Chen *et al.*
Protection and Control of Modern Power Systems (2023) 8:16
https://doi.org/10.1186/s41601-023-00287-w

Protection and Control of Modern Power Systems

ORIGINAL RESEARCH

Open Access

Detection of false data injection attacks on power systems using graph edge-conditioned convolutional networks

Bairen Chen, Q. H. Wu, Mengshi Li and Kaishun Xiahou*

Abstract

State estimation plays a vital role in the stable operation of modern power systems, but it is vulnerable to cyber attacks. False data injection attacks (FDIA), one of the most common cyber attacks, can tamper with measurement data and bypass the bad data detection (BDD) mechanism, leading to incorrect results of power system state estimation (PSSE). This paper presents a detection framework of FDIA for PSSE based on graph edge-conditioned convolutional networks (GECCN), which use topology information, node features and edge features. Through deep graph architecture, the correlation of sample data is effectively mined to establish the mapping relationship between the estimated values of measurements and the actual states of power systems. In addition, the edge-conditioned convolution operation allows processing data sets with different graph structures. Case studies are undertaken on the IEEE 14-bus system under different attack intensities and degrees to evaluate the performance of GECCN. Simulation results show that GECCN has better detection performance than convolutional neural networks, deep neural networks and support vector machine. Moreover, the satisfactory detection performance obtained with the data sets of the IEEE 14-bus, 30-bus and 118-bus systems verifies the effective scalability of GECCN.

Keywords Power system state estimation (PSSE), Bad data detection (BDD), False data injection attacks (FDIA), Graph edge-conditioned convolutional networks (GECCN)

1 Introduction

With the increased access of various sensing and communication devices, traditional power grids are gradually being transformed to smart grids. Comprehensive and detailed information of power grids can be obtained in real time [1]. However, due to the high dependence on cyber systems, there are great concerns about the reliability and security of smart grids. Cyber security plays an increasingly important role in the operation of power systems [2]. In addition, the frequent cyber attacks in recent years have sounded the alarm for the security of power systems [3–5].

Data integrity attacks are one of the common forms of cyber attacks, and mainly hinder the normal data exchange of power grids by injecting false data or illegally tampering with data. False data injection attacks (FDIA) [6] are the typical attack methods. FDIA can bypass the bad data detection (BDD) mechanism of power system state estimation (PSSE) and mislead the state estimation results, thus it is considered to be one of the most challenging threats to the safe operation of power systems.

FDIA for PSSE was first introduced in [7], and since then, a lot of researches have been devoted to the detection and defense against FDIA [8, 9]. At different times, researchers have conducted comprehensive investigations of such cyber attacks [10–13]. Most previous cyber attack detection algorithms are model-based methods,

*Correspondence:
Kaishun Xiahou
xiahouks@scut.edu.cn
School of Electric Power Engineering, South China University of Technology, Guangzhou 510641, China

such as Kalman filter [14], Kullback-Leibler distance [15], adaptive nonparametric cumulative sum [16], etc. However, the performance of model-based methods is often sensitive to the parameters and a slight uncertainty of the parameters leads to poor detection results. Moreover, model-based methods need choose the threshold, which is difficult to set when the system is subject to dynamic changes or loading variations [12]. In addition, model-based approaches fail to make good use of the increasing data of power grids.

Because of the continuous development of high-performance computing hardware, data-driven methods like machine learning have been applied to researches in various fields [17–19]. Using the historical data accumulated by power grids and the real-time data continuously obtained by measurement equipment, the application of data-driven methods to various power system scenarios is also emerging. Researches have been carried out on various scenarios for FDIA and the development of corresponding detection strategies. In [20], support vector machine (SVM) and the deviation in measurements are used to detect FDIA. However, the situation of sample imbalance, the impact of different attack intensities and degrees, the comparison of different models, and the scalability of the models are not considered. Reference [21] uses SVM, k-nearest neighbor, extended nearest neighbor models on the basis of [20] to design detectors for FDIA. Considering the situation of sample imbalance, different attack intensities and degrees, the performances of the three models are evaluated and compared. However, the scalability of the detector is still ignored in [21] and it is difficult to implement effectively in large-scale data sets with imbalanced samples. Neural network models have obvious advantages in dealing with the above problems. In [22, 23], conditional deep belief network model and gated recurrent unit are each used to perform detection of FDIA in real time. These neural network structures can deeply extract data information and improve detection accuracy. Nevertheless, in view of the problem of model scalability, the above two models are only suitable for specific data sets with a single and fixed structure, while they are not well adapted to data sets with different structures. Also, the data-driven models proposed in [20–23] force the input data into a table format, while the topology of power systems is not effectively used.

Graph neural networks have attracted great attention in the field of deep learning in recent years. They have achieved excellent performance in graph structure data such as citation recommendation, link prediction, protein structure inference, chemical molecular property classification, etc [24, 25]. A variety of well-known types have been derived, such as graph convolutional networks (GCN) [26, 27], graph attention networks [28], etc. However, the commonly used graph neural network models such as GCN do not make full use of edge features, where graphs may carry a lot of additional information. These models only use binary adjacency matrices to indicate whether there are connections between nodes. In order to utilize edge features, reference [29] introduces dynamic edge-conditioned filters [30] in convolutional neural networks on graphs. These can also process graphs of different sizes and connectivity.

Similar to general non-Euclidean structures such as transportation networks and social networks, power systems can also be abstracted into graphs composed of nodes and edges. The methods based on GCN have been explored in several fields of smart grids [31, 32]. Nevertheless, to the best of our knowledge, the application in the field of FDIA is still in its infancy.

This paper proposes a detection method for FDIA based on graph edge-conditioned convolutional networks (GECCN) [29], which incorporates dynamic edge-conditioned filters [30] into the convolution operation of the graph structure. Case studies are mainly carried out on the IEEE 14-bus system to demonstrate the effectiveness and validity of the GECCN model. The main contributions of this paper are as follows.

(1) A detection framework based on GECCN is proposed for FDIA, which takes advantages of the GCN model by considering the information of system topology structure and node features, and also makes up for the drawback of GCN which fails to employ the edge feature information of power systems.
(2) The edge-conditioned convolution (ECC) operation is employed in GECCN to improve the adaptation to the data sets which contain different topology structures of power systems, thus enhancing the scalability of the detection framework.
(3) The detection performance of the GECCN model is investigated and compared with commonly used data-driven models under different attack intensities and degrees.

The rest of this paper is organized as follows. In Sect. 2, the basic concepts, including PSSE, BDD and FDIA, are briefly introduced. ECC operation, the structure of GECCN model and the proposed detection framework for FDIA are elaborated in Sect. 3. Section 4 gives the case studies and simulation results, and finally, Sect. 5 concludes the paper.

2 False data injection attacks against state estimation

In this section, the PSSE and BDD mechanisms are first introduced. Then, the construction method of FDIA in the case of state estimation is given.

2.1 Power system state estimation

State estimation is a process to estimate the state variables of power systems by eliminating inaccuracies and errors from meter measurements. The results of state estimation are typically used in security assessment, such as contingency analysis, preventive control, security constrained optimal power flow, etc [33].

In order to simplify the calculation process of obtaining the system estimated states, this paper considers the linear form of state estimation, which can be expressed as

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{z} = (z_1, z_2, ..., z_m)^\top \in \mathbb{R}^m$ represents the meter measurements, $\boldsymbol{x} = (x_1, x_2, ..., x_n)^\top \in \mathbb{R}^n$ represents the system state variables, and $\boldsymbol{e} = (e_1, e_2, ..., e_m)^\top \in \mathbb{R}^m$ represents the measurement errors which are often assumed to be additive white Gaussian noise with variance σ^2 (i.e., $e_i \sim N(0, \sigma_i^2), i = 1, 2, ..., m$). $\boldsymbol{H} \in \mathbb{R}^{m \times n}$ is the measurement Jacobian matrix, which is determined by grid topology and line parameters. Note that the number of valid measurements m should be greater than the number of system state variables n to ensure system observability (i.e., $m > n$).

The weighted least squares (WLS) method is one of the most commonly used algorithms for state estimation. The WLS method minimizes the objective function $J(\boldsymbol{x})$, as

$$\min J(\boldsymbol{x}) = (\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x})^\top \boldsymbol{R}^{-1} (\boldsymbol{z} - \boldsymbol{H}\boldsymbol{x}) \tag{2}$$

and the estimated state variables $\hat{\boldsymbol{x}}$ are computed by

$$\hat{\boldsymbol{x}} = (\boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{z} \tag{3}$$

where $\boldsymbol{R}$ is a diagonal matrix composed of variances of the measurement errors (i.e., $R_{ii} = \sigma_i^2$).

2.2 Bad data detection

Measurements may contain bad data for various reasons. In addition to noises caused by finite accuracy of the meters and the telecommunication medium, bad data may also be introduced by equipment failure, wrong connections, and communication system interference, etc. BDD is intended to detect, identify and eliminate bad measurements [33].

The largest normalized residual (LNR) test is widely used in BDD, and is formalized as

$$\boldsymbol{r}^{\text{nor}} = \frac{|\boldsymbol{r}|}{\sqrt{R_{ii} S_{ii}}} \tag{4}$$

where $\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{H}\hat{\boldsymbol{x}}$ represents the measurement residual vector, $\boldsymbol{S} = \boldsymbol{I} - \boldsymbol{H}(\boldsymbol{G}^{-1}\boldsymbol{H}^\top \boldsymbol{R}^{-1})$ represents the residual sensitivity matrix, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{G} = \boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H}$ is the gain matrix.

Find k such that r_k^{nor} is the largest among all $r_i^{\text{nor}} \in \boldsymbol{r}^{\text{nor}}, i = 1, 2, ..., m$. If $r_k^{\text{nor}} \geq \tau$, where τ is a chosen identification threshold, the estimated state variables are considered to be affected by bad data, otherwise they are trustworthy.

2.3 False data injection attacks

In FDIA, the attackers can hack into a subset of meters and send changed readings to force the state estimator to obtain false estimated state variables [7]. Specifically, they intend to mislead the operator to consider a compromised $\hat{\boldsymbol{x}}_{\text{att}} = \hat{\boldsymbol{x}} + \boldsymbol{c}$ as the current estimated state variables, where $\boldsymbol{c} \neq \boldsymbol{0}$ is the deviation vector of the estimated state variables before and after the attack.

To achieve this goal, the attacker changes the received measurements $\boldsymbol{z}$ at the control center to $\boldsymbol{z}_{\text{att}} = \boldsymbol{z} + \boldsymbol{a}$, where $\boldsymbol{z}_{\text{att}}$ is the compromised measurement vector, and $\boldsymbol{a}$ is the injected attack vector. The vector of the false estimated state variables $\hat{\boldsymbol{x}}_{\text{att}}$ obtained from (3) is

$$\begin{aligned} \hat{\boldsymbol{x}}_{\text{att}} &= (\boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{z}_{\text{att}} \\ &= (\boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^\top \boldsymbol{R}^{-1} (\boldsymbol{z} + \boldsymbol{a}) \\ &= \hat{\boldsymbol{x}} + (\boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{a}. \end{aligned} \tag{5}$$

To bypass the BDD mechanism, the attackers deliberately design the attacker vector $\boldsymbol{a}$ that satisfies $\boldsymbol{a} = \boldsymbol{H}\boldsymbol{c}$, and the false measurement residual vector $\boldsymbol{r}_{\text{att}}$ is the same as the normal measurement residual vector $\boldsymbol{r}$, as shown below

$$\begin{aligned} \boldsymbol{r}_{\text{att}} &= \boldsymbol{z}_{\text{att}} - \boldsymbol{H}\hat{\boldsymbol{x}}_{\text{att}} \\ &= \boldsymbol{z} + \boldsymbol{a} - \boldsymbol{H}\hat{\boldsymbol{x}} - \boldsymbol{H}(\boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{a} \\ &= \boldsymbol{z} + \boldsymbol{H}\boldsymbol{c} - \boldsymbol{H}\hat{\boldsymbol{x}} - \boldsymbol{H}(\boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H}\boldsymbol{c} \\ &= \boldsymbol{z} + \boldsymbol{H}\boldsymbol{c} - \boldsymbol{H}\hat{\boldsymbol{x}} - \boldsymbol{H}\boldsymbol{c} = \boldsymbol{r}. \end{aligned} \tag{6}$$

Therefore, $\boldsymbol{r}_{\text{att}}^{\text{nor}} = \boldsymbol{r}^{\text{nor}}$, and this well-designed unobservable FDIA can tamper with measurement data without being detected.

3 The graph edge-conditioned convolutional network detection framework

In this section, ECC operation is first introduced, then the overall structure of GECCN is presented and finally the proposed detection framework is given.

3.1 Edge-conditioned convolution

In power systems, buses and transmission lines can be regarded as nodes and edges in the graph, respectively. An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is considered, where $\mathcal{V}$ is a set of nodes with $|\mathcal{V}| = p$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges with $|\mathcal{E}| = q$. $l \in \{0, 1, ..., l_{\max}\}$ is the layer index of the feed-forward networks. The nodes and edges in the graph have corresponding features, that is, there are functions $\mathcal{X}^l : \mathcal{V} \mapsto \mathbb{R}^{d_l}$ and $\mathcal{L} : \mathcal{E} \mapsto \mathbb{R}^s$ to assign features to each node and each edge. These two functions can be regarded as matrices $\mathcal{X} \in \mathbb{R}^{p \times d_l}$ and $\mathcal{L} \in \mathbb{R}^{q \times s}$ ($\mathcal{L} \in \mathbb{R}^{p \times p \times s}$), where $\mathcal{X}^0$ is one of the inputs. The neighborhood of node i is defined as $\mathcal{N}(i) = \{j; (j, i) \in \mathcal{E}\} \cup \{i\}$, which contains all adjacent nodes and i itself. The filtered signal $\mathcal{X}^l(i) \in \mathbb{R}^{d_l}$ at node i is normally calculated as a weighted sum of signals $\mathcal{X}^{l-1}(j) \in \mathbb{R}^{d_{l-1}}$ in its neighborhood, $j \in \mathcal{N}(i)$. This commutative aggregation can solve the problem of undefined vertex ordering and varying neighborhood sizes, but it smooths out structural information. In order to retain the structural information, each filtering weight is proposed to be conditioned on the respective edge feature [30]. The method is to define a filter-generating network $\mathcal{F}^l : \mathbb{R}^s \mapsto \mathbb{R}^{d_l \times d_{l-1}}$, and the given edge feature $\mathcal{L}(j, i)$ outputs the edge-specific weight matrix $\Theta^l_{ji} \in \mathbb{R}^{d_l \times d_{l-1}}$ [29].

As shown in Fig. 1, the neighborhood of bus 1 in the IEEE 14-bus system is taken as an example to illustrate ECC. According to the system topology, $\mathcal{N}(1) = \{2; 5; 1\}$ and edge features include $\mathcal{L}(1, 1)$, $\mathcal{L}(2, 1)$, $\mathcal{L}(5, 1)$. The feature $\mathcal{X}^l(1)$ on bus 1 in the l^{th} network layer is computed as a weighted sum of features $\mathcal{X}^{l-1}(\cdot)$ on the set of its predecessor nodes. The specific weight matrices are dynamically generated by the filter-generating network $\mathcal{F}^l$ based on the corresponding edge features $\mathcal{L}(\cdot)$.

The form of ECC operation is given as

$$\begin{aligned}\mathcal{X}^l(i) &= \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathcal{F}^l(\mathcal{L}(j, i); w^l) \mathcal{X}^{l-1}(j) + b^l \\ &= \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \Theta^l_{ji} \mathcal{X}^{l-1}(j) + b^l\end{aligned} \tag{7}$$

where b^l is the learnable bias and w^l is the learnable network weights. Note that b^l and w^l are model parameters updated only during training. The filter-generating network $\mathcal{F}^l$ is parameterized by w^l and it can be implemented with any differentiable architecture, where multilayer perceptrons are used by default. The edge-specific weight matrix Θ^l_{ji} contains dynamically generated parameters for an edge feature in a particular input graph.

3.2 The configuration of graph edge-conditioned convolutional networks for FDIA

The GECCN model used for detection is shown in Fig. 2. The input includes node features $\mathcal{X}$ and edge features $\mathcal{L}$. The adjacency matrix used to represent the graph structure is included in $\mathcal{L}$. The output is the probability of the sample being subject to FDIA.

The network configuration can be described as [ECC(32) – Acti(tanh) – BN]×l – GSP – FC(1) – Acti(sigmoid). ECC(32) denotes an edge-conditioned convolutional layer with 32 output channels, where Conv represents the convolution operation and the filter-generating network $\mathcal{F}$ is configured as FC(16) – Acti(tanh) – FC(32). The specific operational form of ECC has been shown in (7). Acti(tanh/sigmoid) denotes a *tanh/sigmoid* activation function to increase nonlinearity. BN denotes the batch normalization layer, which is used to accelerate learning convergence. [ECC – Acti – BN] is defined as a block, and l is the layer index defined above and can be used to represent the number of blocks, which will be determined in Sect. 4.2.1. GSP denotes the global sum pooling layer, which pools a graph by computing the sum of its node features. FC(16/32/1) is the fully-connected layer with 16/32/1 output channels.

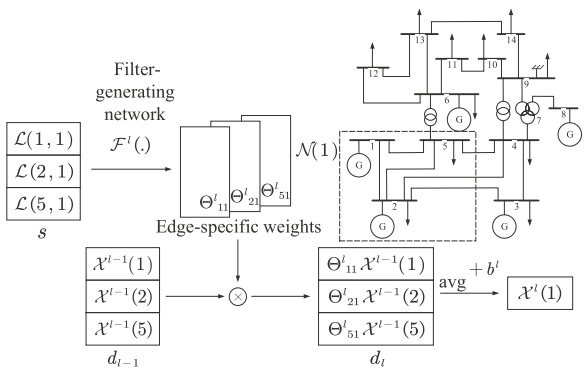


Fig. 1 Illustration of ECC on bus 1 of the IEEE 14-bus system

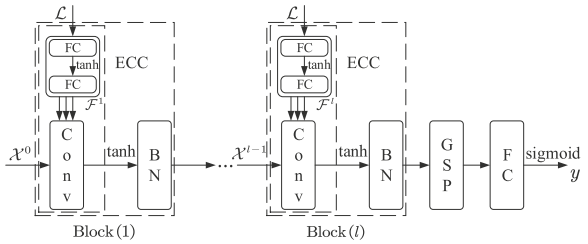


Fig. 2 Structure of GECCN

3.3 The proposed detection framework

The proposed detection framework for FDIA is depicted in Fig. 3. The black boxes and characters represent the normal state estimation process, as shown in Sects. 2.1 and 2.2. The red box and characters represent the process of FDIA for PSSE, which is illustrated in Sect. 2.3. The main process of data generation is to collect measurements and combine them with the network topology to realize PSSE. Then, BDD is performed to detect and delete the unqualified data. Finally, a valid data set of estimated measurements is obtained, which contains normal data (label 0) and compromised data (label 1). The blue box and characters represent the GECCN-based FDIA detection framework. The specific process is to divide the valid data set into training and testing sets in the ratio of 4:1. The training set is used for GECCN model training. After training, the testing set is used for model evaluation, and the detection performance is tested using evaluation indicators. The specific evaluation indicators will be described in Sect. 4.1.2.

4 Case studies

In this section, the performance of the proposed detection framework for FDIA based on the GECCN model is evaluated. Data generation and simulation settings are first introduced, and then the effectiveness of the detection framework is verified by 4 case studies.

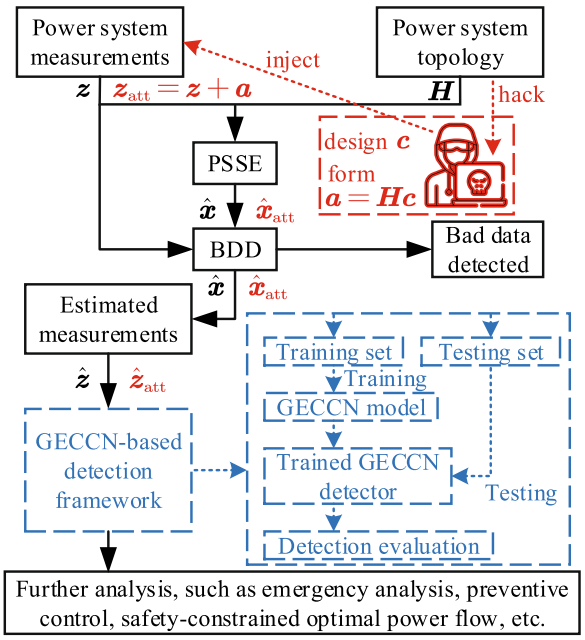


Fig. 3 Detection framework for FDIA

4.1 Data generation and simulation settings

4.1.1 Data generation

The test data in this paper are generated by simulation using the MATGRID toolkit [34]. The data set contains normal samples labeled 0 and compromised samples labeled 1. The values of each sample are the estimated values of measurements after PSSE and BDD.

The data of normal state estimation are generated as follows. First, each sample should determine the operating point by setting, for each load, a random value between 80% and 120% of the initial value. After obtaining a set of different operating points, power flow is performed and the corresponding exact solutions are corrupted by the additive white Gaussian noises with specific variances to form the measurements $\boldsymbol{z}$ required for state estimation. Since the measurements are assumed to be obtained based on supervisory control and data acquisition (SCADA), all variances are set to 0.01. Then, PSSE and BDD are performed. The threshold τ in BDD is set to 3.0 in accordance to [33]. Finally, the corresponding valid estimated measurements $\hat{\boldsymbol{z}}$ can be obtained.

For the generation of compromised data, the above process of normal PSSE needs to be performed first. Then, the $\boldsymbol{H}$ matrix is extracted according to Sect. 2.3, and the state variable deviation vector $\boldsymbol{c}$ is determined by attack intensities and degrees. The attack intensities are set to 3 types with standard deviation $\sigma = 0.01, 0.1, 1$ of normal distribution, representing low, medium, and high attack intensities. It can be expressed as $ai = 0.01/0.1/1$, where ai is the specific value of the standard deviation. The attack degrees are defined as γ_i, where i represents the number of compromised buses. It can be expressed as $\gamma_i = Randperm(n, i)$, where the $Randperm(n, i)$ function returns a vector containing i unique integers randomly selected from 1 to n. In general, the value of n is the number of system buses and i is the number of compromised buses. The specific form of the deviation vector is expressed as $\boldsymbol{c}(\gamma_i) = Normrnd(0, ai, [1, i])$, where the $Normrnd(0, ai, [1, i])$ function generates i random values of normal distribution with mean 0 and standard deviation ai, and then fills in the corresponding bus. Taking the IEEE 14-bus system as an example, the attack intensity is randomly set to medium intensity, that is, $ai = 0.1$, and the attack degree is randomly set to γ_3, that is, 3 randomly selected buses are subject to FDIA. Run the $Randperm(n, i)$ function, if $\gamma_3 = Randperm(14, 3) = [2, 6, 11]$, it means that the corresponding bus will suffer from medium-intensity attacks, and $\boldsymbol{c}$ can be randomly generated as $\boldsymbol{c} = [0, -0.121, 0, 0, 0, 0.072, 0, 0, 0, 0, 0.163, 0, 0, 0]^\top$. After that, the attack vector $\boldsymbol{a} = \boldsymbol{Hc}$ is formed and injected into the measurements $\boldsymbol{z}$ ($\boldsymbol{z}_{att} = \boldsymbol{z} + \boldsymbol{a}$). After PSSE and BDD,

the estimated values of the compromised measurements $\hat{\boldsymbol{z}}_{\text{att}}$ are obtained.

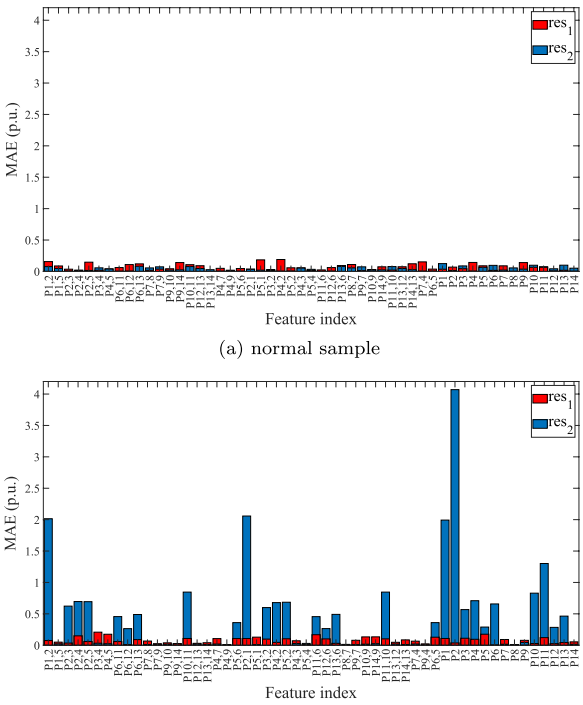
In summary, the pseudocode for data generation is illustrated in **Algorithm 1**.

Algorithm 1: Pseudocode for data generation

Input: $\boldsymbol{z}/\boldsymbol{z}_{\text{att}}$
Output: $\hat{\boldsymbol{z}}/\hat{\boldsymbol{z}}_{\text{att}}$
$nb \leftarrow 14/30/118$; $trig \leftarrow 0/1$; // number of IEEE system bus nb; label $trig$, 0 normal, 1 FDIA
$ai \leftarrow 0.01/0.1/1$; $ad \leftarrow 1/2/.../nb$; // the standard deviation value of attack intensity ai; the value of attack degree ad
$vn \leftarrow 0$; $ns \leftarrow ...$; $\boldsymbol{vr} \leftarrow []$; // number of valid samples vn; total number of samples ns; all value results $\boldsymbol{vr}$
while $vn \leq ns$ **do**
  $\boldsymbol{ll} \leftarrow 0.8 + 0.4 \cdot Rand(nb, 1)$; // load level $\boldsymbol{ll}$
  $\boldsymbol{bm} \leftarrow Load(\texttt{MATGRID}.nb)$; // benchmark load value $\boldsymbol{bm}$
  $\boldsymbol{nbm} \leftarrow \boldsymbol{bm} \cdot \boldsymbol{ll}$; // new load value $\boldsymbol{nbm}$
  $\boldsymbol{z} \leftarrow \texttt{MATGRID}.Rungen(\boldsymbol{nbm}, 0.01)$; // generate $\boldsymbol{z}$
  if $trig == 0$ **then**
    $\hat{\boldsymbol{z}} \leftarrow \texttt{MATGRID}.Runse(\boldsymbol{z}, Bdd)$; // apply PSSE & BDD, and generate $\hat{\boldsymbol{z}}$
  else if $trig == 1$ **then**
    $\boldsymbol{ad} \leftarrow Randperm(nb, ad)$; $\boldsymbol{H} \leftarrow \texttt{MATGRID}.nb(H)$; $\boldsymbol{c} \leftarrow Zeros(nb, 1)$;
    $\boldsymbol{c}(\boldsymbol{ad}) \leftarrow Normrnd(0, ai, [1, Size(\boldsymbol{ad}, 2)])$; // construct $\boldsymbol{c}$ from ai & $\boldsymbol{ad}$
    $\boldsymbol{a} \leftarrow \boldsymbol{H} \cdot \boldsymbol{c}$; $\boldsymbol{z}_{\text{att}} \leftarrow \boldsymbol{z} + \boldsymbol{a}$; // construct $\boldsymbol{a}$, apply FDIA, and get $\boldsymbol{z}_{\text{att}}$
    $\hat{\boldsymbol{z}}_{\text{att}} \leftarrow \texttt{MATGRID}.Runse(\boldsymbol{z}_{\text{att}}, Bdd)$; // apply PSSE & BDD, and get $\hat{\boldsymbol{z}}_{\text{att}}$
  else
    $Disp$(Please set $trig$ to 0/1.); **break**;
  end
  if $Size(\hat{\boldsymbol{z}}/\hat{\boldsymbol{z}}_{\text{att}}) == Size(\boldsymbol{z})$ **then** // check if the sample is valid
    $\boldsymbol{vr} \leftarrow [\boldsymbol{vr}; \hat{\boldsymbol{z}}/\hat{\boldsymbol{z}}_{\text{att}}]$; $vn \leftarrow vn + 1$; // current sample $\hat{\boldsymbol{z}}/\hat{\boldsymbol{z}}_{\text{att}}$ added to $\boldsymbol{vr}$
  end
end
$Save(\boldsymbol{vr}, trig)$; // save $\boldsymbol{vr}$ & $trig$

A normal sample and a compromised sample from the data set are randomly selected for analysis. These two samples have the same initial operating point. Defining res_1 as the residual between estimated and measurement values, and res_2 as the residual between estimated and real values. The comparison of res_1 (red bar) and res_2 (blue bar) under normal and compromised samples is presented in Fig. 4.

For the PSSE of the IEEE 14-bus system, the number of feature indices is 54, including 40 active power flows of the transmission lines and 14 active power injections of the buses. For the compromised sample, using the example mentioned above, the attack intensity is medium and the attack targets are buses $\{2, 6, 11\}$. From Fig. 4a, it can be seen that with a normal sample, res_1 and res_2 are not very different. The mean absolute error (MAE) of each



(a) normal sample

(b) compromised sample

Fig. 4 Comparison under **a** normal sample and **b** compromised sample

feature index is around 0.1 p.u.. However, from Fig. 4b, it can be seen that with a compromised sample, res_2 of several feature indices are significantly larger than the corresponding res_1. In other words, the deviation between the estimated values and real values is large, while the deviation between the estimated values and measurement values is too small to be detected by the BDD mechanism. This is clearly harmful to power systems.

4.1.2 Simulation settings

Simulation settings include parameter settings and evaluation index determination.

Parameter settings are first introduced. The proposed model is built on the Spektral toolkit [35], which is a Python library for graph deep learning based on Tensorflow and Keras. The batch size of the entire simulation is set to 32, and the value of epoch is set to 100. The Adam optimizer with a default learning rate 0.001 is used, which is considered to be one of the most popular optimizers in deep learning. For the loss function, since it is essentially a supervised binary classification problem, the binary cross-entropy loss function is chosen. All the simulations are run on an I7-7700 CPU and an Nvidia GeForce RTX 2080.

For the evaluation indices of the experimental results, accuracy (Acc), precision (Pre), recall (Reca) and F_1 score

Table 1 Confusion Matrix for Binary Classification

Data class	Classified as positive	Classified as negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

(F_1) are chosen. The definitions of true positive (TP), false positive (FP), false negative (FN) and true negative (TN) are shown in Table 1. In the simulations, the FDIA sample is determined as positive (label 1) and the normal sample is determined as negative (label 0).

The four indices are defined as

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{8a}$$

$$\mathrm{Pre} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{8b}$$

$$\mathrm{Reca} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{8c}$$

$$\mathrm{F}_1 = \frac{2 \cdot \mathrm{Pre} \cdot \mathrm{Reca}}{\mathrm{Pre} + \mathrm{Reca}} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{8d}$$

where Acc indicates the overall effectiveness of a classifier, and Pre denotes class agreement of the data labels with the positive labels given by the classifier and higher Pre value means higher detection accuracy. Reca represents the effectiveness of a classifier to identify positive labels, and F_1 shows the relationship between the positive labels of data and those given by the classifier, the value of which is expected to be as high as possible.

4.2 Performance evaluation of GECCN

4.2.1 Determination of the number of blocks in the proposed GECCN model

In this case, the optimal number of blocks in the GECCN model based on the IEEE 14-bus system is determined first. As mentioned in Sect. 3.2, a block contains the ECC layer, tanh activation and BN layer. Block(l) means there are l blocks stacked. In the tests, 7 scenarios are set, from 1 block to 7 blocks, to determine the optimal number.

From **Algorithm 1**, normal and compromised samples are generated. The number of normal samples is set to 50000. For the compromised samples, since the number of samples is much smaller than that of normal samples, the ratio of 1:10 is considered and thus 5000 FDIA samples are generated. Among them, the attack intensity is selected by a random function at three different intensities, and the attack target is also set by a random function. Thus, a total of 55000 valid samples are obtained, which are divided into training and testing sets in a 4:1 ratio.

The results of various evaluation indices for each scenario are presented in Table 2. It can be seen that Acc and Pre are basically above 99%. When there are 3 blocks, the value of Acc is the highest, reaching 99.5%, whereas when there are 2 blocks, Pre reaches the highest value. Reca remains at around 90% and reaches a highest value of 94.5122% when $l = 3$. Compared with Pre, Reca should have more attention paid, because its value is related to the size of FN in the confusion matrix. As FN represents the number of compromised samples misjudged as normal samples, it is expected to be as small as possible, so as to obtain a larger value of Reca. The value of F_1 is largely maintained between 94% and 97%, and it is an important evaluation index in an imbalanced sample data set. When the number of blocks is 3, F_1 has a maximum value of 97.1279%. It can be seen that when 3 blocks are stacked, Acc, Reca, and F_1 all reach the highest values, while the corresponding value of Pre is only 0.0011% lower than the maximum value of Pre. The value of Pre is related to FP, and FP represents the number of normal samples misjudged as compromised samples. The influence of FP is not that great compared to FN. In summary, 3 blocks are chosen as the final number of layers for the subsequent case studies.

The dimension reduction and visualization operations are intended to be performed on the output of the intermediate layers of the proposed model, which consists of 3 blocks (each block contains [ECC – Acti(tanh) – BN]) and 1 GSP layer. The output visualization is depicted in Fig. 5. The operation is to first reduce high-dimensional data to 20 dimensions using principal component analysis (PCA) followed by t-SNE to 2d-space. In Fig. 5, label 0 (blue dots) represents normal samples and label 1 (orange dots) represents compromised samples. It can be seen that after each intermediate layer, the clustering effect of the two types of samples in the new feature space becomes more and more obvious, i.e., one is that the boundary of different types is increasingly clear, and

Table 2 Results of Different Numbers of Blocks

Block(l)	Acc	Pre	Reca	F_1
Block(1)	0.989636	0.990011	0.894684	0.939937
Block(2)	0.992455	**0.998937**	0.919765	0.957718
Block(3)	**0.995000**	0.998926	**0.945122**	**0.971279**
Block(4)	0.992636	0.998907	0.919517	0.957569
Block(5)	0.992273	0.998913	0.916251	0.955798
Block(6)	0.992182	0.998891	0.913793	0.954449
Block(7)	0.991818	0.993443	0.915408	0.952830

the other is that the samples of the same type are gradually aggregated together.

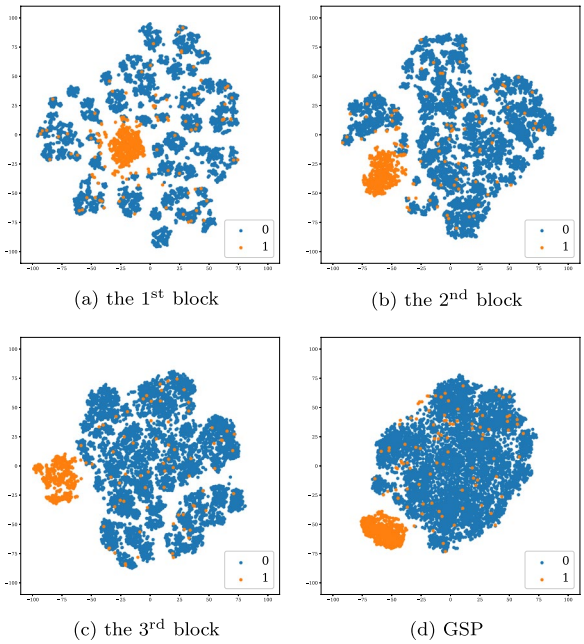
4.2.2 Detection performance under different attack intensities and degrees

In this case, the GECCN model is used to evaluate the detection performance under different attack intensities and degrees in the IEEE 14-bus system. As mentioned in Sect. 4.1.1, three different intensities are set, and the attack degrees refer to the number of non-zero values in the vector $\boldsymbol{c}$. For the IEEE 14-bus system, up to 14 different attack degrees can be set, that is, 1 to 14 points are randomly selected from the vector $\boldsymbol{c}$ to fill in the corresponding non-zero values of attack intensity. Attack degrees are defined as γ_i, where i ranges from 1 to 14 in the IEEE 14-bus system. A larger value of i means a deeper attack degree.

Since the number of attack simulation scenarios is the attack intensities × attack degrees, there are a total of $3 \times 14 = 42$ simulation scenarios for model performance evaluation. The number of compromised samples in each scenario is set to 5000, and the number of normal samples is set to 50000. Mixing normal samples and compromised samples, the data set of each scenario will be divided into training and testing sets in the same 4:1 ratio for model training and testing. The results are presented in Table 3.

It can be seen that under different attack intensities and degrees, Acc and Pre are basically above 99%. Reca and F_1 of low attack intensity initially increase and then slightly decrease as the attack degree deepens. The Reca values of medium and high attack intensities are basically 1 after the attack degrees are deeper than γ_2, that is, the number of compromised samples misjudged as normal samples is 0, which shows excellent performance. The F_1 values of medium and high attack intensities basically show upward trends with the increase of attack degree, and are higher than 99% when the attack degree is deeper than γ_2.

The numbers of FP and FN in the simulation are depicted in Fig. 6. It can be seen from Fig. 6a that



(a) the 1st block (b) the 2nd block

(c) the 3rd block (d) GSP

Fig. 5 Output visualization of the intermediate layers

Table 3 Detection Results of Different Attack Intensities and Attack Degrees

Attack	Low attack intensity				Medium attack intensity				High attack intensity			
Degrees	Acc	Pre	Reca	F_1	Acc	Pre	Reca	F_1	Acc	Pre	Reca	F_1
γ_1	0.9918	0.9955	0.9109	0.9513	0.9941	0.9938	0.9418	0.9671	0.9940	0.9926	0.9408	0.9660
γ_2	0.9976	0.9916	0.9813	0.9864	0.9995	0.9939	1.0000	0.9969	0.9995	0.9951	0.9990	0.9971
γ_3	0.9992	0.9931	0.9980	0.9955	0.9995	0.9939	1.0000	0.9970	0.9998	0.9980	1.0000	0.9990
γ_4	0.9989	0.9886	0.9990	0.9938	0.9990	0.9888	1.0000	0.9944	0.9994	0.9927	1.0000	0.9964
γ_5	0.9993	0.9941	0.9980	0.9961	0.9993	0.9921	1.0000	0.9960	0.9994	0.9930	1.0000	0.9965
γ_6	0.9993	0.9939	0.9980	0.9960	0.9995	0.9940	1.0000	0.9970	0.9993	0.9918	1.0000	0.9959
γ_7	0.9992	0.9924	0.9990	0.9957	0.9998	0.9980	1.0000	0.9990	0.9992	0.9913	1.0000	0.9956
γ_8	0.9993	0.9922	1.0000	0.9961	0.9997	0.9971	1.0000	0.9985	0.9994	0.9928	1.0000	0.9964
γ_9	0.9993	0.9942	0.9981	0.9961	0.9995	0.9937	1.0000	0.9968	0.9995	0.9941	1.0000	0.9971
γ_{10}	0.9993	0.9930	0.9990	0.9960	0.9996	0.9961	1.0000	0.9980	0.9998	0.9980	1.0000	0.9990
γ_{11}	0.9993	0.9916	1.0000	0.9958	0.9995	0.9941	1.0000	0.9970	0.9993	0.9919	1.0000	0.9959
γ_{12}	0.9984	0.9937	0.9875	0.9906	0.9997	0.9969	1.0000	0.9985	0.9993	0.9923	1.0000	0.9961
γ_{13}	0.9982	0.9927	0.9866	0.9896	0.9995	0.9937	1.0000	0.9968	0.9992	0.9912	1.0000	0.9956
γ_{14}	0.9979	0.9937	0.9824	0.9880	0.9995	0.9949	1.0000	0.9974	0.9994	0.9930	1.0000	0.9965

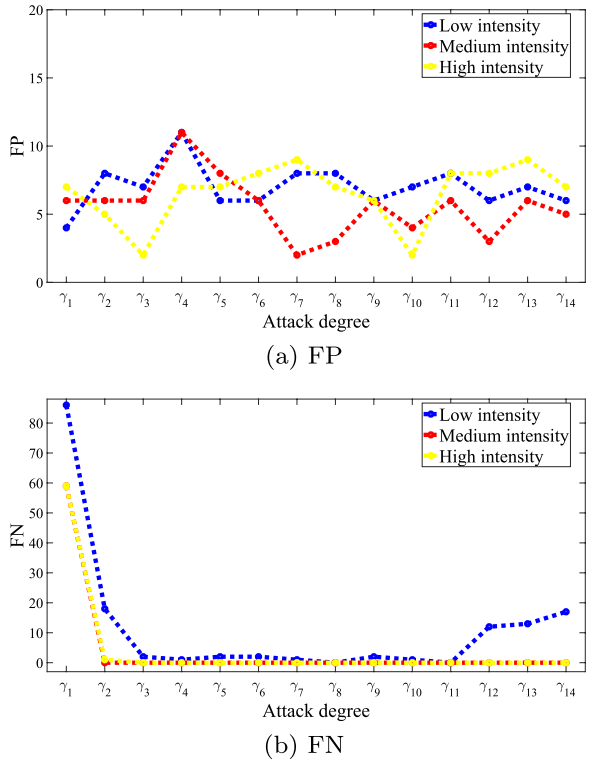


Fig. 6 The number of **a** FP and **b** FN obtained under different attack intensities and degrees

different attack intensities and degrees have little impact on FP, with the number of FP being lower than 15 in each scenario. FP refers to the number of normal samples that are misjudged to be compromised samples. This does not cause great harm to the systems in actual situations since in any case compromised samples need to be further checked and eliminated. From Fig. 6b, FN is largely maintained at 0 when the attack degree is deeper than γ_2 under medium and high attack intensities, that is, the values of Reca in Table 3 are 1. For low attack intensity, FN gradually decreases as the attack degree deepens, and then slightly increases after γ_{12}. It is estimated that the attack intensity is relatively low and the attack degrees cover almost the entire system. In other words, most values are only slightly increased, causing the detection model to misjudge compromised samples to be the normal samples with relatively high initial operating points.

In summary, the detection of FDIA under low attack intensity is more difficult than that under medium and high intensities. The values of the four types of evaluation indices generally show an upward trend as the attack degree deepens. The overall mean of Acc is 99.88%, Pre is 99.36%, Reca is 99.33%, and F_1 is 99.33%. It can be concluded that the proposed GECCN model has excellent detection performance under different attack intensities and degrees.

4.2.3 Detection performance compared with other methods

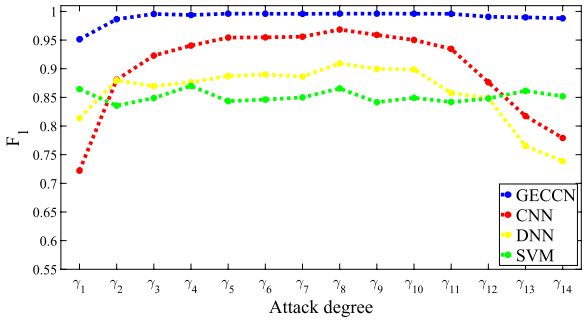
In this case, the GECCN model is compared with convolutional neural networks (CNN), deep neural networks (DNN) and support vector machine (SVM) models under different attack intensities and degrees. F_1 is selected as the evaluation index.

The structure of the CNN model in this paper is configured as Conv1D(5,32) × 3 – Flatten – FC(64) – FC(32) – Acti(tanh) – FC(1) – Acti(sigmoid). Conv1D(5,32) denotes 1-dimensional convolution with kernel size of 5 and 32 output channels. There are 3 Conv1D layers stacked. Flatten represents the flatten layer. The structure of the DNN model is configured as FC(128) – FC(64) – FC(32) – Acti(tanh) – FC(1) – Acti(sigmoid). The structure of the SVM model uses a layer of Random Fourier Features with a linear layer for approximate replacement. The layer of the Random Fourier Features has an output dimension of 4096, a scale of 20 and Gaussian kernel initialization. The linear layer is FC(1) – Acti(sigmoid). The training loss of the SVM uses hinge loss. The data sets and parameters used in training and testing of these models are all consistent with those in Sects. 4.2.2 and 4.1.2. The F_1 results of different models obtained under different attack intensities and degrees are shown in Fig. 7.

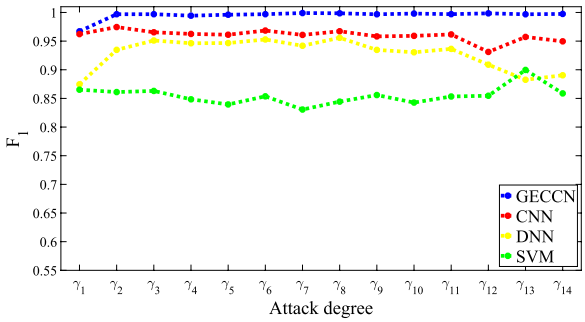
It can be seen from Fig. 7 that F_1 of the SVM is maintained between 80% and 90% under different attack intensities and degrees. F_1 of the DNN fluctuates greatly under low attack intensity, even dropping below 80% when the attack degree exceeds γ_{13}, but remains above 85% under medium and high attack intensities. The detection performance of the DNN is comparable to that of the SVM under low attack intensity but is superior under medium and high intensities. The detection performance of the CNN is better than that of the DNN. F_1 of the CNN under medium and high intensities is around 95%. The detection performance of the CNN is not as good as the SVM when the attack degree is very small or very big under low attack intensity. The F_1 of the GECCN model is higher than the corresponding F_1 of other models regardless of the attack intensity or degree, with F_1 being above 99.5% in most scenarios. Therefore, it can be concluded that the GECCN model has better performance in detection for FDIA than the others.

4.2.4 Scalability performance of detection framework on different systems

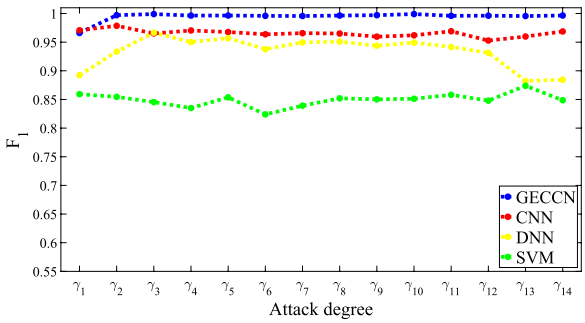
In this case, the GECCN model is applied to different systems to test its detection performance. As



(a) low attack intensity



(b) medium attack intensity



(c) high attack intensity

Fig. 7 Comparisons of different models in different attack degrees under **a** low, **b** medium and **c** high attack intensities

mentioned earlier, ECC can handle graphs of different sizes and connectivity, that is, it can be used for data sets with different graph structures. Data sets with a single topology structure and a mixed structure of different systems are used for testing. The performances of the IEEE 14-bus, IEEE 30-bus and IEEE 118-bus systems are evaluated. Finally, data sets of these three systems are mixed together (the new data set has different graph structures) and the new data set is used for performance evaluation. The data generation methods and simulation settings of the IEEE 30-bus and IEEE 118-bus systems are the same as those of the IEEE 14-bus system. The test results are shown in Fig. 8.

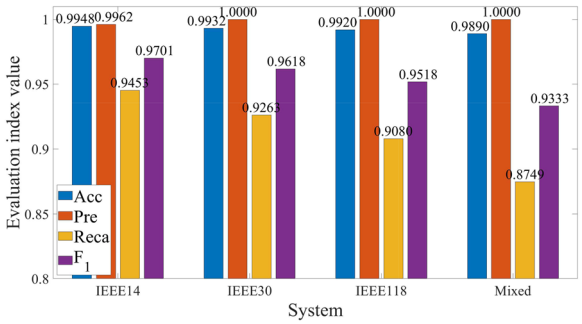


Fig. 8 Scalability performance on different systems

It can be seen that in the single-structure IEEE 14-bus, IEEE 30-bus and IEEE 118-bus systems, Acc and Pre are all above 99%. As the system scale increases, Reca and F_1 gradually decrease, but they remain all above 90%. This is because that as the scale of the system increases, small attack degrees of low attack intensity will become more difficult to detect, which becomes easier to misjudge compromised samples as normal samples, resulting in the decline of Reca and F_1. For the detection of the mixed data set, Acc and Pre are both above 98%. Although Reca is lower than that obtained from data set with single-structure, it is still above 87%. F_1 reaches 93.33%, which is a satisfactory detection performance. Therefore, it can be concluded that the scalability performance of the detection framework on different systems is effective.

5 Conclusion

This paper has proposed a GECCN-based detection framework for FDIA of power systems. From the simulation studies carried out mainly on the IEEE 14-bus system under the conditions of different attack intensities and degrees, conclusions can be drawn as follows.

When determining the optimal number of blocks and verifying through the visualization of the intermediate layers, the GECCN model shows excellent comprehensive detection performance under different attack intensities and degrees. Under different attack intensities, the rates of evaluation indices generally show upward trends as the attack degree deepens. The overall average value of Acc is 99.88%, Pre is 99.36%, Reca is 99.33%, and F_1 is 99.33%. This is due to the fact that the model makes full use of various information such as topology structure, node features and edge features of power systems. Moreover, the number of FP is not affected by various attack scenarios, and the number of FN is almost 0 under medium and high attack intensities. Besides, compared with the commonly used data-driven models such as CNN, DNN, and SVM, the GECCN model has better detection performance and the results of the selected evaluation index

are close to 1 under various attack scenarios. In addition, the ECC operation of GECCN is capable of handling the data sets which contain different structures of the IEEE 14-bus, IEEE 30-bus and IEEE 118-bus systems, thereby maintaining satisfactory detection results obtained under various testing systems and ensuring the scalability of the proposed detection framework.

Acknowledgements
Not applicable.

Author contributions
BC: Investigation, Methodology, Software, Writing - original draft. QHW: Supervision, Writing - review & editing. ML: Writing - review & editing. KX: Data curation, Writing - review & editing, Funding acquisition. All authors read and approved the final manuscript.

Funding
This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B010166004 and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020A1515111100 and in part by the National Natural Science Foundation of China under Grant 52207106 and in part by the Open Fund of State Key Laboratory of Operation and Control of Renewable Energy & Storage Systems (China Electric Power Research Institute) under Grant KJ80-21-001.

Availability of data and materials
All data generated or analysed during this study are included in this published article.

Declarations

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 21 August 2022 Accepted: 7 March 2023
Published online: 07 April 2023

References

1. Morello, R., Mukhopadhyay, S. C., Liu, Z., Slomovitz, D., & Samantaray, S. R. (2017). Advances on sensing technologies for smart cities and power grids: A review. *IEEE Sensors Journal, 17*(23), 7596–7610.
2. Humayed, A., Lin, J., Li, F., & Luo, B. (2017). Cyber-physical systems security-a survey. *IEEE Internet of Things Journal, 4*(6), 1802–1831.
3. Liang, G., Weller, S. R., Zhao, J., Luo, F., & Dong, Z. Y. (2016). The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems, 32*(4), 3317–3318.
4. Xiahou, K., Liu, Y., & Wu, Q. (2020). Robust load frequency control of power systems against random time-delay attacks. *IEEE Transactions on Smart Grid, 12*(1), 909–911.
5. Zhou, T., Xiahou, K., Zhang, L., & Wu, Q. (2021). Multi-agent-based hierarchical detection and mitigation of cyber attacks in power systems. *International Journal of Electrical Power & Energy Systems, 125*, 106516.
6. Hug, G., & Giampapa, J. A. (2012). Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks. *IEEE Transactions on Smart Grid, 3*(3), 1362–1370.
7. Liu, Y., Ning, P., & Reiter, M. K. (2011). False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC), 14*(1), 1–33.
8. Xiahou, K., Liu, Y., & Wu, Q. (2021). Decentralized detection and mitigation of multiple false data injection attacks in multiarea power systems. *IEEE Journal of Emerging and Selected Topics in Industrial Electronics, 3*(1), 101–112.
9. Chen, C., Cui, M., Fang, X., Ren, B., & Chen, Y. (2020). Load altering attack-tolerant defense strategy for load frequency control system. *Applied Energy, 280*, 116015.
10. Yang, Q., Yang, J., Yu, W., An, D., Zhang, N., & Zhao, W. (2013). On false data-injection attacks against power system state estimation: Modeling and countermeasures. *IEEE Transactions on Parallel and Distributed Systems, 25*(3), 717–729.
11. Liang, G., Zhao, J., Luo, F., Weller, S. R., & Dong, Z. Y. (2016). A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid, 8*(4), 1630–1638.
12. Musleh, A. S., Chen, G., & Dong, Z. Y. (2019). A survey on the detection algorithms for false data injection attacks in smart grids. *IEEE Transactions on Smart Grid, 11*(3), 2218–2234.
13. Xu, Y. (2020). A review of cyber security risks of power systems: From static to dynamic false data attacks. *Protection and Control of Modern Power Systems, 5*(1), 1–12.
14. Manandhar, K., Cao, X., Hu, F., & Liu, Y. (2014). Detection of faults and attacks including false data injection attack in smart grid using kalman filter. *IEEE Transactions on Control of Network Systems, 1*(4), 370–379.
15. Chaojun, G., Jirutitijaroen, P., & Motani, M. (2015). Detecting false data injection attacks in ac state estimation. *IEEE Transactions on Smart Grid, 6*(5), 2476–2483.
16. Zhou, T., Xiahou, K., Zhang, L., & Wu, Q. (2020). Real-time detection of cyber-physical false data injection attacks on power systems. *IEEE Transactions on Industrial Informatics, 17*(10), 6810–6819.
17. Simonyan, K., & Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
18. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT press.
19. Xue, Z., Xiahou, K., Li, M., Ji, T., & Wu, Q. (2019). Diagnosis of multiple open-circuit switch faults based on long short-term memory network for dfig-based wind turbine systems. *IEEE Journal of Emerging and Selected Topics in Power Electronics, 8*(3), 2600–2610.
20. Esmalifalak, M., Liu, L., Nguyen, N., Zheng, R., & Han, Z. (2014). Detecting stealthy false data injection using machine learning in smart grid. *IEEE Systems Journal, 11*(3), 1644–1652.
21. Yan, J., Tang, B., & He, H.: Detection of false data attacks in smart grid with supervised learning. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 1395–1402 (2016). IEEE
22. He, Y., Mendis, G. J., & Wei, J. (2017). Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism. *IEEE Transactions on Smart Grid, 8*(5), 2505–2516.
23. James, J., Hou, Y., & Li, V. O. (2018). Online false data injection attack detection with wavelet transform and deep neural networks. *IEEE Transactions on Industrial Informatics, 14*(7), 3271–3280.
24. You, J., Ying, Z., & Leskovec, J. (2020). Design space for graph neural networks. *Advances in Neural Information Processing Systems, 33*, 17009–17021.
25. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems, 32*(1), 4–24.
26. Defferrard, M., Bresson, X., & Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. Advances in Neural Information Processing Systems **29** (2016)
27. Kipf, T.N., & Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
28. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
29. Simonovsky, M., & Komodakis, N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3693–3702 (2017)
30. Jia, X., De Brabandere, B., Tuytelaars, T., & Gool, L.V.: Dynamic filter networks. Advances in Neural Information Processing Systems **29** (2016)
31. Chen, K., Hu, J., Zhang, Y., Yu, Z., & He, J. (2019). Fault location in power distribution systems via deep graph convolutional networks. *IEEE Journal on Selected Areas in Communications, 38*(1), 119–131.

32. Liao, W., Yang, D., Wang, Y., & Ren, X. (2020). Fault diagnosis of power transformers using graph convolutional network. *CSEE Journal of Power and Energy Systems, 7*(2), 241–249.
33. Abur, A., & Expósito, A. G. (2004). *Power System State Estimation: Theory and Implementation*. New York: CRC Press.
34. Cosovic, M.: MATGRID. (2019). https://github.com/mcosovic/MATGRID
35. Grattarola, D., & Alippi, C. (2021). Graph neural networks in tensorflow and keras with spektral. *IEEE Computational Intelligence Magazine, 16*(1), 99–106.