PowerFDNet: Deep Learning-Based Stealthy False Data Injection Attack Detection for AC-model Transmission Systems

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

Smart grids are vulnerable to stealthy false data injection attacks (SFDIAs), as SFDIAs can bypass residual-based bad data detection mechanisms. Methods based on deep learning technology have shown promising accuracy in the detection of SFDIAs. However, most existing methods rely on the temporal structure of a sequence of measurements but do not take account of the spatial structure between buses and transmission lines. To address this issue, we propose a spatiotemporal deep network, PowerFDNet, for the SFDIA detection in AC-model power grids. The PowerFDNet consists of two sub-architectures: spatial architecture (SA) and temporal architecture (TA). The SA is aimed at extracting representations of bus/line measurements and modeling the spatial structure based on their representations. The TA is aimed at modeling the temporal structure of a sequence of measurements. Therefore, the proposed PowerFDNet can effectively model the spatiotemporal structure of measurements. Case studies on the detection of SFDIAs on the benchmark smart grids show that the PowerFDNet achieved significant improvement compared with the state-of-the-art SFDIA detection methods. In addition, an IoT-oriented lightweight prototype of size 52 MB is implemented and tested for mobile devices, which demonstrates the potential applications on mobile devices. The trained model will be available at https://github.com/HubYZ/PowerFDNet.

Index Terms—Stealthy false data injection attack (SFDIA) detection, Bad data detection, Spatiotemporal deep learning network.
To address this issue, this paper proposes a spatiotemporal deep learning network (named PowerFDNet) for SFDIA detection in AC power systems. The proposed PowerFDNet contains two key sub-architectures: spatial architecture (SA) and temporal architecture (TA). The SA aims to extract representations from bus/line measurements and model the spatial structure of the measurements by learning their representations. The TA aims to learn the temporal structure of time-series measurements and make the decision. To naturally model the spatiotemporal structure information, measurements with and without SFDIAs are utilized as the training data. The comprehensive experiments evaluated on the benchmark power grids demonstrate that the proposed PowerFDNet achieves significant improvement in detection accuracy ($F_1$, recall, and precision) in comparison with the state-of-the-art methods.

The main contributions of this paper are summarized as follows:

1) Compared to most existing approaches that mainly model temporal structure information for SFDIA detection, we propose a new network to learn the global spatiotemporal structure information of measurements.

2) Compared to most existing approaches that take measurements as one-dimensional input, we propose well-designed residual-based sub-networks to learn multidimensional representations for buses and lines separately. Specifically, we propose using well-designed convolutional layers and residual connections to model the multidimensional representations. As an advantage, this facilitates the subsequent spatiotemporal structure learning.

3) To capture the temporal structure of the representations of a sequence of measurements, we design a booster-refiner feature encoder based on long short-term memory (LSTM) architecture. The booster-refiner encoder first models the bus/line measurement data relationship with rich features and then refines the high-dimensional feature.

4) We generate and release a comprehensive SFDIA dataset for facilitating research works in this area. The SFDIA dataset is generated for the power grids in the SimBench dataset [21], which contains a wide variety of power grids with high, medium, and low voltage, as well as the corresponding load and generator profiles in 15-minute resolution for a whole year. Our released dataset can provide a more realistic evaluation setting. Therefore, with this SFDIA dataset, researchers in this area can focus on the design and analysis of SFDIA detection.

5) An IoT-oriented lightweight prototype of size around 52 MB, with an optimized mobile model of size around 8.5 MB, is implemented and tested for mobile devices, which demonstrates the potential applications on mobile devices.

The rest of this paper is organized as follows: Section II reviews related works on deep learning-based SFDIA detection; Background knowledge about state estimation, bad data detection, and the SFDIA are presented in Section III; Section IV describes the proposed PowerFDNet in detail; the experiments and results are organized in Section V; and at the end, we concluded the paper in Section VI.

II. RELATED WORKS

Bad data detection is one of the essential functions of estate estimation to detect measurement errors. Those measurements’ errors may occur due to various reasons, such as the finite accuracy of meters, the telecommunication medium, and meters’ failure [22]–[24]. Liu et al. [25] validated that well-defined error data, as known as stealthy false data injection attack (SFDIA), can bypass the residual-based bad measurement detection in DC power grids; and in 2012, Hug et al. [5] established this type of attack to AC power grids.

Some methods managed to detect SFDIAs by statistical methods [26], [27], sparse optimization [28], graph theory [29], Kalman filter [30], time-series simulation [31], state forecasting [32], [33], and machine learning [10], [11], [14], [34]–[36]. For example, Ozay et al. [10] investigated SVM-based algorithms to classify measurements as being either secure or attacked. The experimental results demonstrate that machine learning algorithms perform better on SFDIA detection than detection algorithms that employ state vector estimation. He et al. [11] proposed a real-time SFDIA detection method based on restricted Boltzmann machines [37]. The advantage is that historical measurements are used to capture features for SFDIA detection. However, this method did not take account of the spatial structure information between transmission lines and buses. Wang et al. [14] proposed a CNN-based method to detect SFDIA attacks. The advantage is that this method attempts to identify the attack locations. However, this method failed to consider the temporal structure information. Besides, the aforementioned methods are mainly developed to detect SFDIAs in DC-model power systems.

In recent years, with the development of deep learning techniques, researchers have explored the application of deep learning to AC model power systems [15]–[20]. Kundu et al. [17] presented a detection approach based on an auto-encoder by modeling the relations between system states and measurements. The advantage is that historical measurements are used to detect SFDIAs. The disadvantage is that it did not take into account the relationship between line measurements and bus measurements. Zhang et al. [20] extended this auto-encoder scheme by embedding a generative adversarial network. One of the common points between the two methods is that the measurements are taken as the input feature. Different from that, Yu et al. [19] proposed an SFDIA detection method, which instead utilizes state variables (i.e., bus voltage angles and magnitudes) as the input feature. However, that incurs two potential risks. One is that the original spatiotemporal structure of measurements may be lost, as the model is learned from the estimated state variables instead of the measurements. Another one is that state variables estimated from false measurements may be incorrect to the power grids at the current time. Recently, Yin et al. [8] proposed a sub-grid-oriented microservice-based supervising network through privacy-preserving collaborative learning to detect SFDIAs. However, this method mainly considered the local spatiotemporal relationship of measurement in sub-grids in the
privacy-preserving setting. The work will focus on modeling the global spatiotemporal structure in a non-privacy-preserving setting.

III. BACKGROUND

In this section, we firstly provide the brief background knowledge related to the residual-based bad measurement data detection and the SFDIA. Then, we introduce an approach to generate the SFDIAs against AC-model power grids. Some key notations in this paper are listed in Table I.

| Symbol | Comments |
|--------|----------|
| SFDIAs | stealthy false data injection attacks |
| SA     | spatial architecture |
| TA     | temporal architecture |
| z      | meter measurements |
| x      | state variables |
| e      | measurement errors |
| $P_{ik}$ | active power flow from bus $i$ to bus $k$ |
| $Q_{ik}$ | reactive power flow from bus $i$ to bus $k$ |
| $P_t$   | active power injection at bus $i$ |
| $Q_t$   | reactive power injection at bus $i$ |
| $r$     | measurement residual |
| $H(x)$ | power functions of the state variables |
| $m_b$  | the number of monitored buses |
| $c_b$  | the maximum number of measurements at each monitored bus |
| $m_l$  | the number of monitored lines |
| $c_l$  | the maximum number of measurements at each monitored line |
| $T$    | a time window |
| $t_k$  | a time step |
| $z_{ik}$ | bus measurements collected at time $t_k$ |
| $z_{ik}$ | line measurements collected at time $t_k$ |
| $Z_{ik}$ | a sequence of measurements collected in the time window $T$ |

TABLE I: Key notations

A. AC State Estimation

The major objective of the state estimation is to determine optimal power states (i.e., bus voltage and angle) based on a set of redundant measurements for the power system [22]. The measurements are usually comprised of bus measurements and line measurements. The bus measurements typically consist of active/reactive power injection (i.e., bus load and generation) and bus voltage magnitude. The line measurements typically consist of active/reactive power flows measured at two sides of the transmission lines and line current flow magnitudes. In an AC-model power grid, the nonlinear formulation between the state variable $x$ and the measurement $z$ can be expressed by [22]:

$$z = H(x) + \varepsilon,$$

(1)

where $z \in \mathbb{R}^m$ denotes the measurements, $\varepsilon \in \mathbb{R}^m$ denotes the errors of the measurements, $x = [\theta_2, \theta_3, \cdots, \theta_n, V_1, V_2, \cdots, V_n]^T \in \mathbb{R}^{2n-1}$ denotes bus voltage angles and magnitudes for a $n$-bus power grid, and $H(x)$ is the nonlinear vector function of the state variables. $h_i(x) \in H(x)$ is the formula of the measurement $z_i$ related to the state variable $x_i$. It is assumed that the error $\varepsilon_i \in \varepsilon$ is independent and draws from a Gaussian distribution $\mathcal{N}(0, \sigma^2_e)$.

The nonlinear functions of each measurement and power states are presented as follows. The active and reactive power flows from bus $i$ to bus $k$, $P_{ik}$ and $Q_{ik}$, can be expressed by [22]

$$\begin{align*}
P_{ik} &= V_i^2(g_{ik} + g_{si}) - V_i V_k(b_{ik} \sin \theta_{ik} + g_{ik} \cos \theta_{ik}), \\
Q_{ik} &= -V_i^2(b_{ik} - b_{si}) - V_i V_k(g_{ik} \sin \theta_{ik} - b_{ik} \cos \theta_{ik}),
\end{align*}$$

(2)

and, line current flow magnitude from bus $i$ to bus $k$, $I_{ik}$, can be expressed by [22]

$$I_{ik} = \sqrt{P_{ik}^2 + Q_{ik}^2}/V_i.$$  

(3)

The active and reactive power injections at bus $i$, $P_i$ and $Q_i$, can be expressed by

$$\begin{align*}
P_i &= V_i \sum_{k \in \Omega_i} V_k (B_{ik} \sin \theta_{ik} + G_{ik} \cos \theta_{ik}), \\
Q_i &= V_i \sum_{k \in \Omega_i} V_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}),
\end{align*}$$

(4)

where $\theta_i$ and $V_i$ represent the state variables for bus $i$, $\theta_k$ and $V_k$ represent the state variables for bus $k$, $\theta_{ik} = \theta_i - \theta_k$, $g_{ik} + j b_{ik}$ is the admittance of the line connecting buses $i$ and $k$, and $g_{si} + j b_{si}$ is the admittance of the shunt line at bus $i$. $G_{ik} + j B_{ik}$ is the $ik$th element of its complex bus admittance matrix. $\Omega_i$ denotes the set of adjacent buses that are directly connected to bus $i$.

The state estimation is to find the optimal solution $\hat{x}$ based on the measurements through minimizing the following weighted least squares problem:

$$\mathcal{L}(x) = (z - H(x))^T \Lambda^{-1} (z - H(x)),$$

(5)

where $\Lambda = \text{diag}[\sigma_1^2, \sigma_2^2, \cdots, \sigma_m^2]$ denotes the weight matrix whose element represents the variance of the measurements at the corresponding electricity meter. The minimization of the objective function $\mathcal{L}(x)$ can be solved by iterative approaches (e.g., the Newton-Raphson algorithm), which can be expressed by

$$\hat{x} = \arg \min_x \mathcal{L}(x),$$

(6)

where $\hat{x}$ denotes the optimal power system state estimated from the measurements.

B. Residual-based Bad Measurement Detection

Errors in the measurements may be introduced due to distinct reasons such as meter malfunction, signal transmission interference, and cyberattacks. The bad data detection is aimed at determining whether or not the measurements contain significant errors. The values of state variables estimated from normal meter measurements should be close to the true state values, while the state values estimated from bad measurements may be significantly different from the true values. Thus, one popular method is to calculate the residual
Eq. (12) can then be expressed as follows \[5\]

(13) can bypass the detection mechanism. Therefore, the malicious attack measurement obtained by Eq. (12) \[5\],

Hence, we have Eq. (12) \[5\],

\[
\|z_{bad} - H(\hat{x}_a)\| = \|z + a - H(\hat{x} + c)\| = \left\| \begin{bmatrix} z_1 \\ z_2 + a_2 \end{bmatrix} - \begin{bmatrix} H_1(\hat{x}_1) \\ H_2(\hat{x}_1, \hat{x}_2 + c_2) \end{bmatrix} \right\| = \|z - H(\hat{x})\| = r.
\]

Therefore, the malicious attack measurement obtained by Eq. (13) can bypass the detection mechanism.

IV. PROPOSED POWERFDNET

In this section, we present details of the proposed PowerFDNet to detect SFDIAs in AC-model power grids. The PowerFDNet is aimed at classifying the measurements to determine whether measurements are maliciously modified, which consists of two key sub-architectures: a spatial architecture (SA) and a temporal architecture (TA). The SA aims to extract representations from bus/line measurements and model the spatial structure of measurements by learning their representations (Section IV-B). The TA aims to model the temporal structure of the measurements by learning the intermediate feature obtained by the SA and making a final prediction (Section IV-C).

A. Measurement Data

In this subsection, we provide the details about the organization of the measurements. Common measurement variables and notations such as in [8], [19], [22] are adopted. Typically, the measurements for power systems include line measurements and bus measurements. The line measurements commonly contain active and reactive power flow data measured at the two sides of transmission lines and line current flow magnitudes, which are summarized as follows:

- \(P_l\), active power flow measurement at the ‘in’ side,
- \(P_o\), active power flow measurement at the ‘out’ side,
- \(Q_l\), reactive power flow measurement at the ‘in’ side,
- \(Q_o\), reactive power flow measurement at the ‘out’ side,
- \(I_l\), current flow magnitude measurement at the ‘in’ side,
- \(I_o\), current flow magnitude measurement at the ‘out’ side.

The typical bus measurements are summarized as follows:

- \(P\), bus active power injection measurement,
- \(Q\), bus reactive power injection measurement,
- \(V\), bus voltage magnitude measurement.

Let \(z_{tk}^b \in \mathbb{R}^{m_b \times 1 \times c_b}\) denote the bus measurements collected at time \(t_k\), where \(m_b\) denotes the number of monitored buses and \(c_b\) denotes the maximum number of measurements at each monitored bus. Similarly, let \(z_{tk}^l \in \mathbb{R}^{m_l \times 1 \times c_l}\) denote the line measurements collected at time \(t_k\), where \(m_l\) denotes the number of monitored lines and \(c_l\) denotes the maximum number of measurements at each monitored line. Measurements with less than \(c_l/c_b\) data on monitored buses/lines are padded with 0. Hence, the measurements of a power grid collected at time \(t_k\) can be expressed by \(z_{tk}\), which is composed of \(z_{tk}^b\) and \(z_{tk}^l\), formulated by \(z_{tk} = \{z_{tk}^l, z_{tk}^b\} \). Let \(Z_{tk} = \{z_{tk}, z_{tk-1}, \ldots, z_{tk-(T-1)}\}\) denote a sequence of measurements, where \(T\) is a constant value. For convenience, \(Z_{tk}\) is also equivalently expressed by \(Z_{tk} = \{Z_{tk}^l, Z_{tk}^b\}\), where

\[
Z_{tk}^b = \{z_{tk-(T-1)}^b, z_{tk-(T-2)}^b, \ldots, z_{tk}^b\} \in \mathbb{R}^{T \times m_b \times 1 \times c_b}
\]

is the time-series bus measurements and
\[
Z_{tk}^l = \{z_{tk-(T-1)}^l, z_{tk-(T-2)}^l, \ldots, z_{tk}^l\} \in \mathbb{R}^{T \times m_l \times 1 \times c_l}
\]
is the time-series line measurements. We define the label for $Z_{t+k}$ as follows:

$$y_{t+k} = \begin{cases} 
0, & \text{if } z_{t+k} \text{ is normal measurement data that has not been attacked;} \\
1, & \text{if } z_{t+k} \text{ is attacked by an SFDIA and can bypass the residual-based detection.}
\end{cases}$$

(15)

In this paper, the commercial power system analysis software PowerFactory 2017 SP4 \(^1\) was used to conduct the residual-based bad measurement detection.

### B. Spatial Architecture (SA)

The SA is aimed at modeling the spatial structure between buses and lines by learning the representations of their measurements. The architecture of SA is shown in Fig. 1. It is composed of three sub-networks: one for bus measurement representation, one for line measurement representation, and the other one for spatial structure modeling. Different from the methods [11], [14], [17], [19], [20] that squeeze line and bus measurements into a one-dimensional input, we propose extracting the representations for lines and buses separately. As an advantage, the SA can effectively learn the hidden features from bus and line measurements and model their spatial structures. Compared with the method in [8], we utilize residual blocks to extract the bus/line representations and deploy deep layers to make them suitable for modeling the global spatial structure of the large-scale power grids.

1) Sub-network for Representation of Bus Measurement Data

This sub-network is aimed at extracting representations of bus measurements. The architecture is shown in Fig. 1. The parameters are summarized in Table II. To learn to learn the residual from the measurements [38], the intermediate features are concatenated with the original measurement in the second and third layers. The fourth and fifth layers are aimed at extracting the representations of bus measurements from the intermediate feature space.

Specifically, for bus measurement $Z_{t+k}^b$, the first layer $\text{Conv}_{b_1}$ utilizes $c_b$ convolution filters of size $1 \times c_b$ to extract a residual feature for each bus, formulated by

$$b_{t+k, o_j}^b = \text{Conv}_{b_1}(z_{t+k}^b) = \phi(W_{o_j} \ast z_{t+k}^b + a_{o_j}),$$

(16)

where $a_{o_j}$ is an additive bias for each output channel, $W_{o_j} \in \mathbb{R}^{1 \times c_b}$, $z_{t+k}^b \in \mathbb{R}^{m_b \times 1 \times c_b}$, $b_{t+k, o_j}^b \in \mathbb{R}^{m_b \times 1}$, and $\phi$ denotes the convolutional operator. $\phi$ denotes the batch normalization, and $\phi$ denotes the exponential linear unit (ELU) [39]. Then, the residual feature and the bus measurements are concatenated into a two-channel data, which are fed into the next layer. The last layer $\text{Conv}_{b_5}$ utilizes four $1 \times 6$ convolution filters to extract the representation for each bus. Therefore, a representation of four feature values is obtained for each bus. For the input bus measurements $Z_{t+k}^b$, its representation is denoted by $B_{t+k}^b$.

2) Sub-network for Representation of Line Measurement Data

This sub-network is proposed to extract the representations for the line measurements. The network architecture is shown in Fig. 1. The parameters are summarized in Table III. In the second and third layers, the previous results are concatenated with the original measurement to learn the residual from the line measurements. The fourth and fifth layers are designed to learn the line representations from the intermediate feature space.

Similar to the sub-network for the bus representation, a representation of four feature values is learned for each line.

---

\(^1\)https://www.digsilent.de/en/powerfactory.html
TABLE III: Configuration of the sub-network for the representation of line measurements

| Layer     | Kernel Size | Out Channels | Groups | Stride | Padding | BN/AF/Reshape |
|-----------|-------------|--------------|--------|--------|----------|---------------|
| Conv1     | 1 × c_l    | m_l          | m_l    | 1      | [0, 1]   | Yes/ELU/No    |
| Conv2     | 2 × c_l    | m_l          | m_l    | 1      | [0, 1]   | Yes/ELU/No    |
| Conv3     | 2 × c_l    | 12m_l        | m_l    | 1      | [0, 0]   | Yes/ELU/Yes   |
| Conv4     | 1 × 3      | m_l          | m_l    | 2      | [0, 1]   | Yes/ELU/No    |
| Conv5     | 1 × 3      | m_l          | m_l    | 1      | [0, 0]   | Yes/ELU/Yes   |

BN: Batch normalization  AF: Activation function

Therefore, for the input line measurements $Z_{t_k}$, its representation is denoted by $L_{t_k} \in \mathbb{R}^{T \times m_l \times 1 \times 4}$.

3) Sub-network for Modeling Spatial Structure

This sub-network is proposed to learn the spatial structure of measurements from the measurement representations, as shown in Fig. 1. The parameters are summarized in Table IV. To learn the spatial structure between each line/bus and the remaining ones, we propose utilizing larger filters with the size of $(m_h + m_l) \times 1$. Therefore, output features learned from the bus/line representations can reflect such relationships.

TABLE IV: Configuration of the sub-network for modeling the spatial structure

| Layer     | Kernel Size | Out Channels | Groups | Stride | Padding | BN/AF/Reshape |
|-----------|-------------|--------------|--------|--------|----------|---------------|
| Conv1     | (m_h + m_l) × 1 | 256          | 1      | 1      | [0, 0]   | Yes/ELU/No    |
| Conv2     | 1 × 4       | 256          | 1      | 1      | [0, 0]   | Yes/ELU/Yes   |
| Conv3     | 256 × 1     | 128          | 1      | 1      | [0, 0]   | Yes/ELU/Yes   |

BN: Batch normalization  AF: Activation function

Specifically, the bus and line representations are firstly reshaped and concatenated to fully represent the input measurements, expressed by

$$H_{t_k} = \{B_{t_k}, L_{t_k}\} \in \mathbb{R}^{T \times 1 \times (m_h + m_l) \times 4}.$$  \hspace{1cm} (16)

Then, three layers are designed to learn the spatial structure of line/bus measurements, which is expressed by

$$S_{t_k} = \text{Conv}_{bd_3}(\text{Conv}_{bd_2}(\text{Conv}_{bd_1}(H_{t_k}))),$$  \hspace{1cm} (17)

where

$$S_{t_k} = \{s_{t_k}, s_{t_k-1}, \ldots, s_{t_k-(T-1)}, \ldots, s_{t_k-1}, s_{t_k}\} \in \mathbb{R}^{T \times 128}.$$  \hspace{1cm} (18)

C. Temporal Architecture (TA)

The TA is used to model the temporal structure of a sequence of measurements in a time window by learning their intermediate features obtained by the SA. The architecture of TA is shown in Fig. 2. It is composed of four LSTM layers, one fully connected layer, and a sigmoid layer. The output represents the probability that the measurement $z_{t_k}$ at time step $t_k$ is an SFDIA. To effectively model the temporal structure information, we incorporate the LSTM architecture shown in Fig. 3 into the proposed PowerFDNet. It has two major advantages: one is to represent temporal structure information of time-series measurements [40], and another is to avoid gradient vanishing and exploding [41]. This architecture forms a booster-refiner encoder that can use rich features to model the large-scale spatial representations of bus and line measurement variables and then refines them with a more salient and condensed feature vector representation.

Recall that for the input measurements $Z_{t_k}$, the spatial structure information $S_{t_k}$ is learned by the SA. For convenience, the input data for time step $t$ is denoted by $X_t \in S_{t_k}, d = 128$, and $h = 256$. Specifically, the calculation flow in the $f_{TA}$ is expressed as follows:

$$F_t = \sigma(X_t W_{xf} + \tilde{H}_{t-1} W_{hf} + b_f),$$

where $W_{xf} \in \mathbb{R}^{d \times h}, W_{hf} \in \mathbb{R}^{h \times h}, b_f \in \mathbb{R}^{1 \times h}$, and $\sigma$ denotes the sigmoid function;

$$I_t = \sigma(X_t W_{xi} + \tilde{H}_{t-1} W_{hi} + b_i),$$

where $W_{xi} \in \mathbb{R}^{d \times h}, W_{hi} \in \mathbb{R}^{h \times h}, b_i \in \mathbb{R}^{1 \times h}$;

$$C_t = \tanh(X_t W_{xc} + \tilde{H}_{t-1} W_{hc} + b_c),$$

where $W_{xc} \in \mathbb{R}^{d \times h}, W_{hc} \in \mathbb{R}^{h \times h}, b_c \in \mathbb{R}^{1 \times h}$;

$$O_t = \sigma(X_t W_{xo} + \tilde{H}_{t-1} W_{ho} + b_o),$$

where $W_{xo} \in \mathbb{R}^{d \times h}, W_{ho} \in \mathbb{R}^{h \times h}, b_o \in \mathbb{R}^{1 \times h}$;

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t,$$

where $C_{t-1} \in \mathbb{R}^{1 \times h}$ and $\odot$ denotes the Hadamard product. Then, the output of the $f_{LSTM}$ is expressed by

$$\tilde{H}_t = O_t \odot \tanh(C_t).$$  \hspace{1cm} (18)

After the four LSTM layers, the feature map $X_{out} \in \mathbb{R}^{T \times d}$ is obtained. To detect the SFDIA, the feature data for the current time step is detached, represented by $x_{tp} \in \mathbb{R}^{1 \times d}$. This feature is then processed by layer $f_{TA}$ with a sigmoid activation, formulated by

$$y_p = \sigma(x_{tp} W + a),$$  \hspace{1cm} (19)

where $W \in \mathbb{R}^{d \times 1}$ and $a \in \mathbb{R}$.

D. Loss Function

The binary cross entropy error is utilized to train the proposed PowerFDNet, which is expressed by

$$f_{loss} = - \sum_{i=1}^{N} (y_i \log y_{ip} + (1 - y_i) \log(1 - y_{ip})),$$  \hspace{1cm} (20)
Where \( N \) is the mini-batch size, \( y_p \) is the prediction result obtained by Eq. (19) for the measurements \( z_t \), and \( y_i \) is the corresponding ground truth label (in Eq. (15)). In the training stage, the optimization algorithm of Adam [43] is used to update the network weights, with an initial learning rate of \( 1 \times 10^{-4} \). The learning rate is dynamically adjusted in the training stage by the ReduceLROnPlateau scheduler. The popular deep learning framework Pytorch-1.9.0 was used to construct the PowerFDNet for the model training and testing. The trained network model of the proposed PowerFDNet will be available online at https://github.com/FrankYinXF/PowerFDNet.

V. CASE STUDIES

A. Dataset

In the experiments, two benchmark power systems from the public SimBench dataset [21] were used to assess the SFDIA detection. There are three main reasons. First, these power grids contain detailed data, especially time-series demand profiles for an entire year that are generated every 15 minutes (e.g., 35,136 demand profiles). Therefore, it can be convenient to use these power grids to simulate a power grid with dynamical power load and generation. Second, measurements for buses and lines have been defined and tested in these SimBench power grids with high voltage and extra-high voltage [21]. Therefore, this dataset can be conveniently used to evaluate the SFDIA detection. Finally, the SimBench power grids originated from the German power systems. To some extent, it provides realistic power grids for evaluating the SFDIA detection.

The two benchmark power grids, ‘1-HV-mixed–0-no_sw’ and ‘1-EHV-mixed–0-no_sw’, were utilized to evaluate the performance of SFDIA detection. The ‘1-HV-mixed–0-no_sw’ is a high voltage level grid with 110 KV transmission lines, denoted by Grid-HV, which is monitored by 355 measurements, with 35,136 profiles for dynamical power load and generation. More details are shown in Table VI. The ‘1-EHV-mixed–0-no_sw’ is an extra-high voltage level grid with 220-380 KV transmission lines, denoted by Grid-EHV, which is monitored by 3,956 measurements, with 35,136 profiles for dynamical power load and generation. More details are shown in Table VII.

The open-source software Pandapower\(^3\) and SimBench\(^4\) and the commercial software PowerFactory 2017 SP4\(^5\) were used in the SFDIA data generation stage for the power flow calculation and the bad data detection.

\(^2\)https://pytorch.org/docs/1.9.0/
\(^3\)https://www.pandapower.org/
\(^4\)https://simbench.readthedocs.io/en/stable/about/installation.html
\(^5\)https://www.digsilent.de/en/powerfactory.html
1) Time-series Measurements Generated on the Grid-HV and Grid-EHV

The genuine values of these measurements in the two power grids were obtained by calculating the power flow using the commercial software PowerFactory 2017 SP4. Hence, there are 35,136 normal measurement samples for each power grid. Details of measurement data are presented in Section IV-A. For Grid-HV, each measurement sample collected at a time step contains 192 measurement values for buses and 163 measurement values for transmission lines. For Grid-EHV, each measurement sample measured at a time step contains 1,698 measurement values for buses and 2,254 measurement values for transmission lines. The measurement noises are assumed to follow Gaussian distributions and are configured to be less than 1% for voltage magnitude and less than 2% for active/reactive power injection and power flow [22].

2) SFDIA Measurement Generation

The generation of the SFDIA measurement is based on the method proposed in [5], [8]. The attacks were launched on a target bus by maliciously modifying either its voltage angle ($V_a$) or voltage magnitude ($V_m$). To comprehensively evaluate the performance of the SFDIA detection, three types of SFDIAs are designed and summarized as follows:

- Type-A that the rate of the active power injection change on the target bus is in the range of (50%, 100%),
- Type-B that the rate of the active power injection change on the target bus is in the range of (25%, 50%), and
- Type-C that the rate of the active power injection change on the target bus is in the range of (5%, 25%).

Therefore, Type-A SFDIA will lead to a large change in power injection, Type-B SFDIA will lead to a medium change, and Type-C SFDIA will lead to a relatively small change in power injection. At each time step, six buses with injection are randomly selected as the target buses to launch these three types of SFDIAs. Therefore, there are totally $35,136 \times 6 = 210,816$ attacked measurement samples, summaries as follows:

- 35,136 Type-A SFDIA attack measurements by attacking the bus magnitude $V_m$,
- 35,136 Type-A SFDIA attack measurements by attacking the bus angle $V_a$,
- 35,136 Type-B SFDIA attack measurements by attacking the bus magnitude $V_m$,
- 35,136 Type-C SFDIA attack measurements by attacking the bus magnitude $V_m$, and
- 35,136 Type-C SFDIA attack measurements by attacking the bus angle $V_a$.

All of these SFDIA measurements have bypassed the bad measurement detection function of PowerFactory 2017 SP4. These three types of SFDIA datasets generated for Grid-HV and Grid-EHV in this experiment will be publicly available online at https://github.com/FrankYinXF/PowerFDNet. Fig. 4 shows the statistical information from time step 300 to 500 about Type-A SFDIA attack measurements on Grid-HV in terms of the change of $V_m$, the rate of $P$ change, and the change of $P$. Fig. 5 shows a normal measurement sample (corresponding to $z$ in Eq. (10)), an SFDIA sample obtained (corresponding to $z_{bad}$ in Eq. (10)) by attacking the normal sample, and the change (corresponding to $\alpha$ in Eq. (10)) between the two samples.

![Fig. 4: Statistical information about Type-A SFDIA measurements at target buses in Grid-HV: (a) the change of voltage magnitude, (b) the distribution of the rate of active power change, and (c) the change of active power.](image-url)
3) Training and Testing Dataset
As introduced in Section V-A2, each power grid generates 35,136 normal measurement samples and 210,816 SFDIA samples with three types of attacks. The normal measurements are labeled as 0, and the SFDIA measurements are labeled as 1, as expressed in Eq. (15). For each grid, 29,952 normal measurements for the first 312 days of one year are grouped as training data, and the remaining 54 days’ normal measurements (namely 5,184 samples) are utilized for testing. The SFDIA measurements are grouped in a similar way, e.g., 179,712 for training and 31,104 for testing (each type of SFDIA contains 10,368 testing samples, with 5,184 SFDIA samples by modifying $V m$ and 5,184 SFDIA samples by modifying $Va$). The advantage of this way of data partitioning is that the testing data is completely fresh to the trained model so that the detection cases can realistically simulate the real-world situation.

B. Performance Metrics
Three commonly used metrics were applied to assess the SFDIA detection [17], [19], [20], which are expressed by:

\[
\begin{align*}
\text{Precision} & = \frac{N_{tp}}{N_{tp} + N_{fp}}, \\
\text{Recall} & = \frac{N_{tp}}{N_{tp} + N_{fn}}, \\
F_1 & = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},
\end{align*}
\]

where $N_{fp}$ indicates the number of false positive, $N_{tp}$ the number of true positive, $N_{fn}$ the number of false negative, and $N_{tn}$ the number of true negative, which are summarized in Table VIII. A normal measurement sample is defined as negative, while a sample attacked by the SFDIA is defined as positive. Hence, $N_{fn} + N_{tp}$ is the total number of real positive samples in the data set, and $N_{fp} + N_{tn}$ is the total number of real negative samples in the data set.

| Approaches | Grid-HV | Grid-EHV |
|------------|---------|----------|
| M-I [19]   | 95.954  | 93.615   |
| M-II [17]  | 96.572  | 95.930   |
| PowerFDNet | 99.557  | 99.778   |

TABLE IX: Comparison of detection accuracy of the Type-A SFDIAs evaluated on Grid-HV and Grid-EHV, respectively.

C. Evaluation of SFDIA Detection
In the experiment, we compared the SFDIA detection accuracy of the proposed PowerFDNet with two state-of-the-art approaches, M-I [19] and M-II [17]. As introduced in Section V-A2 and Section V-A3, there are 5,184 normal samples in the test stage. Each type of SFDIA contains 10,368 samples, where 5,184 samples are obtained by modifying the voltage magnitude of a target bus and the other 5,184 samples are obtained by modifying the voltage angle of a target bus. Table IX and Table X compare the accuracy of Type-A SFDIA detection evaluated on Grid-HV and Grid-EHV, respectively.

TABLE X: Comparison of detection accuracy of the Type-A SFDIAs evaluated on Grid-EHV, between the proposed PowerFDNet with the two state-of-the-art approaches.

| Approaches | Grid-EHV | Grid-EHV |
|------------|---------|----------|
| M-I [19]   | 95.635  | 93.615   |
| M-II [17]  | 96.259  | 95.303   |
| PowerFDNet | 99.422  | 99.508   |

As shown in Table IX, in the case of the Type-A SFDIA detection, it is clear that the proposed PowerFDNet achieves the best $F_1$ of 99.668%, the best recall of 99.778%, and the best precision of 99.557% on Grid-HV. The precision achieved by the proposed method is about 3.756% higher than M-I and 3.091% higher than M-II, respectively. The recall achieved by the proposed method is about 5.885% higher than M-I and 4.012% higher than M-II, respectively. The $F_1$ score achieved by the proposed method is about 4.819% higher than M-I and 3.551% higher than M-II, respectively. Similar detection performance is also achieved on Grid-EHV, as summarized in Table X. Our method obtains the best $F_1$ of approximately 99.465%, which is around 5.127% higher than M-I and about 3.849% higher than M-II, respectively. The best precision obtained by our method is approximately 99.422%, which is about 3.960% higher than M-I and 3.286% higher than M-II, respectively. The best recall achieved by our method is approximately 99.508%, which is about 6.295% higher...
than M-I and 4.413% higher than M-II, respectively. That demonstrates that for the Type-A SFDIAs with large rates of power injection change at target buses, the PowerFDNet achieved the highest SFDIA detection accuracy in terms of the precision, recall, and $F_1$ score when compared to the two state-of-the-art approaches.

2) Case B: Type-B SFDIA Detection

The Type-B SFDIA detection is to assess the detection accuracy of SFDIAs that are launched by attacking the target bus through a medium modification in the range of (25%, 50%) to the bus active power injection. Because there is less modification in the power injection in the Type-B SFDIAs, it is harder to detect the Type-B SFDIAs than to detect the Type-A SFDIAs. Table XI and Table XII compare the detection accuracy of the Type-B SFDIAs evaluated on Grid-HV and Grid-EHV, respectively.

TABLE XI: Comparison of the detection accuracy of the Type-B SFDIAs evaluated on Grid-HV between the proposed PowerFDNet with the two state-of-the-art approaches.

| Approaches | Grid-HV |       |       |
|------------|---------|-------|-------|
|            | Precision (%) | Recall (%) | $F_1$ (%) |
| M-I [19]   | 95.851        | 94.030        | 94.932 |
| M-II [17]  | 96.412        | 95.621        | 96.015 |
| PowerFDNet | 99.461        | 99.576        | 99.518 |

TABLE XII: Comparison of the detection accuracy of the Type-B SFDIAs evaluated on Grid-EHV between the proposed PowerFDNet with the two state-of-the-art approaches.

| Approaches | Grid-EHV |       |       |
|------------|---------|-------|-------|
|            | Precision (%) | Recall (%) | $F_1$ (%) |
| M-I [19]   | 95.541        | 93.200        | 94.356 |
| M-II [17]  | 96.170        | 94.927        | 95.544 |
| PowerFDNet | 99.363        | 99.296        | 99.329 |

As shown in Table XI, in the case of Type-B SFDIA detection, it is clear to see that the proposed PowerFDNet obtains the best $F_1$ of 99.518%, the best precision of 99.461%, and the best recall of 99.576% on Grid-HV. The precision achieved by our method is about 3.766% higher than M-I and 3.162% higher than M-II, respectively. The recall obtained by our method is about 5.898% higher than M-I and 4.136% higher than M-II, respectively. The $F_1$ score obtained by our method is approximately 4.831% higher than M-I and 3.649% higher than M-II, respectively. Similar detection performance is also achieved on Grid-EHV, as summarized in Table XII. Our method obtains the best $F_1$ of approximately 99.329%, which is around 5.271% higher than M-I and about 3.962% higher than M-II, respectively. The best precision obtained by our method is approximately 99.363%, which is about 4.001% higher than M-I and 3.321% higher than M-II, respectively. The best recall achieved by our method is approximately 99.296%, which is about 6.540% higher than M-I and 4.603% higher than M-II, respectively.

3) Case C: Type-C SFDIA Detection

The Type-C SFDIA detection is to assess the detection accuracy of SFDIAs that are launched by attacking the target bus through a relatively small modification in the range of (5%, 25%) to the bus active power injection. Compared with the other two SFDIAs, the Type-C SFDIA measurements have a smaller modification in the bus active power. Hence, it is more difficult to detect the Type-C SFDIAs. Table XIII and Table XIV summarize the detection accuracy of the Type-C SFDIAs evaluated on Grid-HV and Grid-EHV, respectively. As clearly shown in Table XIII and Table XIV, the detection accuracy (precision, recall, and $F_1$) of all the three methods is slightly lower than that evaluated on the Type-A and Type-B SFDIAs. Compared to the other two methods, the proposed PowerFDNet achieved the highest detection accuracy in terms of $F_1$, recall, and precision on Grid-HV and Grid-EHV.

TABLE XIII: Comparison of the detection accuracy of the Type-C SFDIAs evaluated on Grid-HV between the proposed PowerFDNet with the two state-of-the-art approaches.

| Approaches | Grid-HV |       |       |
|------------|---------|-------|-------|
|            | Precision (%) | Recall (%) | $F_1$ (%) |
| M-I [19]   | 95.748        | 93.837        | 94.783 |
| M-II [17]  | 96.311        | 95.428        | 95.867 |
| PowerFDNet | 99.373        | 99.402        | 99.388 |

TABLE XIV: Comparison of the detection accuracy of the Type-C SFDIAs evaluated on Grid-EHV between the proposed PowerFDNet with the two state-of-the-art approaches.

| Approaches | Grid-EHV |       |       |
|------------|---------|-------|-------|
|            | Precision (%) | Recall (%) | $F_1$ (%) |
| M-I [19]   | 95.423        | 92.892        | 94.140 |
| M-II [17]  | 96.092        | 94.637        | 95.359 |
| PowerFDNet | 99.324        | 99.209        | 99.267 |

Compared to M-I evaluated on Grid-HV, our method improved by approximately 4.858% in $F_1$ score, approximately 3.786% in precision, and about 5.931% in the recall, respectively. Compared with M-II evaluated on Grid-HV, our method obtained an improvement of approximately 3.672%, 4.164%, and 3.180% in terms of $F_1$ score, recall, and precision, respectively. Compared with M-I evaluated on Grid-EHV, our method improved by approximately 5.446% in $F_1$, about 6.801% in the recall, and around 4.089% in precision, respectively. Compared with M-II evaluated on Grid-EHV, our method obtained an improvement of approximately 4.097%, 5.446%, and 3.649% in terms of $F_1$, recall, and precision, respectively. That demonstrates that for the Type-C SFDIAs with small rates of power injection change at target buses, the PowerFDNet achieved the highest SFDIA detection accuracy ($F_1$, recall, and precision) compared to the two state-of-the-art approaches.
D. An IoT-oriented Prototype of the SFDIA detection

A lightweight IoT-oriented SFDIA detection prototype was implemented in the Android-based mobile platform, as shown in Fig. 6. The lightweight prototype is around of size 52 MB, with the optimized mobile model of size 8.5 MB. The optimized lightweight model is achieved by PyTorch, which provides such a utility to easily create serializable and optimizable models. The testing time for one sample by the prototype is about 0.2 seconds in an android emulator of Pixel XL API 30. The popular deep learning framework PyTorch 1.10.0 and PyTorch_android_lite:1.10.0 were used to implement this IoT-oriented prototype.

VI. CONCLUSION

In this paper, we proposed a spatiotemporal deep learning-based PowerFDNet for successful SFDIA detection in AC-model power systems. To model the spatiotemporal structure information between buses and lines, we designed two sub-architectures: the SA for the spatial structure learning and the TA for the temporal structure learning. In the SA, we firstly model the bus measurements and the line measurements separately, so that the model can effectively represent these two types of measurements. Then, a sub-network is designed to capture the spatial structure information between buses and lines and to preliminarily capture the patterns of SFDIA measurements. Further, the TA based on the LSTM is designed to effectively learn the temporal structure information of the preliminary features obtained by the SA. The proposed PowerFDNet is comprehensively evaluated on two realistic benchmark power grids. The experimental results demonstrate that the PowerFDNet achieves significant improvement in terms of $F_1$, recall, and precision compared with the two state-of-the-art SFDIA detection approaches. In addition, an IoT-oriented lightweight prototype of size 52 MB is implemented and tested for mobile devices, which demonstrates the potential applications on mobile devices.

ACKNOWLEDGMENT

This research was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government. This project was supported by ARC Discovery Grant with project ID DP190103660 and ARC Linkage Grant with project ID LP180100663.

REFERENCES

[1] L. Che, X. Liu, Z. Li, and Y. Wen, “False data injection attacks induced sequential outages in power systems,” IEEE Transactions on Power Systems, vol. 34, pp. 1513–1523, 2019.
[2] Z. Zhao, Y. Huang, Z. Zhen, and Y. Li, “Data-driven false data-injection attack design and detection in cyber-physical systems,” IEEE Transactions on Cybernetics, pp. 1–9, 2020.
[3] J. Liang, L. Sankar, and O. Koutr, “Vulnerability analysis and consequences of false data injection attack on power system state estimation,” IEEE Transactions on Power Systems, vol. 31, pp. 3864–3872, 2016.
[4] J. Hu and A. V. Vasilakos, “Energy big data analytics and security: Challenges and opportunities,” IEEE Transactions on Smart Grid, vol. 7, pp. 2423–2436, 2016.
[5] G. Hug and J. A. Giampapa, “Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks,” IEEE Transactions on Smart Grid, vol. 3, pp. 1362–1370, 2012.
[6] 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 and Computation: Practice and Experience, vol. 34, no. 7, p. e5956, 2022.
[7] M. Yang, H. Zhang, C. Peng, and Y. Wang, “A penalty-based adaptive secure estimation for power systems under false data injection attacks,” Information Sciences, vol. 508, pp. 380–392, 2020.
[8] X. Yin, Y. Zhu, and J. Hu, “A sub-grid-oriented privacy-preserving microservice framework based on deep neural network for false data injection attack detection in smart grids,” IEEE Transactions on Industrial Informatics, pp. 1–1, 2021.
[9] 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, vol. 14, pp. 1–33, 2011.
[10] 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 Transactions on Neural Networks and Learning Systems, vol. 27, pp. 1773–1786, 2016.
[11] 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 Transactions on Smart Grid, vol. 8, pp. 2505–2516, 2017.
[12] J. Sakhnini, H. Karimipour, and A. Dehghantanha, “Smart grid cyber attacks detection using supervised learning and heuristic feature selection,” in IEEE International Conference on Smart Energy Grid Engineering (SEGE), 2019, pp. 108–112.
[13] A. Sayghe, Y. Hu, I. Zografopoulos, X. Liu, R. G. Dutta, Y. Jin, and C. Konstantinou, “Survey of machine learning methods for detecting false data injection attacks in power systems,” IET Smart Grid, vol. 3, pp. 581–595, 2020.
[14] 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, pp. 8218–8227, 2020.
[15] Q. Deng and J. Sun, “False data injection attack detection in a power grid using rnn,” in Annual Conference of the IEEE Industrial Electronics Society, 2018, pp. 5983–5988.
[16] M. Lu, L. Wang, Z. Cao, Y. Zhao, and X. Xui, “False data injection attacks detection on power systems with convolutional neural network,” in Journal of Physics: Conference Series, vol. 1633, 2020, p. 012134.
[17] A. Kundu, A. Sahu, E. Serpedin, and K. Davis, “A3D: Attention-based autoencoder anomaly detector for false data injection attacks,” Electric Power Systems Research, vol. 189, p. 106795, 2020.

[18] M. Ashrafulazman, S. Das, Y. Chakchhouk, S. Shiva, and F. T. Sheldon, “Detecting stealthy false data injection attacks in the smart grid using ensemble-based machine learning,” Computers & Security, vol. 97, p. 101994, 2020.

[19] 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 Transactions on Industrial Informatics, vol. 14, pp. 3271–3280, 2018.

[20] Y. Zhang, J. Wang, and B. Chen, “Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach,” IEEE Transactions on Smart Grid, vol. 9, p. 101994, 2020.

[21] S. Meinecke, D. Sarajlic, S. R. Drauz, A. Klettke, L.-P. Lauven, A. Abur and A. G. Exposito, “Strategic protection against data injection attacks,” Power Systems Research, vol. 5, pp. 13787–13798, 2017.

[22] T. T. Kim and H. V. Poor, “Strategic protection against data injection attacks on power grids,” IEEE Transactions on Smart Grid, vol. 2, pp. 326–333, 2011.

[23] A.-Y. Lu and G.-H. Yang, “False data injection attacks against state estimation in the presence of sensor failures,” Information Sciences, vol. 508, pp. 92–104, 2020.

[24] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” in ACM Conference on Computer and Communications Security, 2009, p. 21–32.

[25] S. A. Foroutan and F. R. Salmisi, “Detection of false data injection attacks against state estimation in smart grids based on a mixture gaussian distribution learning method,” IET Cyber-Physical Systems: Theory & Applications, vol. 2, pp. 161–171, 2017.

[26] S. K. Singh, K. Khatana, R. Bose, B. K. Panigrahi, and A. Joshi, “Joint-transformation-based detection of false data injection attacks in smart grid,” IEEE Transactions on Industrial Informatics, vol. 14, pp. 89–97, 2018.

[27] L. Liu, M. Esmailifalak, Q. Ding, V. A. Emeshiz, and Z. Han, “Detecting false data injection attacks on power grid by sparse optimization,” IEEE Transactions on Smart Grid, vol. 5, pp. 612–621, 2014.

[28] 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 Transactions on Control of Network Systems, vol. 1, pp. 370–379, 2014.

[29] O. A. Beg, T. T. Johnson, and A. Davoudi, “Detection of false-data injection attacks in cyber-physical dc microgrids,” IEEE Transactions on Industrial Informatics, vol. 13, pp. 2693–2703, 2017.

[30] J. Zhao, G. Zhang, M. L. Scala, Z. Y. Dong, C. Chen, and J. Wang, “Short-term state forecasting-aided method for detection of smart grid general false data injection attacks,” IEEE Transactions on Smart Grid, vol. 8, pp. 1580–1590, 2017.

[31] R. Xu, R. Wang, Z. Guan, L. Wu, J. Wu, and X. Du, “Achieving efficient detection against false data injection attacks in smart grid,” IEEE Access, vol. 5, pp. 13787–13798, 2017.

[32] U. Adhikari, T. H. Morris, and S. Pan, “Applying non-nested generalized exemplars classification for cyber-power event and intrusion detection,” IEEE Transactions on Smart Grid, vol. 9, pp. 3928–3941, 2018.

[33] M. Esmailifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, “Detecting stealthy false data injection using machine learning in smart grid,” IEEE Systems Journal, vol. 11, pp. 1644–1652, 2017.

[34] K. Khatana, B. K. Panigrahi, and A. Joshi, “AI-based approach to identify compromised meters in data integrity attacks on smart grid,” IET Generation, Transmission & Distribution, vol. 12, pp. 1052–1066, 2018.

[35] G. E. Hinton, A Practical Guide to Training Restricted Boltzmann Machines. Springer Berlin Heidelberg, 2012, pp. 599–619.

[36] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[37] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (ELUs),” 2015.

[38] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, “Modeling long-and short-term temporal patterns with deep neural networks,” in International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 95–104.

[39] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory. MIT Press, 1997, vol. 9.

[40] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, Dive into Deep Learning, 2020.

[41] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in International Conference for Learning Representations, 2015.