A Subgrid-Oriented Privacy-Preserving Microservice Framework Based on Deep Neural Network for False Data Injection Attack Detection in Smart Grids

Xuefei Yin, Yanming Zhu, and Jiankun Hu, *Senior Member, IEEE*

***Abstract*—False data injection attacks (FDIAs) have recently become a major threat to smart grids. Most of the existing FDIA detection methods have focused on modeling the temporal relationship of time-series measurement data but have paid less attention to the spatial relationship between bus/line measurement data and have failed to consider the relationship between subgrids. To address these issues, in this article, we propose a subgrid-oriented microservice framework by integrating a well-designed spatial–temporal neural network for FDIA detection in ac-model power systems. First, a well-designed neural network is developed to model the spatial–temporal relationship of bus/line measurements for subgrids. A microservice-based supervising network is then proposed for integrating the representation features obtained from subgrids for the collaborative detection of FDIAs. To evaluate the proposed framework, three types of FDIA datasets are generated based on a public benchmark power grid. Case studies on the FDIA datasets show that our method outperforms state-of-the-art methods for FDIA detection in these datasets.**

***Index Terms*—Bad data detection, deep learning, false data injection attack (FDIA), FDIA detection, microservice, privacy preserving.**

I. Introduction

In recent years, false data injection attacks (FDIAs) have drawn the attention of researchers to the vulnerability of cyber-physical smart grids [1]–[3]. Well-designed FDIAs have the capability to circumvent conventional residual-based bad data detection of a state estimation [4], resulting in serious consequences for physical power grids. Compared with certain cyber-attacks (for example, malware attacks), FDIAs aim at maliciously manipulating measurement data that are generated by sensors. FDIAs can pass the residual-based bad data detection because the maliciously modified measurements can perfectly obey the power flow equations [4]. To address this issue, many data-driven methods have been proposed [5], and among such methods, machine-learning-based approaches have achieved a state-of-the-art detection performance [6].

Despite their success, most of the existing methods [7]–[9] are specifically focused on dc-model power systems and are not well suited to real-world power systems, which are based on the ac model. For example, Wang *et al.* [8] proposed a CNN-based method for detecting FDIAs in dc-model power systems by capturing the inconsistency and co-occurrence dependence in malicious measurement data. In recent years, to effectively detect FDIAs in ac-model systems, some deep-learning-based methods have been proposed [10]–[14]. Kundu *et al.* [11] proposed an autoencoder-based unsupervised learning method to detect FDIAs in ac-model systems. In addition, Zhang *et al.* [14] proposed a semisupervised deep learning approach by integrating an autoencoder into a generative adversarial network. Compared with the method in [11], labeled false measurement data are used to train the network model. In both methods, the measurement data are used as the training dataset. Different from these two methods [11], [14], Yu *et al.* [13] proposed using state variable values estimated from measurement data as the training dataset to train a deep neural network for FDIA detection. However, the estimated state values as the training dataset may incur some potential risks. For example, the second-hand state values estimated from the measurement data may suffer from data noise.

Although these machine-learning-based methods have achieved some success in detecting FDIAs in ac-model systems, the accuracy can be further improved. Most of those methods focus on temporal relationships between time-series measurement data and pay less attention to the spatial relationship of the measurement data between buses and transmission lines. For example, the autoencoder-based method proposed in [11] only considers the temporal relationship of normal

Manuscript received March 25, 2021; revised July 6, 2021; accepted July 29, 2021. Date of publication August 6, 2021; date of current version December 6, 2021. This work was supported by the Australian Research Council under Discovery Grant DP190103660 and under Linkage Grant LP180100663. Paper no. TII-21-1347. *(Corresponding author: Jiankun Hu.)*

Xuefei Yin and Jiankun Hu are with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia (e-mail: xuefei.yin@unsw.edu.au; j.hu@adfa.edu.au).

Yanming Zhu is with the School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: yanming.zhu@unsw.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3102332.

Digital Object Identifier 10.1109/TII.2021.3102332

time-series measurement data. Because the measurement data at a particular time step are treated as a 1-D input, the spatial relationship between buses and lines is not considered in the power grid. In the method proposed in [14], the measurement data captured at a particular time step for a power grid are simply treated as a 1-D sample, resulting in a loss of spatial relationships between buses and lines. Yu *et al.* [13] proposed a gated recurrent unit (GRU) network-based detector to model the temporal relationship between time-series measurement data. Compared with the two aforementioned methods [11], [14], one major difference is that the estimated state values instead of the original measurement data are used as the training dataset to train the FDIA detector. However, the estimated state values obtained from measurement data cannot thoroughly represent the spatial–temporal data pattern of the measurement data.

As discussed above, most of the aforementioned state-of-the-art methods [10]–[14] focus on modeling the temporal relationships in a power grid from time-series measurement data using autoencoders, a recurrent neural network, or a GRU; however, they fail to consider the spatial relationships between buses and lines. In addition, existing machine-learning-based FDIA detection methods attempt to learn a distinctive data pattern or distribution for the entire power grid between the normal measurement data and the malicious measurement modified using the FDIAs. Most, if not all, of these existing methods fail to consider the mutual relationship between subgrids of a power grid. Because a power grid is a meshed physical system, a mutual relationship exists between subgrids. Hence, data distribution of the measurement data in each subgrid can be utilized to collaboratively detect the FDIA patterns.

With the rapid development of microservices, some studies have explored the application of microservice technology in the field of smart grids [15]–[17]. Although a microservice-based architecture offers many benefits for smart grids, with the deregulation of the power systems [18], a power grid system is run by many different companies competing with each other. Therefore, the privacy of their local system data needs to be protected. To address these issues, we propose a subgrid-oriented privacy-preserving microservice framework integrating a well-designed spatial–temporal neural network for FDIA detection in ac power systems. The experimental results based on the public benchmark dataset SimBench [19] show that compared with the state-of-the-art methods, the proposed framework achieves significant improvements in terms of precision, recall, and F_1 score.

The main contributions of this article are summarized as follows.

1) Conventional centralized methods do not offer data privacy protection of local measurement data. We propose a novel subgrid-oriented microservice framework for FDIA detection in smart grids by collaboratively learning the relationship between a specific subgrid and the remaining subgrids. Subgrid-level features are learned using subgrid models applied to represent the subgrids. A supervising model is designed to integrate these features and collaboratively detect FDIAs. The proposed framework has three major benefits: data privacy preservation, parallel computing, and low latency.
2) Compared with most of the existing methods that focus on temporal relationship between the measurement data, we propose a novel spatial–temporal neural network to learn a subgrid-level representative feature to represent the spatial–temporal relationship between time-series bus/line measurement data. Network layers are designed to learn equal dimension representations for all bus/line measurement data. Fully connected perceptron layers are designed to model the spatial relationship between the bus/line representations. Long short-term memory layers are integrated into the neural network to effectively learn the temporal relationship from time-series measurement data.
3) Compared with most of the existing methods that do not consider the spatial relationship between bus and line measurement data, we propose a neural network architecture to model such spatial relationship. Because a smart grid is a meshed physical network, where a bus/line is mutually dependent upon its connecting buses/lines, we propose the use of a fully connected perceptron layer to model such a relationship between one bus/line and the remaining buses/lines.
4) Compared with the existing methods that treat the measurement as a 1-D input, we propose an efficient learning mechanism for bus and line measurement data of different dimensions to facilitate the subsequent model learning. With this mechanism, we propose learning the bus and line measurements separately, and thus, the approach is flexible in terms of the neural network design. To facilitate the subsequent model training, we propose learning an equal dimension representative feature for all bus/line measurement data.

The rest of this article is organized as follows. Section II reviews related state-of-the-art methods on machine-learning-based FDIA detection. Section III provides necessary background knowledge regarding FDIA in ac-model power systems. The proposed framework is presented in detail in Section IV. The experimental setting and evaluation results are then covered in Section V. Finally, Section VI concludes this article.

II. Related Work

In smart grids, errors in measurement data may be generated for various reasons, such as a poor telecommunication medium, meters with finite accuracy, and reading failures [20]. This type of error can usually be efficiently detected and removed by the residual-based bad data detection function in the state estimation. However, Liu *et al.* [4] proved that a type of well-designed malicious measurement vector generated according to system equations, i.e., FDIA, can successfully circumvent the conventional residual-based bad data detector in dc-model power systems; in addition, the study in [21] expanded this attack to ac-model power systems [4]. In recent years, various methods have been developed to efficiently detect and defend against FDIAs [22]. However, most of these studies focus on protecting some special meters or encrypting the measurement data instead of detecting the FDIAs from the measurement data [23], [24]. Some methods have been developed to detect FDIAs, such as statistic-based method [25], graph-theory-based methods [26], [27], a time-series simulation [28], and machine-learning-based methods [7]–[9]. However, most of the aforementioned methods focus on FDIA detection for dc-model power systems. In this study, an FDIA detection neural network is proposed for ac-model power systems.

In recent years, to effectively detect FDIAs in ac-model power systems, some deep-learning-based methods have been proposed [10]–[14]. Kundu *et al.* [11] developed an attention-based autoencoder detector by capturing the relationships between normal measurement data. Its key hypothesis is that an autoencoder model trained on clean measurement data can be used to infer whether measurement data are positive or negative. As an advantage, historical time-series measurement data are used to train the detector. However, this detector is only trained on normal measurement data without FDIA measurement data. In addition, this method does not consider the spatial relationship between buses and lines. By comparison, we proposed a neural network based on fully connected perceptron layers to capture the spatial relationship. Furthermore, not only the normal measurement data but also malicious measurement data are utilized to train the neural network. The experiment results show that the proposed neural network achieves a better accuracy in FDIA detection in terms of the recall, precision, and F_1 score. Zhang *et al.* [14] proposed a semisupervised deep learning approach by integrating an autoencoder into a generative adversarial network. The autoencoder is used to reduce the dimensions of the input data and extract representative features. The generative adversarial network is then used to capture the nonconformity between malicious and normal measurement data. Unlike the method in [11], labeled FDIA measurement data are used in this method to train the network model. Compared with our framework, this method does not consider the spatial relationship between buses and lines. In addition, this method does not take into account the mutual relationship between subgrids. Different from these two methods [11], [14], Yu *et al.* [13] proposed using state values estimated from measurement data as the training dataset instead of the measurement data to train a deep neural network for FDIA detection. However, estimated state values as the training dataset may incur some potential risks. For example, the second-hand state values estimated from the measurement data may suffer from data noise. Therefore, to avoid this risk, first-hand measurement data are utilized to train the proposed neural network.

With the rapid development of smart grids and microservices, some research studies are exploring the application of microservice technology in the field of smart grids [15]–[17]. Liang *et al.* [15] proposed a cloud-based microservice architecture for a real-time data process in the supervisory control and data acquisition (SCADA) power control system. As an advantage, in this article, the functions of the data collection, processing, storage, interaction, and display in the SCADA system are analyzed. In addition, a front collection subsystem and front collection service are discussed in the proposed cloud-based SCADA microservice system. However, this article mainly focuses on the microservice application in the SCADA control system from the concept level. Huang *et al.* [16] investigated the design scheme of a microservice architecture, elaborated on key technologies of the microservice, and proposed a microservice architecture for a power grid dispatching control system. The experimental results show that the proposed microservice system achieves an improvement in terms of fault tolerance, maintainability, and scalability. Lyu *et al.* [17] proposed a microservice-based architecture for an energy management system. As an advantage, this architecture improves the load performance and scalability of an energy management system. Power systems are currently being deregulated in many countries. Data privacy protection for these competing local operators has become an emergent issue [18].

III. Preliminary Knowledge

In this section, we mainly provide some necessary background knowledge on a state estimation and the bad data detection mechanism in ac-model power systems. Following that, we present a general approach to the design of FDIAs applied against conventional bad data detection.

A. AC State Estimation

State estimation is an essential function in a modern power management system. The function of the state estimation is to determine the optimal state for a power system according to proper measurement data. For example, in an N-bus system, there are $n = 2N - 1$ values in state $\mathbf{x}$, as denoted by $\mathbf{x} = [\theta_2, \theta_3, \ldots, \theta_N, V_1, V_2, \ldots, V_N]^T$, where V_i and θ_i are the bus states at bus i, and the phase angle θ_1 as the reference angle is normally set to zero radians. The state of a power system can be thoroughly described by the voltage magnitudes and phase angles at all buses [29]. Then, an entire power grid can be mathematically modeled according to its optimal state, grid topology, and physical parameters of various electrical equipment. The measurement data usually include bus and line measurement data. Bus measurement data are typically comprised of the voltage magnitude, active power injection, and reactive power injection. The line measurement data are typically comprised of an active/reactive power flow at two ends of the lines. For ac power systems, the nonlinear relationship between the measurement data and the states can be formulated as follows [13]:

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

where $\mathbf{z} \in \mathbb{R}^m$ is the measurement vector at a time step, $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{e} \in \mathbb{R}^m$ is a measurement error vector with a zero mean, and $\mathbf{h}(\mathbf{x})$ is a set of m nonlinear power functions of the measurement and state. Each error $e_i \in \mathbf{e}$ for measurement $z_i \in \mathbf{z}$ is assumed to be independent and follow a zero-mean Gaussian distribution $\mathcal{N}(0, \sigma_i^2)$. Here, $h_i(\mathbf{x}) \in \mathbf{h}(\mathbf{x})$ represents the power function between the measurement i and state vector $\mathbf{x}$. Specifically, given $\mathbf{x} = [\theta_2, \theta_3, \ldots, \theta_N, V_1, V_2, \ldots, V_N]^T$, the power flows at a line connecting buses i and j can be formulated as

$$P_{ij} = V_i^2(g_{si} + g_{ij}) - V_i V_j (g_{ij} \cos\theta_{ij} + b_{ij} \sin\theta_{ij})$$

$$Q_{ij} = -V_i^2(b_{si} + b_{ij}) - V_i V_j (g_{ij} \sin\theta_{ij} - b_{ij} \cos\theta_{ij})$$

where $\theta_{ij} = \theta_i - \theta_j$. $g_{ij} + jb_{ij}$ and $g_{si} + jb_{si}$ are the parameters.

The state estimation is an optimization process used to determine the optimal state vector $\mathbf{x}$ by solving the following weighted least-squares optimization problem:

$$\mathbf{J}(\mathbf{x}) = (\mathbf{z} - \mathbf{h}(\mathbf{x}))^{\mathbf{T}} \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\mathbf{x})) \tag{2}$$

where $\mathbf{R} = \text{diag}[\sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2]$ is a weight matrix, whose elements indicate the measurement accuracy of those measurements. The function $\mathbf{J}(\mathbf{x})$ can be minimized using iterative approximation methods (such as the Newton–Raphson method). The state estimation can then be formulated using

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathbf{J}(\mathbf{x}) \tag{3}$$

where $\hat{\mathbf{x}}$ indicates the optimal states estimated on the measurement data $\mathbf{z}$.

B. Residual-Based Bad Data Detection

Because measurement data (bad data) with large noise may lead to significant errors in the state values obtained from the state estimation process, bad data detection is, therefore, developed to detect whether the measurement data contain bad data. One commonly used method is the *Chi-square* test [30].

The residual is defined as the difference between the original measurement values $\mathbf{z}$ and values obtained from the power functions $\mathbf{h}(\hat{\mathbf{x}})$ with the optimal state values $\hat{\mathbf{x}}$, formulated using

$$\mathbf{r} = \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}). \tag{4}$$

Let $\gamma_i = r_i/\sigma_i$, where $r_i \in \mathbf{r}$. The variable γ_i then follows the standard normal distribution $\mathcal{N}(0,1)$. Here, Υ is defined as the sum of the square of γ_i, which is formulated by

$$\Upsilon = \sum_{i=1}^{m} \gamma_i^2. \tag{5}$$

Then, Υ follows a chi-square distribution χ^2_{m-n} with $m-n$ degrees of freedom. According to the theory of χ^2 testing, the value of Υ can be utilized to determine whether the measurement data contain bad data [30]. Therefore, bad data can be detected and removed from the measurement data; in addition, correct state values can be obtained by reconducting the state estimation process according to the clean measurement data.

C. False Data Injection Attack

Because measurement data maliciously modified using FDIAs are generated according to the power functions, they can circumvent the residual-based bad data detection mechanism. According to (1), malicious measurement data can be generated by deliberately manipulating certain special measurements. The stealth FDIA can be deduced from the residual-based bad data detection mechanism, as follows [21]:

$$\begin{aligned} \|\mathbf{z_a} - \mathbf{h}(\hat{\mathbf{x}}_{\mathbf{bad}})\| &= \|\mathbf{z} + \mathbf{a} - \mathbf{h}(\hat{\mathbf{x}} + \mathbf{c})\| \\ &= \left\| \begin{pmatrix} \mathbf{z_1} \\ \mathbf{z_2} + \mathbf{a_2} \end{pmatrix} - \begin{pmatrix} \mathbf{h_1}(\hat{\mathbf{x}}_\mathbf{1}) \\ \mathbf{h_2}(\hat{\mathbf{x}}_\mathbf{1}, \hat{\mathbf{x}}_\mathbf{2} + \mathbf{c}) \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} \mathbf{z_1} \\ \mathbf{z_2} \end{pmatrix} - \begin{pmatrix} \mathbf{h_1}(\hat{\mathbf{x}}_\mathbf{1}) \\ \mathbf{h_2}(\hat{\mathbf{x}}_\mathbf{1}, \hat{\mathbf{x}}_\mathbf{2}) \end{pmatrix} \right\| \\ &= \|\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})\| \end{aligned} \tag{6}$$

where variables with a subscript of 1 indicate those that stay untouched during the attack, and variables with a subscript of 2 indicate those that will be maliciously modified (one of the effective approaches used to determine the attacked measurements is presented in [21]). Vector $\mathbf{c}$ denotes an attack vector against the selected state variables; in addition, the vector $\mathbf{a}$ denotes the required changes in the attacked measurements. If (7) is satisfied, the attack is, thus, a stealth attack and can circumvent the residual-based bad data detection mechanism as follows:

$$\mathbf{a_2} = \mathbf{h_2}(\hat{\mathbf{x}}_\mathbf{1}, \hat{\mathbf{x}}_\mathbf{2} + \mathbf{c}) - \mathbf{h_2}(\hat{\mathbf{x}}_\mathbf{1}, \hat{\mathbf{x}}_\mathbf{2}). \tag{7}$$

Therefore, the malicious attack measurement can fool the bad data detection mechanism.

IV. Proposed Framework

In this section, we present details of the proposed subgrid-oriented microservice framework for FDIA detection in ac-model power systems. The FDIA detection task is formulated as a multilabel classification problem to detect whether measurement data are malicious. The proposed microservice framework is composed of two main components: a subgrid-level spatial–temporal architecture as a subgrid microservice module, denoted by $\mathbf{M}_{\text{sub}}^{\text{ST}}$, and a supervising architecture as the supervising microservice module, denoted by $\mathbf{M}_{\text{sup}}$. The $\mathbf{M}_{\text{sub}}^{\text{ST}}$ aims to learn a representative feature for a subgrid to represent the spatial–temporal relationship between time-series bus/line measurement data. In addition, $\mathbf{M}_{\text{sup}}$ aims to integrate these representative features and collaboratively detect FDIAs in the power grid. Compared with federated learning, one similarity is that there is no training data sharing during the entire training process. Therefore, it is helpful for preserving data privacy. One of differences is that there is no model sharing in the proposed framework, while federated learning will share the global model with other local clients during the training process. Therefore, the proposed framework further strengths privacy preservation in the deregulated smart grids. Fig. 1 provides an overview of the proposed microservice framework.

A. Subgrids and Measurement Data

In this framework, a power grid G is logically represented by several subgrids, denoted by $G = \{G_1, G_2, \ldots, G_\tau\}$, where τ denotes the number of subgrids. For each subgrid G_i, we design a spatial–temporal neural network $\mathbf{M}_{\text{sub-}G_i}^{\text{ST}}$ to learn a representative feature $\mathbf{O}_{G_i}$ and represent the spatial–temporal

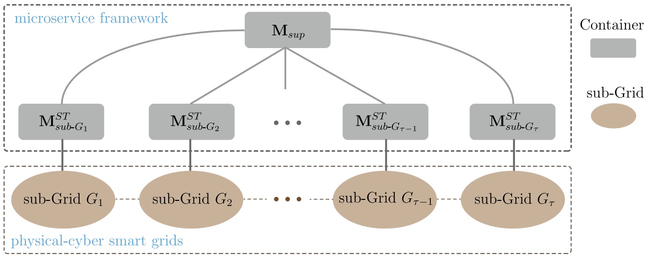

microservice framework
$\mathbf{M}_{sup}$
Container
sub-Grid
$\mathbf{M}^{ST}_{sub\text{-}G_1}$
$\mathbf{M}^{ST}_{sub\text{-}G_2}$
$\mathbf{M}^{ST}_{sub\text{-}G_{\tau-1}}$
$\mathbf{M}^{ST}_{sub\text{-}G_\tau}$
sub-Grid G_1
sub-Grid G_2
sub-Grid $G_{\tau-1}$
sub-Grid G_τ
physical-cyber smart grids

Fig. 1. Diagram of the proposed microservice framework.

relationship between time-series bus/line measurement data. Because the bus and line measurement data are usually of different dimensions, we design network layers to learn equal dimension representations for them. Then, we design neural architectures to model the spatial–temporal relationship between bus/line representations, as denoted by $\mathbf{O}_{G_i} = f_t^{G_i}(f_s^{G_i}(D^b_{G_i}, D^l_{G_i})) = \mathbf{M}^{\text{ST}}_{\text{sub-}G_i}(D^b_{G_i}, D^l_{G_i})$, where $f_s^{G_i}$ denotes the spatial architecture, $f_t^{G_i}$ denotes the temporal architecture, and $D^b_{G_i}$ and $D^l_{G_i}$ stand for the bus and line measurement data, respectively.

For an n_b-bus system with n_l lines, the measurement data normally consist of bus and line measurement data. The measurement data at bus b_i are typically comprised of the voltage magnitude V_i, active power injection P_i, and reactive power injection Q_i. The measurement data at line l_{ij} connecting buses b_i and b_j are typically composed of the following:

1) P_s, the active power flow at the "*from*" side;
2) Q_s, the reactive power flow at the "*from*" side;
3) P_t, the active power flow at the "*to*" side;
4) Q_t, the reactive power flow at the "*to*" side.

We denote the bus measurement data at time t_i as $\mathbf{z}^b_{t_i} \in \mathbb{R}^{n_b \times 3}$ and denote the line measurement data at time t_i as $\mathbf{z}^l_{t_i} \in \mathbb{R}^{n_l \times 4}$. Hence, the entire measurement data at time t_i can be denoted as $\mathbf{z}_{t_i}$, which is composed of $\mathbf{z}^b_{t_i}$ and $\mathbf{z}^l_{t_i}$, as formulated using $\mathbf{z}_{t_i} = \{\mathbf{z}^b_{t_i}, \mathbf{z}^l_{t_i}\}$. For the time-series measurement data, we denote this as $\mathbf{Z}_{t_i} = \{\mathbf{z}_{t_{i-K}}, \ldots, \mathbf{z}_{t_{i-1}}, \mathbf{z}_{t_i}\}$, where K is the time window. For convenience, we also represent it as $\mathbf{Z}_{t_i} = \{\mathbf{Z}^b_{t_i}, \mathbf{Z}^l_{t_i}\}$, where $\mathbf{Z}^b_{t_i} = \{\mathbf{z}^b_{t_{i-K}}, \ldots, \mathbf{z}^b_{t_{i-1}}, \mathbf{z}^b_{t_i}\}$ and $\mathbf{Z}^l_{t_i} = \{\mathbf{z}^l_{t_{i-K}}, \ldots, \mathbf{z}^l_{t_{i-1}}, \mathbf{z}^l_{t_i}\}$. For measurement data $\mathbf{z}_{t_i}$, we define its label as follows:

$$y_{t_i} = \begin{cases} 1, & \text{if } \mathbf{z}_{t_i} \text{ is maliciously manipulated by FDIAs} \\ & \text{and can circumvent the bad data detection} \\ 0, & \text{if } \mathbf{z}_{t_i} \text{ is normal measurements without FDIAs} \end{cases}. \tag{8}$$

During the experiments, the bad data detection mechanism embodied in the commercial software PowerFactory-2017-SP4[1] is utilized to generate the labeled data.

B. $\mathbf{M}^{ST}_{sub}$: Subgrid-Level Spatial–Temporal Microservice Architecture

The proposed module $\mathbf{M}^{\text{ST}}_{\text{sub}}$ aims to learn a representative feature $\mathbf{O}_{G_i}$ for each subgrid G_i to represent the spatial–temporal relationship between time-series bus/line measurement data. The model architecture is shown in Fig. 2. The corresponding values of the related parameters are summarized in Table I, where $f_*^{c\text{-}n\text{-}a}$ indicates a convolution layer followed by a batch normalization and an activation exponential linear unit (ELU), and $f_*^{l\text{-}a}$ indicates a linear perceptron layer followed by an activation ELU.

TABLE I
CONFIGURATION OF PARAMETERS IN $\mathbf{M}^{\text{ST}}_{\text{sub}}$

Layer	**Kernel Size**	**Kernel Numbers**
$f_{1\text{-}b}^{c\text{-}n\text{-}a}$	1×3	6
$f_{1\text{-}l}^{c\text{-}n\text{-}a}$	1×4	6
$f_2^{c\text{-}n\text{-}a}$	1×6	3
$f_3^{l\text{-}a}$	$n_{bl} \times 3$	n_{bl}
$f_4^{c\text{-}n\text{-}a}$	$n_{bl} \times 1$	40
Layer	**Input Numbers**	**Hidden Numbers**
f_5^{LSTM}	40	40
f_6^{LSTM}	40	20
f_7^{LSTM}	20	20

In the $\mathbf{M}^{\text{ST}}_{\text{sub}}$ module, we first design two network layers, $f_{1\text{-}b}^{c\text{-}n\text{-}a}$ and $f_{1\text{-}l}^{c\text{-}n\text{-}a}$, to learn equal dimension representations for buses and lines. Compared with the methods in the literature [11], [13], [14], whereas bus and line measurements are mixed as a 1-D input, our module has the following advantages: 1) the measurement data for each bus and line are treated separately, which can retain the topology information, and 2) equal dimension presentations are learned to represent each bus and line facilitating the subsequent model learning. The construction of the network layers is formulated as follows:

$$\begin{cases} \mathbf{O}^b_1 = & f_{1\text{-}b}^{c\text{-}n\text{-}a}(\mathbf{Z}^b_{t_i}) = f^a(f^n(\mathbf{Z}^b_{t_i} * \mathbf{W}^b_1 + \mathbf{a}^b_1)) \\ \mathbf{O}^l_1 = & f_{1\text{-}l}^{c\text{-}n\text{-}a}(\mathbf{Z}^l_{t_i}) = f^a(f^n(\mathbf{Z}^l_{t_i} * \mathbf{W}^l_1 + \mathbf{a}^l_1)) \end{cases}$$

where $\mathbf{W}^b_1 \in \mathbb{R}^{3\times 6}$, $\mathbf{W}^l_1 \in \mathbb{R}^{4\times 6}$, and $\mathbf{a}^b_1$ and $\mathbf{a}^l_1$ are the additive bias. Here, $\mathbf{O}_1$ is obtained by concatenating $\mathbf{O}^b_1$ and $\mathbf{O}^l_1$, where $n_{bl} = n_b + n_l$. The layer $f_2^{c\text{-}n\text{-}a}$ is designed to further model hidden representative features, which are formulated using

$$\mathbf{O}_2 = f_2^{c\text{-}n\text{-}a}(\mathbf{O}_1) = f^a(f^n(\mathbf{O}_1 * \mathbf{W}_2 + \mathbf{a}_2))$$

where $\mathbf{W}_2 \in \mathbb{R}^{6\times 3}$ and $\mathbf{a}_2$ is the additive bias. To model the spatial relationship between a bus/line and the remains of the buses/lines, we design the layer $f_3^{l\text{-}a}$, which is formulated using

$$\mathbf{O}_3 = f_3^{l\text{-}a}(\mathbf{O}_2) = f^a(\mathbf{O}_2 * \mathbf{W}_3 + \mathbf{a}_3)$$

where $\mathbf{W}_3 \in \mathbb{R}^{3\times n_{bl} \times n_{bl}}$, and $\mathbf{a}_3$ is the additive bias. The layer $f_4^{c\text{-}n\text{-}a}$ aims to model a fixed-length spatial representation for measurement data at a time step, which is formulated using

$$\mathbf{O}_4 = f_4^{c\text{-}n\text{-}a}(\mathbf{O}_3) = f^a(f^n(\mathbf{O}_3 * \mathbf{W}_4 + \mathbf{a}_4))$$

where $\mathbf{W}_4 \in \mathbb{R}^{n_{bl}\times 40}$, and $\mathbf{a}_4$ is the additive bias. To model the temporal relationship between time-series spatial representations obtained from the layer $f_4^{c\text{-}n\text{-}a}$, three LSTM unites are

[1] [Online]. Available: https://www.digsilent.de/en/newsreader/digsilent-releases-powerfactory-2017-sp4.html

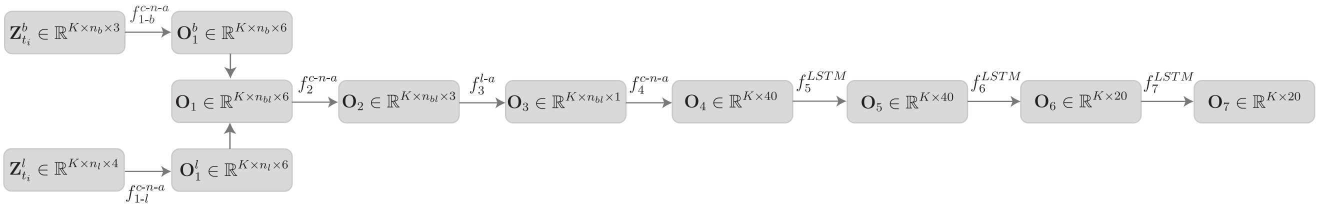

$\mathbf{Z}^b_{t_i} \in \mathbb{R}^{K\times n_b \times 3}$
$f^{c\text{-}n\text{-}a}_{1\text{-}b}$
$\mathbf{O}^b_1 \in \mathbb{R}^{K\times n_b \times 6}$
$\mathbf{O}_1 \in \mathbb{R}^{K\times n_{bl} \times 6}$
$f^{c\text{-}n\text{-}a}_2$
$\mathbf{O}_2 \in \mathbb{R}^{K\times n_{bl} \times 3}$
$f^{l\text{-}a}_3$
$\mathbf{O}_3 \in \mathbb{R}^{K\times n_{bl} \times 1}$
$f^{c\text{-}n\text{-}a}_4$
$\mathbf{O}_4 \in \mathbb{R}^{K\times 40}$
f^{LSTM}_5
$\mathbf{O}_5 \in \mathbb{R}^{K\times 40}$
f^{LSTM}_6
$\mathbf{O}_6 \in \mathbb{R}^{K\times 20}$
f^{LSTM}_7
$\mathbf{O}_7 \in \mathbb{R}^{K\times 20}$
$\mathbf{Z}^l_{t_i} \in \mathbb{R}^{K\times n_l \times 4}$
$\mathbf{O}^l_1 \in \mathbb{R}^{K\times n_l \times 6}$
$f^{c\text{-}n\text{-}a}_{1\text{-}l}$

Fig. 2. Module $\mathbf{M}^{\text{ST}}_{\text{sub}}$, i.e., subgrid-level spatial–temporal neural network architecture for each subgrid.

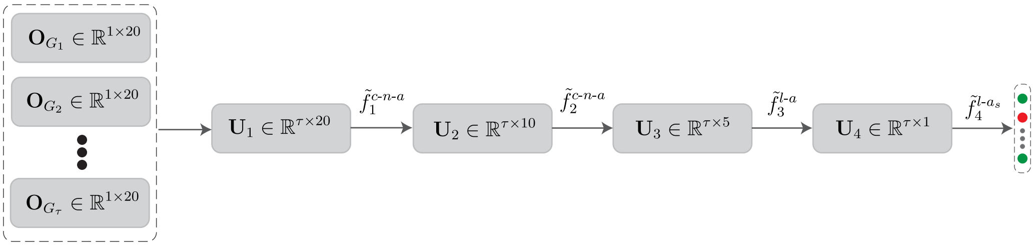

$\mathbf{O}_{G_1} \in \mathbb{R}^{1\times 20}$
$\mathbf{O}_{G_2} \in \mathbb{R}^{1\times 20}$
$\mathbf{O}_{G_\tau} \in \mathbb{R}^{1\times 20}$
$\mathbf{U}_1 \in \mathbb{R}^{\tau\times 20}$
$\tilde{f}^{c\text{-}n\text{-}a}_1$
$\mathbf{U}_2 \in \mathbb{R}^{\tau\times 10}$
$\tilde{f}^{c\text{-}n\text{-}a}_2$
$\mathbf{U}_3 \in \mathbb{R}^{\tau\times 5}$
$\tilde{f}^{l\text{-}a}_3$
$\mathbf{U}_4 \in \mathbb{R}^{\tau\times 1}$
$\tilde{f}^{l\text{-}a_s}_4$

Fig. 3. Module $\mathbf{M}_{\text{sup}}$, supervising neural network architecture.

integrated into the framework, which is formulated using

$$\begin{cases} \mathbf{O}_5 = & f^{\text{LSTM}}_5(\mathbf{O}_4) \\ \mathbf{O}_6 = & f^{\text{LSTM}}_6(\mathbf{O}_5). \\ \mathbf{O}_7 = & f^{\text{LSTM}}_7(\mathbf{O}_6) \end{cases}$$

As an advantage, LSTM can successfully capture temporal information of a time-series data and can avoid gradient vanishing and exploding. Because we focus on the status of the power grid at the current moment t_i, the feature vector at the last row in $\mathbf{O}_7$, denoted by $\mathbf{O}_{G_i}$, is, therefore, obtained as the representative feature for the corresponding subgrid.

In summary, by applying the module $\mathbf{M}^{\text{ST}}_{\text{sub}}$ to each subgrid of $G = \{G_1, G_2, \ldots, G_\tau\}$, we can obtain a set of fixed-length representative features, $\{\mathbf{O}_{G_1}, \mathbf{O}_{G_2}, \ldots, \mathbf{O}_{G_\tau}\}$, which is formulated using

$$\mathbf{O}_{G_i} = \mathbf{M}^{\text{ST}}_{\text{sub-}G_i}(\mathbf{Z}_{t_i}).$$

C. $\mathbf{M}_{sup}$: Supervising Microservice Architecture

The proposed module $\mathbf{M}_{\text{sup}}$ aims to integrate these representative features and collaboratively detect FDIAs in the power grid G. The model architecture is shown in Fig. 3. The corresponding values of the related parameters are summarized in Table II, where $\tilde{f}^{c\text{-}n\text{-}a}_*$ indicates a convolution layer followed by a batch normalization and an activation ELU, $\tilde{f}^{l\text{-}a}_3$ indicates a linear perceptron layer followed by an activation ELU, and $\tilde{f}^{l\text{-}a_s}_4$ indicates a linear perceptron layer followed by an activation function sigmoid.

Through the modules $\mathbf{M}^{\text{ST}}_{\text{sub-}G_*}$, the subgrid-level representative features $\{\mathbf{O}_{G_1}, \mathbf{O}_{G_2}, \ldots, \mathbf{O}_{G_\tau}\}$ are learned to represent the status of each subgrid. Here, $\mathbf{U}_1$ is obtained by concatenating $\{\mathbf{O}_{G_1}, \mathbf{O}_{G_2}, \ldots, \mathbf{O}_{G_\tau}\}$. The first two layers $\tilde{f}^{c\text{-}n\text{-}a}_1$ and $\tilde{f}^{c\text{-}n\text{-}a}_2$ are designed to further extract the hidden features for each subgrid and simultaneously reduce the dimension of the features, which is formulated using

TABLE II
CONFIGURATION OF THE PARAMETERS IN $\mathbf{M}^{\text{ST}}_{\text{sub}}$

Layer	Kernel Size	Kernel Numbers
$\tilde{f}^{c\text{-}n\text{-}a}_1$	1×20	10
$\tilde{f}^{c\text{-}n\text{-}a}_2$	1×10	5
$\tilde{f}^{l\text{-}a}_3$	$\tau \times 5$	τ
$\tilde{f}^{l\text{-}a_s}_4$	$\tau \times 1$	τ

$$\begin{cases} \mathbf{U}_2 = & \tilde{f}^{c\text{-}n\text{-}a}_1(\mathbf{U}_1) = \tilde{f}^a(\tilde{f}^n(\mathbf{U}_1 * \tilde{\mathbf{W}}_1 + \tilde{\mathbf{a}}_1)) \\ \mathbf{U}_3 = & \tilde{f}^{c\text{-}n\text{-}a}_2(\mathbf{U}_2) = \tilde{f}^a(\tilde{f}^n(\mathbf{U}_2 * \tilde{\mathbf{W}}_2 + \tilde{\mathbf{a}}_2)) \end{cases}$$

where $\tilde{\mathbf{W}}_1 \in \mathbb{R}^{20\times 10}$, $\tilde{\mathbf{W}}_2 \in \mathbb{R}^{10\times 5}$, and $\tilde{\mathbf{a}}_1$ and $\tilde{\mathbf{a}}_2$ are the additive bias. To model the interactive relationship between a subgrid and the remains of the subgrids, we design a network layer $\tilde{f}^{c\text{-}n\text{-}a}_3$, which is formulated using

$$\mathbf{U}_4 = \tilde{f}^{c\text{-}n\text{-}a}_3(\mathbf{U}_3) = \tilde{f}^a(\tilde{f}^n(\mathbf{U}_3 * \tilde{\mathbf{W}}_3 + \tilde{\mathbf{a}}_3))$$

where $\tilde{\mathbf{W}}_3 \in \mathbb{R}^{5\times\tau\times\tau}$, and $\tilde{\mathbf{a}}_3$ is the additive bias. The last network layer $\tilde{f}^{l\text{-}a_s}_4$ aims to map the learned representative features to a multilabel classification output, with each element in the output denoting whether the measurement data for the corresponding subgrid is attacked by the FDIAs. Specifically, the layer $\tilde{f}^{l\text{-}a_s}_4$ models the representative features using a linear perceptron layer followed by an activation function sigmoid, which is formulated using

$$\mathbf{y}_p = \tilde{f}^{l\text{-}a_s}_4(\mathbf{U}_4) = \tilde{f}^{a_s}(\mathbf{U}_4 * \tilde{\mathbf{W}}_4 + \tilde{\mathbf{a}}_4) \qquad (9)$$

where $\tilde{\mathbf{W}}_4 \in \mathbb{R}^{\tau\times\tau}$, and $\tilde{\mathbf{a}}_4$ is the additive bias.

D. Loss Function

The training goal is to minimize the multilabel cross entropy error between the true label and the model prediction, which is formulated using

$$f_{\text{loss}} = -\sum_{i=1}^{\tau}(y_i \log y_{p,i}+(1-y_i)\log(1-y_{p,i})) \qquad (10)$$

where τ is the number of classes/subgrids, $y_i \in \mathbf{y}$ is the true label for the current measurement data, and $y_{p,i} \in \mathbf{y}_p$ is the corresponding prediction result obtained through (9). The open-source machine learning software Pytorch-1.7.0[2] is utilized to implement and train the proposed neural network.

E. Summary

In the proposed framework, as shown in Fig 1, a smart grid is first divided into subgrids, denoted by $G = \{G_1, G_2, \ldots, G_\tau\}$. During the training stage, for each subgrid, a spatial–temporal neural network, as shown in Fig. 2 or Table I, is trained to learn a representative feature and represent the spatial–temporal relationship between the time-series bus/line measurement data. A supervising network, as shown in Fig. 3 or Table II, is trained to integrate these representative features and collaboratively detect FDIAs in the power grid. During the test stage, microservices are deployed by subgrid operators to run the corresponding subgrid models, whereas a supervising microservice is deployed to communicate with other microservices of the subgrid and run the supervising module for FDIA detection.

V. EXPERIMENTS

A. Dataset for FDIA Detection

During the experiment, two power grids from the public benchmark dataset SimBench[19] are utilized to evaluate the performance of the proposed framework on FDIA detection. The SimBench datasets used in this study are derived from German power systems. The grid dataset contains exhaustive time-series profiles for the load and generation with a resolution of 15 min for one year. This means there are 35 136 time steps of the measurement data, which are useful and convenient for evaluating the performance of FDIA detection. In addition, this dataset contains power grids with different levels of voltage, including extra-high-, high-, medium-, and low-voltage levels. It is, therefore, convenient to utilize this dataset and verify FDIA detection in power grids of different voltage levels.

The details regarding these two grids are summarized in Tables III and IV. The commercial software PowerFactory-2017-SP4 is utilized to calculate the power flow and detect the bad data during the stages of normal and FDIA measurement generation.

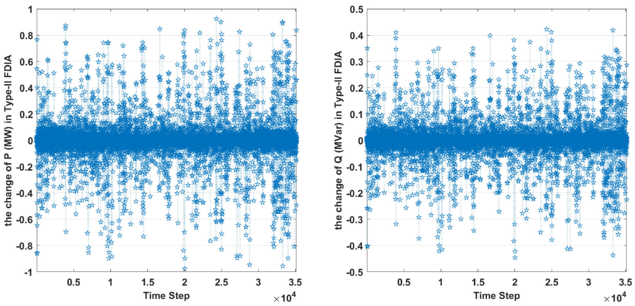
The method in [21] presented in Section III-C is used to establish the FDIAs. Three types of FDIAs are designed to comprehensively evaluate the performance of the proposed framework, categorized by the rate of change of the active power injection (P_i) on the target bus i.

TABLE III
GRID-A WITH CODE *1-MV-RURAL–0-NO_SW*

Element	Numbers	Comments
Bus	95	all of the buses are in service
Lines	101	6 lines are out of service
profiles	35,136	load and generation for one year
Bus measurement	285	95×3
Line measurement	380	$(101-6) \times 4$
sub-grids	5	19 buses for each sub-grid

TABLE IV
GRID-B WITH CODE *1-HVMV-URBAN-3.201-2-NO_SW*

Element	Numbers	Comments
Bus	255	all of the buses are in service
Lines	303	11 lines are out of service
profiles	35,136	load and generation for one year
Bus measurement	765	255×3
Line measurement	1,168	$(303-11) \times 4$
sub-grids	10	around 25 buses for each sub-grid



(a) (b)

Fig. 4. Statistical information on Type-II FDIA data in Grid-A. (a) Active power change at the target buses. (b) Reactive power change at the target buses.

1) Type-I FDIAs: The rate is within the range of $(50\%, 100\%]$.
2) Type-II FDIAs: The rate is within the range of $(25\%, 50\%]$.
3) Type-III FDIAs: The rate is within the range of $(5\%, 25\%]$.

For each type of FDIA, two FDIA datasets are generated by maliciously manipulating the voltage magnitude Vm or phase angle Va at the target bus, which is summarized as follows:

1) 35 136 Type-I FDIA data by manipulating Vm;
2) 35 136 Type-I FDIA data by manipulating Va;
3) 35 136 Type-II FDIA data by modifying Vm;
4) 35 136 Type-II FDIA data by manipulating Va;
5) 35 136 Type-III FDIA data by manipulating Vm;
6) 35 136 Type-III FDIA data by manipulating Va.

As an illustrated example, the statistical information on the Type-II FDIA data for Grid-A is shown in Fig. 4 .

[2] [Online]. Available: https://pytorch.org/

B. Deployment of the Proposed Framework

As shown in Fig. 1, in this framework, a widely distributed power grid is first divided into region-based subgrids. In the experiments, Grid-A is divided into five subgrids, and Grid-B is divided into ten subgrids, as shown in Tables III and IV, respectively. For each subgrid, a Docker-container-based microservice is deployed to run the corresponding local model $\mathrm{M}_{\mathrm{sub}}^{\mathrm{ST}}$. A Docker-container-based supervising microservice is deployed by the control center to communicate with the microservices of the other subgrids and run the supervising module $\mathrm{M}_{\mathrm{sup}}$ for FDIA detection. Each microservice runs in a Docker container located in a Pod in the open-source container orchestration engine Kubernetes.[3] These subgrid microservices are run in parallel with the necessary computing resources. The open-source software Kubernetes-1.2.0[4] and Docker-20.10.5[5] are utilized to deploy the proposed microservice framework. The proposed framework is suitable to be deployed in the low-resourced Internet of Things environment. This is because it contains subgrid models and a supervising model. Each subgrid model is running locally and independently. In our experimental setting, the storage of a subgrid model is less than 6 MB. Regarding the runtime support, the PyTorch Mobile runtime beta release[6] has provided such a function to deploy trained machine learning models in mobile devices or edge devices (such as laptop or Android/iOS phones).

C. Evaluation Metrics

The commonly used metrics, precision (Prec), recall (Rec), and F_1 score (F_1), are utilized to evaluate the performance of the FDIA detection [11], [13], [14], which is formulated as follows:

$$\mathrm{Prec} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$$

$$\mathrm{Rec} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$$

$$F_1 = 2 \times (\mathrm{Pre} \times \mathrm{Rec})/(\mathrm{Pre} + \mathrm{Rec})$$

where TP denotes the number of true FDIA samples predicted as FDIA data, FP denotes the number of normal samples predicted as FDIA data, TN denotes the number of true normal samples predicted as normal data, and FN denotes the number of FDIA samples predicted as normal data. Hence, TP + FN equals the total number of true FDIA samples in the dataset, and TN + FP equals the total number of true normal samples in the dataset.

D. Performance of FDIA Detection

In this section, we compare the FDIA detection performance of the proposed framework with that of two state-of-the-art methods: M-2020 [11] published in 2020 and M-2021 [14] published in 2021. M-2020 [11] proposed an autoencoder-based unsupervised learning method, which is trained using only clean or normal measurement data. In addition, M-2021 [14] proposed an FDIA method based on a generative adversarial network and an autoencoder.

TABLE V
COMPARISON OF TYPE-I FDIA DETECTION ON GRID-A

Methods	Grid-A		
	Prec (%)	*Rec* (%)	F_1 (%)
M-2020 [11]	96.804	96.113	96.457
M-2021 [14]	97.630	97.743	97.687
Ours	**99.154**	**99.479**	**99.316**

TABLE VI
COMPARISON OF TYPE-I FDIA DETECTION ON GRID-B

Methods	Grid-B		
	Prec (%)	*Rec* (%)	F_1 (%)
M-2020 [11]	96.794	95.795	96.292
M-2021 [14]	97.623	97.454	97.538
Ours	**99.162**	**99.354**	**99.258**

1) Case I: Type-I FDIA Detection:

Type-I FDIA datasets with large power injection within the range of $(50\%, 100\%]$ are first utilized to evaluate the performance of FDIA detection. Tables V and VI summarize a comparison between the proposed framework and the state-of-the-art methods, respectively.

As shown in Table V, the proposed framework achieves the highest precision of 99.154%, recall of 99.479%, and an F_1 score of 99.316%. Regarding the F_1 score, the proposed framework improves by approximately 2.96% and 1.67%, respectively, compared with M-2020 [11] and M-2021 [14]. Regarding the recall, the proposed framework achieves an improvement of approximately 3.50% in comparison with M-2020 and an improvement of 1.78% compared with M-2021. For the precision, the proposed framework improves by approximately 2.43% in comparison to M-2020 and 1.56% compared to M-2021.

As shown in Table VI, the proposed framework achieves the highest precision of 99.162%, recall of 99.354%, and F_1 score of 99.258%. Compared with M-2020, the proposed framework improves by approximately 2.45% in terms of precision, 3.72% in recall, and 3.08% in F_1 score. Compared with M-2021, the proposed framework achieves improvements of approximately 1.58%, 1.95%, and 1.76% in precision, recall, and F_1 score, respectively.

These improvements in precision, recall, and F_1 score can be explained as follows. In M-2020, the temporal relationship between historical measurement data is modeled using an autoencoder network. However, only normal measurement data are used to train the network. M-2021 improves this method and integrates malicious measurement data into the training process. As shown in Tables V and VI, M-2021 achieves a better performance in terms of precision, recall, and F_1 score. Compared with M-2021, the proposed framework takes into account the feature distributions of the subgrids and spatial–temporal relationship of the bus/line measurement data. From Tables V and

[3] [Online]. Available: https://kubernetes.io/docs/home/
[4] [Online]. Available: https://v1-20.docs.kubernetes.io/docs/home/
[5] [Online]. Available: https://www.docker.com/
[6] [Online]. Available: https://pytorch.org/mobile/home/

TABLE VII
COMPARISON OF TYPE-II DETECTION ON GRID-A

Methods	Grid-A		
	$Prec$ (%)	Rec (%)	F_1 (%)
M-2020 [11]	96.794	95.804	96.297
M-2021 [14]	97.624	97.473	97.548
Ours	**99.235**	**98.814**	**99.024**

TABLE VIII
COMPARISON OF TYPE-II DETECTION ON GRID-B

Methods	Grid-B		
	$Prec$ (%)	Rec (%)	F_1 (%)
M-2020 [11]	96.781	95.409	96.090
M-2021 [14]	97.615	97.106	97.360
Ours	**99.234**	**98.717**	**98.975**

TABLE IX
COMPARISON OF TYPE-III DETECTION ON GRID-A

Methods	Grid-A		
	$Prec$ (%)	Rec (%)	F_1 (%)
M-2020 [11]	96.778	95.322	96.045
M-2021 [14]	97.608	96.827	97.216
Ours	**99.203**	**98.409**	**98.804**

TABLE X
COMPARISON OF TYPE-III DETECTION ON GRID-B

Methods	Grid-B		
	$Prec$ (%)	Rec (%)	F_1 (%)
M-2020 [11]	96.761	94.801	95.771
M-2021 [14]	97.597	96.354	96.971
Ours	**99.202**	**98.274**	**98.735**

VI, we can see that the proposed framework achieves significant improvements in terms of the precision, recall, and F_1 score.

2) Case II: Type-II FDIA Detection: Type-II FDIA datasets with moderate power injection within the range of $(25\%, 50\%]$ are utilized to evaluate the performance of FDIA detection. Because the power injection change ratio is less than in a Type-I FDIA, the detection of a Type-II FDIA is more difficult than that of a Type-I FDIA. Tables VII and VIII summarize the comparison between the proposed framework with other state-of-the-art methods.

As shown in Tables VII and VIII, the proposed approach achieves the best performance in terms of precision, recall, and F_1 score at 99.235%, 98.814%, and 99.024%, respectively, on Grid-A and 99.234%, 98.717%, and 98.975%, respectively, on Grid-B. Compared with M-2020, the proposed framework achieves improvements of approximately 2.52% and 2.53% in terms of precision on Grid-A and Grid-B, respectively, approximately 3.14% and 3.47% in terms of recall, and approximately 2.83% and 3.00% in terms of F_1 score. Compared with M-2021, the proposed framework improves by approximately 1.65% and 1.66% in terms of precision on Grid-A and Grid-B, respectively, approximately 1.38% and 1.66% in terms of recall, and approximately 1.51% and 1.66% in terms of F_1 score.

Compared with the detection performance on Type-I FDIA datasets, there is a slight degradation in recall and F_1 score for all three methods. As an advantage, the proposed framework continuously achieves the best accuracy in comparison with the other two methods.

3) Case III: Type-III FDIA Detection: The Type-III FDIA datasets with a smaller change in power injection within the range of $(5\%, 25\%]$ are utilized to evaluate the performance of FDIA detection. As the power injection change ratio is smaller than that of the Type-I and Type-II FDIAs, the measurement data are not as disturbed. Hence, the detection of the Type-III FDIA is more difficult than that of the other two types of FDIAs. Tables IX and X summarize the comparison among the proposed framework and the state-of-the-art methods, respectively.

As shown in Tables IX and X, the proposed approach achieves the best performance in terms of precision, recall, and F_1 score at 99.203%, 98.409%, and 98.804%, respectively, on Grid-A and 99.202%, 98.274%, and 98.735%, respectively, on Grid-B. Compared with M-2020, the proposed framework achieves improvements of approximately 2.51% and 2.52% in terms of precision on Grid-A and Grid-B, respectively, approximately 3.24% and 3.66% in terms of recall, and approximately 2.87% and 3.09% in terms of F_1 score. Compared with M-2021, the proposed framework improves by approximately 1.63% and 1.64% in terms of precision on Grid-A and Grid-B, respectively, approximately 1.63% and 1.99% in terms of recall, and approximately 1.63% and 1.82% in terms of F_1 score.

E. Advantages of the Proposed Framework

1) Data Privacy: The proposed microservice-based framework provides a scheme for data privacy preservation. In traditional methods, all measurement data collected from the power grids need to be transmitted to the control center. There is no guarantee of data privacy. In the proposed framework, data privacy is mainly guaranteed by a scheme, in which the measurement data collected by each subgrid operator are not shared with the other operators. During the entire training process, measurement data for each subgrid are consistently held by the local operator; in addition, only a feature representation for each subgrid is transmitted to the supervising module for collaborative training. Therefore, the proposed framework is privacy preserving.

2) Comparison of Latency: Because smart grids are usually distributed remotely, the massive measurement data continuously generated at each time step need to be transmitted to the operation center for subsequent processes. In the proposed framework, measurement data for each subgrid are transmitted to a local operator instead of a distant control center. Therefore, at this stage, the proposed framework will significantly reduce the data latency. For example, for Grid-B in our experiments, the amount of transmission data for a fixed length feature representation is only 1% of the transmitted raw measurement data required for the conventional centralized methods. In addition, the proposed framework can effectively utilize parallel computations for each subgrid. Therefore, compared with conventional methods in which all measurement data are transmitted to the distant control center, our proposed method undoubtedly achieves a low latency.

3) Detection Performance: As discussed in Section V-D, compared with state-of-the-art methods, the proposed framework achieves significant improvements in terms of the precision, recall, and F_1 score evaluated on two benchmark datasets in three types of FDIAs.

4) Efficient Utilization of Computing Resources: The proposed framework can efficiently utilize computing resources. First, models trained for subgrids can efficiently run in microservices in a parallel manner. Each subgrid microservice is independent of the others. Second, because the container-based microservices in the proposed framework only need to be allocated necessary computing resources, this framework can efficiently utilize such resources.

F. Limitation and Future Research Direction

This article proposes a subgrid-oriented privacy-preserving microservice framework based on a deep neural network for FDIA detection in smart grids. Compared with conventional methods, the proposed framework can effectively model the feature distributions between subgrids and extract spatial–temporal representations for the subgrids. The experimental results evaluated on the three types of FDIAs show that the proposed framework achieves a state-of-the-art performance. However, as one limitation of the proposed framework, it cannot be used to identify the exact location of the FDIAs (which bus or line is attacked) in the subgrids. In other words, the proposed framework can only detect which subgrids are attacked. Identification of more specific locations of buses and lines being attacked will be a good research question.

VI. Conclusion

In this article, we proposed an efficient subgrid-oriented privacy-preserving microservice framework based on deep neural networks for FDIA detection in smart grids. Our article focused on the FDIAs, which are usually launched by external attackers who aim to compromise the power system by injecting well-designed false data into the measurement data. As the false data are designed according to power flow equations, FDIAs can pass the conventional residual-based bad data detection system. To defend against FDIAs from external attackers, we proposed the subgrid-oriented deep neural network framework by considering: 1) the spatial–temporal relationship between buses and lines; 2) the feature distributions between subgrids; and 3) the normal measurement data and malicious measurement data. Hence, the framework can detect FDIAs by learning patterns from the normal and malicious measurement data. Experimental results show that the proposed framework achieved a state-of-the-art performance. When the microservices are deployed in the cloud, cloud security mechanisms such as the existing 4Cs (cloud, clusters, containers, and code) Cloud Native Security tools can be used against various external/insider attacks. How to improve these security tools is beyond the scope of this article. Our framework can provide local system privacy protection against attacks from insiders (participants in the framework as local operators), because of its local model training function. The comprehensive experimental results on two public power grids showed that, compared with the state-of-the-art methods, the proposed framework achieves significant improvement in terms of the precision, recall, and F_1 score.

Acknowledgement

This work was undertaken with the assistance of resources and services from the National Computational Infrastructure, supported by the Australian Government.

References

[1] N. N. Tran, H. R. Pota, Q. N. Tran, X. Yin, and J. Hu, "Designing false data injection attacks penetrating AC-based bad data detection system and FDI dataset generation," *Concurrency Comput.: Pract. Experience*, 2020, Art. no. e5956.

[2] G. Q. Liang, J. H. Zhao, F. J. 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.

[3] J. Hu and A. V. Vasilakos, "Energy big data analytics and security: Challenges and opportunities," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2423–2436, Sep. 2016.

[4] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, pp. 1–33, 2011.

[5] 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.

[6] A. Sayghe *et al.*, "Survey of machine learning methods for detecting false data injection attacks in power systems," *IET Smart Grid*, vol. 3, pp. 581–595, 2020.

[7] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.

[8] S. Y. 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 Things J.*, vol. 7, no. 9, pp. 8218–8227, Sep. 2020.

[9] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2505–2516, Sep. 2017.

[10] M. Lu, L. Wang, Z. Cao, Y. Zhao, and X. Sui, "False data injection attacks detection on power systems with convolutional neural network," *J. Phys.: Conf. Ser.*, vol. 1633, 2020, Art. no. 012134.

[11] A. Kundu, A. Sahu, E. Serpedin, and K. Davis, "A3D: Attention-based auto-encoder anomaly detector for false data injection attacks," *Elect. Power Syst. Res.*, vol. 189, 2020, Art. no. 106795.

[12] M. Ashrafuzzaman *et al.*, "Detecting stealthy false data injection attacks in the smart grid using ensemble-based machine learning," *Comput. Secur.*, vol. 97, 2020, Art. no. 101994.

[13] 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.

[14] 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.

[15] S. Liang, C. He, W. Fang, Z. Zhou, Y. Li, and Y. Wang, "A SCADA platform architecture for cloud-based micro-service system with real time data process," in *Proc. IOP Conf. Ser.: Mater. Sci. Eng.*, 2019, Art. no. 042056.

[16] L. Huang, W. Zhuang, M. Sun, and H. Zhang, "Research and application of microservice in power grid dispatching control system," in *Proc. IEEE Inf. Technol., Netw., Electron. Autom. Control Conf.*, 2020, pp. 1895–1899.

[17] Z. Lyu, H. Wei, X. Bai, and C. Lian, "Microservice-based architecture for an energy management system," *IEEE Syst. J.*, vol. 14, no. 4, pp. 5061–5072, Dec. 2020.

[18] R. D. Christie, B. F. Wollenberg, and I. Wangensteen, "Transmission management in the deregulated environment," *Proc. IEEE*, vol. 88, no. 2, pp. 170–195, Feb. 2000.

[19] C. Spalthoff *et al.*, "Simbench: Open source time series of power load, storage and generation for the simulation of electrical distribution grids," in *Proc. Int. ETG Congr.*, 2019, pp. 1–6.

[20] A. Y. Lu and G. H. Yang, "False data injection attacks against state estimation in the presence of sensor failures," *Inf. Sci.*, vol. 508, pp. 92–104, 2020.

[21] G. Hug and J. A. Giampapa, "Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1362–1370, Sep. 2012.

[22] R. Deng, G. Xiao, R. Lu, H. Liang, and A. V. Vasilakos, "False data injection on state estimation in power system–Attacks, impacts, and defense: A survey," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 411–423, Apr. 2017.

[23] M. A. Mustafa, S. Cleemput, A. Aly, and A. Abidin, "A secure and privacy-preserving protocol for smart metering operational data collection," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6481–6490, Nov. 2019.

[24] B. B. Li, G. X. Xiao, R. X. Lu, R. L. Deng, and H. Y. Bao, "On feasibility and limitations of detecting false data injection attacks on power grid state estimation using D-FACTS devices," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 854–864, Feb. 2020.

[25] S. K. Singh, K. Khanna, R. Bose, B. K. Panigrahi, and A. Joshi, "Joint-transformation-based detection of false data injection attacks in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 89–97, Jan. 2018.

[26] Y. Guan and X. Ge, "Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 48–59, Mar. 2018.

[27] M. Jorjani, H. Seifi, and A. Y. Varjani, "A graph theory-based approach to detect false data injection attacks in power system ac state estimation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2465–2475, Apr. 2021.

[28] O. A. Beg, T. T. Johnson, and A. Davoudi, "Detection of false-data injection attacks in cyber-physical DC microgrids," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2693–2703, Oct. 2017.

[29] Z. Kazemi, A. A. Safavi, F. Naseri, L. Urbas, and P. Setoodeh, "A secure hybrid dynamic-state estimation approach for power systems under false data injection attacks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7275–7286, Dec. 2020.

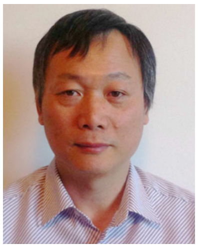
[30] A. Monticelli, *State Estimation in Electric Power Systems: A Generalized Approach*. New York, NY, USA: Springer, 2012.

Xuefei Yin received the B.S. degree in mathematics from Liaoning University, Liaoning, China, in 2011, the M.E. degree in computer science from Tianjin University, Tianjin, China, in 2014, and the Ph.D. degree in computer science from the University of New South Wales at Canberra, Canberra, ACT, Australia, in 2019.

He is currently a Research Associate with the University of New South Wales at Canberra. He has authored or coauthored articles published in top journals including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and *ACM Computing Surveys*. His research interests include biometrics, pattern recognition, privacy preserving, and intrusion detection.

Yanming Zhu received the B.E. degree from Shandong Agricultural University, Tai'an, China, in 2010, the M.E. degree from Tianjin University, Tianjin, China, in 2014, and the Ph.D. degree from the University of New South Wales, Sydney, NSW, Australia, in 2019, all in computer science.

She is currently a Research Fellow with the University of New South Wales. She has authored or coauthored articles published in top journals including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Pattern Recognition*, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, *ACM Computing Surveys*, and *Bioinformatics*. Her research interests include deep learning, biometrics, and biomedical image analysis.



Jiankun Hu (Senior Member, IEEE) received the bachelor's degree in industrial automation from Hunan University, Changsha, China, in 1983, the Ph.D. degree in engineering from the Harbin Institute of Technology, Harbin, China, in 1993, and the master's degree in computer science and software engineering from Monash University, Melbourne, VIC, Australia, in 2000.

He is currently a Full Professor with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT, Australia. He has authored or coauthored many articles published in top journals including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, PATTERN RECOGNITION, AND IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. His research interests include cybersecurity covering intrusion detection, sensor key management, and biometrics authentication.

Pof. Hu is an invited expert of the Australia Attorney-General's Office assisting the draft of the Australia National Identity Management Policy. He was the recipient of ten Australian Research Council (ARC) Grants. He has served for the Panel on Mathematics, Information and Computing Sciences, ARC Excellence in Research for Australia Evaluation Committee, in 2012. He was on the Editorial Board of up to seven international journals, including IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.