electronics

MDPI

Article

A Deep Learning-Based Classification Scheme for False Data Injection Attack Detection in Power System

Yucheng Ding [1,*], Kang Ma [2], Tianjiao Pu [1], Xinying Wang [1], Ran Li [3] and Dongxia Zhang [1]

[1] China Electric Power Research Institute, Beijing 100192, China; tjpu@epri.sgcc.com.cn (T.P.); wangxinying@epri.sgcc.com.cn (X.W.); zhangdx@epri.sgcc.com.cn (D.Z.)
[2] Department of Electronic and Electrical Engineering, University of Bath, Bath BA2 7AY, UK; K.Ma@bath.ac.uk
[3] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; rl272@bath.ac.uk
* Correspondence: dingyucheng007@gmail.com; Tel.: +86-188-1110-9859

Abstract: A smart grid improves power grid efficiency by using modern information and communication technologies. However, at the same time, due to the dependence on information technology and the deep integration of electrical components and computing information in cyber space, the system might become increasingly vulnerable to cyber-attacks. Among various emerging security problems, a false data injection attack (FDIA) is a new type of attack against the state estimation. In this article, a deep learning-based identification scheme is developed to detect and mitigate information corruption. The scheme implements a conditional deep belief network (CDBN) to analyze time-series input data and leverages captured features to detect the FDIA. The performance of our detection mechanism is validated by using the IEEE 14-bus test system for simulation. Different attack scenarios and parameters are set to demonstrate the feasibility and effectiveness of the developed scheme. Compared with the artificial neural network (ANN) and the support vector machine (SVM), the experimental analyses indicate that the results of our detection mechanism are better than those of the other two in terms of FDIA detection accuracy and robustness.

Keywords: conditional deep belief network; cyber security; false data injection attacks detection; feature extraction; deep learning; smart grids; state estimation

Citation: Ding, Y.; Ma, K.; Pu, T.; Wang, X.; Li, R.; Zhang, D. A Deep Learning-Based Classification Scheme for False Data Injection Attack Detection in Power System. *Electronics* **2021**, *10*, 1459. https://doi.org/10.3390/electronics10121459

Academic Editor: Rui Pedro Lopes

Received: 27 April 2021
Accepted: 2 June 2021
Published: 18 June 2021

1. Introduction

The power system is a complex and interconnected network that transfers electrical energy from generators to users [1,2]. The power grid is continuously operated and monitored by a supervisory control and data acquisition system (SCADA) to ensure a normal operating condition. In particular, the state of the power system is estimated by the measured value, and the system operators use the estimated state to control the actual operation [3–5].

By integrating various advanced communication technologies, the power system is moving towards the direction of the smart grid [6–8]. However, due to the deep integration of the cyber space with the physical space, the power grid is facing increasing security challenges. In addition, massive real-time power system data has brought about the transformative potential and challenge of protecting smart grid systems. Physical security and cyber security are two significant aspects of power system security. Physical security is the ability of a power system to maintain continuous supply in the event of equipment breakdowns. Cyber security refers to the security of a SCADA system that maintains the operation of the power system. Recently, cyber-attacks have gradually threatened modern power systems due to the ubiquitous use of communication technologies [9–11]. Besides, because of the close interlinking between the physical and SCADA systems, the physical security of power systems can be compromised by cyber security vulnerabilities [12–14].

Cyber-attacks have led to numerous incidents and have been concerned by both power system operators and users. They can undermine or even completely disrupt the control system of the power grid. For instance, in 2010, the Iranian nuclear power plant was invaded by a Stuxnet worm that falsely altered the system status, which spread across the whole SCADA system and disrupted system protection strategies. On December 23th, 2015, three Ukrainian regional electric power distribution companies experienced a cyber-attack, which caused power outages affecting nearly 225,000 customers for several hours. Barely a month after the incident, ransomware attacked the Israel Electric Authority through online phishing. The events are fresh examples of the vulnerability of a highly automated smart grid to cyber-attacks. Generally, there are three major types of cyber-attacks: denial of service attack (DoS), replay attack (RA), and false data injection attack (FDIA). The DoS attack occurs when an attacker inserts artificial loads to the service source such that the normal trend of service will be no longer accessible to legitimate requests [15]. The RA involves an attacker replacing the current data with the measurement of a certain period of time before the control center can make correct decisions about the current system state [16]. The FDIA means that an attacker can access the current power system configuration and manipulate the stored data and measurements. This article focuses on the FDIA, which is regarded as a severe threat to the SCADA system.

There is a growing body of literature that recognizes the FDIA. Studies over the past decades have provided valuable information on the FDIA scenarios and the corresponding detection strategies. Bobba et al. [17] investigated the detection of the FDIA by a strategically selected set of measurements and state variables. The authors show that it is useful to defend against such attacks by protecting a set of basic measurements. Pasqualetti et al. [18] proposed a mathematical framework for cyber-physical systems, characterized fundamental monitoring limitations from system-theoretic and graph-theoretic perspectives, and designed centralized and distributed attack detection and identification monitors. In Reference [19], the authors introduced the attack model with the least amount of effort and formulated the attack strategy, in which several meters are selected for manipulation to cause the maximum damage. To defend against the attacks, the authors also investigated the protection-based defence and detection-based defence. In Reference [20], the problem of false data detection was modelled as a matrix separation problem. The nuclear norm minimization method and low rank matrix factorization method are presented. The authors in [21] introduced two distributed detection methods: distributed observable island detection (DOID) algorithm and distributed time approaching detection (DTAD) algorithm. In Reference [22], the equivalent measurement transformation and the residual researching method are utilized to identify false data. However, to some extent, the above-mentioned traditional methods strongly depend on the prescribed bad data detection threshold and are sensitive to environmental noise. Moreover, they are easily affected by the attack intensity, i.e., the smaller the attack intensity, the lower the detection accuracy.

With the rapid development of artificial intelligence technology, the FDIA detection method based on artificial intelligence has also been widely studied. In Reference [23], the authors designed a support vector machine (SVM) based on the the alternating direction method of multipliers, which can effectively identify whether the power system is under attack. Multilayer perceptrons (MLPs), as deep learning models, have been used to detect attacks in [24–26]. They treated the FDIA detection problem as a supervised classification problem. In Reference [27], the authors combined discrete wavelet transform(DWT), dropout with recurrent neural network(RNN), extracted the hidden time-frequency domain characteristics, solved the overfitting problem, and increased the accuracy of FDIA detection. This type of artificial intelligence detection model automatically processes features and the detection accuracy is often higher than the traditional methods, but the training of the model relies on large sample datasets and requires too much computation. In addition, the existing works do not consider the impact of historical measurements on the current situation.

In the recent past, a deep belief network (DBN) was proposed as an unsupervised learning method to learn the hierarchical representations and correlation from real-time data [28,29]. It is one of the basic deep learning technologies built by stacking restricted Boltzmann machines (RBMs) [30–32]. By implementing automatic feature extraction, the DBN can achieve higher efficiency and accuracy than traditional machine learning algorithms [33–35]. Although the DBN demonstrates good performance in static modelling, it encounters challenges in capturing complicated temporal dynamics from time-series input [36]. In light of this, this paper develops an extended version of the DBN, called the conditional deep belief network (CDBN), which updates a conditional Gaussian–Bernoulli RBM (CGBRBM) to model temporal data [37–39]. The CDBN-based approach can then identify the hidden correlation and estimate the reliability of the measurement data. The main contributions of this paper are as follows:

- The standard DBN is improved to deal with the continuous real-time series data of the power system flexibly and extract the time correlation.
- A CDBN-based FDIA detection scheme is proposed to evaluate the reliability of the measurement and ensure the safe and stable operation of the power grid.
- By simulating different attack scenarios, the performance of the proposed scheme is evaluated from multiple aspects to ensure its feasibility and effectiveness.

Section 2 presents the system model, the state estimation, and the conventional bad data detection (BDD) system. Section 3 mathematically models the FDIA. Section 4 presents the basic principles of the CDBN and formulates a deep learning-based detection scheme. Section 5 performs case studies to evaluate the performance and effectiveness of the developed methodology. The last section draws conclusions and suggests future work.

2. System Model

2.1. State Estimation in Power Systems

Generation, transmission, and distribution are the three main parts of the power system. In a power grid, the control centre must monitor the state of all buses and nodes to make operational decisions as quickly as possible. However, it is impossible to measure all the data directly. On this subject, the control centre estimates the operating conditions of the system by collecting the readings from remote meters.

Let $\mathbf{z} = [z_1, z_2, \ldots, z_m]^T$ be an $m \times 1$ vector of all measurements, including loads and power injections at buses, power flows at transmission lines, and so on. $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ denotes an $n \times 1$ state vector, where $m \gg n$. $\mathbf{e} = [e_1, e_2, \ldots, e_m]^T$, where $e_m \sim \mathcal{N}(0, \sigma_m^2)$ is the measurement error. We have

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

where $h(\cdot)$ shows the nonlinear relationship between the measurement $\mathbf{z}$ and the state $\mathbf{x}$. In a DC power flow model, Equation (1) can be written in the form of a linear matrix:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e} \tag{2}$$

where $\mathbf{H}$ is an $m \times n$ Jacobian matrix, and $e \sim \mathcal{N}(0, \sigma^2)$ is the environmental noise. On this basis, the state vector can be calculated by

$$\mathbf{x} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{z} \tag{3}$$

where

$$\mathbf{W} = \begin{bmatrix} \sigma_1^{-2} & & & \\ & \sigma_2^{-2} & & \\ & & \ddots & \\ & & & \sigma_m^{-2} \end{bmatrix} \tag{4}$$

2.2. Conventional Bad Data Detection

Erroneous data measurements can occur for a variety of reasons (e.g., device misconfiguration and malicious attacks). These measurements could get incorrect state estimates. Therefore, they must be recognized and removed in time. The BDD system can eliminate some random errors. When detecting and identifying the erroneous data, the L2-norm of measurement residual is first calculated. By comparing the calculated result r with a prescribed threshold τ, it reports normal data measurements if

$$\mathrm{r} = \| \mathbf{z} - \mathbf{H}\mathbf{x} \| < \tau \tag{5}$$

holds, or bad ones otherwise.

3. False Data Injection Attack

When an adversary launches the FDIA, he can manipulate the measurement $\mathbf{z}$ to cause an arbitrary change in the estimated value without being detected by the BDD system [40]. Figure 1 presents the process when the state estimation is attacked. Under the condition of the FDIA, an original measurement $\mathbf{z}$ can be replaced by a compromised $\mathbf{z}_a$, where $\mathbf{z}_a = \mathbf{z} + \mathbf{a}$ and $\mathbf{a}$ is an $m \times 1$ malicious data vector. If so, the result of the state estimation then becomes $\mathbf{x}_a$. In general, the BDD system is likely to recognize the random attack vector $\mathbf{a}$. However, in [40], it was found that a few well-designed attack vectors (such as $\mathbf{a} = \mathbf{Hc}$) can bypass the BDD because the injected false data do not affect the residue:

$$\mathbf{z}_a - \mathbf{H}\mathbf{x}_a = \mathbf{z} + \mathbf{a} - \mathbf{H}(\mathbf{x} + \mathbf{c}) = \mathbf{z} - \mathbf{H}\mathbf{x} \tag{6}$$

where

$$\begin{aligned} \mathbf{x}_a &= \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{z}\mathbf{a} \\ &= \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}(\mathbf{z} + \mathbf{a}) \\ &= \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{z} + \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{a} \\ &= \mathbf{x} + \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{H}\mathbf{c} \\ &= \mathbf{x} + \mathbf{c} \end{aligned} \tag{7}$$

and $\mathbf{c} = [c_1,\ c_2,\ \ldots, c_n]^{\mathrm{T}}$ is an arbitrary $n \times 1$ vector. Therefore, the attack is stealthy and can inject any malicious data into the state estimation.

However, adversaries can usually only compromise a limited number of measurements, so two main realistic attack scenarios are considered as follows:

1. Least-effort attack [19]: $k = 1$, adversaries manipulate the minimum number of measurements to launch the FDIA;
2. Multiple attacks [40]: $k > 1$, adversaries can compromise up to k measurements to launch the FDIA;

where k is the number of attacked measurements. However, the FDIAs are not constrained by these two scenarios. In the IEEE 14-bus test system, Figure 2 shows the difference in the economic dispatch of the power system before and after the measurement $\mathbf{z}$ is attacked. We can see that the total generation and the production cost are higher than those of the original case. Furthermore, as the attack intensity increases, the difference increases accordingly. We find that the FDIA can leave the system out of control and even cause security risks. Our developed scheme can specifically detect this kind of attack.

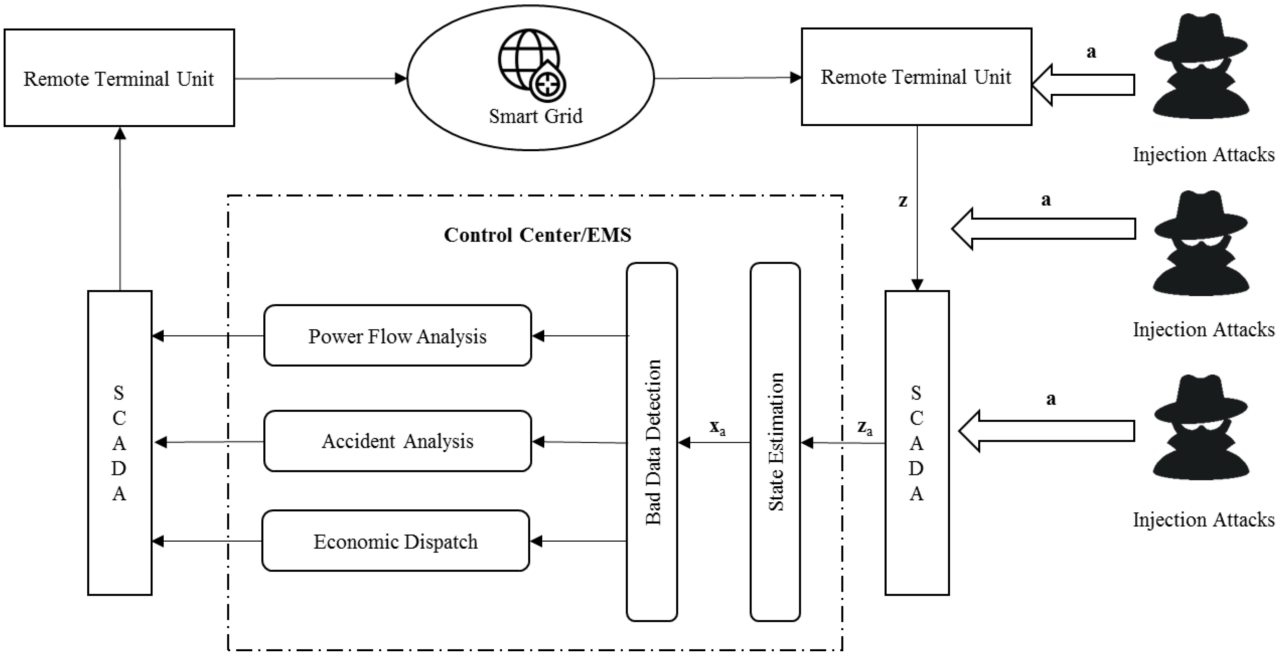

Figure 1. The state estimation under the attack.

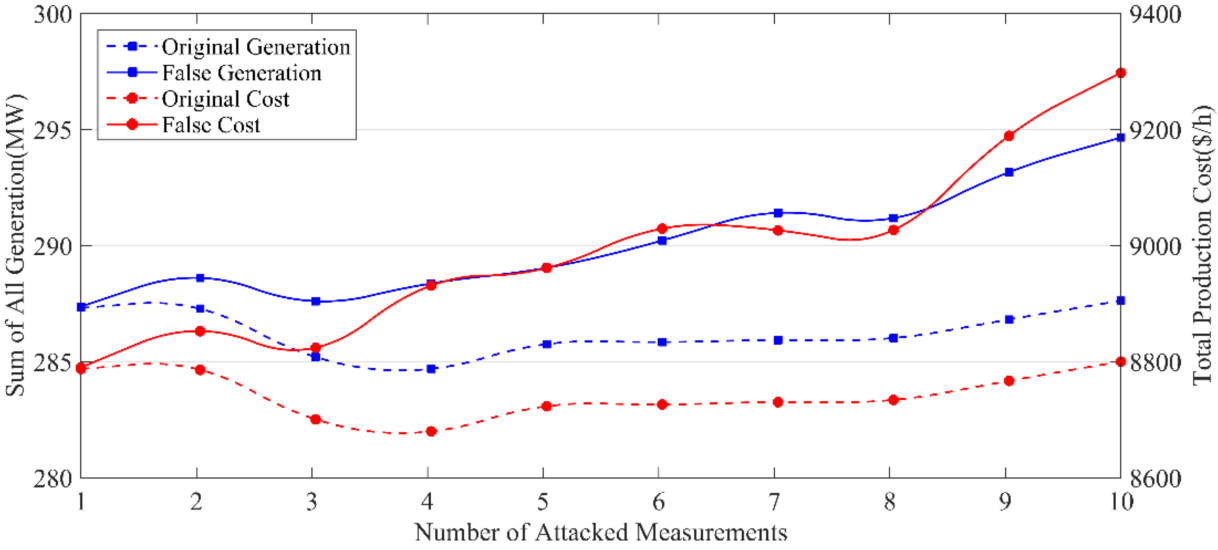

Figure 2. The immediate damaging effect to the power system.

4. Deep Learning-Based Identification Scheme

In order to detect the FDIA, a deep learning-based identification scheme is developed. We propose a CDBN by combining a conventional DBN with a CGBRBM, which can process real-valued data and consider the impact of previous measurements on current detection results. Figure 3 shows the framework of the CDBN. We employ a CGBRBM and stack $K-1$ standard RBMs on top, where K is the number of hidden layers. To indicate whether the measurements are attacked by the FDIA, a BP output unit is added at the end of the scheme to make it a binary classifier.

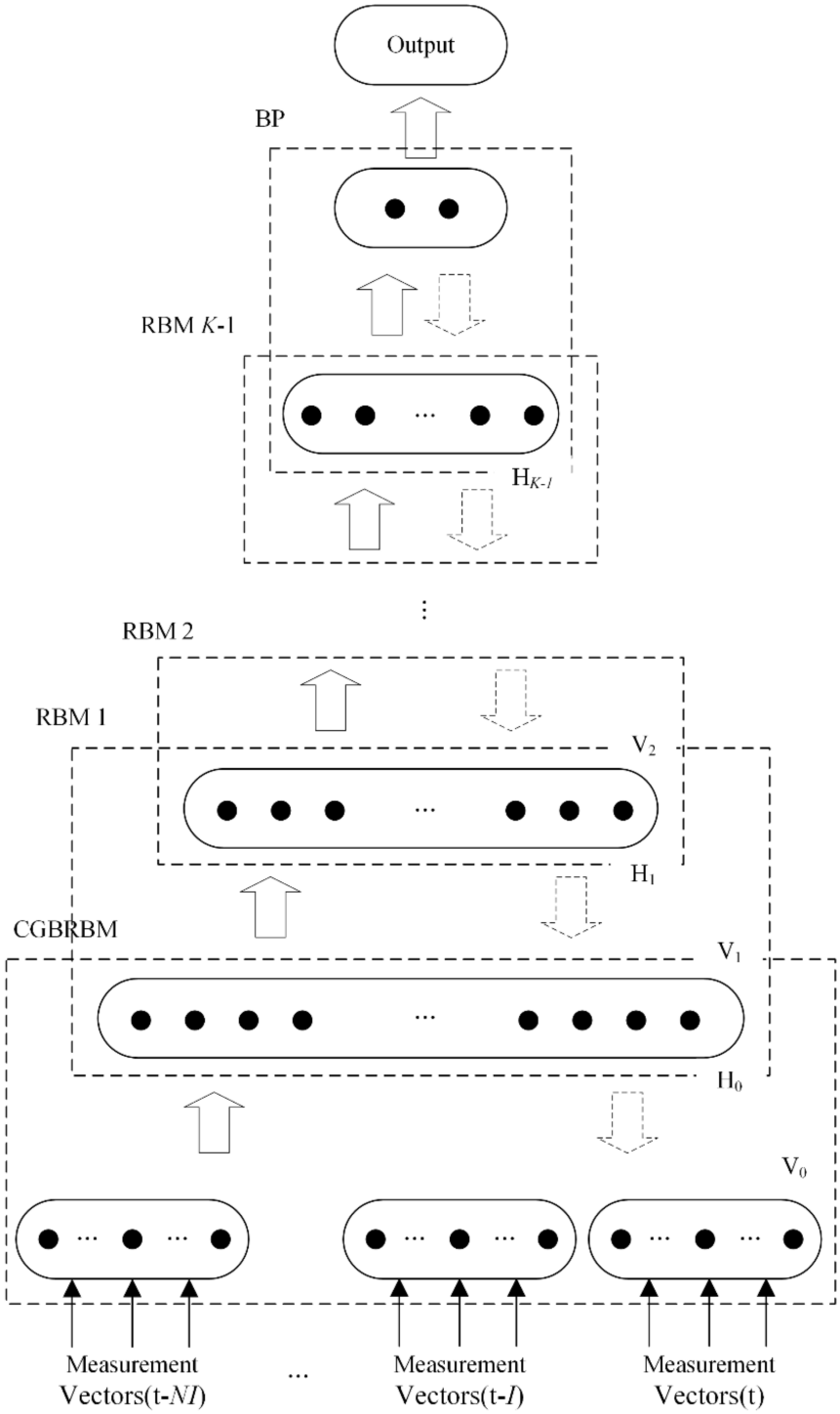

Figure 3. The structure of the CDBN.

4.1. Conventional RBM

The RBM is a two-layer neural network, which is the core of the CDBN. As Figure 4 shows, its two layers are the visible layer and the hidden layer. The units between adjacent layers are connected, but there is no connection inside each layer. The visible layer corresponds to the measurement, and the hidden layer can represent feature extraction.

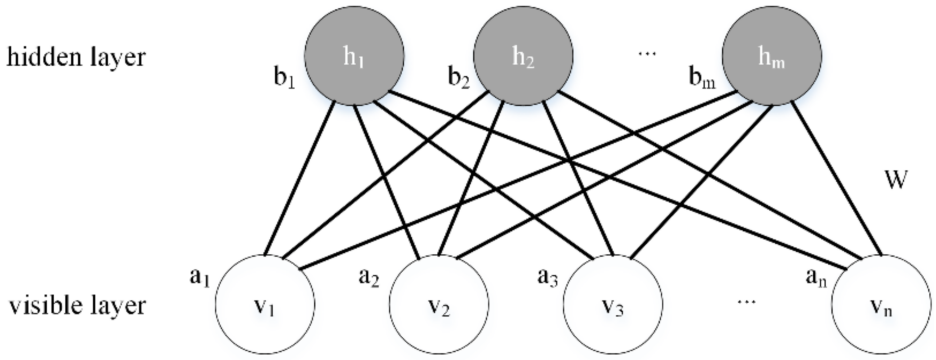

Figure 4. The structure of the RBM.

The RBM is an energy-based undirected generation model, and its system energy is

$$\mathrm{E}(\mathbf{v},\ \mathbf{h}) = \sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j - \sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j \tag{8}$$

where v_i and h_j are the state of visible unit i and hidden unit j, w_{ij} is the weight between them, a_i and b_j index the standard biases, n and m are the numbers of visible and hidden units, respectively. According to the property of the RBM, given the state of the visible layer, the activation probability of the jth hidden unit is:

$$P\left(h_j = 1 \mid \mathbf{v}\right) = sigm\left(\sum_{i=1}^{n} w_{ij} v_i + b_j\right) \tag{9}$$

Similarly, given the state of the hidden layer, the activation probability of the ith visible unit is:

$$P\left(v_i = 1 \mid \mathbf{h}\right) = sigm\left(\sum_{j=1}^{m} w_{ij} h_j + a_i\right) \tag{10}$$

where $sigm(x) = 1/(1 + \exp(\mathrm{x}))$.

The goal of the RBM training is to obtain the parameters to maximize the likelihood function by gradient descent. By calculating the derivative of the log-likelihood, the weights and the biases can be updated as follows:

$$\begin{aligned} w_{ij} &= w_{ij} + \varepsilon\left(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{model}\right) \\ a_i &= a_i + \varepsilon\left(\langle v_i\rangle_{data} - \langle v_i\rangle_{model}\right) \\ b_j &= b_j + \varepsilon\left(\langle h_j\rangle_{data} - \langle h_j\rangle_{model}\right) \end{aligned} \tag{11}$$

where ε is the learning rate, $\langle\cdot\rangle_{data}$ and $\langle\cdot\rangle_{model}$ are the expectations calculated from the data and model distributions, respectively. $\langle\cdot\rangle_{data}$ is easily obtained by Equations (9) and (10). However, getting $\langle\cdot\rangle_{model}$ is much more difficult. To simplify the process, Hinton proposed an efficient and straightforward contrast divergence (CD) algorithm based on Gibbs sampling [41].

4.2. Conditional Gaussian-Bernoulli RBM

In the standard type of the RBM, input data are binary and static, but the measurements in the power system are usually real-valued and time-series data. To address this limitation, we adopt a conditional Gaussian–Bernoulli RBM(CGBRBM) as the basis for the detection algorithm.

It can be seen from Figure 5 that the CGBRBM is a variant of the conventional RBM. First, the input units are linear with Gaussian noise, whereas the hidden units are still

binary. The second improvement is that the time-series data can be modelled by considering the visible variables in previous time steps. The energy function of the CGBRBM is:

$$\mathrm{E}(\mathbf{v}_t, \ldots, \mathbf{v}_{t-NI}, \mathbf{h}) = \sum_{i=1}^{n} \frac{(v_{i,t} - a_{i,t})^2}{2\sigma_i^2} - \sum_{j=1}^{m} b_{j,t} h_j - \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{v_{i,t}}{\sigma_i} w_{ij} h_j \tag{12}$$

where $v_{i,t}$ is the ith real-valued visible element at time step t, h_j is the state of hidden unit j, w_{ij} expresses the weight between $v_{i,t}$ and h_j, σ_i is the standard deviation of the ith visible element, N is the size of the observation window at the previous time, I represents the time interval between two adjacent time steps, n and m are the numbers of visible and hidden units, respectively. $a_t = a + \sum_{k=1}^{N} v_{t-kI} A_k$ and $b_t = b + \sum_{k=1}^{N} v_{t-kI} B_k$ represent the dynamic biases from the past to the visible bias vector **a** and the hidden bias vector b, where $k = 1, \ldots, N$, v_{t-kI} is the kth previous visible vector, A_k and B_k are the weight matrices of the kth previous visible vector to the current visible unit and the hidden unit, respectively. According to Equation (12), the corresponding activation probabilities become

$$P\left(h_j = 1 \mid \mathbf{v}_t, \ldots, \mathbf{v}_{t-NI}\right) = sigm\left(\sum_{i=1}^{n} w_{ij} \frac{v_{i,t}}{\sigma_i} + b_{j,t}\right) \tag{13}$$

$$P\left(v_{i,t} = 1 \mid \mathbf{h}\right) = \mathcal{N}\left(\sum_{j=1}^{m} w_{ij} h_j + a_{i,t}, \sigma_i^2\right) \tag{14}$$

where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian with mean μ and variance σ^2. In practice, when ${\sigma_i}^2$ is fixed to 1, it can make the learning work better [37]. So, in this case, similar to the conventional RBM, by using the CD algorithm, we can update the weights and the biases as follows:

$$\begin{gathered} w_{ij} = w_{ij} + \varepsilon\left(\langle v_{i,t} h_j \rangle_{data} - \langle v_{i,t} h_j \rangle_{model}\right) \\ a_{ijk} = a_{ijk} + \varepsilon\left(\langle v_{i,t-kI} v_{j,t} \rangle_{data} - \langle v_{i,t-kI} v_{j,t} \rangle_{model}\right) \\ b_{ijk} = b_{ijk} + \varepsilon\left(\langle v_{i,t-kI} h_j \rangle_{data} - \langle v_{i,t-kI} h_j \rangle_{model}\right) \\ a_{i,t} = a_{i,t} + \varepsilon\left(\langle v_{i,t} \rangle_{data} - \langle v_{i,t} \rangle_{model}\right) \\ b_j = b_j + \varepsilon\left(\langle h_j \rangle_{data} - \langle h_j \rangle_{model}\right) \end{gathered} \tag{15}$$

where a_{ijk} and b_{ijk} are the elements of A_k and B_k.

4.3. CDBN

The CDBN is a probability generation model. It is a deep learning classifier composed of the CGBRBM, the RBM, and the BP [42]. As Figure 3 shows, the data are first input into the CGBRBM at the bottom for training and feature extraction. Then, the extracted features are used as the input values of another RBM. In this way, more RBM layers can be stacked [28]. The training process of the CDBN model consists of two steps [30]: layer-wise unsupervised learning and fine-turning.

The first step is an unsupervised learning process. By using the CD algorithm, the RBM of each layer is trained layer-by-layer. Finally, we get the CDBN with a few layers, the parameters of which are suitable for extracting the characteristics of this type of data [31].

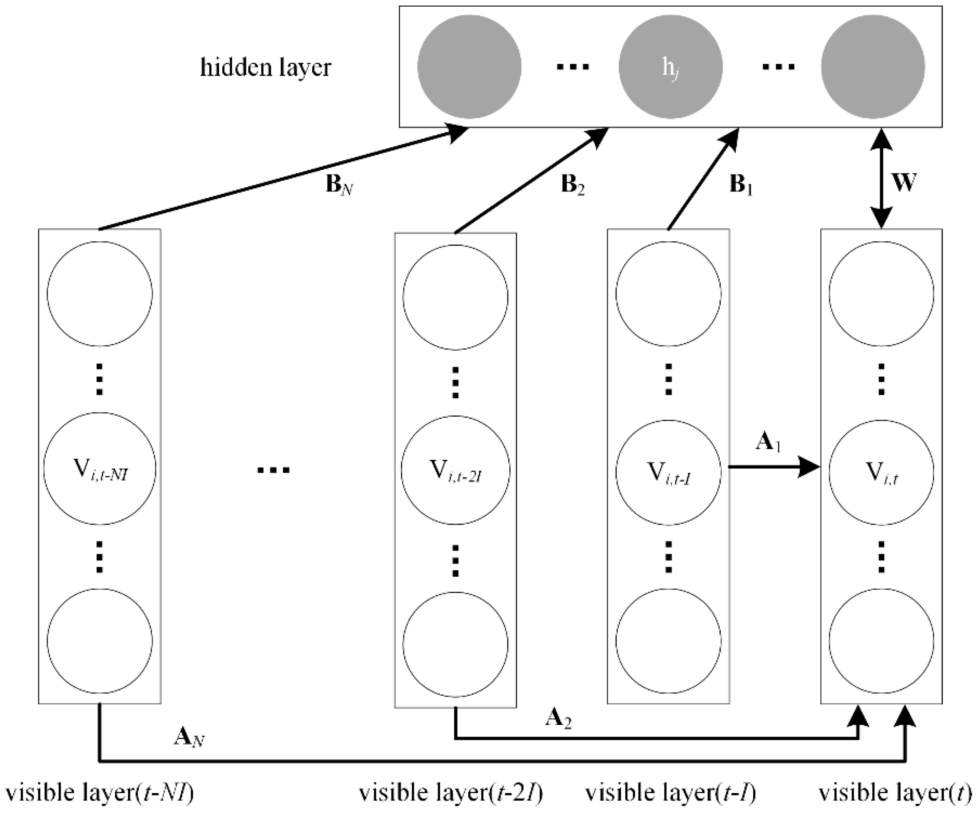

Figure 5. The structure of the CGBRBM.

In order to optimize the parameters mapped to each layer, the whole CDBN model should be fine-tuned. This process uses the labelled data and the BP network for top-down supervised learning. The binary output node can be calculated by Equation (9), and it can be utilized to represent the compromised label and the normal one. In the calculation of the kth hidden layer, the weights and the biases are updated in the following:

$$\begin{cases} \Delta W_{k,i,j} = -\eta \delta_{k,j} p_{k-1,i} \\ \Delta b_{k,j} = -\eta \delta_{k,j} \end{cases} \tag{16}$$

where η is the learning rate, $p_{k-1,i}$ is the ith activation probability of the $(k-1)$th hidden layer, and

$$\delta_{k,j} = p_{k,j}\left(1 - p_{k-1,j}\right) \sum_{h}^{H} \delta_{k+1,h} W_{k+1,j,h} \tag{17}$$

where $p_{k,j}$ is the jth activation probability of the kth hidden layer, $W_{k+1,j,h}$ is the jhth element of the $(k+1)$th layer weight matrix, H is the number of elements. Correspondingly, for the output layer, the updated values of the weights and the biases are as follows:

$$\begin{cases} \Delta W_{i,o} = -\eta \delta_o p_{K,i} \\ \Delta b_o = -\eta \delta_o \end{cases} \tag{18}$$

where $p_{K,i}$ is the ith activation probability of the last RBM layer, and

$$\delta_o = p_o(1 - p_o)(l_o - L), \tag{19}$$

where p_o is the activation probability of the output layer, l_o and L represent the predicted value and the actual one, respectively.

As shown in Figure 6, the detection process of our scheme can be mainly divided into three steps: data preprocessing stage, training stage, and testing stage. The first stage is to

obtain the measurement vector **z** and inject the attack vector **a** into it according to a certain proportion. After the normalization process, some sample data are selected as the training set and others as the test set. Next, by completing layer-wise unsupervised learning and fine-turning, the model is trained in the second stage. Finally, the trained model is used to predict whether the sample data in the test set is under attack. By comparing with the actual value, the accuracy of our developed scheme can be evaluated.

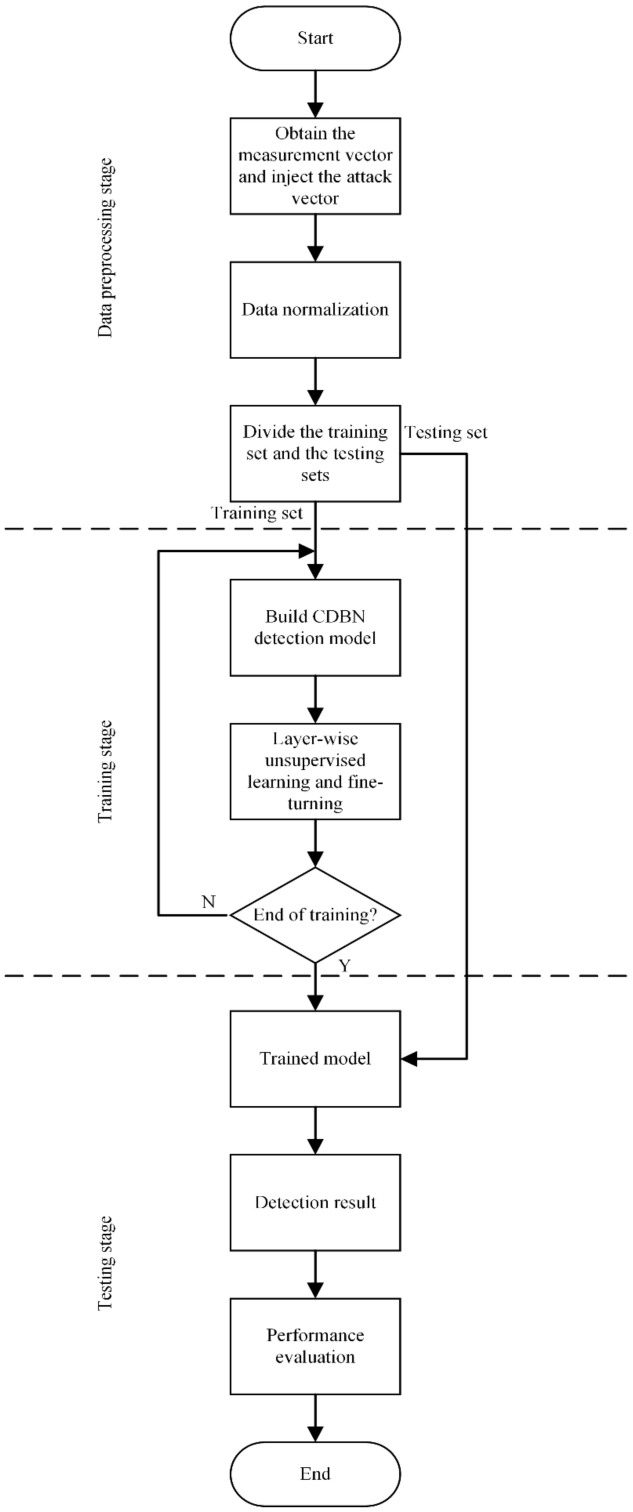

Figure 6. The CDBN-based detection model flow chart.

5. Simulation

In this simulation, the performance of our developed scheme is evaluated in the IEEE 14-bus test system. All the data used in the simulation, including the vector of measurements and the Jacobian matrix H, are based on the MATPOWER 7.1. MATPOWER is an open-source Matlab (R2017b, MathWorks, Natick, Massachusetts, USA) power system simulation package, which has been widely used in research and education for solving power flow and optimal power flow problems. Included are numerous example power flow and optimal power flow cases. It can simulate most power system scenarios, and the generated data are consistent with the actual situation, which can satisfy the verification of algorithm performance.

In the IEEE 14-bus test system, by changing the active and reactive power of the load, we first use MATPOWER to complete the power flow calculation for 30,000 consecutive moments. Then, some values (including the branch power flow, the active and reactive power of the generator, and the node voltage, a total of 39 values) are selected from the calculation results of each power flow, and Gaussian noise (such as $(0,\ 0.25)$) is injected into them. Finally, the calculation result is regarded as the measurement of state estimation. There are 30,000 measurements in total, and the number of elements in each measurement is 39. Next, according to the method in [40], the FDIA is launched randomly on 15,000 measurements. The measurement residual after the attack is guaranteed to be less than the prescribed threshold τ, so as to avoid bad data detection. These 30,000 measurements are divided into three parts on average, which are used as the training set, the verification set, and the test set, respectively. For the above two scenarios (least-effort attack and multiple attacks), we consider the following three aspects to evaluate the performance of the mechanism. Each value of the simulation is an average among 30 independent trials.

5.1. Experimental Results

5.1.1. Structural Design

I. Effect of the height and width of the CDBN

We first study the effect of the number of hidden layers and the number of units per layer on the performance of our developed scheme. In this simulation, the number of attacked measurements k is set to 1, the size of the observation window *(N)* is 1, the time interval *(I)* is 2, and the number of hidden layers is changed from 2 to 5, the hidden layer units range from 20 to 60. From Figure 7, when there are three hidden layers and the number of units in each layer is 30, we can see that the accuracy can be up to 97.3%.

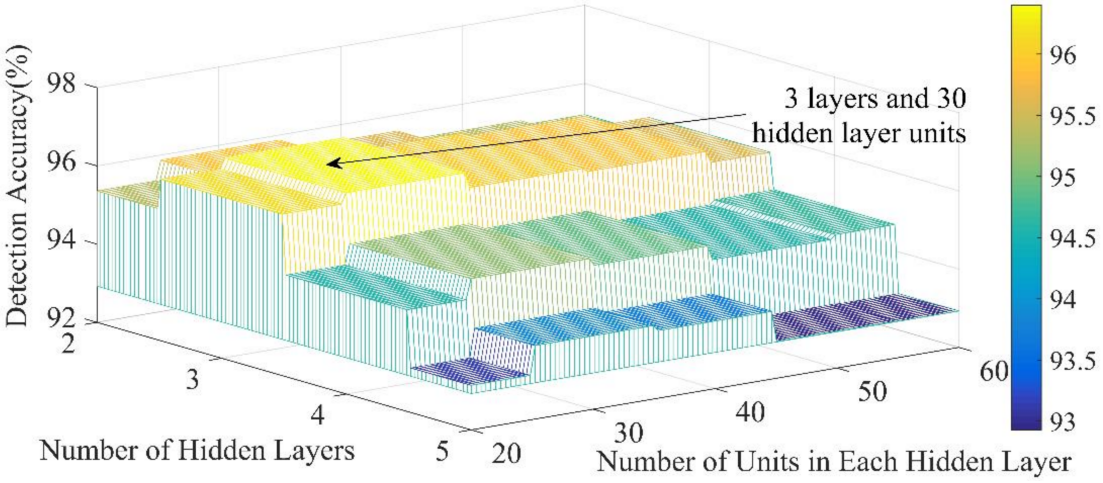

Figure 7. Accuracy of different hidden layers and different hidden layer units.

II. Effect of the Observation Window Structure

Next, we consider the effect of the size of the observation window at the previous time (N) and the time interval (I) between two adjacent time steps on the effectiveness of our scheme, where N and I are defined before. According to the conclusion of the previous section, we build a CDBN structure with three hidden layers and 30 units in each layer. The range of N is set from 1 to 4, and I is increased from 1 to 5. We can see the simulation results in Figure 8. Considering the accuracy and the availability, $N = 1$ and $I = 2$ represent a reasonable choice for detecting the FDIA.

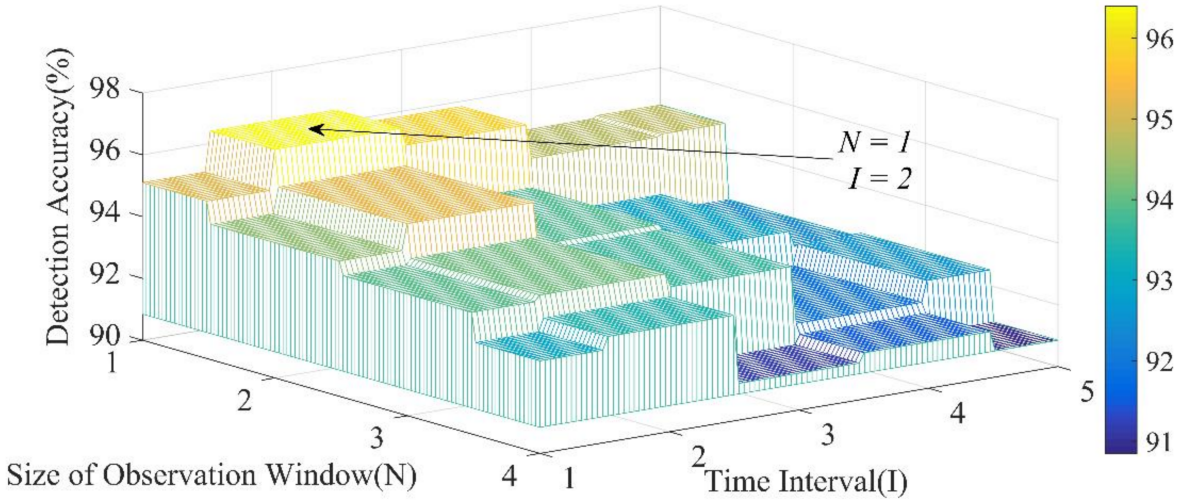

Figure 8. Accuracy of different N and I.

5.1.2. Multi-Scenario Validation

In this experiment, we discuss the accuracy of our developed scheme in the least-effort attack ($k = 1$) and multiple attacks ($k > 1$), respectively. According to Section 5.1.1, we simulate a 3-layer CDBN model with 30 units per layer, set N to 1, and I to 2. Besides, by using the same data set, we compare the performance of our method with the ANN and the SVM, where the ANN consists of a hidden layer with 30 units and the radial basis function (RBF) kernel is used in the SVM. Figure 9 shows the detection results. Specifically, when $k = 1, 4, 7, 10$, the receiver operating characteristics (ROC) curves of the method are shown in Figure 10 [43]. ROC is one of the essential metrics for evaluating the performance of a classification model.

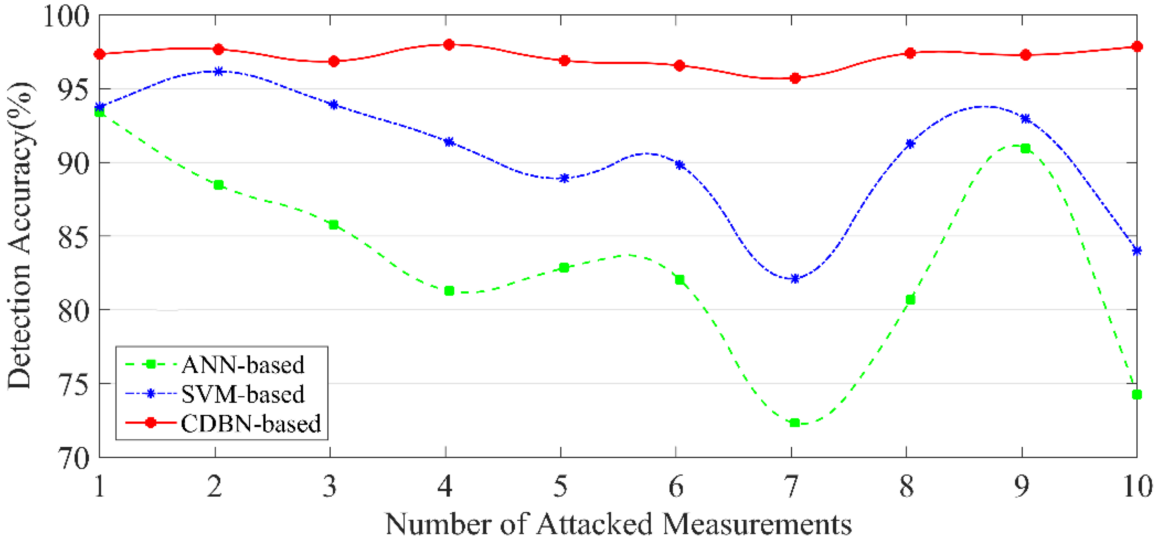

Figure 9. Accuracy of different number of attacked measurements.

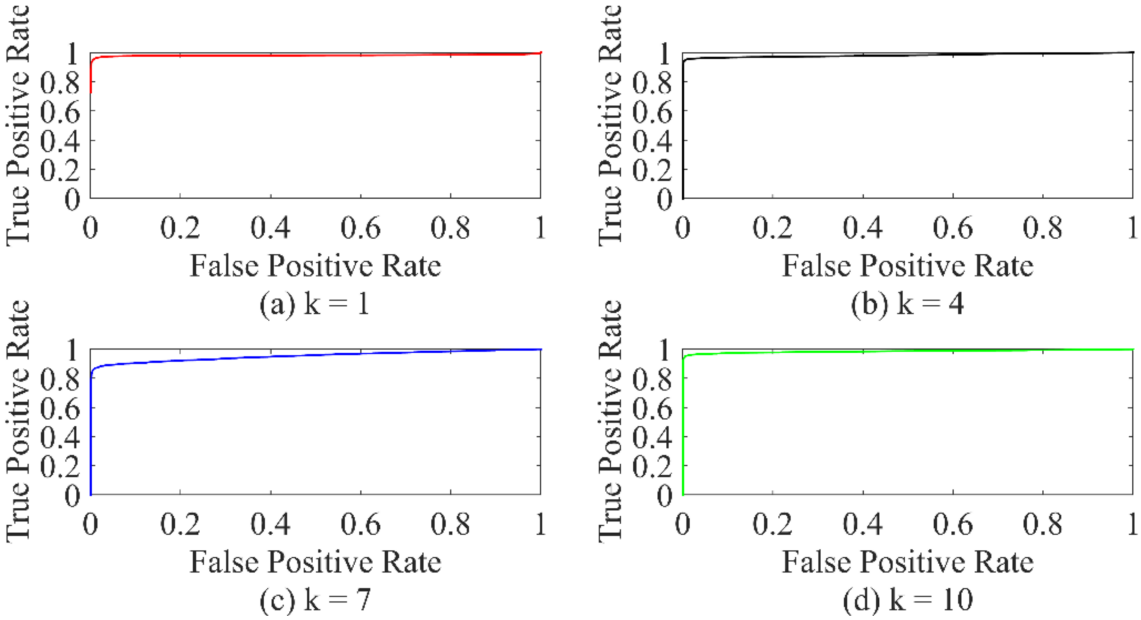

Figure 10. ROC curve when k = 1,4,7,10.

5.1.3. Robustness Validation

In the previous experiment, we set $\mathcal{N}(0,\ 0.25)$ as the environmental noise. It means that we use a Gaussian with mean 0 and variance 0.25 as the environmental noise. However, the real environment may be much worse. To evaluate the robustness, in this part, we fix the number of attacked measurements (k) to 4, and the standard deviation σ of environmental noise ~$\mathcal{N}(0,\ \sigma)$ changes from 0.25 to 2.5. The settings of the other structural parameters are the same as Section 5.1.2. Figure 11 compares the accuracy obtained from the ANN, the SVM, and our developed scheme.

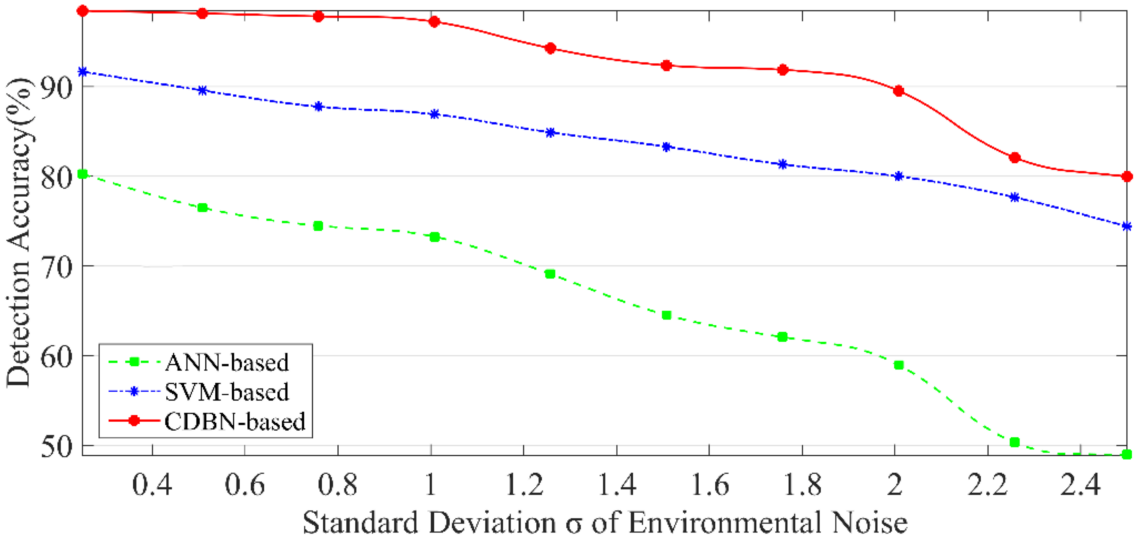

Figure 11. Accuracy of different standard deviation σ.

5.2. Analysis of Results

In the verification of the CDBN structure, the number of hidden layers, the number of units per layer, the size of the observation window (N), and the time interval (I) are four important parameters. The function of depth is to abstract layer by layer and extract features continuously, while the function of width is to allow each layer of RBM to learn

more features. Generally speaking, the performance of an algorithm is more sensitive to depth, and an appropriate width is easier to improve performance. Setting too few or too many layers and hidden units may cause under-fitting and over-fitting, which will decrease the accuracy [44]. If N is larger or smaller than what is required, the observation window cannot adequately reflect the recent changes in the measurements. Similarly, a larger I tends to smooth out or even ignore some short-term but critical fluctuations, whereas a smaller I may cause this change to be too dramatic and lose its reference value [45]. So, by choosing the appropriate parameters, the accuracy of the mechanism can be significantly improved. According to the experimental results, we simulate a three-layer CDBN model with 30 units per layer, set N to 1, and I to 2.

In the multi-scenario validation, we can find that the accuracy of our CDBN-based method is higher than the other two. Moreover, with the increase of k, the detection performance is stable, and the accuracy can reach up to 98.4%. The area under curve (AUC) is close to 1. The developed scheme not only considers the time correlation of the measurements, but the structure of deep learning also makes the feature extraction more accurate. It can be inferred that the CDBN model has good performance and can accurately identify FDIA.

In the robustness validation, as the noise level increases, the accuracy of the three methods decreases. It is understandable. Because the higher the noise level, the harder it is to distinguish between normal and compromised measurements. However, the accuracy of the developed method is always the highest of the three. It can be concluded that the CDBN-model can deal with more severe situations and is more suitable for FDIA detection in actual power scenarios. Especially when $\sigma < 2.0$, the accuracy can be more than 90%. That is to say, when the difference caused by the environmental noise is smaller than that caused by the FDIA, our CDBN-based method is competent and has good robustness.

Although the detection accuracy is high, there are still some FDIAs undetected. There are three main reasons for this:

1. The choice of the parameters
2. The presence of environmental noise
3. Insufficient data

In conclusion, our developed scheme has the advantages of high detection accuracy, stable performance, and good robustness. It has great practical value in FDIA detection.

6. Conclusions

This article presents an in-depth study of the state estimation, analyzes the basic principles of the FDIA, and focuses on the detection of power system cyber-attacks. By integrating the DBN structure with the CGBRBM, which can process time-series real-valued measurement data, we introduce a deep learning-based scheme to recognize the potential FDIA for maintaining the stability of the smart grid. It can extract the high-dimensional temporal behaviour features from the input data to construct a classification model and perform detection. In the simulation, we first optimize the model parameters suitable for the FDIA detection. By simulating two realistic attack scenarios, according to the determined optimal parameters, the performance is then demonstrated. The results indicate that our scheme can efficiently detect the FDIA and achieve better accuracy and robustness than the ANN and the SVM. In our future work, more sophisticated attack scenarios will be investigated based on the developed mechanism. Additionally, to be more widely used in the field of the FDIA detection, we will explore our scheme in the AC power system model.

Author Contributions: Conceptualization, T.P. and D.Z.; methodology, Y.D.; software, Y.D.; validation, R.L. and X.W.; formal analysis, D.Z.; investigation, K.M.; resources, T.P.; data curation, Y.D.; writing—original draft preparation, Y.D.; writing—review and editing, K.M.; visualization, R.L.; supervision, T.P.; project administration, T.P.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Project Research on Forecasting Method of Smart Grid Big Data Based on Random Projection Neural Networks supported by National Natural Science Foundation of China, grant number 61703379.

Data Availability Statement: The model and data used to support the results of this study are available from the corresponding author upon request.

Acknowledgments: The authors would like to thank all of the editors and anonymous reviewers for their careful reading and insightful remarks.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

ANN	Artificial neural network
AUC	Area under curve
BDD	Bad data detection
CDBN	Conditional deep belief network
CGBRBM	Conditional Gaussian-Bernoulli RBM
CD	Contrast divergence
DOID	Distributed observable island detection
DTAD	Distributed time approaching detection
DBN	Deep belief network
FDIA	False data injection attack
RBM	Restricted boltzmann machine
ROC	Receiver operating characteristics
SVM	Support vector machine
SCADA	Supervisory control and data acquisition system
a	Attack vector
a_i, b_j	Standard biases
a_t	Dynamic biases from the past to the visible bias vector
A_k	Weight matrices of the kth previous visible vector to the current visible unit
a_{ijk}	Elements of A_k
B_k	Weight matrices of the kth previous visible vector to the current hidden unit
b_{ijk}	Elements of B_k
b_t	Dynamic biases from the past to the hidden bias vector
c	Arbitrary vector added to the state variable
$\mathbf{e}$	Measurement error vector
$\mathbf{H}$	Jacobian matrix
H	Number of elements in each layer of CDBN
h_j	State of hidden unit j
$h(\cdot)$	Nonlinear relationship between the measurement $\mathbf{z}$ and the state $\mathbf{x}$
I	Time interval between two adjacent time steps
k	Number of attacked measurements
l_o	Predicted value
L	Actual value
N	Size of the observation window at the previous time
$\mathcal{N}(\mu,\ \sigma^2)$	Gaussian with mean μ and variance σ^2
n, m	Numbers of visible and hidden units
$p_{k-1,i}$	ith activation probability of the $(k-1)$th hidden layer
$p_{k,j}$	jth activation probability of the kth hidden layer
p_o	Activation probability of the output layer
v_i	State of visible unit i
$v_{i,t}$	ith real-valued visible element at time step t
w_{ij}	Weight between unit i and unit j
$W_{k+1,j,h}$	jhth element of the $(k+1)$th layer weight matrix
$\mathbf{x}$	State vector

$\mathbf{x}_a$	Compromised state vector
$\mathbf{z}_a$	Compromised vector of all measurements
$\mathbf{z}$	Vector of all measurements
τ	Threshold of BDD system
ε, η	Learning rate
$\langle \cdot \rangle_{data}$	Expectations calculated from the data
$\langle \cdot \rangle_{model}$	Expectations calculated from the model distributions
σ_i	Standard deviation of the ith visible element

References

1. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart grid—The new and improved power grid: A survey. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 944–980. [CrossRef]
2. Deng, R.; Yang, Z.; Chow, M.-Y.; Chen, J. A survey on demand response in smart grids: Mathematical models and approaches. *IEEE Trans. Ind. Inform.* **2015**, *11*, 570–582. [CrossRef]
3. Wu, F.F. Power system state estimation: A survey. *Int. J. Electr. Power Energy Syst.* **1990**, *12*, 80–87. [CrossRef]
4. Monticelli, A. Electric power system state estimation. *Proc. IEEE* **2000**, *88*, 262–282. [CrossRef]
5. Abur, A.; Exposito, A.G. *Power System State Estimation-Theory and Implementation*; Marcel Dekker Inc.: New York, NY, USA, 2004.
6. Mylonas, E.; Tzanis, N.; Birbas, M.; Birbas, A. An Automatic Design Framework for Real-Time Power System Simulators Supporting Smart Grid Applications. *Electronics* **2020**, *9*, 299. [CrossRef]
7. Deng, R.; Yang, Z.; Chen, J.; Asr, N.R.; Chow, M.-Y. Residential Energy Consumption Scheduling: A Coupled-Constraint Game Approach. *IEEE Trans. Smart Grid* **2014**, *5*, 1340–1350. [CrossRef]
8. Zhao, C.; He, J.; Cheng, P.; Chen, J. Consensus-Based Energy Management in Smart Grid With Transmission Losses and Directed Communication. *IEEE Trans. Smart Grid* **2017**, *8*, 2049–2061. [CrossRef]
9. Wadhawan, Y.; Almajali, A.; Neuman, C. A Comprehensive Analysis of Smart Grid Systems against Cyber-Physical Attacks. *Electronics* **2018**, *7*, 249. [CrossRef]
10. Sorebo, G.N.; Echols, M.C. *Smart Grid Security: An End-to-End View of Security in the New Electrical Grid*; CRC Press: Boca Raton, FL, USA, 2016; Volume 7.
11. Wood, A.J.; Wollenberg, B.F.; Sheblé, G.B. *Power Generation, Operation, and Control*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2013; Volume 7.
12. Sridhar, S.; Hahn, A.; Govindarasu, M. Cyber–Physical System Security for the Electric Power Grid. *Proc. IEEE* **2012**, *100*, 210–224. [CrossRef]
13. Teixeira, A.; Amin, S.; Sandberg, H.; Johansson, K.H.; Sastry, S.S. Cyber security analysis of state estimators in electric power systems. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 5991–5998.
14. Soe, Y.N.; Feng, Y.; Santosa, P.I.; Hartanto, R.; Sakurai, K. Towards a Lightweight Detection System for Cyber Attacks in the IoT Environment Using Corresponding Features. *Electronics* **2020**, *9*, 144. [CrossRef]
15. Chen, W.; Ding, D.; Dong, H.; Wei, G. Distributed Resilient Filtering for Power Systems Subject to Denial-of-Service Attacks. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1688–1697. [CrossRef]
16. Hosseinzadeh, M.; Sinopoli, B.; Garone, E. Feasibility and Detection of Replay Attack in Networked Constrained Cyber-Physical Systems. In Proceedings of the 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 24–27 September 2019; pp. 712–717.
17. Bobba, R.B.; Rogers, K.M.; Wang, Q.; Khurana, H.; Nahrstedt, K.; Overbye, T.J. Detecting false data injection attacks on dc state estimation. In Proceedings of the Preprints of the First Workshop on Secure Control Systems, Stockholm, Sweden, 12 April 2010.
18. Pasqualetti, F.; Dörfler, F.; Bullo, F. Attack Detection and Identification in Cyber-Physical Systems. *IEEE Trans. Autom. Control.* **2013**, *58*, 2715–2729. [CrossRef]
19. Yang, Q.; Yang, J.; Yu, W.; An, D.; Zhang, N.; Zhao, W. On False Data-Injection Attacks against Power System State Estimation: Modeling and Countermeasures. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 717–729. [CrossRef]
20. Liu, L.; Esmalifalak, M.; Ding, Q.; Emesih, V.A.; Han, Z. Detecting False Data Injection Attacks on Power Grid by Sparse Optimization. *IEEE Trans. Smart Grid* **2014**, *5*, 612–621. [CrossRef]
21. Liu, Y.; Yan, L.; Ren, J.-W.; Su, D. Research on Efficient Detection Methods for False Data Injection in Smart Grid. In Proceedings of the 2014 International Conference on Wireless Communication and Sensor Network, Wuhan, China, 13–14 December 2014; pp. 188–192.
22. Hu, Z.; Wang, Y.; Tian, X.; Yang, X.; Meng, D.; Fan, R. False data injection attacks identification for smart grids. In Proceedings of the 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), Beirut, Lebanon, 29 April–1 May 2015; pp. 139–143.
23. Esmalifalak, M.; Liu, L.; Nguyen, N.; Zheng, R.; Han, Z. Detecting Stealthy False Data Injection Using Machine Learning in Smart Grid. *IEEE Syst. J.* **2017**, *11*, 1644–1652. [CrossRef]
24. Mohammadpourfard, M.; Sami, A.; Seifi, A.R. A statistical unsupervised method against false data injection attacks: A visualization-based approach. *Expert Syst. Appl.* **2017**, *84*, 242–261. [CrossRef]

25. Tabakhpour, A.; Abdelaziz, M.M.A. Neural Network Model for False Data Detection in Power System State Estimation. In Proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 5–8 May 2019; pp. 1–5.
26. Liagkou, V.; Kavvadas, V.; Chronopoulos, S.K.; Tafiadis, D.; Christofilakis, V.; Peppas, K.P. Attack Detection for Healthcare Monitoring Systems Using Mechanical Learning in Virtual Private Networks over Optical Transport Layer Architecture. *Computation* **2019**, *7*, 24. [CrossRef]
27. Yu, J.J.Q.; Hou, Y.; Li, V.O.K. Online False Data Injection Attack Detection With Wavelet Transform and Deep Neural Networks. *IEEE Trans. Ind. Informatics* **2018**, *14*, 3271–3280. [CrossRef]
28. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
29. Kaabi, R.; Bouchouicha, M.; Mouelhi, A.; Sayadi, M.; Moreau, E. An Efficient Smoke Detection Algorithm Based on Deep Belief Network Classifier Using Energy and Intensity Features. *Electronics* **2020**, *9*, 1390. [CrossRef]
30. Fischer, A.; Igel, C. An introduction to restricted Boltzmann machines. In Proceedings of the Iberoamerican Congress on Pattern Recognition, Buenos Aires, Argentina, 3–6 September 2012; pp. 14–36.
31. Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. Greedy layer-wise training of deep networks. In Proceedings of the Advances in neural information processing systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 153–160.
32. Aldwairi, T.; Perera, D.; Novotny, M.A. Measuring the Impact of Accurate Feature Selection on the Performance of RBM in Comparison to State of the Art Machine Learning Algorithms. *Electronics* **2020**, *9*, 1167. [CrossRef]
33. Lee, H.; Grosse, R.; Ranganath, R.; Ng, A.Y. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Commun. ACM* **2011**, *54*, 95–103. [CrossRef]
34. Siddiqui, S.; Nesbitt, R.; Shakir, M.Z.; Khan, A.A.; Khan, A.A.; Khan, K.K.; Ramzan, N. Artificial Neural Network (ANN) Enabled Internet of Things (IoT) Architecture for Music Therapy. *Electronics* **2020**, *9*, 2019. [CrossRef]
35. Chaeikar, S.S.; Manaf, A.A.; Alarood, A.A.; Zamani, M. PFW: Polygonal Fuzzy Weighted—An SVM Kernel for the Classification of Overlapping Data Groups. *Electronics* **2020**, *9*, 615. [CrossRef]
36. Chen, X.-W.; Lin, X. Big Data Deep Learning: Challenges and Perspectives. *IEEE Access* **2014**, *2*, 514–525. [CrossRef]
37. Taylor, G.W.; Hinton, G.E.; Roweis, S.T. Modeling human motion using binary latent variables. In Proceedings of the Advances in neural information processing systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 1345–1352.
38. Wan, R.; Mei, S.; Wang, J.; Liu, M.; Yang, F. Multivariate Temporal Convolutional Network: A Deep Neural Networks Approach for Multivariate Time Series Forecasting. *Electronics* **2019**, *8*, 876. [CrossRef]
39. Wei, J.; Mendis, G.J. A deep learning-based cyber-physical strategy to mitigate false data injection attack in smart grids. In Proceedings of the 2016 Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG), Vienna, Austria, 12 April 2016; pp. 1–6.
40. Liu, Y.; Ning, P.; Reiter, M.K. False data injection attacks against state estimation in electric power grids. *ACM Trans. Inf. Syst. Secur.* **2011**, *14*, 1–33. [CrossRef]
41. Hinton, G.E. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Comput.* **2002**, *14*, 1771–1800. [CrossRef]
42. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
43. Gönen, M. Single continuous predictor. In *Analyzing Receiver Operating Characteristic Curves with SAS*; SAS Publishing: Cary, NC, USA, 2007; Volume 3, pp. 15–36.
44. Ke, J.; Liu, X. Empirical Analysis of Optimal Hidden Neurons in Neural Network Modeling for Stock Prediction. In Proceedings of the 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Wuhan, China, 19–20 December 2008; Volume 2, pp. 828–832.
45. Deypir, M.; Sadreddini, M.H.; Hashemi, S. Towards a variable size sliding window model for frequent itemset mining over data streams. *Comput. Ind. Eng.* **2012**, *63*, 161–172. [CrossRef]