centralized in the cloud. An overview of each component and its responsibilities during both the training and inference stages is provided below:

1) Local Feature Extractor at Grid Edge Computing Units: The first part of the grid edge computing unit is the local feature extractor $\mathcal{M}_l$ as shown in Fig. 1. During the training stage, each $\mathcal{M}_l$ is responsible for processing the local raw measurements, transmitting the obtained intermediate features to the cloud for further processing. Each $\mathcal{M}_l$ is updated based on the received gradients from the cloud through the backpropagation process. During the inference stage, it is only responsible for extracting and broadcasting the local features to the cloud. The details about the training and inference stages are given in Section III-B and Section III-C, respectively.

2) Global Feature Extractor at Cloud: The global feature extractor $\mathcal{M}_g$ is allocated to the cloud and responsible for extracting the global features. $\mathcal{M}_g$ receives the extracted local features from each $\mathcal{M}_l$ as its input. During the training and inference stages, it is responsible for extracting global features and transmitting them to the feature classifiers at the grid edge. Moreover, it is also responsible for calculation and broadcasting the gradients required to update the proposed detection models through the backpropagation. In the inference stage, the output of the $\mathcal{M}_g$ is sent to each feature classifier located in the grid edge computing units where the decision is taken. $\mathcal{M}_g$ only receives the intermediate features and gradients with respect to its output, thereby addressing data-sharing concerns.

3) Feature Classifier at Grid Edge Computing Units: The feature classifier $\mathcal{M}_c$ is the second and final component of the grid edge computing unit, responsible for orchestrating the conclusive processing stage and leveraging the output from the $\mathcal{M}_g$ to generate the output at each node. During the training, each $\mathcal{M}_c$ is updated via backpropagation and calculates the gradients and sends them back to the cloud to accomplish the backpropagation process. During the inference stage, each $\mathcal{M}_c$ processes the extracted global features received from the $\mathcal{M}_g$.

Data-sharing concerns are mitigated since the raw measurements and the labels are not exchanged; instead, the raw measurements and labels are solely processed by each $\mathcal{M}_l$ and $\mathcal{M}_c$ respectively, at each node. Overall, each $\mathcal{M}_c$ processes the extracted global/spatial relationships between the extracted local features from the nodes' measurements and decides whether its respective node's measurements are compromised.

B. Training Mechanisim of the Proposed Framework

The fundamental concept of the proposed framework is that each grid edge computing unit consisting of $\mathcal{M}_l$ and $\mathcal{M}_c$ collaborates with the $\mathcal{M}_g$ at the cloud to collaboratively train the FDIA localization models at each node, effectively addressing data-sharing concerns by not sharing raw data and labels. The overall training procedure of the proposed framework based on split learning in U-shape configuration is described in 9 steps, as shown below. Furthermore, its mechanism is explained in the pseudo-code provided in Algorithm 1.

Step 1. Initializing the models: The proposed framework begins by initializing the $\mathcal{M}_g$ at the cloud and each $\mathcal{M}_l$ and $\mathcal{M}_c$ in each grid edge computing unit with their parameters θ_g, θ_l, θ_c respectively. The number of epochs E, learning rates η_l, η_g, η_c for $\mathcal{M}_l$, $\mathcal{M}_g$, $\mathcal{M}_c$ respectively and loss function $\mathcal{L}$ for each $\mathcal{M}_c$ are also initialized. Each iteration begins with a randomly selected subset ξ^t, where t denotes the iteration number.

Step 2. Local feature processing at the grid edge: Upon completion of the initialization, each grid edge computing unit executes forward propagation using its designated local feature extractor $\mathcal{M}_l$ and the raw measurements $\mathbf{X}_n^{\xi^t}$ representing the selected mini-batch data for node n at training round t. This process yields the intermediate features $\boldsymbol{Z}_n^t$ from each local feature extractor during training round t, calculated as follows:

$$\boldsymbol{Z}_n^t = \mathcal{M}_{l,n}\left(\theta_{l,n}^t; \mathbf{X}_n^{\xi^t}\right) \tag{12}$$

Step 3. Broadcasting and aggregating of intermediate features: After executing forward propagation with each $\mathcal{M}_l$ in each grid edge computing unit, the intermediate features $\boldsymbol{Z}_n^t$ are broadcasted to the cloud and concatenated as follows:

$$\boldsymbol{Z}_{con}^t \leftarrow \text{Concat}\left(\boldsymbol{Z}_1^t, \ldots, \boldsymbol{Z}_n^t\right) \tag{13}$$

Step 4. Global feature processing at the cloud: The global feature extractor $\mathcal{M}_g$ continues performing forward propagation with the concatenated intermediate features $\boldsymbol{Z}_{con}^t$ and obtains the intermediate features of the global feature extractor as follows:

$$\boldsymbol{Z}_{global}^t = \mathcal{M}_g\left(\theta_g^t; \boldsymbol{Z}_{con}^t\right) \tag{14}$$

The $\boldsymbol{Z}_{global}^t$ is then transmitted to each feature classifier $\mathcal{M}_c$ in the grid edge computing units.

Step 5. Executing and updating the feature classifiers at the grid edge: Each $\mathcal{M}_c$ processes the output of the global feature extractor $\boldsymbol{Z}_{global}^t$ received from $\mathcal{M}_g$ to obtain the prediction result as follows:

$$\hat{\boldsymbol{y}}_n^t = \mathcal{M}_{c,n}\left(\theta_{c,n}^t; \boldsymbol{Z}_{global}^t\right) \tag{15}$$

Subsequently, the loss is calculated using the predicted output $\hat{\boldsymbol{y}}_n^t$ and the ground truth labels $\boldsymbol{y}_n^t$, with $\mathcal{L}_n$ chosen as the cross-entropy loss function for feature classifier $\mathcal{M}_{c,n}$ at node n. Backpropagation is then initiated, and each $\mathcal{M}_c$ are updated as follows:

$$g_{\mathcal{M}_{c,n}}^t \leftarrow \nabla_{\mathcal{M}_{c,n}} \mathcal{L}_n(\hat{\boldsymbol{y}}_n^t, \boldsymbol{y}_n^t) \tag{16}$$

$$\theta_{c,n}^{t+1} = \theta_{c,n}^t - \eta_c g_{\mathcal{M}_{c,n}}^t \tag{17}$$

where $g_{\mathcal{M}_{c,n}}^t$ is the gradient of the loss function $\mathcal{L}_n$ with respect to the feature classifier $\mathcal{M}_{c,n}$ at training round t and η_c is the learning rate of $\mathcal{M}_{c,n}$.

Step 6. Broadcasting the parameters from the grid edge to the cloud: Each $\mathcal{M}_c$ subsequently computes the gradient of the loss with respect to the intermediate output of the global feature extractor $\boldsymbol{Z}_{global}^t$ as $\nabla_{\boldsymbol{Z}_{global}^t} \mathcal{L}_n(\hat{\boldsymbol{y}}_n^t, \boldsymbol{y}_n)$. These gradients from each $\mathcal{M}_c$ are then transmitted back to the $\mathcal{M}_g$, enabling $\mathcal{M}_g$ to be updated through backpropagation.

Step 7. Updating the global feature extractor at the cloud: The $\mathcal{M}_g$ first receives the gradients $\nabla_{\boldsymbol{Z}_{global}^t} \mathcal{L}_n$ from each grid edge computing unit. Then, the average of these gradients is calculated across the feature classifiers, which in turn yields the gradient of the loss with respect to $\mathcal{M}_g$ by utilizing the gradients of the $\boldsymbol{Z}_{global}$ with respect to the $\mathcal{M}_g$.

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

This gradient is then used to update $\mathcal{M}_g$ as follows:

$$g^t_{\mathcal{M}_g} \leftarrow \nabla_{\mathcal{M}_g}\mathcal{L} \tag{18}$$

$$\theta^{t+1}_g = \theta^t_g - \eta_g g^t_{\mathcal{M}_g} \tag{19}$$

where $g^t_{\mathcal{M}_g}$ is the gradient of the loss function with respect to the global feature extractor $\mathcal{M}_g$ at training round t and η_g is the learning rate for $\mathcal{M}_g$.

Step 8. Broadcasting the parameters from the cloud back to the grid edge: After updating $\mathcal{M}_g$, the gradient of each loss function with respect to the respective intermediate feature $\boldsymbol{Z}^t_n$ of each local feature extractor is obtained as $g^t_{\boldsymbol{Z}_n} \leftarrow \nabla_{\boldsymbol{Z}^t_n}\mathcal{L}_n(\hat{\boldsymbol{y}^t_n}, \boldsymbol{y}_n)$ based on $\nabla_{\mathcal{M}_g}\mathcal{L}_n$ and $\nabla_{\boldsymbol{Z}^t_n}\mathcal{M}_g$. Then, each $g^t_{\boldsymbol{Z}_n}$ is broadcasted back to its respective $\mathcal{M}_l$ to complete the backpropagation at the grid edge.

Step 9. Updating the local feature extractors at the grid edge: Each $\mathcal{M}_l$ then computes the gradient of the loss function with respect to their local feature extractors as $g^t_{\mathcal{M}_{l,n}} \leftarrow \nabla_{\mathcal{M}_{l,n}}\mathcal{L}_n(\boldsymbol{y}^t_n, \boldsymbol{y}_n)$ based on the received gradients $\nabla_{\boldsymbol{Z}^t_n}\mathcal{L}_n$ and $\nabla_{\mathcal{M}_{l,n}}\boldsymbol{Z}^t_n$. Subsequently, each $g^t_{\mathcal{M}_l}$ is utilized to update the local feature extractors as follows:

$$\theta^{t+1}_{l,n} = \theta^t_{l,n} - \eta_l g^t_{\mathcal{M}_{l,n}} \tag{20}$$

Upon the completion of this step, a single round of collaborative training is completed. This process repeats from step two until satisfactory performance is attained. Consequently, the training of the collaborative FDIA localization models $\mathcal{M}_n = \{\mathcal{M}_{l,n}; \mathcal{M}_g; \mathcal{M}_{c,n}\}$ are finalized.

To summarize, each step fosters collaborative knowledge sharing while alleviating data-sharing issues as the raw data and labels are not broadcasted, leading to incremental improvements in the performance of the FDIA localization models across successive training iterations. After the training is completed, the grid edge computing units and the cloud continue their collaborative operations in real-time, as further elaborated in the next section.

C. Inference Stage and Implementation of the Proposed Framework

1) Inference Stage: The local feature extractors and feature classifiers at the grid edge and the global feature extractor at the cloud perform FDIA localization in collaboration during the inference stage based on the raw measurements at each node, as depicted in Fig. 2. The pseudo-code representing the inference stage is also given in 2. First, each $\mathcal{M}_l$ processes the raw measurements at each node and broadcasts the extracted local features Z_n from each node to $\mathcal{M}_g$. The extracted Z_n from each node is then aggregated and processed by $\mathcal{M}_g$ to extract the global features Z_{global}. Subsequently, Z_{global} is sent to each $\mathcal{M}_c$ at the grid edge, where the final decision is taken, thereby determining whether the measurements at each node is compromised by any FDIA.

2) Implementation: In the proposed framework, for both the training and inference stages, a sample-wise consensus is necessary between the data-owners since the complementary features at each node belonging to the same sample, need to be processed together. This enables the extraction of the global features. This requirement is an additional processing stage and a potential challenge compared to the other frameworks

Algorithm 1 FDIA Localization Framework - Training Stage

Require: Number of training round T, raw measurements from each node at the grid edge $\mathbf{X}_n | n \in N$
Ensure: FDIA Localization Framework

Initialization:
The local feature extractors $\mathcal{M}_{l,n}$ and the feature classifiers $\mathcal{M}_{c,n}$ at the grid edge and the global feature extractor $\mathcal{M}_g$ at the cloud
Learning rates η_l, η_g, η_c for $\mathcal{M}_l$, $\mathcal{M}_g$, $\mathcal{M}_c$ respectively
for $t = 1$ to T **do**
  for each mini-batch ξ^t **do**
    GRID EDGE:
    for $n = 1$ to N **in parallel do**
      $\boldsymbol{Z}^t_n = \mathcal{M}_{l,n}\left(\theta^t_{l,n}; \mathbf{X}^{\xi^t}_n\right)$
      ▷ intermediate features from $\mathcal{M}_{l,n}$
      Send each $\boldsymbol{Z}^t_n$ to the $\mathcal{M}_g$
    end for
    CLOUD:
    $\boldsymbol{Z}^t_{con} \leftarrow \text{Concatenate}\left(\boldsymbol{Z}^t_1, \ldots, \boldsymbol{Z}^t_N\right)$
    $\boldsymbol{Z}^t_{global} = \mathcal{M}_g\left(\theta^t_g; \boldsymbol{Z}^t_{con}\right)$
    Send $\boldsymbol{Z}^t_{global}$ to each $\mathcal{M}_{c,n}$
    GRID EDGE:
    for $n = 1$ to N **in parallel do**
      $\hat{\boldsymbol{y}}^t_n = \mathcal{M}_{c,n}\left(\theta^t_{c,n}; \boldsymbol{Z}^t_{global}\right)$ ▷
      Calculate the loss $\mathcal{L}_n(\hat{\boldsymbol{y}^t_n}, \boldsymbol{y}^t_n)$ at $\mathcal{M}_{c,n}$
      $g^t_{\mathcal{M}_{c,n}} \leftarrow \nabla_{\mathcal{M}_{c,n}}\mathcal{L}_n(\hat{\boldsymbol{y}^t_n}, \boldsymbol{y}^t_n)$
      $\theta^{t+1}_{c,n} = \theta^t_{c,n} - \eta_c g^t_{\mathcal{M}_{c,n}}$ ▷ Update $\mathcal{M}_{c,n}$
      Calculate $g^t_{\boldsymbol{Z}^t_{global},n} \leftarrow \nabla_{\boldsymbol{Z}^t_{global}}\mathcal{L}_n(\hat{\boldsymbol{y}^t_n}, \boldsymbol{y}_n)$
      Send $g^t_{\boldsymbol{Z}^t_{global},n}$ to the cloud
    end for
    CLOUD:
    Take the average $\frac{1}{N}\sum_{n=1}^{N} g^t_{\boldsymbol{Z}^t_{global},n}$
    Calculate $g^t_{\mathcal{M}_g} \leftarrow \nabla_{\mathcal{M}_g}\mathcal{L}$
    $\theta^{t+1}_g = \theta^t_g - \eta_g g^t_{\mathcal{M}_g}$ ▷ Update $\mathcal{M}_g$
    for $n = 1$ to N **in parallel do**
      $g^t_{\boldsymbol{Z}^t_n,n} \leftarrow \nabla_{\boldsymbol{Z}^t_n}\mathcal{L}_n(\hat{\boldsymbol{y}^t_n}, \boldsymbol{y}_n)$
      Send $g^t_{\boldsymbol{Z}^t_n,n}$ to each $\mathcal{M}_{l,n}$
    GRID EDGE:
    for $n = 1$ to N **in parallel do**
      $g^t_{\mathcal{M}_{l,n}} \leftarrow \nabla_{\mathcal{M}_{l,n}}\mathcal{L}_n(\hat{\boldsymbol{y}^t_n}, \boldsymbol{y}_n)$
      $\theta^{t+1}_{l,n} = \theta^t_{l,n} - \eta_l g^t_{\mathcal{M}_{l,n}}$ ▷ Update $\mathcal{M}_{l,n}$
    end for
  end for
end for

such as those based on the FL framework [20], [22]–[24] which only exchange parameters such as model weights with the cloud without depending on a sample-wise consensus during the training stage. It is assumed that the computational capability of each grid edge computing unit is similar which is also a requirement for the FL-based studies [20].

It is worth mentioning that the availability of measurements is directly related to the observability required by the state estimator. Therefore, various scenarios can be utilized, provided that observability is ensured. The measurements are assumed

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

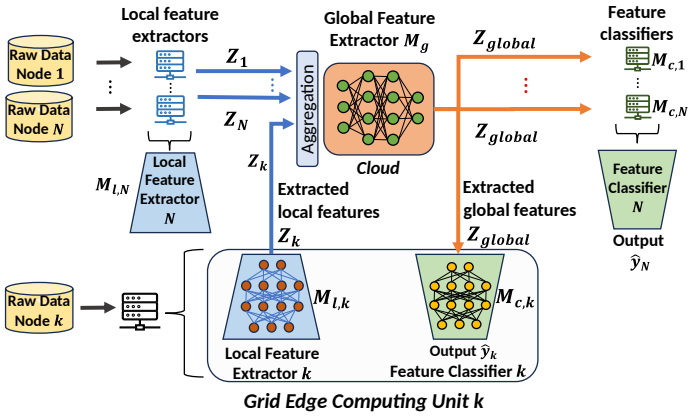


Fig. 2. Inference stage of the proposed framework

Algorithm 2 FDIA Localization Framework - Inference Stage

Require: Completion of the training of the local feature extractors $\mathcal{M}_{l,n}$ and the feature classifiers $\mathcal{M}_{c,n}$ at the grid edge and the global feature extractor $\mathcal{M}_g$ at the cloud

GRID EDGE:
for $n = 1$ to N **in parallel do**
  $\boldsymbol{Z}_n = \mathcal{M}_{l,n}(\theta_{l,n}; \mathbf{X}_n)$
  ▷ Intermediate features from each $\mathcal{M}_{l,n}$
  Send each $\boldsymbol{Z}_n$ to the $\mathcal{M}_g$
end for
CLOUD:
$\boldsymbol{Z}_{con} \leftarrow \text{Concatenate}(\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N)$
$\boldsymbol{Z}_{global} = \mathcal{M}_g(\theta_g; \boldsymbol{Z}_{con})$
Send $\boldsymbol{Z}_{global}$ to each $\mathcal{M}_{c,n}$
GRID EDGE:
for $n = 1$ to N **in parallel do**
  $\hat{\boldsymbol{y}}_n = \mathcal{M}_{c,n}(\theta_{c,n}; \boldsymbol{Z}_{global})$
  ▷ Obtaining the decision at each grid edge
end for

to be part of a supervisory control and data acquisition system. Additionally, the measurements are assumed to be synchronized, as complementary measurements are processed by each local feature extractor, then aggregated in the cloud and processed collectively by the global feature extractor.

It is important to note that the grid edge is considered a single node in the distribution network in this paper. The grid edge computing unit (the local feature extractor and the feature classifier) is only responsible for its respective assigned node. The FDIA localization is exclusively performed for the assigned node based on the measurements available to the grid edge computing unit. However, the grid edge can be a single high-voltage substation or more than one covering a certain region or even a large network under a wide area monitoring systems. The proposed framework is adaptable and can be applied as either a cross-silo or cross-device approach, similar to the FL framework.

D. The Proposed Detection Model

In this section, the complete FDIA localization models $\mathcal{M}_n$ which is split as $\{\mathcal{M}_{l,n}; \mathcal{M}_g; \mathcal{M}_{c,n}\}$ and deployed at the grid edge and the cloud as depicted in Fig. 3 is introduced.

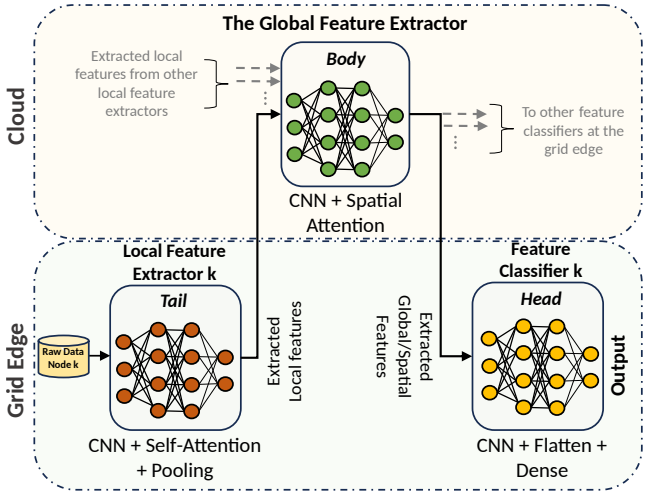


Fig. 3. The proposed deep learning model

It is important to emphasize that each component of the detection model is assigned a specific task. The local feature extractor, situated at the grid edge, is responsible for extracting the local features from the input measurement vector, which may include power injections, power flows, and voltage measurements. This task is accomplished through the integrated self-attention mechanism that assigns weights to the inputs based on their importance. Subsequently, the locally extracted features are concatenated and forwarded to the global feature extractor, which is designed to capture spatial features. This process leverages a CNN model coupled with a spatial attention module to extract spatial relationships among the concatenated local features. Finally, the extracted spatial features are transmitted to each feature classifier, generating outputs at the grid edge. These interactions among the model components are depicted in Fig. 3, where, for simplicity, only a single grid-edge model comprising the local feature extractor and feature classifier, is illustrated.

The input data for the local feature extractor is a vector containing the available measurements at each node. This vector is reshaped into a tensor consisting of the batch size, the number of features, and a single channel as it is processed by the CNN model in the local feature extractor. The tensor shape remains consistent throughout, up to the feature classifier, which produces the final output. The details of each model are thoroughly described in the following section.

1) The local feature extractors: Each $\mathcal{M}_l$ is designed to extract local features from the node measurements at the grid edge. The node measurements can be power injections, power flows and voltage measurements based on the availability on a specific node in the system. Each $\mathcal{M}_l$ consists of a convolution layer which is followed by a pooling layer and the self-attention mechanism [31]. Self-attention mechanism which enables the models to focus on the most important features from the raw measurements is utilized in each $\mathcal{M}_l$.

Each $\mathcal{M}_l$ initiates processing by applying a convolutional layer to its respective raw measurements X^{ξ}, producing a feature map $\boldsymbol{F}_c \in \mathbb{R}^{H \times W \times C}$. Here, C denotes the number

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

of filters employed in the convolutional layers, with H and W specifying the dimensions of the input data's height and width, respectively. The self-attention mechanism is then applied to the $\mathbf{F}_c$. First, 1×1 convolutions are applied to the $\mathbf{F}_c$ to transform the input into Query ($\boldsymbol{Q} = \boldsymbol{W}_Q \boldsymbol{F}_c$), Key ($\boldsymbol{K} = \boldsymbol{W}_K \boldsymbol{F}_c$), and Value ($\boldsymbol{V} = \boldsymbol{W}_V \boldsymbol{F}_c$) matrices, which are the key elements of the self-attention mechanism, where $\{\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V\}$ represent the matrices of parameters for 1x1 convolutions. Then, the attention mechanism is computed by $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{K}^T$. The resulting scores are then scaled and processed by the softmax function as $\mathrm{AT}(\mathbf{Q}, \mathbf{K}) = \mathrm{softmax}(\boldsymbol{S}/\sqrt{d})\mathbf{V}$ yielding an attention map. The attention weights are then obtained as $\mathbf{O} = \mathrm{AT}(\mathbf{Q}, \mathbf{K})\mathbf{V}$ quantifying the total attention score assigned to each element across the entire data. These weighted values are reshaped to their original dimensions and then reintegrated with the original input, yielding the output of the self-attention mechanism. The self-attention mechanism is utilized to focus on significant components of the input, enhancing the model's ability to discern critical relationships and dependencies within the data.

It is important to note that the data obtained from the self-attention mechanism constitutes the local features extracted by each $\mathcal{M}_l$, thereby corresponding to $\boldsymbol{Z}_n^t$ at node n. Each $\boldsymbol{Z}_n^t$ is then sent to the cloud to be processed by $\mathcal{M}_g$, as explained in the next section.

2) The global feature extractor: The $\mathcal{M}_g$ is specifically designed to extract the global features at the cloud. The input of $\mathcal{M}_g$ is the concatenated local features represented as $\boldsymbol{Z}_{con}^t$. Spatial information can be extracted as the local features from each node are concatenated as $\boldsymbol{Z}_{con}^t$ in the cloud. Therefore, a spatial attention module is integrated into the $\mathcal{M}_g$ which consists of a convolutional layer which is followed by a spatial attention module [32] and a pooling layer respectively. The spatial attention module enables the model to capture spatial features leading to enhanced feature representation and more efficient learning.

A spatial attention map $\boldsymbol{M}_s(\boldsymbol{Z}_{con})$ is formed by utilizing the inter-spatial relationships. Initially, operations of average-pooling and max-pooling are conducted along the channel axis, with the resulting outputs being merged to form efficient feature descriptors. Subsequently, a convolution layer is applied to the merged feature descriptors as follows:

$$\boldsymbol{M}_s(\boldsymbol{Z}_{con}) = \sigma\left(f([\mathrm{Avg\,P}(\boldsymbol{Z}_{con}); \mathrm{Max\,P}(\boldsymbol{Z}_{con})])\right) \quad (21)$$

where f represents a convolution operation, AvgP and MaxP represent the average and max-pooling operations, respectively. Consequently, the spatial attention map $\boldsymbol{M}_s$ is generated. Note that in the proposed framework, once the $\boldsymbol{M}_s$ is processed by the pooling layer, it then turns into the extracted spatial features $\boldsymbol{Z}_{global}^t$ by $\mathcal{M}_g$ at the cloud.

3) The feature classifier: Each feature classifier $\mathcal{M}_c$ receives $\boldsymbol{Z}_{global}^t$ as its input and processes it to obtain the output at the grid edge. $\mathcal{M}_c$ is specifically designed to process the global features retrieved from the cloud, enabling precise classification. $\mathcal{M}_c$ incorporates a convolution layer alongside standard layers such as flatten and dense layers. As the classification problem is designed as a binary classification in each $\mathcal{M}_c$, the decision is taken as 0 or 1 representing the attacked or normal label, respectively, at the grid edge.

IV. Results and Discussion

A. Dataset Generation

The load profiles are generated by aggregating the real smart meter data [33] for IEEE 13 and IEEE 37 node test systems [34], which are illustrated in Fig. 5, as in [35]. In this work, the weighted least squares based state estimation integrating a linear approximation and residual-based detection mechanism which is known to be one of the most common techniques [6], as explained in Section II-A, are utilized. Stealthy FDIA explained in Section II-A is adopted to generate malicious measurements. Attacks of varying strength, categorized as weak, medium, and strong, are implemented as outlined in [36] and applied to single or multiple nodes. Fig. 4 illustrates how FDIA can generate false state estimates (e.g. voltage magnitudes) and how different types of FDIA can mislead system operators into perceiving the system as operating closer to or beyond safety thresholds. Such deception could lead to incorrect operational decisions, even when the actual system is in a safe state. Conversely, FDIA can also create a false perception of safety when the system is, in reality, operating in an unsafe condition, thereby delaying actions. Moreover, small incorrectness in voltage estimates may result in large errors in the estimated power flows/currents in branches and incorrect conclusions about overloading of component(s). Meaning that even a weak attack (in terms of voltage deviations) might escalate into a strong one (in terms of power flows/currents/loading rate).

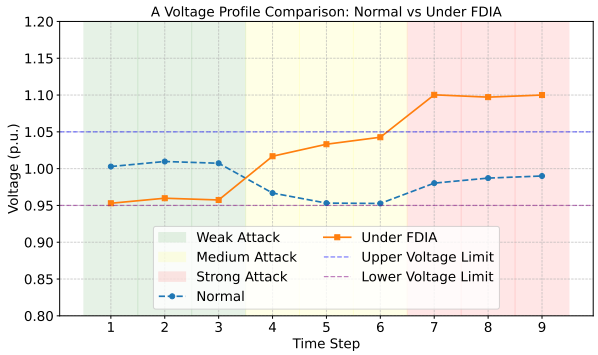


Fig. 4. An example of normal and under FDIA voltages

17520 samples were created for both test systems. FDIAs necessitate altering multiple measurements across various nodes to induce deviations in state variables. Consequently, this results in varying attack ratios at each node. The average attack ratio (number of malicious samples over all samples) across all nodes is obtained as 48% and 43% for the IEEE 13 and IEEE 37 test systems, respectively, after implementing a range of FDIA scenarios as illustrated in Fig. 6. The allocation of the generated measurements to specific nodes is based on the placement of the measurement devices as shown in Fig. 5. Consequently, $\mathbf{X}_n$ is generated for each node n.

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

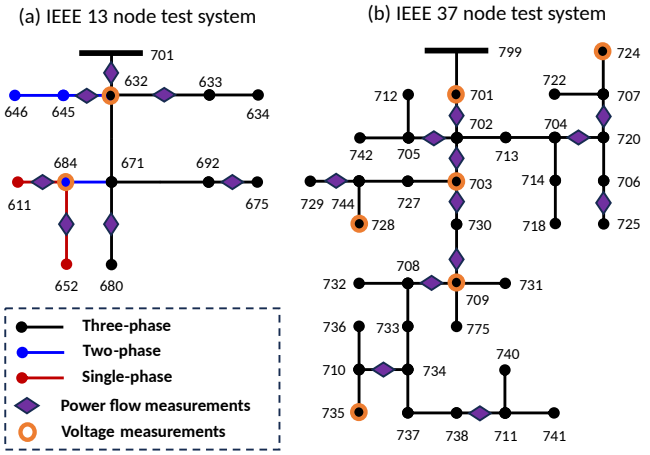


Fig. 5. IEEE 13 and 37 node test systems

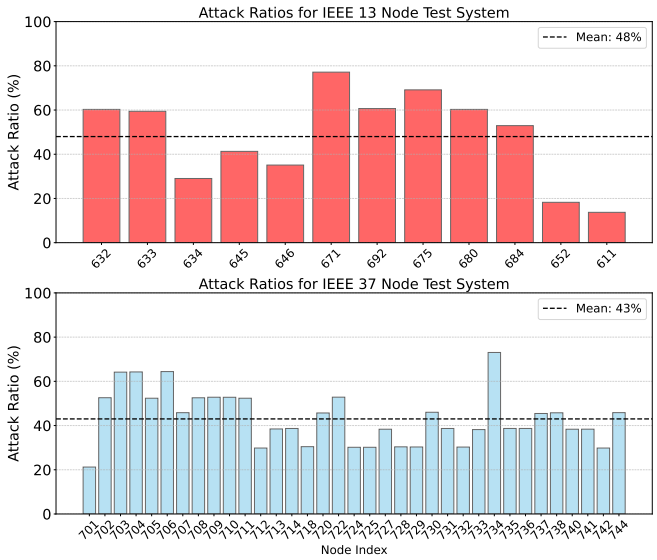


Fig. 6. Attack ratios for IEEE 13 and 37 node test systems

B. Evaluation metrics

The FDIA localization is designed as a binary classification task for each feature classifier $\mathcal{M}_c$ at the grid edge. Malicious and normal samples are denoted by positive and negative classes, respectively. The evaluation metrics given in (22) are utilized.

$$\begin{aligned} Accuracy &= (TP+TN)/(TP+TN+FP+FN) \\ precision &= TP/(TP+FP) \\ recall &= TP/(TP+FN) \\ F1 &= 2\cdot(precision\cdot recall)/(precision+recall) \end{aligned} \quad (22)$$

TN and TP represent the correctly predicted normal and attacked samples, respectively, while FN and FP represent the incorrect predictions of normal and attacked samples, respectively. Accuracy (ACC) indicates the overall detection performance for normal and compromised samples, whereas the F1-score reflects the harmonic mean of precision and recall.

C. Results

In this section, the performance of the proposed framework is evaluated through various analyses.

*1) **Comparison with the State-of-the-art Models**:* The performance of the proposed deep learning model is compared with the state-of-the-art models which are selected as Long Short-Term Memory (LSTM) and Deep Neural Networks (DNN) under the proposed framework. The mean value of each evaluation metric, calculated across all nodes, is presented in Fig. 7 for both test systems.

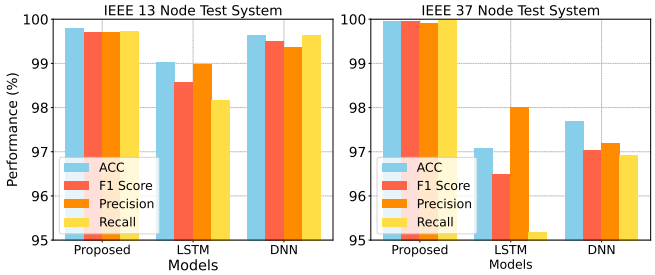


Fig. 7. Performance comparison with the state-of-the-arts models

Fig. 7 demonstrates that the proposed deep learning model excels in all evaluated metrics, achieving accuracies of 99.80% and 99.96%, F1 scores of 99.71% and 99.95%, precisions of 99.70% and 99.91%, and recalls of 99.72% and 99.99% for the IEEE 13-node and IEEE 37-node test systems, respectively. In comparison, the DNN also displays high performance closely rivalling that of the proposed model for the IEEE 13-node test system, achieving an accuracy of 99.64%, an F1 score of 99.50%, a precision of 99.36%, and a recall of 99.64%. However, when applied to the larger IEEE 37-node test system, the performance of the DNN significantly declines. This decline in performance can be attributed to the DNN's inability to capture the spatial relationships within the increased scale of a larger system, unlike the proposed model. This underscores the superior performance of the proposed framework consistently outperforming the other models across both test systems. Furthermore, the LSTM records the poorest performance across both test systems.

Since the mean values of the evaluation metrics calculated across the nodes given in Fig. 7 do not provide a detailed representation for each individual node, the results for several nodes from each test system are detailed in Table II. Although the mean accuracy value for the DNN in the IEEE 13-node test system is 97.68%, the accuracy values at the grid edge computing units are 93.01% and 95.38% for nodes 718 and 735, respectively.

It can be concluded that the ability to extract the local and global/spatial features through the integrated attention mechanism enables the proposed deep learning model to substantially outperform the other models in terms of all evaluation metrics.

*2) **Comparison of Frameworks**:* In this section, the proposed framework is evaluated in comparison to the following frameworks:

1) **Multi-Label Centralized Framework (ML-CF):** In the literature, one of the most common approaches for FDIA

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

TABLE II
PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART MODELS UNDER THE PROPOSED FRAMEWORK

	Node	**Proposed Model**				**LSTM**				**DNN**			
		ACC	Precision	Recall	F1	ACC	Precision	Recall	F1	ACC	Precision	Recall	F1
IEEE 13	645	**99.90**	**99.74**	**99.99**	**99.87**	98.97	99.47	97.94	98.70	98.62	96.64	**99.99**	98.29
	646	**99.94**	**99.85**	**99.99**	**99.92**	99.28	99.69	98.18	98.93	99.43	98.36	**99.99**	99.17
	671	**99.95**	**99.93**	**99.99**	**99.97**	98.62	98.66	99.52	99.09	99.79	99.73	**99.99**	99.86
IEEE 37	718	**99.83**	**99.99**	**99.14**	**99.56**	95.72	98.98	91.55	95.12	93.01	94.89	89.48	92.10
	722	**99.94**	**99.99**	**99.81**	**99.91**	96.69	98.36	90.55	94.30	96.35	91.75	96.60	94.11
	733	**99.91**	**99.99**	**99.88**	**99.94**	92.26	88.41	95.05	91.61	96.95	98.80	96.98	95.25
	735	**99.97**	**99.99**	**99.93**	**99.96**	94.30	98.91	86.32	92.19	95.38	98.76	98.83	98.80
	740	**100**	**99.99**	**99.99**	**99.99**	95.66	97.96	90.83	94.26	96.86	96.96	94.98	95.96

localization is training a deep learning model as a multi-label classifier in the cloud where the decision is made, as in [8]. Therefore, the proposed deep learning model $\mathcal{M} = \{\mathcal{M}_l; \mathcal{M}_g; \mathcal{M}_c\}$ as a whole is trained with the raw measurements from all nodes in the cloud without considering data-sharing concerns.

2) **Multi-Label Vanilla Split Learning Framework (ML-VSL):** The ML-VSL framework is developed by transforming the vanilla split learning framework used in [26] designed to detect the presence of FDIA, into a multi-label classification problem. This transformation adopts the approach outlined in [8] to facilitate FDIA localization and comparison with the framework proposed in this paper. In the ML-VSL framework, each $\mathcal{M}_l$ is stationed at the grid edge, while $\mathcal{M}_g$ and $\mathcal{M}_c$ are combined to form the cloud-side model located at the cloud where multi-label classification occurs. This contrasts with the FL and the proposed frameworks, where decisions are made at each grid edge computing unit.
3) **FL Framework:** The FL framework, as proposed in [20], requires consistency across node measurements. Consequently, to facilitate the comparison, the feature set within the IEEE 37-node test system is standardized across all nodes, adhering to the FL framework's requirements. For instance, in [20], which employs the FL framework, the power injections at each bus and the power flow of a branch connected to the bus are considered as inputs to each edge computing unit, as data uniformity is a requirement of the FL framework. Although multiple branches may be connected to a single bus, or additional measurements may be available for a specific bus, only those measurements that ensure consistency across all buses are selected. Consequently, to implement the FL approach used in [20], the power injection at each node is utilized as input to establish the FL framework. Therefore, a separate dataset is specifically generated to facilitate a comparison with the FL framework which is unable to accommodate the variability in measurements across different nodes. On the other hand, it is important to note that the proposed framework based on split learning in a U-shape configuration does not impose any data-consistency requirements, meaning that any available measurement can be utilized as input. Additionally, the proposed deep learning model, which is split into $\mathcal{M}_l$, $\mathcal{M}_g$, and $\mathcal{M}_c$, is consolidated into a single model as $\mathcal{M}$ within each grid edge computing node and trained under the FL framework.

An overall comparison regarding the information exchange and decision location of the frameworks is also provided in Table III. Although the grid edge computing units in the ML-VSL framework are trained using vanilla split learning, decisions are made in the cloud, necessitating label sharing. This arrangement cannot alleviate data-sharing concerns even though the raw measurements are not shared.

TABLE III
COMPARISON OF FRAMEWORKS
A: RAW MEASUREMENTS, B: MODEL WEIGHTS, C: GRADIENTS, D: INTERMEDIATE FEATURES, E: LABEL

Framework	Information Exchange					Decision Location
	A	B	C	D	E	
ML-CL	✓	x	x	x	x	Cloud
ML-VSL	x	x	✓	✓	✓	Cloud
FL	x	✓	x	x	x	Edge
Proposed	x	x	✓	✓	x	Edge

The mean values of the evaluation metrics, calculated across the nodes, for both test systems, are given in Table IV.

TABLE IV
MEAN VALUE COMPARISON FOR EVALUATION METRICS

	Framework	ACC	Precision	Recall	F1
IEEE 13	Proposed	99.80	99.72	99.71	99.70
	ML-CL	99.56	99.14	99.77	99.45
	ML-VSL	99.56	99.44	99.47	99.22
IEEE 37	Proposed	99.96	99.91	99.99	99.95
	ML-CL	99.73	99.65	99.72	99.59
	ML-VSL	97.72	96.73	97.29	96.22

Table IV shows that the proposed framework outperforms both the ML-CL and ML-VSL frameworks across both test systems, while also successfully mitigating data-sharing concerns, unlike the ML-CL and ML-VSL frameworks. The ML-CL achieves high performance in both test systems; however, because the raw measurements are directly fed to the centralized model, data-sharing concerns are not alleviated. It is also observed that the accuracy of the ML-VSL framework decreases to 97.72% when applied to the IEEE 37-node test system. it can be concluded that despite not exchanging raw measurements, the proposed framework achieves higher performance than the centralized learning framework.

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

To facilitate comparison with the FL framework, a new dataset is generated in alignment with the FL requirements to ensure that identical features are fed to each grid edge computing unit. The proposed framework and the FL framework are then both trained using this dataset. A detailed analysis of the performance of each grid edge computing unit is depicted in Fig. 8 for the IEEE 37-node test system.

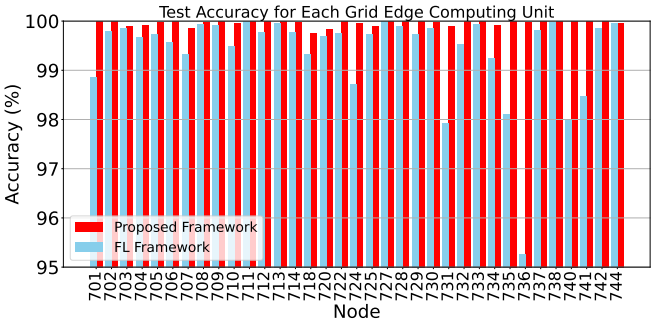


Fig. 8. Comparison of the FL and the proposed frameworks

Fig. 8 shows that nearly all grid edge computing units trained under the proposed framework surpass those trained under the FL framework in performance. For example, the grid edge computing unit at node 736 achieves an accuracy of 95.25% under the FL framework, while it is 99.97% under the proposed framework. Similar disparities in performance, with the proposed framework consistently outperforming the FL framework, are also observed at other nodes, including 731, 735, and 741. The superior performance of the proposed framework can be attributed to more extensive data-sharing capabilities, and the ability to collaborate in the inference stage, which facilitates a more thorough exchange of information between the grid edge and the cloud compared to the FL framework.

*3) **Performance Comparison under Different FDIA Strengths**:* In this section, the performance of the proposed framework is assessed across varying levels of attack strength, with additional comparisons made against the other frameworks. Two distinct datasets are created: one containing only weak attack samples and the other comprising strong attack samples considering the attack strength classifications given in [36]. These datasets are solely used for evaluating the model's performance and are not part of the training process. The results for both test systems are presented in Fig. 9.

Fig. 9 shows that all frameworks exhibit high performance under strong attack conditions for the IEEE 13-node test system. These high performances stem from the substantial variations in state variables caused by the strong FDIAs, leading to significant changes in measurements. However, the ML-VSL framework's performance declines against strong FDIAs in the IEEE 37-node test systems, in contrast to other frameworks. When subjected to weak FDIAs, the proposed framework outperforms both the ML-CL and ML-VSL frameworks whose performance significantly drops. The performance of the proposed framework under weak FDIAs is lower compared to strong FDIAs, as the attacked measurements closely resemble normal measurements due to minimal variations in state variables. Consequently, although the detection rate is lower, the impact of weak FDIAs on the system is less severe compared to strong FDIAs. Additionally, it is observed that the proposed framework performs better in larger systems, demonstrating its capability to extract spatial features effectively.

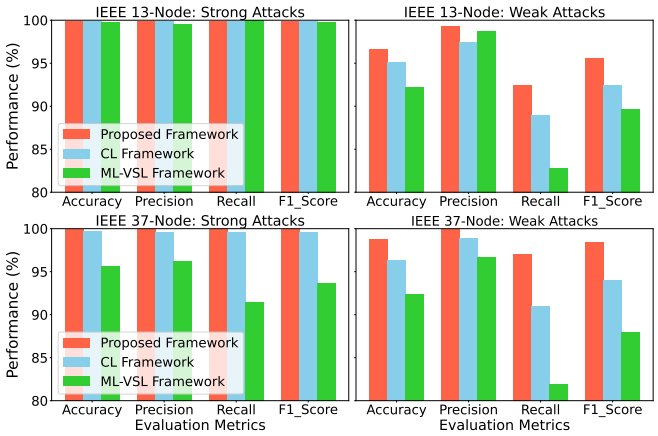


Fig. 9. Performance comparison against weak and strong FDIAs

To compare the proposed framework with the FL framework, two datasets, one for weak attack strength and one for strong attack strength are generated following the FL framework's requirements, as outlined in Section IV-C2. The results for both frameworks on the IEEE 37-node test system are presented in Fig. 10.

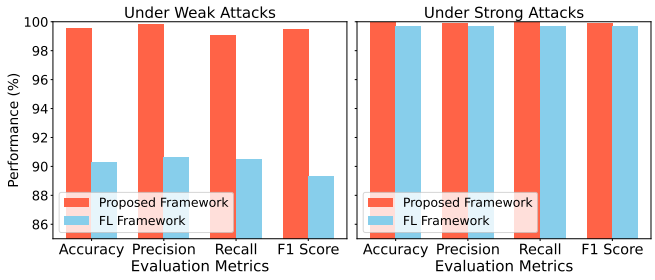


Fig. 10. Performance comparison against weak and strong FDIA

Fig. 10 shows that the proposed framework attains performance metrics exceeding 99% across all evaluation criteria for both strong and weak FDIA scenarios. On the other hand, although the FL framework achieves over 99% in all evaluation metrics under strong FDIAs, it performs poorly under weak FDIAs, with around 90% in all evaluation metrics. It is concluded that the proposed framework is robust to varying FDIA strengths, whereas the FL framework fails to achieve this robustness.

*4) **Effect of Measurement Noise**:* Measurements collected from the field typically exhibit a signal-to-noise ratio (SNR) ranging from 30 dB to 80 dB [37]. To evaluate the performance of the proposed framework under realistic conditions, a comparative analysis of the performance metrics is conducted at different SNR noise levels $\{65, 50, 35, 25\}$. The noise is applied only to the test data, which is not seen during the training stage. The performance of the proposed framework

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

under noisy conditions for the IEEE 13-node and IEEE 37-node test systems is presented in Fig. 11.

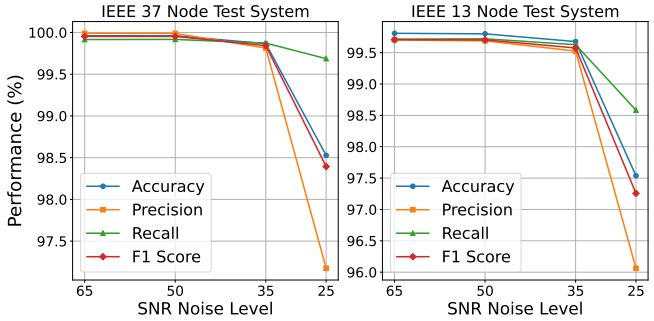


Fig. 11. Performance Comparison Against Noise

Fig. 11 demonstrates that the proposed framework maintains performance levels above 99.5% across all evaluation metrics at noise levels up to 25dB for both test systems. When the noise level reaches 25dB, it shows an accuracy of 98.5% and 97.5%, recall rates of 99.5% and 98.5%, precision of 97% and 96%, and F1 scores of 98.5% and 97.5% for the IEEE 37 and IEEE 13 node test systems, respectively. It is concluded that the proposed framework is resilient to varying noise levels and consistently achieves satisfactory performance, even under heavy noise conditions.

The performance of the proposed framework is also compared with the FL framework. Noise is applied to the test set generated for the FL framework in Section IV-C2. The results are given in Fig. 12.

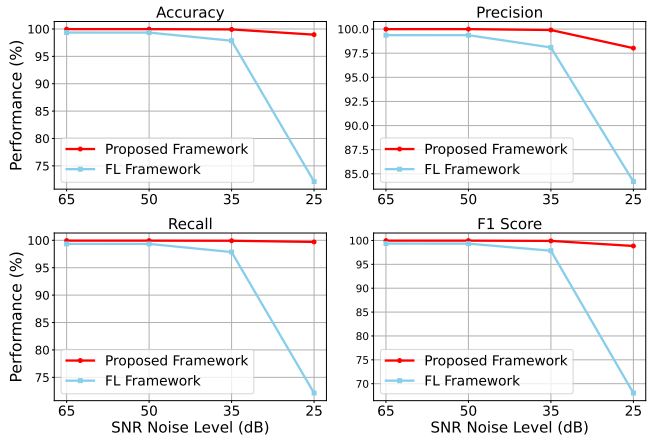


Fig. 12. Performance Comparison Against Noise

Fig. 12 shows that the proposed framework maintains performance levels above 98% in accuracy, precision, and F1 score, and 99.5% in recall under 25dB noise level. In contrast, the FL framework shows around 70% accuracy, recall, and F1 score, with approximately 84% precision under 25dB noise level. In summary, the proposed framework consistently outperforms the FL framework in all performance metrics across various SNR noise levels, demonstrating its robustness in handling noisy measurements. It can be concluded that the proposed framework is effectively trained to remain resilient to varying noise levels, whereas the FL framework fails to maintain resilience in the presence of noise levels exceeding 35 dB.

V. Conclusion

In this paper, a cloud-edge framework based on split learning in U-shape configuration is proposed for locational detection of FDIA in smart grids. The framework enables the use of non-consistent measurement sets across nodes while mitigating data-sharing concerns, as neither raw data nor labels are exchanged. In the proposed framework, the local and global feature extractor models incorporate attention mechanisms to effectively capture local and global features. The proposed model incorporating the attention units, outperforms state-of-the-art models, LSTM and DNN models, across all evaluation metrics within the proposed framework. Furthermore, the proposed framework, based on split learning in a U-shape configuration, exhibited superior performance compared to state-of-the-art frameworks, including federated learning, centralized learning, and vanilla split learning. This superiority is evidenced by its enhanced ability to accurately localize false data injection attacks, robust resilience to varying attack strengths, and effective handling of noisy measurements. Future work will investigate coordinated FDIA which requires less information compared to stealthy FDIAs, along with asynchronous merging techniques for grid edge computing units that have varying computational powers and the challenge of reducing computational overhead in the proposed framework to enable its application to larger distribution networks. It is worth highlighting that no existing studies have specifically examined the impact of different state estimator types on FDIA detection and localization. This gap presents a valuable avenue for future research.

References

[1] U. Inayat, M. F. Zia, S. Mahmood, H. M. Khalid, and M. Benbouzid, "Learning-based methods for cyber attacks detection in IoT systems: A survey on methods, analysis, and future prospects," *Electronics*, vol. 11, no. 9, p. 1502, 2022.

[2] H. M. Khalid, M. M. Qasaymeh, S. M. Muyeen, M. S. E. Moursi, A. M. Foley, T. O. Sweidan, and P. Sanjeevikumar, "WAMS operations in power grids: A track fusion-based mixture density estimation-driven grid resilient approach toward cyberattacks," *IEEE Systems Journal*, vol. 17, no. 3, pp. 3950–3961, 2023.

[3] H. M. Khalid, F. Flitti, M. S. Mahmoud, M. M. Hamdan, S. Muyeen, and Z. Y. Dong, "Wide area monitoring system operations in modern power grids: A median regression function-based state estimation approach towards cyber attacks," *Sustainable Energy, Grids and Networks*, vol. 34, p. 101009, 2023.

[4] A. S. Musleh, H. M. Khalid, S. Muyeen, and A. Al-Durra, "A prediction algorithm to enhance grid resilience toward cyber attacks in WAMCS applications," *IEEE Systems Journal*, vol. 13, no. 1, pp. 710–719, 2017.

[5] H. M. Khalid and J. C.-H. Peng, "A bayesian algorithm to enhance the resilience of WAMS applications against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 2026–2037, 2016.

[6] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, pp. 1–33, 2011.

[7] D. Du, M. Zhu, X. Li, M. Fei, S. Bu, L. Wu, and K. Li, "A review on cybersecurity analysis, attack detection, and attack defense methods in cyber-physical power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 3, pp. 727–743, 2022.

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

[8] S. Wang, S. Bi, and Y.-J. A. Zhang, "Locational detection of the false data injection attack in a smart grid: A multilabel classification approach," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8218–8227, 2020.

[9] D. Mukherjee, "A novel strategy for locational detection of false data injection attack," *Sustainable Energy, Grids and Networks*, vol. 31, p. 100702, 2022.

[10] J. Zhu, W. Meng, M. Sun, J. Yang, and Z. Song, "FLLF: A fast-lightweight location detection framework for false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 15, no. 1, pp. 911–920, 2023.

[11] T. Zhu, J. Wang, Y. Zhu, H. Chen, H. Zhang, and S. Yin, "Power grid network security: A lightweight detection model for composite false data injection attacks using spatiotemporal features," *International Journal of Critical Infrastructure Protection*, vol. 46, p. 100697, 2024.

[12] O. Boyaci, M. R. Narimani, K. R. Davis, M. Ismail, T. J. Overbye, and E. Serpedin, "Joint detection and localization of stealth false data injection attacks in smart grids using graph neural networks," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 807–819, 2021.

[13] S. Peng, Z. Zhang, R. Deng, and P. Cheng, "Localizing false data injection attacks in smart grid: A spectrum-based neural network approach," *IEEE Transactions on Smart Grid*, vol. 14, no. 6, pp. 4827–4838, 2023.

[14] Q. Li, J. Zhang, J. Zhao, J. Ye, W. Song, and F. Li, "Adaptive hierarchical cyber attack detection and localization in active distribution systems," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 2369–2380, 2022.

[15] J. Lin, J. Ma, and J. Zhu, "Hierarchical federated learning for power transformer fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[16] Y. Wang, J. Yan, Z. Yang, Y. Dai, J. Wang, and Y. Geng, "A novel federated transfer learning framework for intelligent diagnosis of insulation defects in gas-insulated switchgear," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[17] X. Cheng, F. Shi, Y. Liu, J. Zhou, X. Liu, and L. Huang, "A class-imbalanced heterogeneous federated learning model for detecting icing on wind turbine blades," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8487–8497, 2022.

[18] H. Moayyed, A. Moradzadeh, B. Mohammadi-Ivatloo, A. P. Aguiar, and R. Ghorbani, "A cyber-secure generalized supermodel for wind power forecasting based on deep federated learning and image processing," *Energy Conversion and Management*, vol. 267, p. 115852, 2022.

[19] W.-T. Lin, G. Chen, and X. Zhou, "Privacy-preserving federated learning for detecting false data injection attacks on power system," *Electric Power Systems Research*, vol. 229, p. 110150, 2024.

[20] 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 Transactions on Smart Grid*, vol. 13, no. 6, pp. 4862–4872, 2022.

[21] F. Kausar, S. Deo, S. Hussain, and Z. Ul Haque, "Federated deep learning model for false data injection attack detection in cyber physical power systems," *Energies*, vol. 17, no. 21, p. 5337, 2024.

[22] M. Abdel-Basset, N. Moustafa, and H. Hawash, "Privacy-preserved generative network for trustworthy anomaly detection in smart grids: A federated semisupervised approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 995–1005, 2022.

[23] 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," *Applied Energy*, vol. 314, p. 118828, 2022.

[24] M. A. Husnoo, A. Anwar, H. T. Reda, N. Hosseinzadeh, S. N. Islam, A. N. Mahmood, and R. Doss, "Feddisc: A computation-efficient federated learning framework for power systems disturbance and cyber attack discrimination," *Energy and AI*, vol. 14, p. 100271, 2023.

[25] X. Yin, Y. Zhu, and J. Hu, "A subgrid-oriented privacy-preserving microservice framework based on deep neural network for false data injection attack detection in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1957–1967, 2021.

[26] M. Kesici, B. Pal, and G. Yang, "Detection of false data injection attacks in distribution networks: A vertical federated learning approach," *IEEE Transactions on Smart Grid*, vol. 15, no. 6, pp. 5952–5964, 2024.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 2017, pp. 1273–1282.

[28] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.

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

[30] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, 2018.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[32] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[33] C. for Energy Regulation (CER), "CER smart metering project-electricity customer behaviour trial, 2009-2010," 2012. [Online]. Available: https://www.ucd.ie/issda/data/commissionforenergyregulationcer/

[34] K. P. Schneider, B. A. Mather, B. C. Pal, C.-W. Ten, G. J. Shirek, H. Zhu, J. C. Fuller, J. L. R. Pereira, L. F. Ochoa, L. R. de Araujo, R. C. Dugan, S. Matthias, S. Paudyal, T. E. McDermott, and W. Kersting, "Analytic considerations and design basis for the ieee distribution test feeders," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3181–3188, 2018.

[35] M. Alduhaymi, R. Singh, F. U. Nazir, B. Pal, and A. Ahmadi, "Forecasting-aided state estimation with deep learning-generated pseudo measurements," *IEEE Transactions on Instrumentation and Measurement*, 2025.

[36] J. James, Y. Hou, and V. O. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3271–3280, 2018.

[37] 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," *International Journal of Electrical Power & Energy Systems*, vol. 145, p. 108612, 2023.

Mert Kesici 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, Turkey, in 2019. He is currently a Maria Skłodowska-Curie Early-Stage Researcher and a PhD candidate at Imperial College London. His research interests include the cyber security of smart grids and the application of privacy-preserving machine learning algorithms to smart grids.

Malek Alduhaymi received the B.Sc. degree in electrical engineering from Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia, in 2015, the M.Sc. degree in electric power systems from the University of Southern California, Los Angeles, CA, USA, in 2019, and the Ph.D. degree in control and power systems from the Imperial College London, London, U.K., in 2024. He is currently an Assistant Professor of Electrical and Engineering, Prince Sattam bin Abdulaziz University. His expertise extends to integrating machine learning and data-driven approaches to enhance renewable energy systems and power system operations. His research interests include advanced power system modeling and analysis, particularly on time-series forecasting, state estimation for active distribution networks, and power system optimization.

Anton Ishchenko received the Ph.D. degree in electrical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands in 2008. From 2008 to 2009 he was employed as a postdoc there. Since 2009 he works as a researcher at Phase to Phase B.V. company, Arnhem, The Netherlands, busy with the development of software for power system computations. His research interests include power system modelling and computations: deterministic and probabilistic power flow, state estimation, power system transients, short-circuit, arc flash, harmonics, etc.

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2025.3580683

Guangya Yang 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 (InnoCyPES) and the Lead Editor of the Power and Energy Society Section in IEEE ACCESS.

Bikash Pal received B.E.E. degree (Hons.) in electrical engineering from Jadavpur University, Calcutta, India, in 1990, the M.E. degree in electrical engineering from Indian Institute of Science, Bengaluru, India, in 1992, and the Ph.D. degree in electrical engineering from the Imperial College London, London, U.K., in 1999. He is currently a Professor with the Department of Electrical and Electronic Engineering, Imperial College London. His current research interests include renewable energy modeling and control, state estimation, and power system dynamics. Dr. Pal is a fellow of the Royal Academy of Engineering, U.K., for the industrial impact of his research in power network control. He was the Vice President of Publications of the IEEE Power and Energy Society from 2019 to 2023. 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.

© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.