sensors

MDPI

Article

Detection of False Data Injection Attacks in Smart Grids Based on Expectation Maximization

Pengfei Hu [1,2], Wengen Gao [1,2,*], Yunfei Li [1,2], Minghui Wu [1,2], Feng Hua [1,2] and Lina Qiao [1,2]

[1] School of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China
[2] Key Laboratory of Advanced Perception and Intelligent Control of High-end Equipment, Chinese Ministry of Education, Wuhu 241000, China
* Correspondence: ahpuchina@ahpu.edu.cn

Abstract: The secure operation of smart grids is closely linked to state estimates that accurately reflect the physical characteristics of the grid. However, well-designed false data injection attacks (FDIAs) can manipulate the process of state estimation by injecting malicious data into the measurement data while bypassing the detection of the security system, ultimately causing the results of state estimation to deviate from secure values. Since FDIAs tampering with the measurement data of some buses will lead to error offset, this paper proposes an attack-detection algorithm based on statistical learning according to the different characteristic parameters of measurement error before and after tampering. In order to detect and classify false data from the measurement data, in this paper, we report the model establishment and estimation of error parameters for the tampered measurement data by combining the the k-means++ algorithm with the expectation maximization (EM) algorithm. At the same time, we located and recorded the bus that the attacker attempted to tamper with. In order to verify the feasibility of the algorithm proposed in this paper, the IEEE 5-bus standard test system and the IEEE 14-bus standard test system were used for simulation analysis. Numerical examples demonstrate that the combined use of the two algorithms can decrease the detection time to less than 0.011883 s and correctly locate the false data with a probability of more than 95%.

Keywords: false data injection attacks; statistical learning methods; attack detection; attack location; smart grid

check for updates

Citation: Hu, P.; Gao, W.; Li, Y.; Wu, M.; Hua, F.; Qiao, L. Detection of False Data Injection Attacks in Smart Grids Based on Expectation Maximization. *Sensors* **2023**, *23*, 1683. https://doi.org/10.3390/s23031683

Academic Editors: Naveen Chilamkurti and Jong-Hyouk Lee

Received: 28 November 2022
Revised: 5 January 2023
Accepted: 29 January 2023
Published: 3 February 2023

1. Introduction

The current power system is continuously monitored by an energy management system (EMS), and a supervisory control and data acquisition (SCADA) system us used to maintain normal and secure operating conditions [1]. In particular, the SCADA system in the control center uses state estimators to process the received measurements. The estimator obtains the best estimate of the system's state by filtering incorrect data. These state estimates are then transmitted to all EMS to control the proper functioning of the physical aspects of the grid, such as the power flow calculation.

The measurements collected by the SCADA system include not only measurement noise due to the limited precision of sensors and communication medium, but also errors due to various problems, such as connecting and calibrating a failed meter. To decrease the effects of noise and error, power system researchers have developed many methods to deal with the measurements during state estimation [2,3]. The basic principle of these methods is to use the redundancy of multiple measurements to identify and eliminate anomalies.

Most of the technologies used to protect grid systems are designed to ensure system reliability, such as preventing random failures. However, more and more attention has been paid to preventing malicious network attacks in the recent proposals for smart grids [4]. The operation and control of smart grids depend on the complex network space of computer, software and communication technology [5]. Since measurement components supported by smart devices, such as smart instruments and sensors, play important roles in confirming the real-time physical states of power systems, they are likely to be targets of attack. These measuring devices widely use Internet-based protocols in communication systems, which are open to external networks and lack of hardware to prevent tampering. In order to promote data sharing, enterprise networks, and even individual users, are allowed to connect to the infrastructure of power grid information [6]. Potential complex malicious attacks increase after these network interfaces are introduced into power systems [7–10]. Liu et al. [11] indicated in 2009 that a new FDIA could bypass bad data detection (BDD) in current SCADA systems and introduce any errors into state estimation without being detected. Malicious covert data injection of network buses will inevitably have a negative impact on power-system state estimation [12,13]. The injection of these malicious data that deviates state estimates away from security values can directly result in serious social and economic losses, and an attacker can utilize the FDIA to manipulate the electricity price of the electric market [14–16], and this attack can even result in regional power shortages [17].

Du et al. [18] proposed a method to extract network parameters from the limited data obtained by phasor measurement units (PMUs) when the network parameters are unknown and then use these parameters to build an AC attack model, finally making the state estimation deviate from the securely value. Most of the classical methods used to construct the attack model focus on tampering measurements, such as the power injected into the bus and the power flow between buses. Liu et al. [19] proposed a method to attack network parameters which reduces the number of attack measurements by coordinating the modifications of parameters and other measurements in the power system. The attack method is still applicable in cases where the topology and line impedance of the network are incomplete. Since it is unrealistic for an attack to modify network parameters directly. Liu et al. [20] proposed a more universally applicable attack model. The concrete approach is to tamper with network parameters indirectly by exploiting the vulnerabilities that exist when the network parameters are incorrectly handled.

Several directions have been taken in the research of detecting FDIAs in smart grids. Although these detection methods differ to varying degrees, they can be broadly classified into two broad categories. Detection methods can be categorized as model-based detection algorithms and data-driven detection algorithms. In response to the situation in which network parameters are attacked, [21] proposed a way to detect network parameter attacks based on the inconsistency of historical data and specified network parameters. However, such methods are no longer applicable in detecting combinatorial attacks. Methods to detect FDIAs using differences in the probability distributions of historical and current measurement data may not be applicable any longer, such as assuming the attack vector is a trapezoidal attack or that spurious data injected do not significantly deviate from the historical trend [22–24]. In addition, such a detection method will easy cause false detection when encountering actual events, such as sudden changes in the load or from the generator. To deal with this situation, a method was proposed in [25] to detect FDIAs using the difference in the residual probability distribution between historical measurement data and that of current measurement data. This method still maintains good detection performance when facing trapezoidal attacks and real events. Chen et al. [26] proposed a scheme to detect data before state estimation by using vector autoregression model. This scheme uses vector autoregressive model to predict and classifiers to detect, which improves the detection rate based on the autoregressive model. Saleh et al. [27] proposed a detection method to detect FDIAs that destroy the state estimation of PMUs. The phase lock value (PLV) is used to judge whether the phase changes between buses are consistent. If the phase change was no longer constant, the data for the PMU were considered to have been manipulated; otherwise, data security at PMUs was considered. The above are several model-based detection methods.

Unlike model-based detection algorithms for FDIAs, machine learning, as a data-driven technique, implies a huge dependence on historical data of the system under test. Yu et al. [28] proposed a false data injection attack detection method for AC state estimation. When FDIAs exist, their spatial and temporal data correlations may deviate from the correlations under normal conditions. By using wavelet transforms and deep neural networks to analyze the estimated states in continuous time, the proposed method can effectively detect this inconsistency. Xun et al. [29] proposed an extreme learning machine (ELM)-based one-class and one-network (OCON) framework for detecting FDIAs. In this framework, the subnetwork of the state identification layer in OCON uses the ELM algorithm to accurately classify false data and normal data. Almasabi et al. [30] proposed a new method to detect FDIAs using moving average, correlation and machine learning algorithms. The experiments showed that the proposed method is able to detect the attacked PMUs and its timing issues with a high detection rate. Most existing machine-learning-based detection methods generally assume that the labels of the training data are known, which may not be consistent with common sense. Since real-life FDIAs are generally considered as rare events, it may be challenging to obtain the identity of the compromised data. An et al. [31] proposed the use of unsupervised integrated autoencoders connected to a Gaussian mixture model (GMM) to accommodate multiple domains. Attention-based potential representation and minimum error reconstruction features are utilized in the hidden space of the integrated autoencoder. The expectation maximization (EM) algorithm is used to estimate the sample density in the GMM. When the estimated sample density exceeds the learning threshold obtained in the training phase, the sample is identified as an outlier. Since the EM algorithm has the disadvantage of being sensitive to initial values, excellent initialization parameters are required for the next iterative step of the calculation. To deal with this challenge, we are required to develop an unsupervised detection approach.

This paper proposes a detection and location method for the false data injection attacks in smart grid. FDIAs threaten the management and control of grids by tampering with the measurement data of the smart grid systems. In fact, the attacker adds an unknown deviation to the measurement data of a system to launch an FDIA. Since the presence of unknown attacks generates error bias, there are different characteristic parameters for the measurement error contained by false data and that of normal data. Therefore, we used the k-means++ algorithm and the expectation maximization (EM) algorithm to estimate the corresponding parameters of the measured data to eliminate the data affected by the FDIA, and finally achieved the purpose of attack detection. The main contributions of this paper can be summarized as follows:

- Since the error models of both measurement vectors and state variables with false data have the characteristics of the Gaussian mixture model (GMM), a false data injection attack detection method based on the k-means++ and expectation maximization (EM) algorithms is proposed.
- To address the fact that the k-means algorithm is sensitive to the initial clustering centers and affects the convergence efficiency, the k-means++ algorithm is proposed to determine the initial estimated parameters of the GMM in a faster iterative approach.
- The k-means++ algorithm is used to preprocess the data to solve the problem of EM algorithm being sensitive to initial values. It also decreases the calculation complexity of the EM algorithm, and finally detects and locates false data rapidly according to the classification results.

2. System Model

For complex information processing of smart grid, it is necessary to generate corresponding mathematical model according to network topology and data of distribution network [32]. The general linear state equation of voltage and current phasors in the smart grid distribution system is as follows [33]:

$$y = \underbrace{\boldsymbol{Hx}}_{=z} + e \tag{1}$$

where $\boldsymbol{y} \in \mathbb{C}^m$ is the original measurement vector of voltage and current phasor; $\boldsymbol{z}$ is the noiseless measurement vector; $\boldsymbol{x} \in \mathbb{C}^n$ is the vector describing the system state variable; $\boldsymbol{H} \in \mathbb{C}^{m\times n}$ is the network topology matrix describing the vicinity of a given working point; $\boldsymbol{e} \in \mathbb{C}^m$ is the measurement error produced by the sensor, where each component is modeled as an independent homodistributed and obeys a complex Gaussian random variable with a zero mean and variance of σ^2.

Attackers use FDIAs to add attack vectors to the measurement vectors to corrupt the measurements available to the operator. The actual measurements after being attacked are

$$\boldsymbol{y_a} = \underbrace{\boldsymbol{Hx}}_{=\boldsymbol{z}} + \boldsymbol{e} + \boldsymbol{a} \tag{2}$$

where $\boldsymbol{a} \in \mathbb{C}^m$ is the attack vector; $\boldsymbol{y_a} \in \mathbb{C}^m$ represents the measurement after being attacked by false data injection.

With the rapid development of synchronous phasor measurement units (PMUs), a smart grid can obtain impeccable phasor measurement values by arranging PMUs on the terminal buses [34]. Using these measurements, the system state variable $\boldsymbol{x}$ can be accurately estimated. However, due to the price factor of PMUs, the device cannot be installed on all transmission buses of the power system, and can only cooperate with other sensors to obtain system measurements. One of the attacks considered under this condition is that during the stable operation of the power system, one of the N phasor measurements in the measurement vector $\boldsymbol{y}$ is continuously attacked; that is, a component in the attack vector $\boldsymbol{a}$ is not zero. In the subsequent measurement acquisition process, we determine whether the phasor measurements are replaced with false data by $K(K \geq 1)$ measurement vectors. To facilitate the calculation, the obtained measurement samples are converted from complex representation to real coordinate representation, and then the actual obtained component of the ith phase measurement of the kth measurement vector $\boldsymbol{y_k} \in \mathbb{R}^{N\times 2}$ is represented as

$$\boldsymbol{y_{i,k}} = \boldsymbol{z_{i,k}} + \boldsymbol{e_{i,k}} \tag{3}$$

where $\boldsymbol{y_{i,k}} \in \mathbb{R}^{1\times 2}$, $\boldsymbol{z_{i,k}} \in \mathbb{R}^{1\times 2}$, $\boldsymbol{e_{i,k}} \in \mathbb{R}^{1\times 2}$. The error distribution of the secure phase measurement is represented by $p_{\boldsymbol{e}}^{(1)}(\boldsymbol{e}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, and the error distribution of the phase measurement tampered with by the attack is represented by $p_{\boldsymbol{e}}^{(2)}(\boldsymbol{e}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma_2})$. In addition, the phasor measurement error distributions belong to two-dimensional Gaussian distributions with unknown parameters.

For ease of calculation, the actual obtained model for the phasor measurement sample of K measurement vectors is written as

$$\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{E} \tag{4}$$

where $\boldsymbol{Y} \in \mathbb{R}^{NK\times 2}$, $\boldsymbol{Z} \in \mathbb{R}^{NK\times 2}$ and $\boldsymbol{E} \in \mathbb{R}^{NK\times 2}$ represent the original measurement, actual measurement and measurement error obtained from K measurements for N phase measurement units, respectively.

$$\boldsymbol{Y} = [\boldsymbol{y}_{1,1}, \cdots, \boldsymbol{y}_{1,K}, \cdots, \boldsymbol{y}_{N,1}, \cdots, \boldsymbol{y}_{N,K}]^T \tag{5}$$

$$\boldsymbol{Z} = [\boldsymbol{z}_{1,1}, \cdots, \boldsymbol{z}_{1,K}, \cdots, \boldsymbol{z}_{N,1}, \cdots, \boldsymbol{z}_{N,K}]^T \tag{6}$$

$$\boldsymbol{E} = [\boldsymbol{e}_{1,1}, \cdots, \boldsymbol{e}_{1,K}, \cdots, \boldsymbol{e}_{N,1}, \cdots, \boldsymbol{e}_{N,K}]^T \tag{7}$$

Power-grid operators generally apply a likelihood ratio test to each measurement to judge whether the measurement is correct. However, there are errors in the measurement data that conform to a Gaussian distribution, and the number of false alarms increases as the number of measurements increases, making it more difficult to detect false data. In this study, we used the method of processing the results of multiple measurements as a set of data. Since interrelated measurement data are linked, the probability of false alarms can be decreased by mathematically determining the relationship between the data. However, the difficulty of this method is also in which calculation method should be used to quickly determine the relationship between the data in the group. An inappropriate method is likely to increase the workload of the detection system and decrease the detection efficiency.

3. Attack Detection

3.1. Maximum Likelihood Estimation

When all measurements $\boldsymbol{Y}$ are considered as a whole, the corresponding measurement error samples $\boldsymbol{E}$ can be seen as coming from two clusters—one with MK correct phasor measurement samples and the other with $(N-M)K$ attacked tampered phasor measurement samples. Without testing, it is impossible to determine which samples of measurements have been tampered with by FDIAs. The probability distribution of the measurement error $\boldsymbol{e}$ for each measurement $\boldsymbol{y}$ according to the assumed statistics can be represented by a Gaussian mixture model (GMM):

$$p(\boldsymbol{e};\boldsymbol{\theta}) = \sum_{l=1}^{2} \alpha_l p_e^{(l)}(\boldsymbol{e};\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l) \tag{8}$$

where $\alpha_1 = M/N$ and $\alpha_2 = (N-M)/N$ are unknown.

In this paper, we derived the distribution parameters of the measurement error by exploiting the asymptotic property of maximum likelihood estimation (MLE). Knowing about the phase measurements associated with the parameters and the actual values derived from the state variables, the maximum likelihood estimate $\boldsymbol{\theta}$ for unknown parameters can be solved by maximizing the log-likelihood function globally. According to the noise model assumed in (8), the log-likelihood function with parameter vector $\boldsymbol{\theta}{=}[\alpha_1,\alpha_2,\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1,\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2]^{\mathrm{T}}$ can be obtained as

$$\begin{aligned}
\mathcal{L}_I(\boldsymbol{\theta};\boldsymbol{E}) &= \ln[p(\boldsymbol{E};\boldsymbol{\theta})] \\
&= \ln\left[\prod_{i=1}^{N}\prod_{k=1}^{K} p(\boldsymbol{e}_{i,k};\boldsymbol{\theta})\right] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} \ln\left[\sum_{l=1}^{2} \alpha_l p_e^{(l)}(\boldsymbol{y}_{i,k}-\boldsymbol{z}_{i,k};\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)\right]
\end{aligned} \tag{9}$$

The maximum likelihood estimate $\hat{\boldsymbol{\theta}}_{ML}$ was obtained by solving

$$\begin{aligned}
&\underset{\boldsymbol{\theta}}{\arg\max} && \mathcal{L}_I(\boldsymbol{\theta};\boldsymbol{E}) \\
&\text{subject to} && \alpha_1 > 0, \alpha_2 > 0 \\
& && \alpha_1 + \alpha_2 = 1 \\
& && \text{and constraints on} \quad \boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l (l = 1,2)
\end{aligned} \tag{10}$$

Since the cost function in (10) is too complex, we would like to use a method to decrease the complexity of calculating the MLE. Therefore, we introduce a complete dataset $\{\boldsymbol{E},\gamma\}$, where

$$\gamma{=}\begin{bmatrix} \gamma_{1,1,1},\cdots,\gamma_{1,K,1},\cdots,\gamma_{N,1,1},\cdots,\gamma_{N,K,1} \\ \gamma_{1,1,2},\cdots,\gamma_{1,K,2},\cdots,\gamma_{N,1,2},\cdots,\gamma_{N,K,2} \end{bmatrix}^{\mathrm{T}} \tag{11}$$

contains $2NK$ random hidden variables whose values reflect which mixed component the random variable in the measurement error $\boldsymbol{E}$ belongs to. $\gamma_{i,k,l}$ is defined as follows:

$$\gamma_{i,k,l} = \begin{cases} 1, \text{ if } \boldsymbol{e}_{i,k} \text{ belong to } p_e^{(l)}(\boldsymbol{e};\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l) \\ 0, \text{ otherwise} \end{cases} \tag{12}$$

With unobserved data $\gamma_{i,k,l}$, the complete data are $(\boldsymbol{e}_{i,k}, \gamma_{i,k,1}, \gamma_{i,k,2})$. More specifically, if $\boldsymbol{e}_{i,k}$ is the measurement error of the security data, then $\boldsymbol{e}_{i,k}$ belongs to the first mixture component $p_e^{(1)}(\boldsymbol{e}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ of the Gaussian mixture model, and its complete data are $(\boldsymbol{e}_{i,k}, 1, 0)$. If $\boldsymbol{e}_{i,k}$ is the measurement error of the false data, then $\boldsymbol{e}_{i,k}$ belongs to the other components of the Gaussian mixture model, denoted as $(\boldsymbol{e}_{i,k}, 0, 1)$. The log-likelihood function for complete data is

$$\begin{aligned}
&\mathcal{L}_C(\boldsymbol{\theta}; \boldsymbol{E}, \gamma) = \ln[p(\boldsymbol{E}, \gamma; \boldsymbol{\theta})] \\
&= \ln\left[\prod_{i=1}^{N}\prod_{k=1}^{K} p(\boldsymbol{e}_{i,k}, \gamma_{i,k,1}, \gamma_{i,k,2}; \boldsymbol{\theta})\right] \\
&= \ln\left\{\prod_{j=1}^{N}\prod_{k=1}^{K}\prod_{l=1}^{2}\left[\alpha_l p_e^{(l)}(\boldsymbol{e}_{i,k}; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)\right]^{\gamma_{i,k,l}}\right\} \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{2}\gamma_{i,k,l}\ln\left[\alpha_l p_e^{(l)}(\boldsymbol{y}_{i,k} - \boldsymbol{z}_{i,k}; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)\right]
\end{aligned} \tag{13}$$

To avoid ambiguity, the original log-likelihood function $\mathcal{L}_I(\boldsymbol{\theta}; \boldsymbol{E})$ in (9) is referred to as the log-likelihood function for incomplete data. Clearly, the newly introduced log-likelihood function $\mathcal{L}_C(\boldsymbol{\theta}; \boldsymbol{E}, \gamma)$ for complete data is much simpler to calculate. For GMM-compliant measurements, the EM algorithm can be used to approximate MLE [35].

3.2. K-Means++ Algorithm

Since the EM algorithm has the disadvantage of being sensitive to initial values, the parameter $\boldsymbol{\theta}$ needs to be initialized in order to proceed to the next iteration of the calculation. The convergence efficiency is greatly decreased by the randomly chosen initial estimated parameter $\boldsymbol{\theta}^{(0)}$ due to the information uncertainty in estimating parameter $\boldsymbol{\theta}$. At the same time, whether to get a global optimal solution is also worth considering. The k-means algorithm classifies data according to the minimum distance criterion, which is commonly used in the clustering of data streams; its advantages are simplicity and rapidity [36]. The k-means++ algorithm determines the initial estimated parameters of the Gaussian mixture model with faster iterating than the k-means algorithm. At the same time, the k-means++ algorithm decreases the sensitivity to the initial clustering center, thereby accelerating the rate of convergence.

The idea of the k-means++ algorithm can be summarized in two steps. In the first step, the only difference between k-means++ and k-means algorithms is that the k-means++ algorithm chooses initial clustering centers that are far away from each other rather than randomly. Therefore, the above characteristics allow the k-means++ algorithm to have faster calculation speed. In the second step, sample points in the dataset are assigned to cluster centers that are nearest to each other to form different clusters and recalculate cluster centers.

In this paper, the workflow of k-means++ algorithm can be summarized as three steps.

The first step is to select the initial cluster center. First, a sample $\boldsymbol{e}$ is randomly selected from the data set $\boldsymbol{E}$ as the initial clustering center $\boldsymbol{c}_1^{(0)}$. Then, the Euclidean distance between each sample $\boldsymbol{e}_{i,k}$ and the currently existing clustering center $\boldsymbol{c}_1^{(0)}$ is calculated and denoted by $D(\boldsymbol{e}_{i,k})$. Next, the probability of each sample being selected as the next cluster center is calculated by using

$$p_{\mathbf{c}}(\boldsymbol{e}_{i,k}) = \frac{D(\boldsymbol{e}_{i,k})^2}{\sum_{i=1}^{N}\sum_{k=1}^{K} D(\boldsymbol{e}_{i,k})^2} \tag{14}$$

Finally, the second initial cluster center $\boldsymbol{c}_2^{(0)}$ is selected according to the roulette wheel selection.

The second step is to assign the dataset. Assign each sample of the dataset to the appropriate cluster center according to the principle of minimum Euclidean distance.

$$\gamma_{i,k,l}^{(n)} = \begin{cases} 1, l = \arg\min\limits_{l} \left\| \boldsymbol{e}_{i,k} - \boldsymbol{c}_l^{(n)} \right\| \\ 0, \text{otherwise} \end{cases} \tag{15}$$

where (15) indicates that $\boldsymbol{e}_{i,k}$ belongs to the $\boldsymbol{c}_l^{(n)}$-centered clustering domain.

The third step is to update the clustering centers. At the $(n+1)$th iteration, the cluster centers of the dataset are recalculated based on the hidden variable $\gamma^{(n+1)}$. The newly calculated cluster centers are then used as the center of mass of the samples belonging to that category.

$$\boldsymbol{c}_l^{(n+1)} = \frac{\sum\limits_{\gamma_{i,k,l}^{(n)}=1} \boldsymbol{e}_{i,k}}{\sum\limits_{i=1}^{N} \sum\limits_{k=1}^{K} \gamma_{i,k,l}^{(n)}} \tag{16}$$

3.3. EM Algorithm

The idea of EM algorithm is to estimate unknown parameters through two iterations: an expectation (E) step and a maximization (M) step. In the first step (E-step), the conditional expectation of the log-likelihood function for complete data is calculated based on the conditional probability of the hidden variable. In the second step (M-step), the conditional expectation obtained by the E-step is maximized for the desired parameters. Using the estimated parameter $\boldsymbol{\theta}$ obtained with the k-means++ algorithm, we proposed the workflow of the EM algorithm for the $(\eta+1)$th iteration thereafter.

Step 1 (E-step): The conditional expectation for defining the log-likelihood function of complete data is as follows:

$$\begin{aligned} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\eta)}\right) &= E\left\{\ln[p(\boldsymbol{E}, \gamma; \boldsymbol{\theta})]; \boldsymbol{E}, \boldsymbol{\theta}^{(\eta)}\right\} \\ &= \sum_{\gamma} \ln[p(\boldsymbol{E}, \gamma; \boldsymbol{\theta})] \Pr\left(\gamma | \boldsymbol{E}; \boldsymbol{\theta}^{(\eta)}\right) \\ &= \sum_{\gamma} \ln[p(\boldsymbol{E}, \gamma; \boldsymbol{\theta})] \hat{\gamma}_{i,k,l}^{(\eta)} \end{aligned} \tag{17}$$

where $\hat{\gamma}_{i,k,l}^{(\eta)}$ is a shorthand form of the conditional probability $\Pr\left(\gamma_{i,k,l}^{(\eta)} = 1 | \boldsymbol{E}; \boldsymbol{\theta}^{(\eta)}\right)$. $\hat{\gamma}_{i,k,l}^{(\eta)}$ denotes the probability that observed data $\boldsymbol{e}_{i,k}$ come from the lth Gaussian sub-model under the current model parameters, called the responsiveness of sub-model l to observed data $\boldsymbol{e}_{i,k}$. $\hat{\gamma}_{i,k,l}^{(\eta)}$ can be calculated from the Bayesian rule of Equation (18).

$$\hat{\gamma}_{i,k,l}^{(\eta)} = \Pr\left(\gamma_{i,k,l}^{(\eta)} = 1 | \boldsymbol{E}; \boldsymbol{\theta}^{(\eta)}\right) = \frac{\alpha_l^{(\eta)} p_e^{(l)}\left(\boldsymbol{e}_{i,k}; \boldsymbol{\mu}_l^{(\eta)}, \boldsymbol{\Sigma}_l^{(\eta)}\right)}{\sum\limits_{l=1}^{2} \alpha_l^{(\eta)} p_e^{(l)}\left(\boldsymbol{e}_{i,k}; \boldsymbol{\mu}_l^{(\eta)}, \boldsymbol{\Sigma}_l^{(\eta)}\right)} \tag{18}$$

Step 2 (M-step): The maximum of function $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\eta)}\right)$ is obtained from Equation (18) with $\boldsymbol{\theta}$ as the vector parameter. The result of the $(\eta+1)$th iteration is

$$\boldsymbol{\theta}^{(\eta+1)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\eta)}\right) \tag{19}$$

4. Algorithm Implementation

The probability density function (PDF) of random variables in measurement error $\boldsymbol{E}$ is

$$p_{\boldsymbol{e}}(\boldsymbol{e}){=}\alpha_1\mathcal{N}(\boldsymbol{e};\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1)+\alpha_2\mathcal{N}(\boldsymbol{e};\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2) \tag{20}$$

The more appropriate initial vector parameter $\boldsymbol{\theta}^{(0)}=\left[\alpha_1^{(0)},\alpha_2^{(0)},\boldsymbol{\mu}_1^{(0)},\boldsymbol{\Sigma}_1^{(0)},\boldsymbol{\mu}_2^{(0)},\boldsymbol{\Sigma}_2^{(0)}\right]^{\mathrm{T}}$ obtained according to the k-means++ algorithm was used for the first iteration of the EM algorithm. The cost function in (17) can be simplified as

$$\Lambda^{(\eta)}(\boldsymbol{\theta})=\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{l=1}^{2}\ln\left[\alpha_l p_{\boldsymbol{e}}^{(l)}(\boldsymbol{e}_{i,k};\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)\right]\hat{\gamma}_{i,k,l}^{(\eta)} \tag{21}$$

In order to maximize the GMM with parameter $\Lambda^{(\eta)}(\boldsymbol{\theta})$, we can solve

$$\frac{\partial}{\partial\alpha_l}\left[\Lambda^{(\eta)}(\boldsymbol{\theta})+\lambda\left(\sum_{l}^{2}\alpha_l-1\right)\right]=0 \tag{22}$$

$$\frac{\partial}{\partial\boldsymbol{\mu}_l}\left[\Lambda^{(\eta)}(\boldsymbol{\theta})\right]{=}0 \tag{23}$$

$$\frac{\partial}{\partial\boldsymbol{\Sigma}_l}\left[\Lambda^{(\eta)}(\boldsymbol{\theta})\right]=0 \tag{24}$$

where λ in (22) is a Lagrange multiplier. In (24), $\boldsymbol{\theta}{=}\left[\alpha_1^{(\eta)},\alpha_2^{(\eta)},\boldsymbol{\mu}_1^{(\eta+1)},\boldsymbol{\Sigma}_1^{(\eta)},\boldsymbol{\mu}_2^{(\eta+1)},\boldsymbol{\Sigma}_2^{(\eta)}\right]^{\mathrm{T}}$. Meanwhile, the solutions of the equations are all in closed form, and the result is

$$\alpha_l^{(\eta+1)}=\frac{\sum\limits_{i=1}^{N}\sum\limits_{k=1}^{K}\hat{\gamma}_{i,k,l}^{(\eta)}}{NK} \tag{25}$$

$$\boldsymbol{\mu}_l^{(\eta+1)}=\frac{\sum\limits_{i=1}^{N}\sum\limits_{k=1}^{K}\boldsymbol{e}_{i,k}\hat{\gamma}_{i,k,l}^{(\eta)}}{\sum\limits_{i=1}^{N}\sum\limits_{k=1}^{K}\hat{\gamma}_{i,k,l}^{(\eta)}} \tag{26}$$

$$\boldsymbol{\Sigma}_l^{(\eta+1)}=\frac{\sum\limits_{i=1}^{N}\sum\limits_{k=1}^{K}\left(\boldsymbol{e}_{i,k}-\boldsymbol{\mu}_l^{(\eta+1)}\right)^{\mathrm{T}}\left(\boldsymbol{e}_{i,k}-\boldsymbol{\mu}_l^{(\eta+1)}\right)\hat{\gamma}_{i,k,l}^{(\eta)}}{\sum\limits_{i=1}^{N}\sum\limits_{k=1}^{K}\hat{\gamma}_{i,k,l}^{(\eta)}} \tag{27}$$

The above calculations are repeated until the log-likelihood function value no longer changes significantly. By rounding the final data $\hat{\gamma}_{i,k,l}^{(\eta+1)}$ of the hidden variable, we obtain the complete data set $\{\boldsymbol{E},\gamma\}$ and the vector parameter $\boldsymbol{\theta}$ of the GMM.

Thus, the pseudo-algorithm of the joint use of k-means++ algorithm and EM algorithm for parameter estimation of GMM is shown in Algorithm 1.

Algorithm 1 Joint k-means++ and EM algorithms for estimating parameters of GMM.

```
Input: $\boldsymbol{Y}$ and $\boldsymbol{Z}$. For each dataset with $i = 1, 2, \ldots, N$, $k = 1, 2, \ldots, K$ .
    Initialize: Iteration index $n$ = 0 for k-means++ algorithm; the EM algorithm's iteration index $\eta$ = 0; convergence tolerance is $\Delta$; and maximum iteration number is $N_{itr}^{\max}$.
    K-means++ algorithm loop:
    (1) A sample point is randomly selected as the initial cluster center $\boldsymbol{c}_1^{(0)}$, and then the second cluster center $\boldsymbol{c}_2^{(0)}$ is selected according to the roulette wheel selection.
    (2) Update $\gamma^{(n)}$ according to Equation (15), and then reclassify the sample points.
    (3) Update Cluster Center $\boldsymbol{c}_l^{(n+1)}$ according to Equation (16).
    (4) If the convergence condition $\boldsymbol{c}_l^{(n+1)} = \boldsymbol{c}_l^{(n)}$ is satisfied, the k-means++ algorithm is terminated. Otherwise, set $n \leftarrow n + 1$ and return to (2).
    Get the initial estimation parameters:
    (1) $\alpha_l^{(0)} = \sum \gamma_{i,k,l}^{(n+1)}$.
    (2) $\boldsymbol{\mu}_l^{(0)} = \boldsymbol{c}_l^{(n+1)}$.
    (3) $\boldsymbol{\Sigma}_l^{(0)} = \mathrm{var}\left(\boldsymbol{E}; \gamma_{i,k,l}^{(n)} = 1\right)$.
    EM algorithm loop:
    (1) Update $\hat{\gamma}^{(\eta)}$ according to Equation (18).
    (2) Parameters $\alpha_l^{(\eta+1)}$, $\boldsymbol{\mu}_l^{(\eta+1)}$, $\boldsymbol{\Sigma}_l^{(\eta+1)}$ are updated according to Equations (25)–(27).
    (3) If the convergence condition $\mathcal{L}_I\left(\boldsymbol{\theta}^{(\eta+1)}; \boldsymbol{E}\right) - \mathcal{L}_I\left(\boldsymbol{\theta}^{(\eta)}; \boldsymbol{E}\right) \leq \Delta$ or $\eta + 1 = N_{itr}^{\max}$ is satisfied, the EM algorithm is terminated. Otherwise, set $\eta \leftarrow \eta + 1$ and return to (1).
Output: $\left\{\boldsymbol{E}, \gamma^{(\eta+1)}\right\}$ and $\boldsymbol{\theta}^{(\eta+1)}$.
```

5. Algorithm Analysis

5.1. Convergence Analysis

The essence of using k-means++ algorithm to calculate new clustering centers is to minimize the sum of squared error (SSE) function:

$$J\left(\boldsymbol{c}_l^{(n+1)}\right) = \sum_{\gamma_{i,k,l}^{(n+1)}=1} \left\| \boldsymbol{e}_{i,k} - \boldsymbol{c}_l^{(n+1)} \right\|^2 \tag{28}$$

As can be found from the algorithm, SSE is a rigorous coordinate descent procedure. Selecting the mean of the current clustering as the new clustering center ensures that SSE will be decreased at each iteration.

$$J\left(\boldsymbol{c}_l^{(n+1)}\right) \leq J\left(\boldsymbol{c}_l^{(n)}\right) \tag{29}$$

Since SSE is monotonically decreasing and has a lower bound, the optimal solution $\boldsymbol{c}_l$ that converges SSE to the minimum can finally be obtained.

For any Gaussian distribution parameter vector $\boldsymbol{\theta}^{(\eta)}$ in the EM algorithm's parameter space, updating $\alpha_1^{(\eta+1)}$, $\alpha_2^{(\eta+1)}$, $\boldsymbol{\mu}_1^{(\eta+1)}$, $\boldsymbol{\Sigma}_1^{(\eta+1)}$, $\boldsymbol{\mu}_2^{(\eta+1)}$, $\boldsymbol{\Sigma}_2^{(\eta+1)}$ is easily verified via the following relationship [37,38]:

$$Q\left(\boldsymbol{\theta}^{(\eta+1)}, \boldsymbol{\theta}^{(\eta)}\right) \geq Q\left(\boldsymbol{\theta}^{(\eta)}, \boldsymbol{\theta}^{(\eta)}\right) \tag{30}$$

Based on the monotonicity of the log-likelihood function $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\eta)}\right)$ for complete data and the boundedness of $p(\boldsymbol{E}; \boldsymbol{\theta})$ in the EM algorithm, it can be proved that the proposed EM algorithm converges to a stationary point $\mathcal{L}_I^*$ of the log-likelihood function $\mathcal{L}_I(\boldsymbol{\theta}; \boldsymbol{E})$ for incomplete data.

5.2. Complexity Analysis

In the complexity analysis, we focused on the iterative process between the k-means++ algorithm and the EM algorithm in the estimation of parameters. Since they consume more computationally, complexity was evaluated with floating point operations (FLOPs).

We define FLOPs in relation to some basic operations as follows:

(1) ε_{add}: FLOPs required for addition.
(2) ε_{sub}: FLOPs required for subtraction.
(3) ε_{mul}: FLOPs required for multiplication.
(4) ε_{div}: FLOPs required for division.
(5) ε_{exp}: FLOPs required for exponential.
(6) ε_{pow}: FLOPs required for square.
(7) ε_{sqrt}: FLOPs required for square root.
(8) ε_{com}: FLOPs required for comparation.
(9) ε_{ass}: FLOPs required for assignment.

Note that the FLOPs used in actual practice may differ depending on the processor.

Since both k-means++ and EM algorithms are iterative, we focused our analysis in a single iterative process. The $(n+1)$th iteration of the k-means++ algorithm to reclassify the dataset according to (15) requires $NK(4\varepsilon_{sub}+4\varepsilon_{pow}+2\varepsilon_{add}+1\varepsilon_{com}+2\varepsilon_{ass})$ flops, and to update the clustering center according to (16) requires $(3NK-5)\varepsilon_{add}+4\varepsilon_{div}+1\varepsilon_{sub}$. We define $FL(\boldsymbol{c})$ as the FLOPs required to estimate cluster center $\boldsymbol{c}$ in one iteration of the k-means++ algorithm.

$$\begin{aligned} FL(\mathbf{c}) = {} & (5NK-5)\varepsilon_{add}+(4NK+1)\varepsilon_{sub} \\ & +4\varepsilon_{div}+4NK\varepsilon_{pow}+NK\varepsilon_{com}+2NK\varepsilon_{ass} \end{aligned} \tag{31}$$

The update of $\hat{\gamma}_{i,k,l}^{(\eta)}$ needs to be evaluated during the $(\eta+1)$th iteration of the EM algorithm, where

$$\alpha_l p_e^{(l)}(\boldsymbol{e}_{i,k};\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l) = \frac{\alpha_l}{2\pi|\boldsymbol{\Sigma}_l|^{1/2}}\cdot\exp\left[-\frac{(\boldsymbol{e}_{i,k}-\boldsymbol{\mu}_l)\boldsymbol{\Sigma}_l^{-1}(\boldsymbol{e}_{i,k}-\boldsymbol{\mu}_l)^{\mathrm{T}}}{2}\right] \tag{32}$$

requires $2((NK+4)\varepsilon_{mul}+(2NK+1)\varepsilon_{sub}+(NK+1)\varepsilon_{div}+(NK+1)\varepsilon_{add}+2NK\varepsilon_{pow}+NK\varepsilon_{exp}+1\varepsilon_{sqrt})$ FLOPs. Equation (18) requires $NK(1\varepsilon_{add}+1\varepsilon_{div}+1\varepsilon_{sub})$ FLOPs. With $\hat{\gamma}_{i,k,l}^{(\eta)}$, we can calculate the Equations (25)–(27), which require $(NK-1)\varepsilon_{add}+1\varepsilon_{div}+1\varepsilon_{sub}$ FLOPs, $2(2(NK-1)\varepsilon_{add}+2NK\varepsilon_{mul}+2\varepsilon_{div})$ FLOPs and $2(2(NK-1)\varepsilon_{add}+2NK\varepsilon_{sub}+2NK\varepsilon_{pow}+2NK\varepsilon_{mul}+2\varepsilon_{div})$ FLOPs, respectively. We define $FL(\boldsymbol{\theta})$ as the FLOPs required to estimate $\boldsymbol{\theta}$ during each EM algorithm iteration.

$$\begin{aligned} FL(\boldsymbol{\theta}) = {} & (12NK+7)\varepsilon_{add}+(9NK+3)\varepsilon_{sub} \\ & +(10NK+8)\varepsilon_{mul}+(3NK+11)\varepsilon_{div} \\ & +2NK\varepsilon_{exp}+4NK\varepsilon_{pow}+2\varepsilon_{sqrt} \end{aligned} \tag{33}$$

Finally, the number of iterations required to achieve convergence is assumed to be N_{itr}^{k} or N_{itr}^{EM} for the k-means++ and EM algorithms, respectively. Then, the FLOPs needed to ultimately estimate the vector parameter $\boldsymbol{\theta}$ are approximately

$$FL \approx N_{itr}^{k}[FL(\boldsymbol{c})]+N_{itr}^{EM}[FL(\boldsymbol{\theta})] \tag{34}$$

6. Simulation Analysis

To verify the feasibility of the proposed algorithm, the simulation in this paper was performed with IEEE 5-bus standard test system and IEEE 14-bus standard test system. The MATLAB R2018b software was used for simulation, and the related data in the MATPOWER 7.1 power simulation package were used for routine power flow calculation. The final operating data were used as the measurement data for the power system. The attack vector was injected into the system first, and then the k-means++ algorithm and EM algorithm were jointly used to verify the feasibility of this detection method.

6.1. Simulation Parameters

The related data modified from the simulation of IEEE 5-bus standard test system are shown in Table 1. The other data were unchanged. We summarize the simulation parameters that were used in the simulation in Table 2, and generated simulation data based on these parameters to test the algorithm.

Table 1. Simulation parameters.

I_{1-2}	**Raw Data**	**Simulation Parameters**
Amplitude/p.u.	2.5078	2.5369
Phase angle/°	-1.8803	-1.1809

Table 2. Simulation parameters.

Parameter	**Value**
N	6
K	100
μ_1	[0 0]
μ_2	[0.03 0.03]
σ	0.01
Δ	10^{-6}
$N_{itr}^{\max}$	100

6.2. Simulation Results

For the 600 data points shown in Figure 1, the measurement errors of some phasors begin to shift when a meter measurement in the power system is tampered with. Figure 2 shows the initial data-clustering results processed by the k-means++ algorithm. The classification results of the GMM and data obtained after the subsequent EM algorithm are shown in Figure 3, and the images of their final classification results are basically consistent with those shown in Figure 1. Figure 4 visualizes the PDF image of the measurement error distribution of GMM, and the figure shows the error offset caused by the false data.

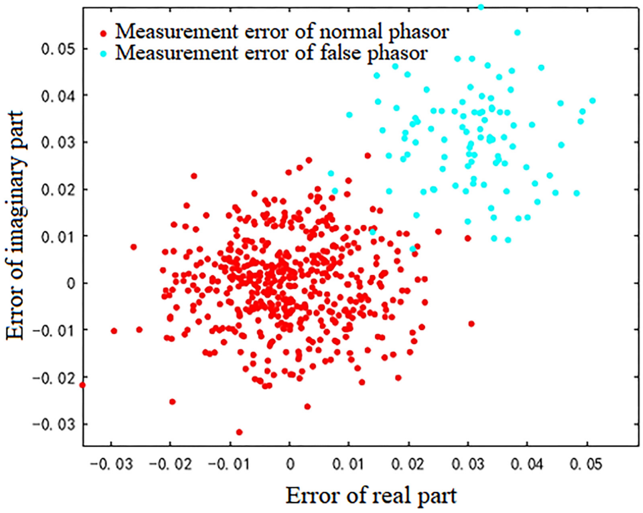

Figure 1. The actual distribution of phase measurement errors after injecting false data.

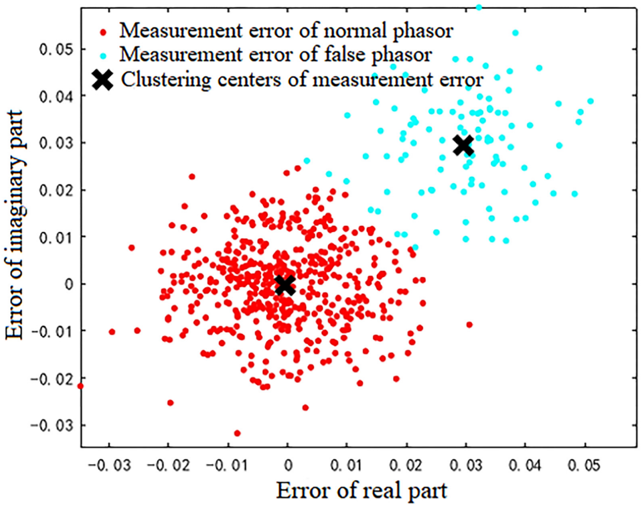

Figure 2. The processing results of the k-means++ algorithm.

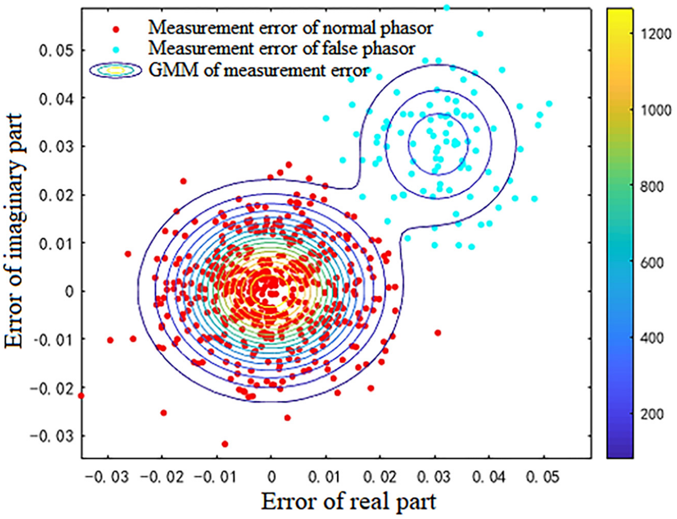

Figure 3. The processing results of the EM algorithm.

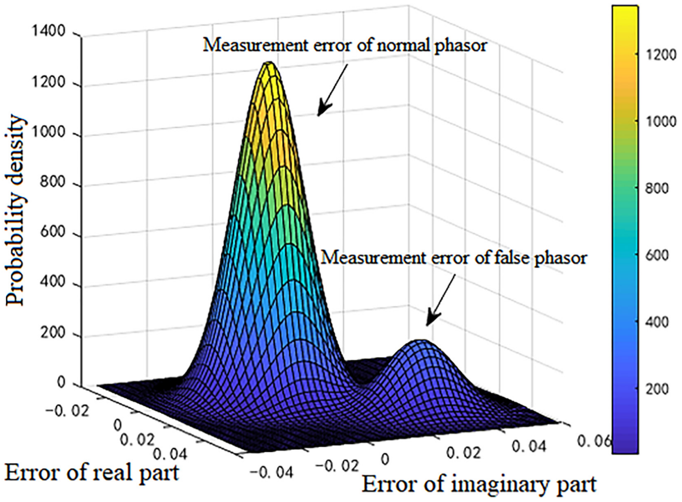

Figure 4. PDF of the GMM of measurement errors.

Figure 5 shows that the sum of squared errors of the model gradually flattens out as the number of iterations monotonically changes when using the k-means++ algorithm for simulation. Figure 6 shows that with the EM algorithm, the logarithmic likelihood function values of the model gradually flatten out as the number of iterations monotonically changes. The simulation results show that both algorithms can take little time to achieve convergence.

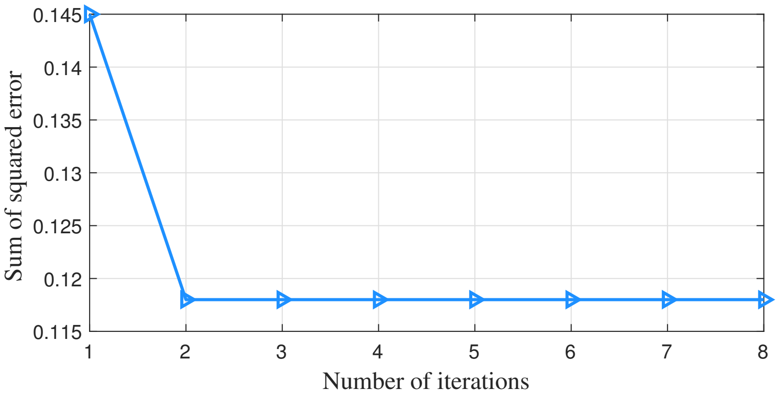

Figure 5. The change in the sum of the squared errors under the k-means++ algorithm.

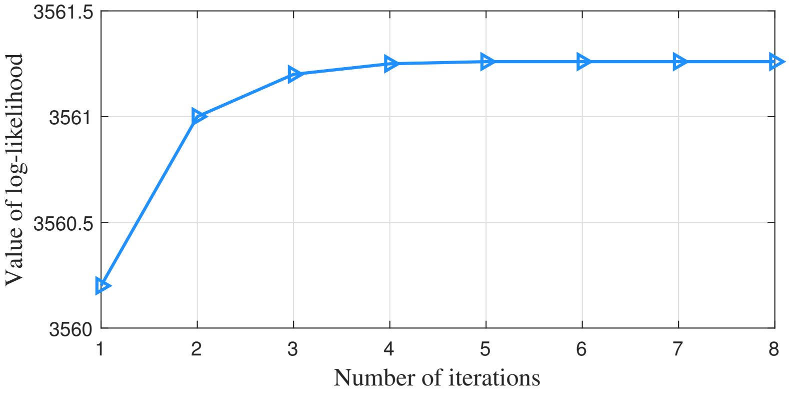

Figure 6. The change in the log-likelihood function value under the EM algorithm.

The simulation result shows in Figure 7 that the detected false data come from the branches I_{1-2} between measurement buses 1 and 2. There was one misdetected measurement datum each in branch I_{1-5} and branch I_{4-5}.

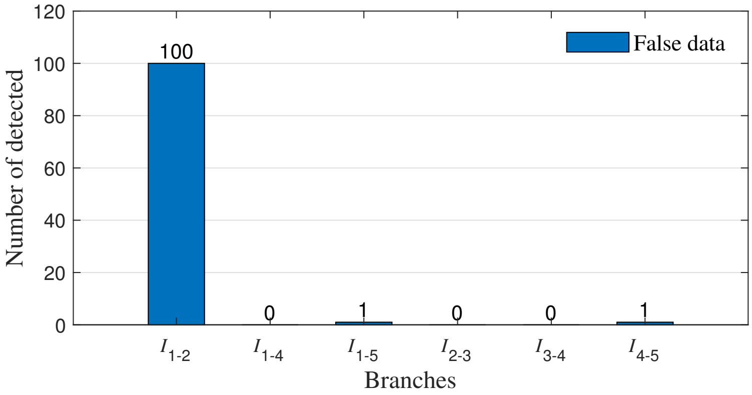

Figure 7. Localization of false data.

For a changing number of measurement buses injected with false data, the average error change of vector parameter $\boldsymbol{\theta}=[\alpha_1, \alpha_2, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2]^{\mathrm{T}}$ in GMM obtained by the detection method in this paper is shown in Figures 8–10. It can be seen that as the false data increase in number, the estimation errors of parameters α_2, $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_2$ of this algorithm decrease continuously.

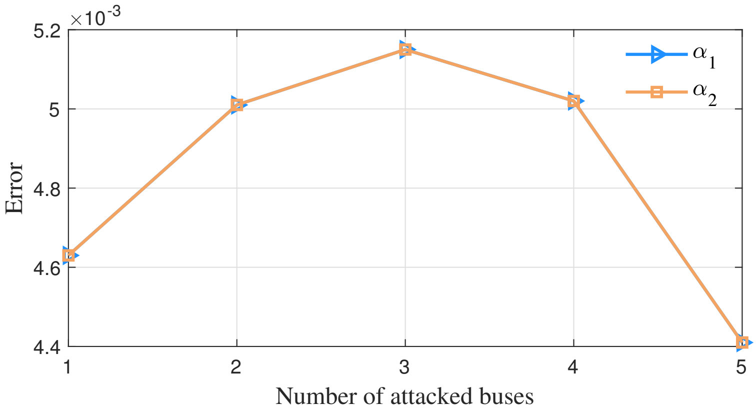

Figure 8. The error variation of the parameter α while the number of attacked buses varies.

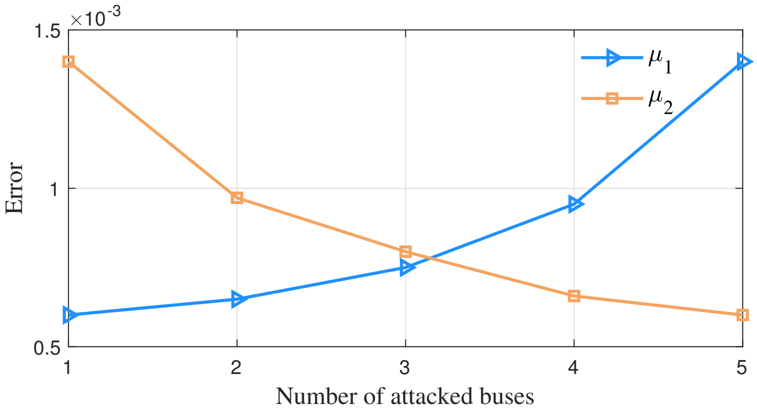

Figure 9. The error variation of the parameter μ while the number of attacked buses varies.

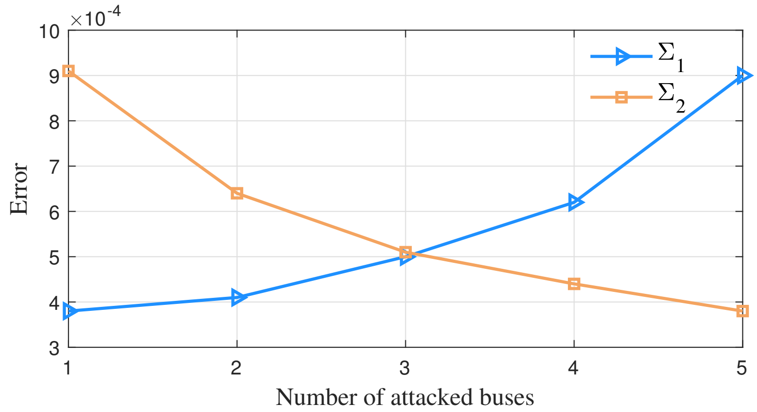

Figure 10. The error variation of the parameter $\boldsymbol{\Sigma}$ while the number of attacked buses varies.

As the proportion of false data in the overall data increases, the probabilities of false data detection, missed detection and false detection by this algorithm change, as shown in Figure 11. It can be seen that the detection rate of the algorithm for false data is basically above 95%, and the detection probability can be further improved to above 99% as the amount of false data increases; thus, the probabilities of false detection and missed detection are normally below 1%.

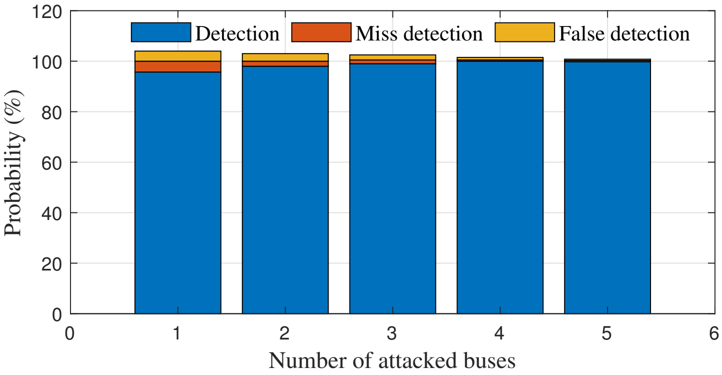

Figure 11. Probability of false data detection.

In order to further verify the rapidity of the algorithm proposed in this paper for detecting false data injection attacks, we have conducted 1000 repeated experiments. The simulated time statistic histogram and normal distribution curve obtained after 1000 repetitions of simulation experiments are shown in Figure 12. From the normal distribution curve in the graph, it can be seen that the algorithm can basically detect false data in 0.011883 s.

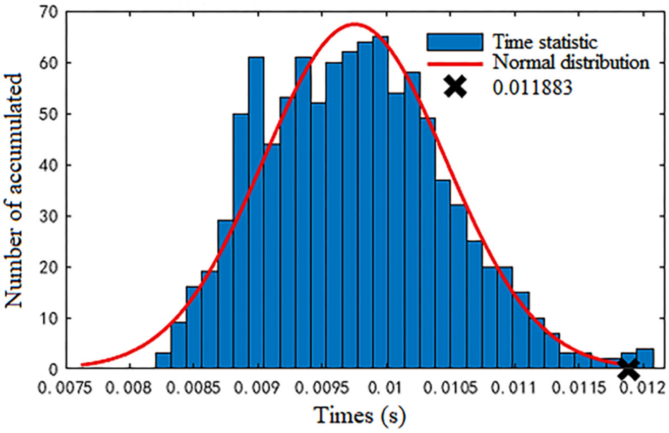

Figure 12. The simulation time statistics of 1000 repeated experiments and their normal distribution.

To verify the feasibility of the proposed algorithm, it was further tested in the IEEE 14-bus standard test system. The measurement errors of active and reactive power of the bus and transmission lines and the errors after being attacked by false data injection are shown in Table 3. The validity of the method was verified by injecting false data into arbitrarily selected measurement units. One thousand sets of quantitative measurement vectors with false data were generated as experimental data according to the Monte Carlo method.

Table 3. The measurement error before and after the power system was attacked.

Types of Measurements	Measurement Error σ before the Attack	Measurement Error σ after the Attack
P_i	0.01	0.015
Q_i	0.01	0.015
P_{ij}	0.008	0.012
Q_{ij}	0.008	0.012

The attack vector injected in this paper against the IEEE 14-bus system was

$$a = [\Delta P_3, \Delta Q_2, \Delta Q_3, \Delta P_{1-2}, \Delta P_{2-3}, \Delta P_{4-2}, \Delta Q_{1-2}, \Delta Q_{2-3}, \Delta Q_{4-2}]^{\mathrm{T}} \qquad (35)$$

Firstly, the measurement errors were used to detect FDIAs. The measurement errors obtained by Monte Carlo method for 1000 instances of normal data were transformed into samples that conformed to the standard normal distribution model, and the measurement error data obtained are shown in Figure 13. All the data conform to the model of standard normal distribution, and the measurement errors of the sample data are not shifted.

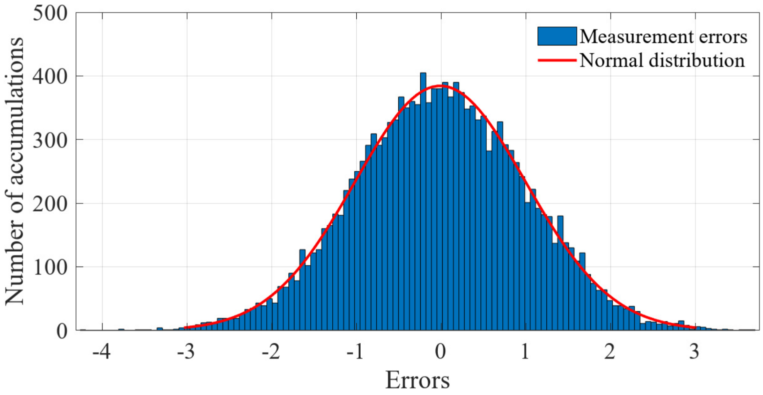

Figure 13. Measurement errors of normal data.

The results of the measurement error after injecting false data are shown in Figure 14. It can be seen in the figure that the FDIAs with Equation (35) as the attack vector made the degree of offset of the measurement error more significant. The results of clustering the measurement errors after the false data injection attack by the k-means++ algorithm are shown in Figure 15.

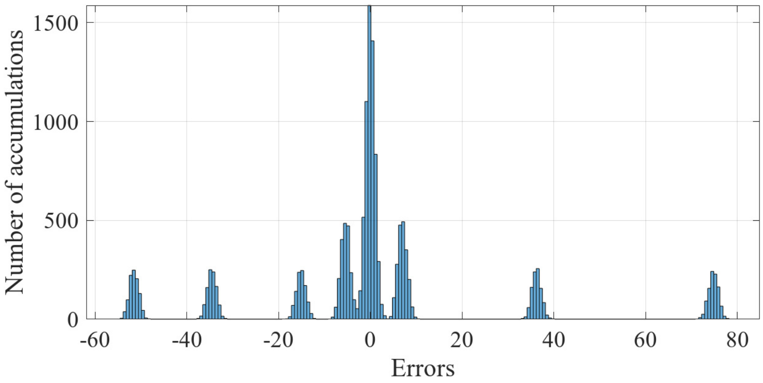

Figure 14. Measurement error of injecting false data.

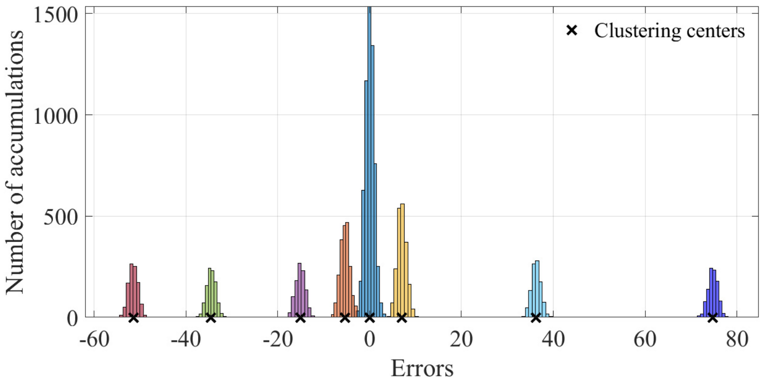

Figure 15. Clustering results of the k-means++ algorithm.

The data preprocessed using the k-means++ algorithm were further iteratively calculated using the EM algorithm. The final PDF image of the GMM of the measurement error was obtained as shown in Figure 16. The results of classifying the sample data of 1000 measurement vectors according to the fitted GMM are shown in Figure 17. From the figure, it can be seen that there is no influence of bias in the normal measurement data, so its error distribution is basically around zero. The data with error deviations were removed and classified by classifying the sample data. It is known that the power measurement data of P_3, Q_2, Q_3, P_{1-2}, P_{2-3}, Q_{2-3}, Q_{1-2} and Q_{4-2} in the power system were tampered with by the attacker through FDIAs. The detection of false data in the measurement data using the algorithm of this paper is shown in Figure 18. A small number of data were identified as normal data because the data in measurement units P_3, Q_3, P_{1-2} and P_{2-3} are more similar to the normal data.

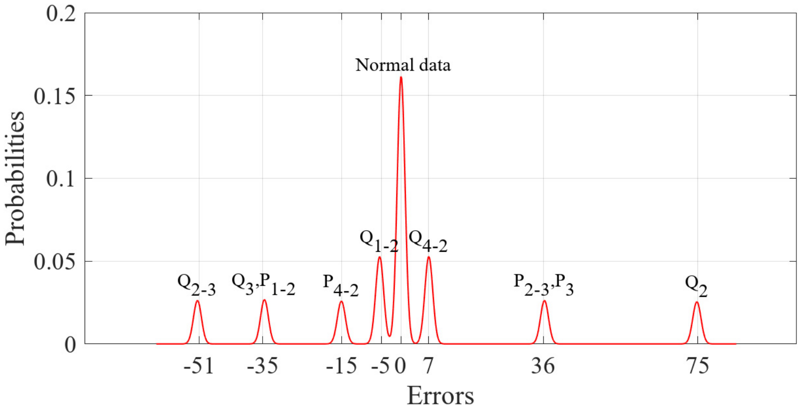

Figure 16. PDF of measurement errors.

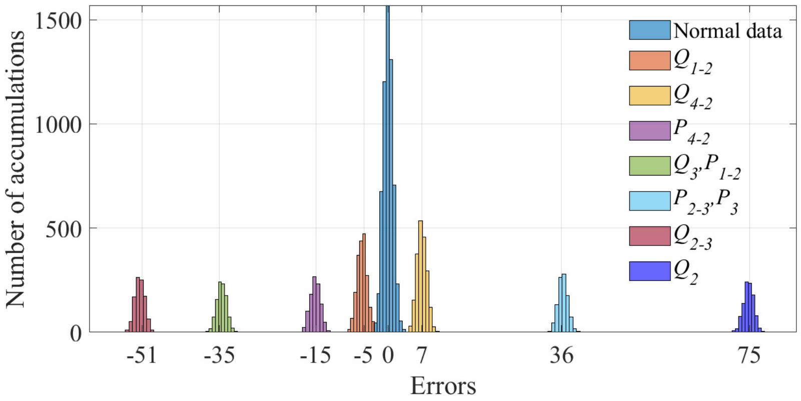

Figure 17. Classification results of the EM algorithm.

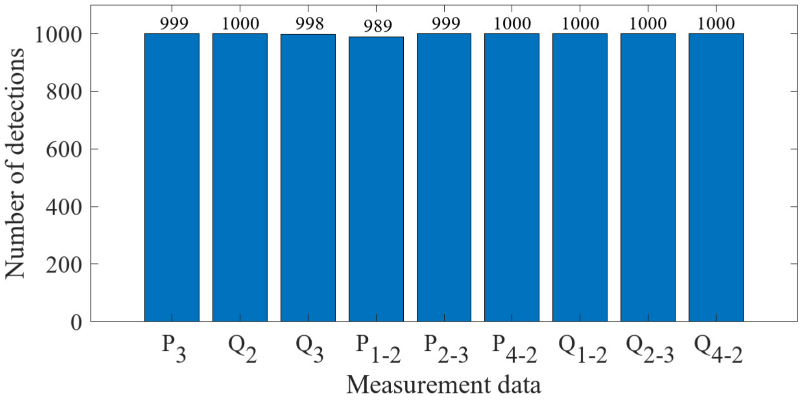

Figure 18. Detection results of false data.

Secondly, we detected FDIAs from the perspective of the results of state estimation. When not under attack, 100 sets were randomly selected from the 1000 sets of measurement data for state estimation. The errors of their state estimation results were transformed into samples that conformed to the model of standard normal distribution, and the obtained estimation errors are shown in Figure 19. All data conform to the model with a standard normal distribution, and none of the sample data are biased by the measurement errors.

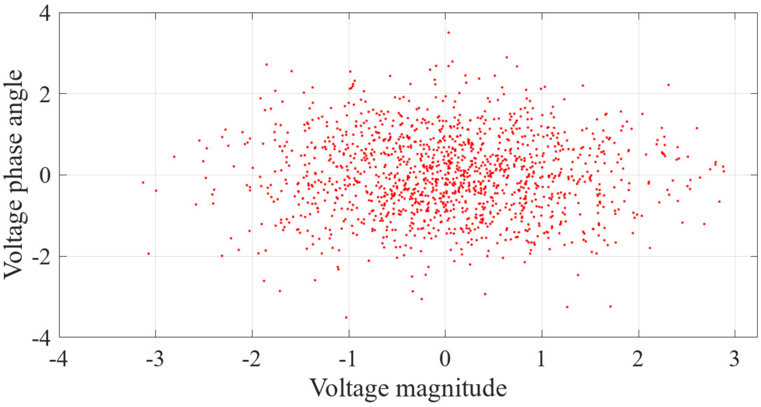

Figure 19. Errors of the state estimation under normal conditions.

The results of its measurement error after injecting false data are shown in Figure 20. From the figure, it can be seen that the voltage amplitude and phase angle of the state estimate of some buses are significantly shifted.

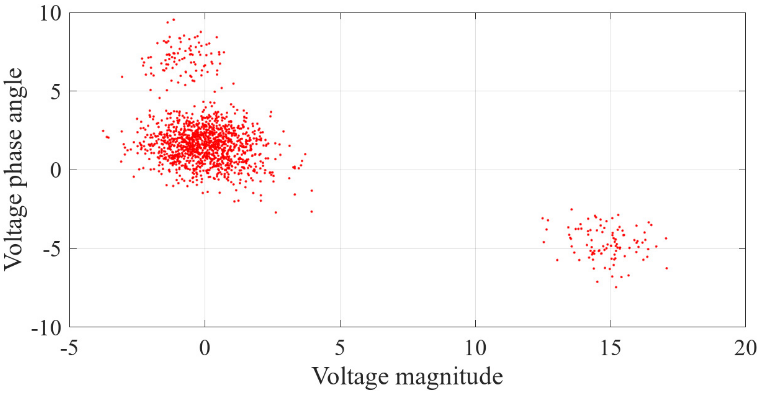

Figure 20. Errors of state estimation after false data injection.

The data preprocessed by the k-means++ algorithm were further iteratively calculated using the EM algorithm, and the final PDF image of the state estimation error conforming to the GMM is shown in Figure 21. The results of classifying the sample data of 100 state variables according to the fitted GMM are shown in Figure 22. From the figure, it can be seen that the data with error deviations were removed and classified by classifying the sample data. The errors of voltage magnitude and phase angle of bus 1 and buses 4–14 are around zero, and their deviations are very small, so they basically have no impact on the power system. The results of the state estimation of bus 3 are mainly the offset of voltage amplitude, which has a mild impact on the power system. The results of the state estimation of bus 2 show large shifts in voltage magnitude and phase angle, indicating that bus 2 was the main target of the FDIAs. The detection of false data in the measurement vector using the algorithm proposed in this paper is shown in Figure 23.

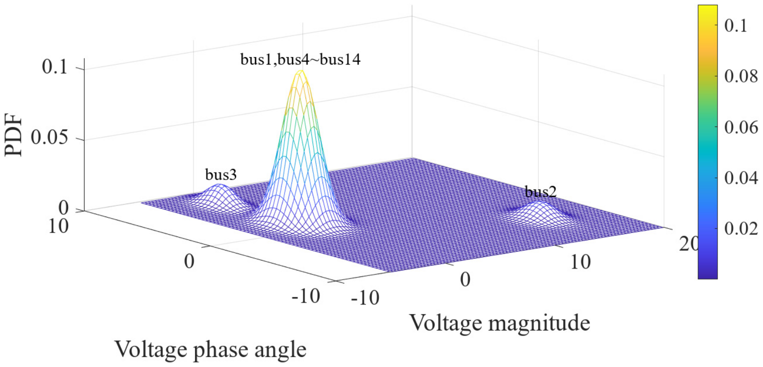

Figure 21. PDF of state estimation errors.

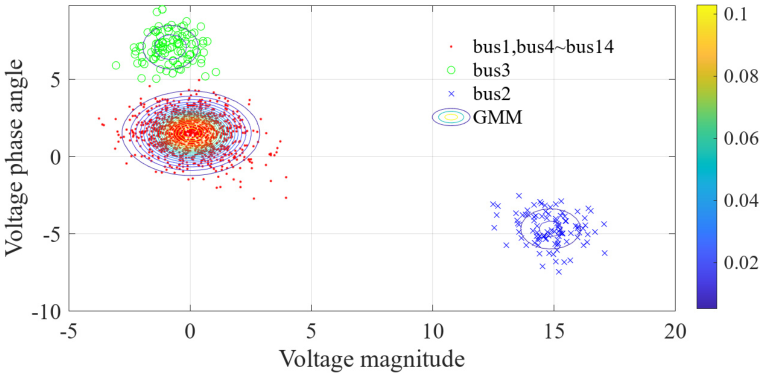

Figure 22. Classification results of the EM algorithm.

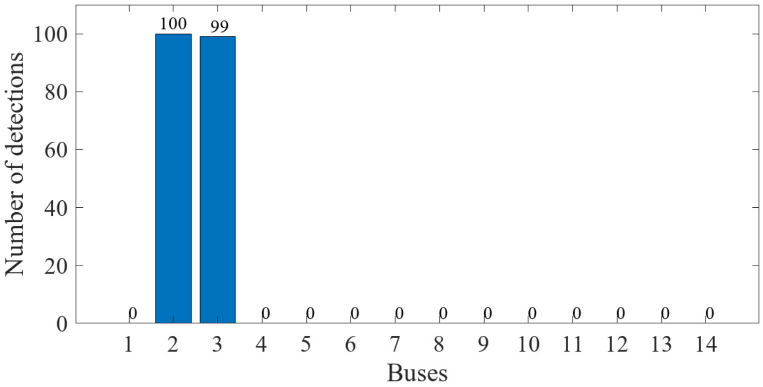

Figure 23. Detection results of false data.

7. Conclusions

Considering that false data injection attacks can disrupt the secure operation of smart grids, we proposed a method to detect and locate false data injection attacks in power systems using statistical learning. By combining the k-means++ algorithm with the EM algorithm, it is possible to accurately model the smart grid bus measurement data within 0.011883s. At the same time, the GMM containing the characteristic parameters of data measurement errors can be obtained. Numerical examples showed that the mathematical model obtained by this joint algorithm provides a detection probability of more than 95% for false data, and can accurately locate the measured buses that are tampered with by FDIAs.

Subsequent research can provide the best choice of GMM with different models by combining the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Silhouette Coefficient (SC), Calinski–Harbasz (CH) score and other methods, so as to build a more perfect model to improve the algorithm in this paper.

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

Funding: This research was funded by the National Natural Science Foundation of China (U21A20146), Natural Science Foundation of AnHui Province (1908085MF215) and Key Research and Development Project of Anhui Province (201904a05020007).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the anonymous reviewers for their valuable comments.

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

References

1. Abur, A.; Exposito, A.G. *Power System State Estimation: Theory and Implementation*; CRC Press: Boca Raton, FL, USA, 2004.
2. Monticelli, A.; Wu,F.F.; Yen,M. Mutiple bad data identwication for state estimation by combinatorial oftimization. *IEEE Trans. Power Deliv.* **1986**, *1*, 361–369. [CrossRef]
3. Granelli, G.P.; Montagna, M. Identification of interacting bad data in the framework of the weighted least square method. *Electr. Power Syst. Res.* **2008**, *78*, 806–814. [CrossRef]
4. Harvey, M.; Long, D.; Reinhard, K. Visualizing nistir 7628, guidelines for smart grid cyber security. In Proceedings of the 2014 Power and Energy Conference at Illinois (PECI), Champaign, IL, USA, 28 February–1 March 2014; pp. 1–8. [CrossRef]
5. Zanero, S. When cyber got real: Challenges in securing cyber-physical systems. In Proceedings of the 2018 IEEE Sensors, New Delhi, India, 28–31 October 2018; pp. 1–4. [CrossRef]
6. Ten, C.W.; Liu, C.C.; Manimaran, G. Vulnerability assessment of cybersecurity for SCADA systems. *IEEE Trans. Power Syst.* **2008**, *23*, 1836–1846. https://ieeexplore.ieee.org/document/4652578. [CrossRef]
7. Khurana, H.; Hadley, M.; Lu, N.; Frincke, D.A. Smart-grid security issues. *IEEE Secur. Priv.* **2010**, *8*, 81–85. MSP.2010.49. [CrossRef]
8. Mo, Y.; Kim, H. J.; Brancik, K.; Dickinson, D.; Lee, H.; Perrig, A.; Sinopoli, B. Cyber–physical security of a smart grid infrastructure. *Proc. IEEE* **2012**, *100*, 195–209. [CrossRef]
9. 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. [CrossRef]
10. Metke, A.R.; Ekl, R.L. Smart grid security technology. In Proceedings of the 2010 Innovative Smart Grid Technologies (ISGT), Gaithersburg, MD, USA, 19–21 January 2010; pp. 1–7. [CrossRef]
11. Liu, Y.; Reiter, M.K.; Ning, P. False data injection attacks against state estimation in electric power grids. In Proceedings of the 2009 ACM Conference on Computer and Communications Security (CCS), Chicago, IL, USA, 9–13 November 2009; pp. 1–33. [CrossRef]
12. Xie, B.; Peng, C.; Zhang, H.; Yang, M. Power system state estimation based on network attack node credibility. *Chin. J. Sci. Instrum.* **2018**, *39*, 157–166. [CrossRef]
13. Ahmadi, N.; Chakhchoukh, Y.; Ishii, H. Power systems decomposition for robustifying state estimation under cyber attacks. *IEEE Trans. Power Syst.* **2021**, *36*, 1922–1933. [CrossRef]
14. Jia, L.; Thomas, R.J.; Tong, L. Impacts of malicious data on real-time price of electricity market operations. In Proceedings of the Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2012; pp. 1907–1914. [CrossRef]
15. Xie, L.; Mo, Y.; Sinopoli, B. Integrity data attacks in power market operations. *IEEE Trans. Smart Grid* **2011**, *2*, 659–666. [CrossRef]
16. Choi, D.H.; Xie, L. Malicious ramp-induced temporal data attack in power market with look-ahead dispatch. In Proceedings of the 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm), Tainan, Taiwan, 5–8 November 2012; pp. 330–335. [CrossRef]
17. Yuan, Y.; Li, Z.; Ren, K. Modeling load redistribution attacks in power systems. *IEEE Trans. Smart Grid* **2011**, *2*, 382–390. [CrossRef]
18. Du, M.; Pierrou, G.; Wang, X.; Kassouf, M. Targeted false data injection attacks against AC state estimation without network parameters. *IEEE Trans. Smart Grid* **2021**, *12*, 5349–5361. [CrossRef]
19. Liu, C.; Liang, H.; Chen, T. Network parameter coordinated false data injection attacks against power system AC state estimation. *IEEE Trans. Smart Grid* **2021**, *12*, 1626–1639. [CrossRef]
20. Liu, C.; He, W.; Deng, R.; Tian, Y.C.; Du, W. False data injection enabled network parameter modifications in power systems: Attack and detection. *IEEE Trans. Ind. Inform.* **2022**, *19*, 177–188. [CrossRef]

21. Molzahn, D.K.; Wang, J. Detection and characterization of intrusions to network parameter data in electric power systems. *IEEE Trans. Smart Grid* **2019**, *10*, 3919–3928. [CrossRef]
22. Chaojun, G.; Jirutitijaroen, P.; Motani, M. Detecting false data injection attacks in AC state estimation. *IEEE Trans. Smart Grid* **2015**, *6*, 2476–2483. [CrossRef]
23. Singh, S.K.; Khanna, K.; Bose, R.; Panigrahi, B.K.; Joshi, A. Joint-transformation-based detection of false data injection attacks in smart grid. *IEEE Trans. Ind. Inform.* **2018**, *14*, 89–97. [CrossRef]
24. Li, B.; Ding, T.; Huang, C.; Zhao, J.; Yang, Y.; Chen, Y. Detecting false data injection attacks against power system state estimation with fast go-decomposition approach. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2892–2904. [CrossRef]
25. Cheng, G.; Lin, Y.; Zhao, J.; Yan, J. A highly discriminative detector against false data injection attacks in AC state estimation. *IEEE Trans. Smart Grid* **2022**, *13*, 2318–2330. [CrossRef]
26. Chen, Y.; Hayawi, K.; Zhao, Q.; Mou, J.; Yang, L.; Tang, J.; Li, Q.; Wen, H. Vector auto-regression-based false data injection attack detection method in edge computing environment. *Sensors* **2022**, *22*, 6789. [CrossRef]
27. Almasabi, S.; Alsuwian, T.; Javed, E.; Irfan, M.; Jalalah, M.; Aljafari, B.; Harraz, F.A. A novel technique to detect false data injection attacks on phasor measurement units. *Sensors* **2021**, *21*, 5791. [CrossRef]
28. Yu, J.Q.; Hou, Y.; Li, V. Online False Data Injection Attack Detection with Wavelet Transform and Deep Neural Networks. *IEEE Trans. Ind. Inform.* **2018**, **14**, 3271–3280.. [CrossRef]
29. Xue, D.; Jing, X.; Liu, H. Detection of False Data Injection Attacks in Smart Grid Utilizing ELM-Based OCON Framework. *IEEE Access* **2019**, *7*, 31762–31773.. [CrossRef]
30. Almasabi, S.; Alsuwian, T.; Awais, M.; Irfan, M.; Jalalah, M.; Aljafari, B.; Harraz, F.A. False Data Injection Detection for Phasor Measurement Units. *Sensors* **2022**, *22*, 3146. [CrossRef] [PubMed]
31. An, P.; Wang Z.; Zhang, C. Ensemble unsupervised autoencoders and Gaussian mixture model for cyberattack detection. *Inf. Process. Manag. Libr. Inf. Retr. Syst. Commun. Netw. Int. J.* **2022**, *59*, 102844.. [CrossRef]
32. Sheng, T.; Wu, W.; Sun, H.; Wang, Z.; Sun, Q.; Ma, J. A fully distributed topology identification approach for active distribution network based on multi-agent framework. In Proceedings of the 2018 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia), Singapore, 22–25 May 2018; pp. 435–440. [CrossRef]
33. Chen, J.C.; Chung, H.M.; Wen, C.K.; Li, W.T.; Teng, J.H. State estimation in smart distribution system with low-precision measurements. *IEEE Access* **2017**, *5*, 22713–22723. [CrossRef]
34. Jiang, J.; Qian, Y. Defense mechanisms against data injection attacks in smart grid networks. *IEEE Commun. Mag.* **2017**, *55*, 76–82. [CrossRef]
35. Sheng, J.; Liu, D. An improved maximum likelihood approach to image reconstruction using ordered subsets and data subdivisions. *IEEE Trans. Nucl. Sci.* **2004**, *51*, 130–135.. [CrossRef]
36. Duan, X.; Sun, G.; Tao, Y. Moving target detection based on genetic k-means algorithm. In Proceedings of the 2011 IEEE 13th International Conference on Communication Technology, Jinan, China, 25–28 September 2011; pp. 819–822. [CrossRef]
37. Watanabe, M.; Yamaguchi, K. *The EM Algorithm and Related Statistical Models*; CRC Press: Boca Raton, FL, USA, 2003. [CrossRef]
38. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.