# Power System State Estimation using Conditional Generative Adversarial Network

Yi He [1], Songjian Chai [1], Zhao Xu [1*], Chun Sing Lai [2]

[1] Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong
[2] Department of Electronic and Computer Engineering, Brunel University London, London, UB8 3PH, UK
[*] eezhaoxu@polyu.edu.hk

**Abstract: Accurate power system state estimation (SE) is essential for power system control, optimization, and security analyses. In this work, a model-free and fully data-driven approach was proposed for power system static SE based on conditional Generative Adversarial Network (GAN). Comparing with the conventional SE approach, i.e., Weighted Least Square (WLS) based methods, any appropriate knowledge of the system model is not required in the proposed method. Without knowing the specific model, GAN can learn the inherent physics of underlying state variables purely relying on historic samples. Once the model has been trained, it can estimate the corresponding system state accurately given the system raw measurements, which are sometimes characterized by incompletions and corruptions in addition to noises. Case studies on the IEEE 118-bus system and a 2746-bus Polish system validate the effectiveness of the proposed approach, and the mean absolute error is less than 1.2e-3 p.u. and 5.3e-3 rad for voltage magnitude and phase angle, respectively, which indicates a high potential for practical applications.**

## 1. Introduction

In power systems, state estimation (SE) plays an important role in contemporary energy management systems (EMS), which is a vital application of the EMS [1, 2], and the accurate SE is essential for power system control, optimization and security analysis. While the system observability is the prerequisite to traditional SE methods, the network is, however, not always fully observable due to e.g. malfunction of measurement devices, missing of measurement data or interference by malicious data attack [3].

In the early years, SE is performed widely based on supervisory control and data acquisition (SCADA) measurements [4, 5]. But nowadays, additional phasor measurement units (PMUs) have been installed as a measurement device to provide faster and more accurate measurements, which adapts to the growing deployment of distributed renewable generators, electric vehicles, and demand response programs[6]. The PMU shows great advantages over conventional SCADA on the following aspects: (1) synchronization. Each PMU measurement is time-stamped and synchronized from the global positioning satellite (GPS) system [7]; (2) higher measurement precision. This is because network buses' voltage phasor can be measured direly and a reference bus with fixed voltage phase angle is not needed to choose any more. (3) higher sampling rates (up to 60 samples/sec), which can capture fast system dynamics while bringing about the huger amount of data as compared with SCADA system (around 1 sample/ 5 secs) [8]. Owing to these merits, the deployment of PMUs makes it possible for real-time monitoring of the smart power grid [9]. In this sense, if every bus of the network is installed with a PMU, the voltage phasors for the entire system can be directly and fully acquired [10]. Yet, it is impossible for a real system with thousands of buses since the PMUs and their networking communication system are costly, which makes the current penetration of PMU is far from the desired level [7, 11].

The studies of SE with PMU measurements have already been extensively carried out, the approaches are either based on the combination of SCADA and PMU measurements [12-20] or purely PMU measurements [8, 21-23]. The former ones can be further divided into two categories: namely hybrid state estimator method and multi-stage method. The hybrid state estimator combines PMU measurements and conventional SCADA measurements by using a nonlinear transformation to connect the traditional state vector in polar form with the voltage phasors in the rectangular form [12-15]. It is proved that incorporation of PMU measurements can significantly improve the SE performance as compared to that with only SCADA measurements under steady-state condition. Nevertheless, the accuracy of those approaches can be compromised due to different time scales between PMUs and SCADA systems [16]. To address the issues of time scale inconsistency, the multi-stage approach is adopted by processing PMU measurements or SCADA measurements in independent stages. A Bar-Shalom-Campo data fusion technique is applied to combine the results of different PMU and SCADA stages in [17, 18]. ~~Different estimation approaches are investigated in the PMU stage such as Schweppe-type Huber Generalized Maximum likelihood (SHGM) [19] and the Least Absolute Value (LAV) [20].~~ All those methods based on the combination of SCADA and PMU measurements adopt various principles to enhance the robustness against gross errors. On the other hand, the purely PMU-based SE method has shown various benefits comparing with purely conventional SCADA based or PMU-SCADA based SE methods [22, 24, 25]: (1) the measurement function is linear as only current or voltage phasors are measured by PMUs, which gets rid of the computationally expensive iterative process in traditional SE with SCADA measurements. (2) no reference bus is needed since the voltage / current phasor phase angle can be directly acquired at the same time with

time-stamped from GPS system [26]. (3) real-time SE can be realized due to high sampling rates and low latency of PMUs.

Aforementioned SE methods are all based on Weighted Least Squares (WLS) or its variants. These methods are well-known non-robust, and a single outlier can severely alter the estimation results. To overcome this drawback, a separate post-estimation bad data processing function is needed to detect and eliminate gross errors [A1]. Therefore, robust SE methods are proposed to enhance the robustness. The least absolute value (LAV) estimator is an alternative technique. By minimizing the $L_1$ norm (rather than $L_2$ in WLS) of measurement residuals, the LAV estimator can be executed via linear programming (LP) solvers and it will detect and eliminate gross errors accordingly [A2]. The authors in [A3] formulated a robust LAV estimator by using PMUs to improve the computation performance of LAV estimator. A hybrid state estimator was proposed in [A4] with the coexistence of PMU measurements and SCADA measurements, to determine the states based on weighted LAV (WLAV). Besides, WLAV is also one of the most common robust estimators with high performance in the aspect of robustness [A5]. Other enhanced estimators for robustness include the method of least median of squares (LMS) and least trimmed squares (LTS), and they aim to restrain the effects of bad data by alternatively using measurements [A6-A7]. The LMS method is proposed depended on the notion that the median of a set of values could be more effective than mean in estimation process though it has got the particular disadvantage of rejecting several normal measurements along with the outliers. The LTS method calculates the sum of squared errors for smallest residuals only.

A main issue with the above-mentioned studies including WLS-based SE methods and robust SE methods is that these methods have a pre-assumption that the system is fully observable, and they cannot work effectively when the system is unobservable. However, in reality, many contingencies can result in system unobservable including measurement loss, line outage, failure of the data concentrator, or failure of the local communication system [27]. The network is unobservable if any state variables cannot be uniquely computed for a given set of measurements and network topology [28]. In unobservable networks, SE cannot be implemented expectedly via WLS-based estimators or robust estimators, hence the system operator cannot monitor any violations or events in these unobservable buses, this may lead to catastrophic outcomes. The traditional solution for unobservability is to use the pseudo-measurements to replace the missing measurement. Their calculation is based on external processes such as historical data, prediction procedures, load curve assessment [16] or derives from interpolated observations [29, 30]. Due to the poor temporal resolution of the pseudo-measurements, the accuracy of SE cannot be guaranteed and may not satisfy the requirement of SE. Besides, it is difficult to implement real-time SE as a result of the sparse data rate of pseudo-measurements with finite source data.

To overcome these issues, in this paper, a novel SE approach using a conditional generative adversarial network (GAN) is proposed. GAN is one of the most promising generative networks under deep learning framework and have attracted great interests in recent years, especially in computer vision research due to its excellent capability in generating realistic images [31] given a collection of indistinct or incomplete images. This inspires us to apply GAN in SE process, where the raw system measurements can be regarded as corrupted images, and the desired system states are corresponding to the real images that can be directly generated through a fine-tuned GAN. Compared to the standard GAN, we apply the conditional GAN (CGAN) to adapt to the SE problem appropriately. The proposed method uses Wasserstein distance rather than the Jensen-Shannon divergence proposed in [31], which can improve the accuracy by capturing all of the patterns in training set instead of a single pattern. The main contributions of this paper are as follows:

1) A novel data-driven, model-free approach for power system SE is proposed. By applying conditional generative adversarial networks, the actual correlations of system states can be well captured, and the system states can be accurately estimated without prior knowledge of the system model.
2) The PMU-based SE method can effectively restore all system states considering the corrupted raw measurements or even missing measurements under contingencies. Thus, the SE process can still be implemented even in an unobservable network. The influence of data contaminations is fully investigated with respect to different data contamination levels and types.
3) The experiment is carried out on a large system, i.e., 2746-bus Polish system. The simulation results validate the effectiveness of the proposed method, all estimated system states are close to true system states. To the best of the authors' knowledge, this is the first work using deep learning models to power system SE processes on a large-scale system.

The proposed method in this paper builds on the previous preliminary study [32]. This work presents a comprehensive study and the proposed method can handle different types and levels of contamination measurements. The case studies are supported by two large power systems, including the IEEE 118-bus system and the 2746-bus Polish system.

The remaining of this paper is organized as follows: Section 2 presents the problem formulation. The proposed method in SE using GAN model is elaborated in Section 3. Results of several case studies performed on the IEEE 118-bus system and a 2746-bus Polish system are presented and analysed in Section 4. Finally, Section 5 concludes the paper.

## 2. Problem Formulation

This section firstly introduces the problem formulation of power system state estimation with different input measurements. Then we will give the mathematical formulations for three state estimation scenarios with different measurements: 1) state estimation with raw measurements; 2) state estimation with corrupted measurements; and 3) state estimation with incomplete measurements.

### *2.1. Power System State Estimation Models*

Conventionally, in the SCADA system, the state variables are magnitude and phase angle of the bus voltage, and most commonly used measurements are the line power flows and bus power injections. As a result, the measurement function between the state variables and the measurements is nonlinear [33]. While for PMUs, the measured quantities are voltage or current phasors, leading to a linear relationship with the state variables. In this work, we assume that the system is observed merely by PMUs, i.e., all the obtained measurements are voltage phasors. The system states thus can be estimated via a linear measurement model instead of the conventional nonlinear measurement function.

Especially, for an $N$-bus power system, the vector of state variables $\mathbf{x}$ is denoted by $[V_1 \ldots, V_j \ldots, V_N, \theta_1 \ldots, \theta_j \ldots, \theta_N]^T$, $j = 1, \ldots, N$, $V_j$ and $\theta_j$ are the voltage magnitude and phase angle of $j$-th bus respectively. Hence, a static system model is formulated for the problem as:

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

where $\mathbf{y}$ represents measurement vector obtained by PMUs; $\mathbf{h}(\cdot)$ is the vector-valued measurement function established based on the state vector $\mathbf{x}$; $\mathbf{e}$ is the measurement error vector that is usually assumed to white noise composed by zero mean with a covariance matrix $\mathbf{R}$.

### 2.2. State Estimation with Raw Measurements

In the system with $M$ PMUs installed, we assume the whole system can be observable by deploying all these PMUs for all system buses, i.e., $M = N$. This corresponds to say that, the voltage phasors $y_j = V_j \angle \theta_j, j = 1, \ldots, N$, of all buses in the system, can be measured to form a raw measurement vector $\mathbf{y}$, which is denoted as $[y_1, \ldots y_j \ldots, y_N]^T$.

We then define $\tilde{\mathbf{x}}$ as the true system states, the training samples are obtained as the pairs of raw measurements and true system states $(\mathbf{y}, \tilde{\mathbf{x}})$. In this model, the inputs are raw measurements $\mathbf{y}$, each of that is labelled with an assigned label (true system states $\tilde{\mathbf{x}}$) as a condition. Once a large number of $(\mathbf{y}, \tilde{\mathbf{x}})$ pairs are collected, the goal is to train a GAN network to generate the estimated system states $\hat{\mathbf{x}}$ that are desired to be close to the true system states $\tilde{\mathbf{x}}$ as much as possible.

### 2.3. State Estimation with Corrupted Measurements

In practice, there is a wide range of factors that might lead to measurement corruption, such as impulsive communication noise, the failures of instruments, cyber-attacks, etc. The developed SE model with high robustness to these potential contaminations is more favoured by decision-makers. In this model, the number of corrupted measurements is assumed as $n_C$, and the vector of corrupt system measurements $\mathbf{y_C}$ is expressed as $[y_1 \ldots y_{n_C} \ldots y_N]^T$.

Similarly, a robust SE model is expected to detect the buses with corrupted measurements and generate the system states that are close to true system operating condition. Through learning with the samples composed of a collection of $(\mathbf{y_C}, \tilde{\mathbf{x}})$ pairs, the underlying SE model can be well trained even when partial measurements are contaminated. As a consequence, once the system measurement is obtained by PMUs, the proposed fully data-driven SE model can render true system states as much as possible irrespective of the contamination type (such as malicious attacks) or degree.

### 2.4. State Estimation with Incomplete Measurements

Incomplete measurement is another common situation faced by system operators. This might attribute to the insufficient PMUs installed due to the high cost of PMUs. Also, the unavailability of getting voltage or current measurements from potential transformers (PTs) or current transformers (CTs), failure of phasor data concentrator (PDC) and failure of a local communication system [27] may all lead to the measurement loss with different levels. Especially, the system is more likely to be unobservable without enough redundancy when the aforementioned situations occur.

In this study, we substitute the missing phasor measurement $y_j$, either resulting from uninstallation of PMUs or measurement loss, with zero. Thus, the vector of incomplete measurements $\mathbf{y_L}$ can be expressed as $[y_1 \ldots 0 \ldots y_N]^T$. $\mathbf{y_L}$ is combined with the corresponding true system state $\tilde{\mathbf{x}}$ to yield the training samples $(\mathbf{y}_L, \tilde{\mathbf{x}})$. Likewise, the generated system states $\hat{\mathbf{x}}$ should be close to the true system states $\tilde{\mathbf{x}}$ to ensure the high model robustness and accuracy.

## 3. The Proposed Method

The proposed fully data-driven methodology based on Wasserstein GAN is presented in this section. Firstly, the basic theory of GAN [34] will be reviewed, in the meanwhile, how the framework of Wasserstein GAN fit into the SE problem is discussed. Later, we will discuss the model establishment process for SE by integrating the synthetic PMU measurement.

### 3.1. Wasserstein GAN

As defined before, $\tilde{\mathbf{x}} = [\tilde{x}_1 \ldots \tilde{x}_N]^T$ represents the true system state, $\{\tilde{x}_j^{(i)}\}_{i=1}^m$ denotes the $i$-th sample of true system state and $m$ is the number of samples. Let $p_{data}(x)$ denote the distribution of true system state. Suppose a group of noise inputs $z$ follow a known distribution $z \sim p_z(z)$, e.g., uniform distribution or jointly Gaussian. The goal of the method is to transform the sample $z$ from the distribution $p_z(z)$ so that it can follow true system state distribution $p_{data}(x)$. To this end, two deep neural networks are trained simultaneously. One is the generator network $G$ expressed as $G(z; \theta^{(G)})$, who is parametrized by $\theta^{(G)}$; the other is the discriminator network $D$ written as $D(x; \theta^{(D)})$, whose function is parametrized by $\theta^{(D)}$. The generator and discriminator are combined to form the GAN network.

*Generator:* When training the generator, a large number of up-sampling operations are implemented to the inputs $z$, the generator outputs are the estimated system

states. The training procedure can be expressed as the following mapping:

$$G(z;\theta^{(G)}): z \to p_G(z) \tag{2}$$

where $p_G(z)$ is the generated distribution which is the estimated system state should be sampled from and also follows the true system state distribution $p_{data}(x)$.

*Discriminator:* The discriminator should be trained with the generator at the same time. Both samples from the generated distribution $p_G(z)$ and the true system state distribution $p_{data}(x)$ are served as discriminator inputs. After plenty of down-sampling operations, the output is a value $p_{real}$ which is continuous and reflects what extent these inputs belong to the true system state distribution $p_{data}(x)$. Likewise, the training process of discriminator can be expressed as a mapping:

$$D(x;\theta^{(D)}): x \to p_{real} \tag{3}$$

where $x$ is the input vector that can be sampled either from $p_G(z)$ or $p_{data}(x)$. The discriminator is expected to learn to distinguish between $p_G(z)$ and $p_{data}(x)$, and to maximize the difference between these two distributions.

In the training stage, $D$ is trained to maximize its capacity of discernment between true system state distribution and estimated state distribution from the generator. We simultaneously train $G$ and $D$ to minimize the difference between these two distributions. The weights of $G$ and $D$ are updated to minimize generator loss function $C_G$ and discriminator loss function $C_D$, respectively. Specifically, a batch of samples collected from distribution $p_z(z)$ are fed into $G$. Meanwhile, a batch of true samples drawn from distribution $p_{data}(x)$ are fed into $D$. A small $C_G$ indicates that the generated samples are more realistic from the discriminator's view. That is, for this work, the generated system states are more similar to the true system states. On the other hand, a small $C_D$ reflects the $D$ does well in distinguishing the discrepancy between the generated system states and the true system states. Also, It indicates that there is a large difference between generated state distribution $p_G(z)$ and true state distribution $p_{data}(x)$. Generator loss function $C_G$ and discriminator loss function $C_D$ can be expressed as [35]:

$$C_G = \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{4}$$

$$C_D = -\mathbb{E}_{x\sim p_{data}(x)}[\log(D(x))] - \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{5}$$

For a given $D$, as a large output value of discriminator $p_{real}$ shows the generated samples are more realistic, the generator should seek to minimize $\log(1-D(G(z)))$ by altering $G$ to generate more realistic samples, which gives the loss function of a generator in (4). For a given $G$, as seen in (5), the discriminator attempts to minimize $-\log(D(x))$, thus giving a large discriminator output value $p_{real}$ In the meanwhile, the network minimizes $-\log(1-D(G(z)))$, which is virtually a reverse of $C_G$. Note that, Eq. (4) equals to minimize $-\log(D(G(z)))$. Hence, the GAN can be formulated by combining these two loss functions as a two-player minimax game with value function $V(G,D)$:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[\log(D(x))] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{6}$$

where $V(G,D)$ is the negative of $C_D$.

At the beginning of training, the performance of generator $G$ is poor and the system state samples generated by $G$ are very different from samples of $p_{data}(x)$, consequently, the discriminator outputs a small value of $p_{real}$ and reject these 'fake' samples with high confidence. Under these circumstances, $C_G$ is small, $C_D$ is large and $V(G,D)$ is also large. As the training goes on, the generator learns to produce more realistic samples and the discriminator learns to distinguish these samples from two different distributions. Finally, $G$ defeats $D$, i.e., the samples generated by $G$ are almost as real as true samples, also, $D$ fails to distinguish samples from $p_G(z)$ and $p_{data}(x)$.

According to the Kantorovich-Rubinstein duality [36], The Wasserstein distance (Earth-Mover distance) is the dual of the minimax objective in (6). $x$ and $y$ are two random variables and $\prod(\mathbb{P}_r,\mathbb{P}_g)$ is the set of all joint distributions $\gamma(x,y)$ whose marginals are $\mathbb{P}_r$ and $\mathbb{P}_g$, respectively. Then the Wasserstein distance between $x$ and $y$ is defined as:

$$W(\mathbb{P}_r,\mathbb{P}_g) = \inf_{\gamma\in\prod(\mathbb{P}_r,\mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \tag{7}$$

the Wasserstein distance can be viewed with the "cost" of the optimal plan that moves all the "mass" $\prod(\mathbb{P}_r,\mathbb{P}_g)$ from location $x$ to location $y$ in order to transform the distribution $\mathbb{P}_r$ into the distribution $\mathbb{P}_g$. $\gamma(x,y)$ can be described as the quantity of the moved "mass" at one time.

The objective of GAN is to make the generated sample distribution $p_z(D(G(z)))$ close to the true system states distribution $p_{data}(D(x))$. Thus, the Wasserstein distance between the true system state and the generated sample can be expressed as:

$$W(D(x),D(G(z))) = \sup_D \mathbb{E}_{x\sim p_{data}(x)}[D(x)] - \mathbb{E}_{z\sim p_z(z)}[D(G(z))] \tag{8}$$

When the Wasserstein distance converges, the optimal plan of moving "mass" is found, and the optimal generator $G^*$ is also found. As reported in the literature [35], the JS divergence applied in original GAN cannot reflect what extent two distribution $p_G(z)$ and $p_{data}(x)$ are close when they are very different from each other, which makes GAN sensitive to the parameters. As a result, the generated system states almost follow the pattern with the highest occurring probability regardless of the inputs. However, applying Wasserstein distance as the loss function of GAN [37] successfully address these limits and gives the accurate distance between two distributions. Therefore, the generator of WGAN can mimic the true system operating scenarios to generate diversified system states rather than the same ones produced by original GAN.

### 3.2. Conditional GAN

The standard GAN model uses merely the noise vector as input and has no extra limitations for the generated output. The standard GAN can be extended to a conditional counterpart where both the generator and discriminator are

conditioned on some extra information [37]. In conditional GAN (CGAN), the generated samples should satisfy this condition $y$.

This architecture is more fit to SE problems where the raw system measurement can be regarded as condition $y$, and the generated system states should be guaranteed to be as close as possible to the true system states while satisfying the corresponding raw measurement. The CGAN is implemented by feeding $y$ into both the generator and discriminator as additional inputs. Eventually, Eq. (8) can be rewritten as:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[D(x|y)] - \mathbb{E}_{z\sim p_z(z)}[D(G(z|y))] \tag{9}$$

Fig. 1 illustrates the CGAN architecture for SE and the algorithm used in the proposed method is described in Algorithm 1.

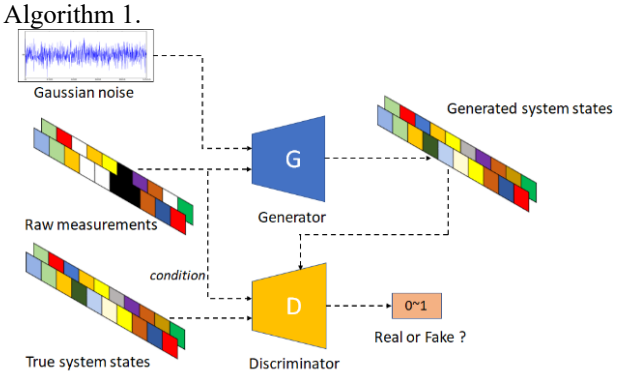



***Fig. 1.*** *The architecture of CGAN that we use, including the input and output of the generator and discriminator, respectively.*

In algorithm 1, $G(z)$ and $D(x)$ are neural networks with parameter $\theta^{(G)}$ and $\theta^{(D)}$, respectively. Both networks consist of multilayer perceptron (MLP), convolution, normalization, max-pooling and Rectified Linear Units (ReLU). The parameters are tuned within several training batches. The training algorithms for discriminator and generator are slightly different, where the former is based on gradient ascend and the latter is gradient descend. Besides, Root Mean Square Propagation (*RMSProp*) algorithm is applied in both generator and discriminator to allow the learning rate to be self-adjustable. *RMSProp* is a method in which the learning rate is adapted for each of the parameters. The idea is to divide the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight [A8]. It should be noted that weight clipping is applied in discriminator training to meet specific conditions and avoid gradient explosion [35]. The model setting for CGAN will be presented in the next section.

**Algorithm 1** CGAN with Wasserstein Distance for SE

**Require:** : $\alpha$, the learning rate; $c$, the clipping parameter; $m$, the batch size; $k_{dis}$, the number of iterations of the discriminator per generator iteration.

**Require:** : $\theta_0^{(D)}$, initial discriminator's parameters; $\theta_0^{(G)}$, initial generator's parameters.

**while** $\theta_0^{(G)}$ has not converged **do**

**for** $t = 0,\ldots,k_{dis}$ **do**

- Sample batch of $m$ noise sample $\{(z^{(i)}, y^{(i)})\}_{i=1}^{m}$ from noise prior distribution $p_z(z)$.
- Sample batch of $m$ examples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ from the true system state data $p_{data}(x)$
- Update the discriminator by ascending its gradient:

$$g_{\theta^{(D)}} \leftarrow \nabla_{\theta^{(D)}} \frac{1}{m}\sum_{i=1}^{m}[D(x^{(i)}|y^{(i)}) - D(G(z^{(i)}|y^{(i)}))]$$

$$\theta^{(D)} \leftarrow \theta^{(D)} + \alpha \cdot RMSProp(\theta^{(D)}, g_{\theta^{(D)}})$$

$$\theta^{(D)} \leftarrow clip(\theta^{(D)}, -c, c)$$

**end for**

- Sample batch of $m$ noise samples $\{(z^{(i)}, y^{(i)})\}_{i=1}^{m}$ from noise prior distribution $p_z(z)$.
- Update the generator by descending its gradient:

$$g_{\theta^{(G)}} \leftarrow -\nabla_{\theta^{(G)}} \frac{1}{m}\sum_{i=1}^{m} D(G(z^{(i)}|y^{(i)}))$$

$$\theta^{(G)} \leftarrow \theta^{(G)} - \alpha \cdot RMSProp(\theta^{(G)}, g_{\theta^{(G)}})$$

**end while**

## 4. Numerical Results

To validate the effectiveness of the proposed SE method the experiment is carried out on two power systems, namely IEEE-118 bus system and the 2746-bus Polish network, respectively. The system data are simulated from MATLAB and MATPOWER toolbox by implementing the Monte Carlo power flow calculations [38]. To obtain distinct system states, we assume the system load satisfies the Gaussian distribution with zero mean and standard deviation of 0.1. Load samples are then drawn and fed into power flow computations to derive true system states. For IEEE-118 bus system and a 2746-bus Polish network, we have 15,000 training examples, respectively. 80% of these samples are used for training and the remaining 20% are used for testing. The batch size is 32, and the number of epochs is 300, therefore, the training of the networks took 112,500 iterations (15000 * 80% / 32 * 300) for each scenario. All the program for the conditional GAN-based SE (CGAN-SE) model is implemented using 'TensorFlow' in Python on PyCharm IDE with NVIDIA GeForce RTX 2080 Ti GPU and its RAM is 11GB [39].

### *4.1. Model Architecture and Training Details*

The generator $G$ consists of 2 fully connected multilayer perceptron (MLP) and 3 de-convolutional layers. The first 2 MLPs are used for up-sampling, and the de-convolutional layers kernel size is $1\times5$ and strides size 2 are used to up-sample the input noise $z$. While the discriminator

*D* has a reversed architecture, whose3 convolutional layers are all with a kernel size of $1\times5$ and stride size of 2. Table I lists the detailed settings of our GAN model on the IEEE 118-bus system.

**Table 1** Proposed GAN model structure

| | Generator *G* | Discriminator *D* |
|---|---|---|
| Input | 100 | 2 * 118 |
| Layer 1 | MLP, 1024 | Conv, 64 |
| Layer 2 | MLP, 512 | Conv, 256 |
| Layer 3 | Conv_transpose, 512 | Conv, 512 |
| Layer 4 | Conv_transpose, 256 | MLP, 1024 |
| Layer 5 | Conv_transpose, 64 | |

The input data are normalized to [-1, 1] to match the output of tanh activation in the last layer of the generator *G*. The models are trained by *RMSProp* optimizer. Random initializations of neuron weights follow a standard normal distribution with zero mean and standard deviation of 0.02. Batch normalization is employed before each layer excluding the input layer to stabilize the inputs to nonlinear activation functions. To be specific, it normalizes each layer's inputs by using zero mean and unit variance. Leaky-ReLU activation is used in the discriminator and ReLU activation is used in the generator excluding the output layer. In this paper, the discriminator *D* is trained for four times and the generator *G* is trained once. Thus, $k_{dis}$ is set as 4 in algorithm 1.

### *4.2. Data Generation*

As mentioned in the early part, we generated the true system states $\tilde{\mathbf{x}}$ via Monte Carlo power flow calculations with different load scenarios. Then, the raw measurements are created by adding a Gaussian noise **e** with zero mean and standard deviation of 0.001 (PMU's precision) to $\tilde{\mathbf{x}}$ . The next section will discuss the fabrication of abnormal measurements considering 3 contamination scenarios, in each of which different contamination level *r*% ranging from 0% to 100% will be considered.

*4.2.1 Corrupted Measurement Data:* Corrupted measurement data refers to the measurements that significantly differ from the normal measurement data. Corrupted voltage magnitude and phase angle measurements are generated by randomly choosing *r*% raw measurements and adding an error with 0.5 mean and 0.05 standard deviation. The rest 1 - *r*% are still raw measurement with only PMU measurement noise.

*4.2.2 Incomplete Measurement Data:* Incomplete measurement data are generating by randomly choosing *r*% raw measurement and setting their voltage magnitude and voltage phase angle with zeros.

*4.2.3 Mixed Measurement Data:* A more common event occurred in practice might be that the collected database is contaminated with a mixture of bad data and missing data. To fabricate this situation, we equally generate the contaminated data for each type. i.e., the mixed contamination data accounting for *r* % of the dataset contains (*r*% / 2) corrupted data and (*r*% / 2) missing data.

### *4.3. Performance Evaluation*

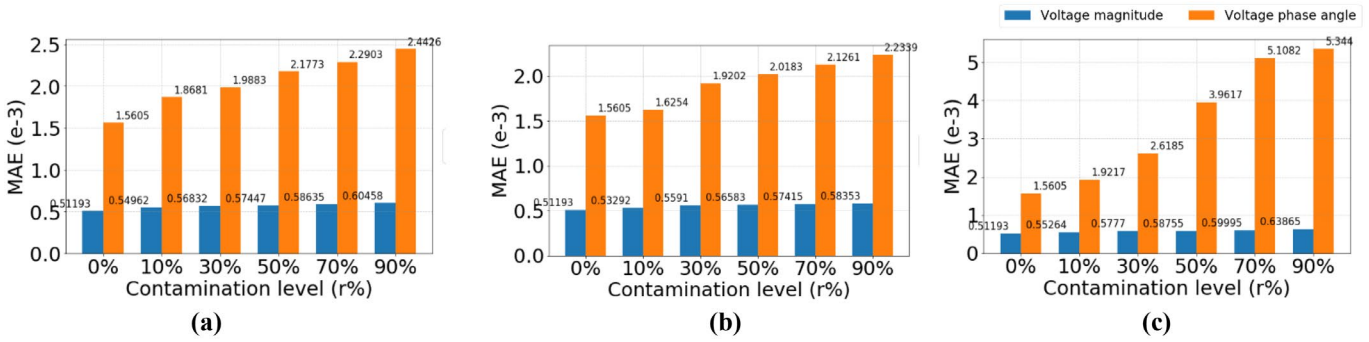
The estimated performance is evaluated by mean average error (MAE) for total error [27]:

$$\mathrm{MAE}=\frac{1}{N*m}\sum_{j=1}^{N}\sum_{i=1}^{m}\left|\hat{x}_j^{(i)}-\tilde{x}_j^{(i)}\right| \tag{10}$$

where $N$ is the number of buses, $m$ is the number of samples, $\hat{x}_j^{(i)}$ and $\tilde{x}_j^{(i)}$ refer to estimated system states and true system states corresponding to *i*-th sample and *j*-th bus, respectively.

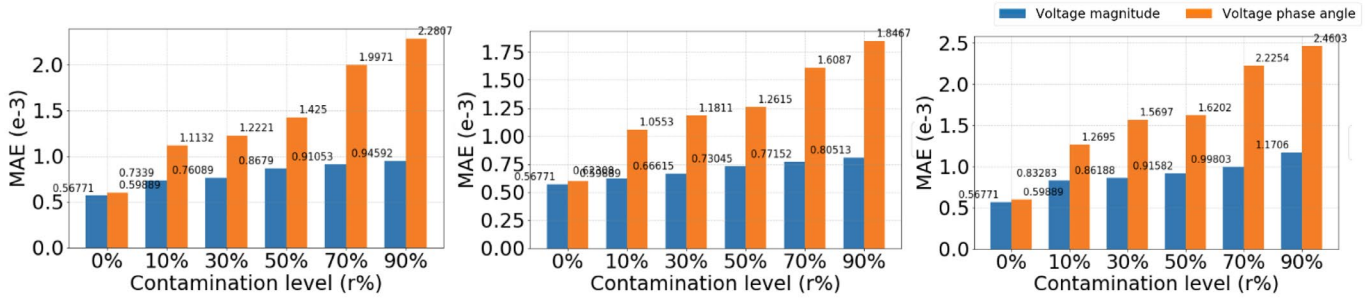
*4.3.1 Overall SE accuracy under different contamination levels and types:* Different contamination levels *r*% ranging from 10% to 90% with 10% increment is examined in this section to give the full-scale analysis of the model accuracy and robustness. The case with raw measurements (*r*% = 0%) is also tested. Additionally, to ensure the experiment is unbiased, the verified model is simulated 10 times for each result. The MAE of the proposed method for both voltage magnitude and phase angle under each contamination scenario, are illustrated in Fig. 2 (IEEE-118 bus system) and Fig. 3 (2746-bus Polish network).Their average value of 10 runs is represented by the blue/orange bars, respectively.

From the results shown in Fig. 2, the MAE of voltage angle is higher than the voltage magnitude under all three contamination scenarios. Especially in Fig. 2 (c), with mixed contamination measurement, the MAE of phase angle is significantly greater than the voltage magnitude. In the scenario with only raw measurement involved (*r*%=0%), the MAE of voltage magnitude and phase angle are 5.1193e-4 and 1.5605e-3, respectively. With the increase of contamination level from 0% to 90%, the MAE of both voltage magnitude and phase angle grows accordingly. Additionally, the MAE of phase angle has an obvious rising trend while that of voltage magnitude ascends slowly. The MAE under mixed contamination situations is greater than that of the other two cases. The MAE with incomplete measurement is slightly smaller as compared to that with corrupted measurement. The tendency of model performance for 2746-bus Polish system under all verified contamination levels is similar to that of 118-bus system, as observed in Fig. 3. Yet, the MAE of voltage magnitude is larger than that in the 118-bus system whereas the error of phase angle is smaller




(a) (b) (c)

***Fig. 2.*** *MAE (e-3) of CGAN-SE on IEEE 118-bus system over 10 runs with respect to various contamination levels of under three measurement contamination scenarios.*
**(a)** Corrupted Measurement, **(b)** Incomplete Measurement, **(c)** Mixed contamination Measurement



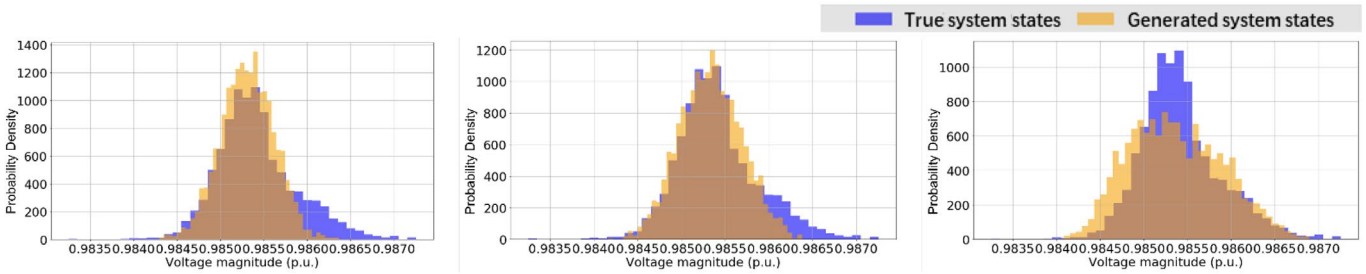

(a) (b) (c)

***Fig. 3.*** *MAE (e-3) of CGAN-SE on 2746-bus Polish system over 10 runs with respect to various contamination levels of under three measurement contamination scenarios.*
**(a)** Corrupted Measurement, **(b)** Incomplete Measurement, **(c)** Mixed contamination Measurement

*4.3.2 Estimated Distribution Assessment:* To investigate the similarity of the distribution between the generated system states and true system states, we randomly choose two PQ-buses to compare their probability density distribution profiles. Fig. 4 and Fig. 5 depict the probability density histograms of generated and true system states at bus 30 (118-bus system) under 30% contamination level, respectively. Also, Fig. 6 and Fig. 7 depict the probability density histograms of generated and true system states at bus 245 (2746-bus system) under 30% contamination level, respectively. The probability in these figures are represented by the individual rectangle areas multiplied by the width of the interval and the Y-axis value and the cumulative rectangle areas are equal to 1. The distribution of voltage magnitude is closer to its true distribution than that of phase angle under three contamination scenarios in both tested systems. On the other hand, CGAN-SE is more effective to handle either incomplete or corrupted measurement than mixed contamination measurement, as the discrepancy of distribution profiles for the former two scenarios is less evident than that for the mixed contamination measurement case. To quantify the similarity between the generated system states and true system states, we calculate the Wasserstein distance between two distributions and the result shows in table 2. The Wasserstein distance is a natural way to compare the probability distributions, and the smaller value means the two distributions are similar. From table 2, the smallest Wasserstein distance is voltage magnitude on IEEE 118-bus system with corrupted measurement and the largest one is voltage phase angle on IEEE 118-bus system with corrupted measurement. Besides, the distance of voltage magnitude is larger than that of voltage phase angle on the same system.

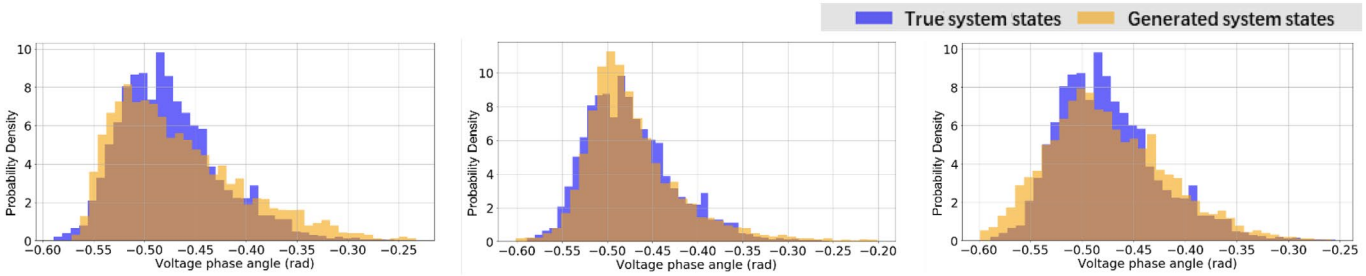
**Table 2** Wasserstein distance with different contamination measurement (30% contamination level) on IEEE 118-bus system and 2746-bus Polish system.

| | Contamination Type (30% level) | | |
|---|---|---|---|
| | Corrupted | Incomplete | Mixed |
| IEEE 118-bus system (voltage magnitude) | 1.4903 | 1.5808 | 2.0910 |
| IEEE 118-bus system (voltage phase angle) | 6.1195 | 6.8061 | 8.7550 |
| 2746-bus Polish system (voltage magnitude) | 2.0592 | 1.9056 | 2.4831 |
| 2746-bus Polish system (voltage phase angle) | 3.9170 | 4.1434 | 5.4669 |



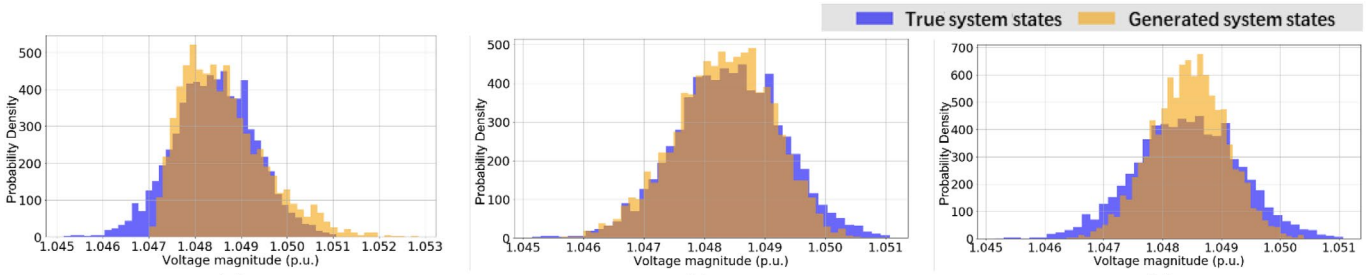

**(a)** **(b)** **(c)**

***Fig. 4.*** *Probability density histograms of the generated system states and true system states with different contamination measurement (30% contamination level) at bus 30 of the IEEE 118-bus system (voltage magnitude).*
**(a), (b), (c)** Probability density distribution of voltage magnitude with 30% level of corrupted measurement, incomplete measurement and mixed contamination measurement, respectively.



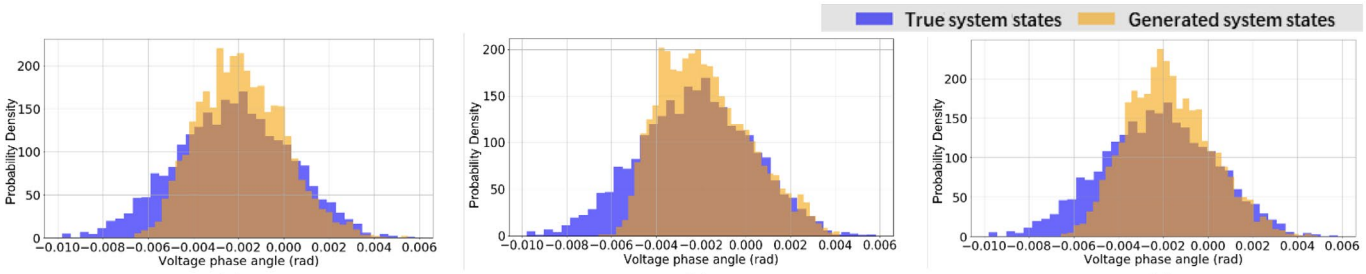

**(a)** **(b)** **(c)**

***Fig. 5.*** *Probability density histograms of the generated system states and true system states with different contamination measurement (30% contamination level) at bus 30 of the IEEE 118-bus system (voltage phase angle).*
**(a), (b), (c)** Probability density distribution of phase angle with 30% level of corrupted measurement, incomplete measurement and mixed contamination measurement, respectively.




**(a)** **(b)** **(c)**

***Fig. 6.*** *Probability density histograms of the generated system states and true system states with different contamination measurement (30% contamination level) at bus 245 of 2746-bus Polish system (voltage magnitude).*
**(a), (b), (c)** Probability density distribution of voltage magnitude with 30% level of corrupted measurement, incomplete measurement, and mixed contamination measurement, respectively.




**(a)** **(b)** **(c)**

***Fig. 7.*** *Probability density histograms of the generated system states and true system states with different contamination measurement (30% contamination level) at bus 245 of 2746-bus Polish system (voltage phase angle).*
**(a), (b), (c)** Probability density distribution of phase angle with 30% level of corrupted measurement, incomplete measurement, and mixed contamination measurement, respectively.

*4.3.3 Spatial Correlation Assessment:* To further validate the quality of generated system states, we study the correlation of buses for voltage magnitude and phase angle, respectively. To exhibit the evolving process of correlation during training, the Pearson correlation coefficient matrix is computed at several training iterations, i.e., 200, 2000 and

20000, under 30% mixed contamination scenario. For the voltage magnitude of 118-bus system, as shown in Fig. 8(a) both generated voltage magnitude and true voltage magnitude show weak spatial correlations as the correlation coefficients tend to be zeros. Thus, CGAN-SE can yield the correlations that almost similar to the trues. This is also confirmed in Fig. 8 (c) on the 2746-bus system. By contrast, the spatial correlation of the phase angle between buses is stronger than that of voltage magnitude. At the beginning of the training, though the correlation profile of phase angles is far from the true ones, with the learning carries on, it can learn the spatial interdependency and finally gives a better result. Additionally, the dark cross line around bus 70 in both generated phase angles and true phase angles means this bus has no spatial correlation with all of the other buses, which represents the phase angle of this bus is a constant. This conforms to the system model that bus 69 is a slack bus. Therefore, CGAN-SE has the capability of learning the spatial correlation of voltage magnitude and phase angle between buses.

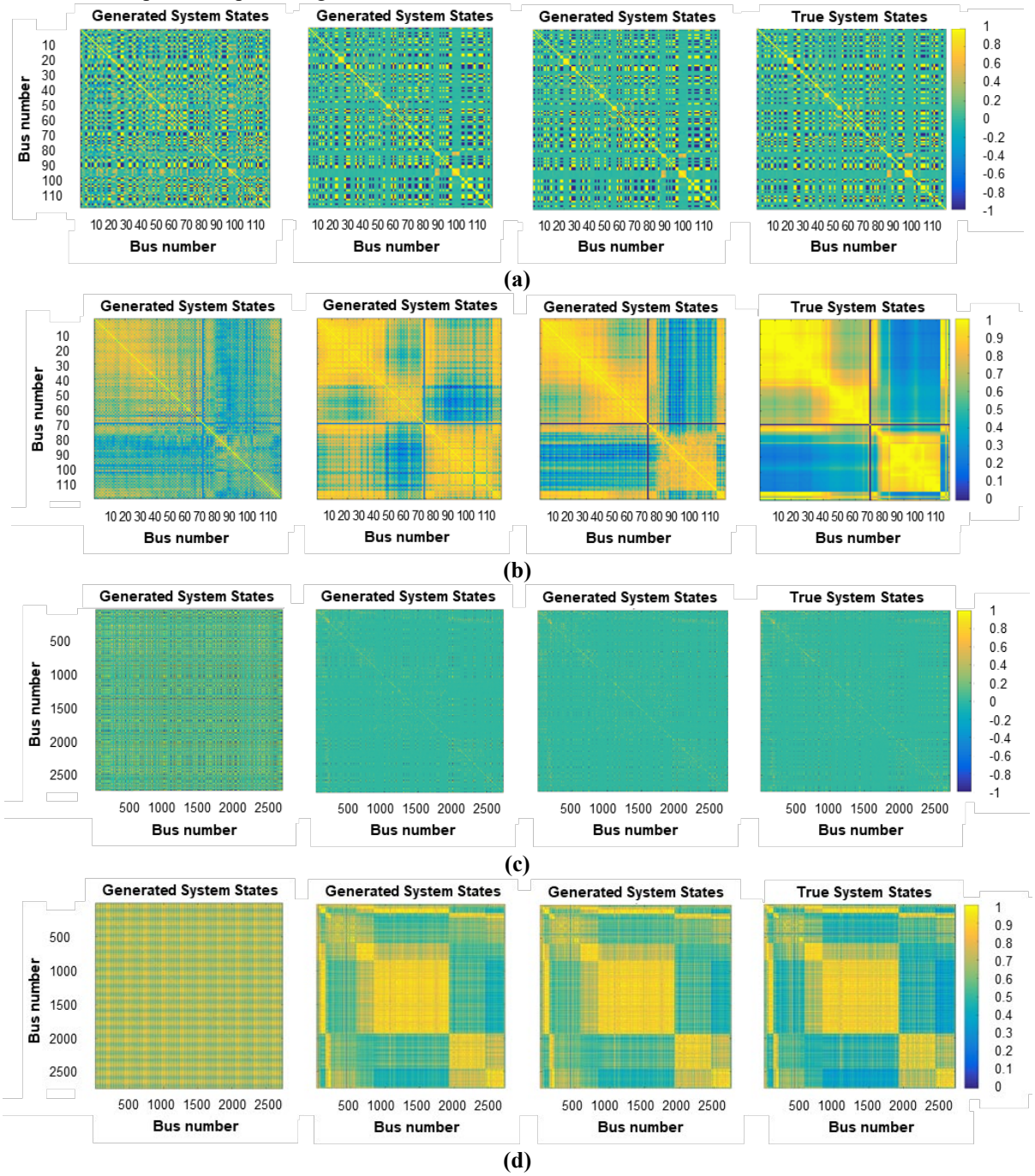



***Fig. 8.*** *The spatial correlation coefficients matrix colourmap for during different training iterations. From left to right: 200 iterations, 2000 iterations, 20000iterations, true system states. All results are tested with 30% contamination level of mixed contamination measurement (the right colour bar is the correlation coefficient).*
**(a)** and **(b)** are the voltage magnitude and phase angle spatial correlation on IEEE 118-bus system, respectively; **(c)** and **(d)** are the voltage magnitude and phase angle spatial correlation on 2746-bus Polish system, respectively.

### 4.4. Accuracy Comparison

As discussed before, only PMU measurements are considered in the network and the measurement model is linear. Thus, the proposed method is compared with a linear WLS state estimator (LWLS-SE) [25]. To investigate the robustness of CGAN-SE, we also implement the WLAV state estimator (WLAV-SE) by minimizing the $L_1$ norm between true system states and estimated states [A9].

For the unobservable scenarios, it can be occurred with incomplete measurements from level 10% to 90%. Since the process of the incomplete measurement data generation is stochastic, and we randomly choose r% raw measurements which are set their voltage magnitude and voltage phase angle with zeros, the system can be unobservable with each level of the incomplete measurement as well as the mixed measurement. For example, if 11 measurements on IEE 118-bus system are lost in an adjacent area, then part of buses will not be monitored, and the system will be unobservable. Besides, with the increased level of the incomplete measurement and the mixed measurement, the more unobservable scenarios are occurred.

Thus, in this section, we compare their MAE of voltage magnitude and phase angle on the IEEE 118-bus system and 2746-bus Polish system, with three types of contamination measurement from level 0% to level 90%.

We use the same measurement data and apply LWLS-SE and WLAV-SE to estimate voltage magnitude and phase angle. Besides, a deep convolutional neural network-based SE (DCNN-SE) method is applied to investigate the advantages of CGAN-SE over other neural networks with deep learning [40]. DCNN is a generator network, which is trained by the same dataset and verified via the same contamination level from 0% to 90%.

The comparison results of LWLS-SE, WLAV-SE, DCNN-SE and CGAN-SE on the IEEE 118-bus system and 2746-bus Polish system are shown in Table 3 and Table 4, respectively. For the scenario of noise only ($r$% = 0), the MAE of both voltage magnitude and phase angle for LWS-SE and WLAV-SE method has the PMU measuring precision close to 0.001. In the cases with corrupted measurements in both two test systems, DCNN-SE achieves significant improvements over the traditional LWLS-SE and WLAV-SE, but inferior to the proposed CGAN-SE method.

In the context of incomplete and mixed contamination measurements, the proposed CGAN-SE consistently outperforms LWLS-SE, WLAV-SE, and DCNN-SE. With increased contamination levels, the MAE of LWLS-SE and WLAV-SE increase dramatically. The reason is that more unobservable scenarios are occurred with the higher incomplete measurement or mixed measurement, especially with the incomplete measurement where the result has the largest MAE. On the other hand, CGAN-SE perform well because it is data-driven and model-free approach. By applying conditional generative adversarial networks, the actual correlations of system states can be well captured, and the system states can be accurately estimated without prior knowledge of the system model. Therefore, even the system is unobservable, CGAN-SE can still estimate system states with small error.

Especially for robust WLAV-SE on the IEEE 118-bus system, the smallest MAE is 9.032e-03 for voltage magnitude and 1.115e-02 for voltage phase angle with 10% corrupted, while its MAE are larger with the incomplete measurement or the higher level corrupted measurement. In these cases, the system operator cannot monitor any violations or events in large biases buses, this may lead to catastrophic outcomes.

**Table 3** Performance comparison with benchmarks on the IEEE 118-bus system.

| | Contamination Type | Contamination Level $r$% | IEEE 118-bus System | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Voltage Magnitude (p.u.) | | | | Phase Angle (rad) | | | |
| | | | LWLS-SE | WLAV-SE | DCNN-SE | CGAN-SE | LWLS-SE | WLAV-SE | DCNN-SE | CGAN-SE |
| MAE | Noise only | 0% | 1.030e-03 | 1.152e-03 | 6.031e-04 | **5.119e-04** | **9.910e-04** | 1.226e-03 | 1.984e-03 | 1.561e-03 |
| | Corrupted | 10% | 5.086e-02 | 9.032e-03 | 8.004e-04 | **5.496e-04** | 6.203e-02 | 1.115e-02 | 4.972e-03 | **1.868e-03** |
| | | 30% | 1.308e-01 | 4.801e-02 | 8.092e-04 | **5.683e-04** | 1.427e-01 | 5.207e-02 | 5.669e-03 | **1.988e-03** |
| | | 50% | 2.205e-01 | 1.104e-01 | 8.102e-04 | **5.745e-04** | 2.282e-01 | 1.331e-01 | 6.242e-03 | **2.177e-03** |
| | | 70% | 3.103e-01 | 2.768e-01 | 8.185e-04 | **5.864e-04** | 3.265e-01 | 2.959e-01 | 6.945e-03 | **2.290e-03** |
| | | 90% | 4.301e-01 | 4.437e-01 | 8.289e-04 | **6.046e-04** | 4.432e-01 | 5.520e-01 | 8.079e-03 | **2.443e-03** |
| | Incomplete | 10% | 9.796e-02 | 1.608e-02 | 8.031e-04 | **5.329e-04** | 1.301e-01 | 1.896e-02 | 4.993e-03 | **1.625e-03** |
| | | 30% | 2.761e-01 | 9.353e-02 | 8.234e-04 | **5.591e-04** | 2.922e-01 | 9.700e-02 | 6.912e-03 | **1.920e-03** |
| | | 50% | 4.631e-01 | 2.097e-01 | 8.281e-04 | **5.658e-04** | 4.750e-01 | 2.277e-01 | 8.187e-03 | **2.018e-03** |
| | | 70% | 5.989e-01 | 5.572e-01 | 8.324e-04 | **5.742e-04** | 6.139e-01 | 5.781e-01 | 9.952e-03 | **2.126e-03** |
| | | 90% | 8.863e-01 | 8.958e-01 | 8.331e-04 | **5.835e-04** | 8.930e-01 | 8.894e-01 | 1.486e-02 | **2.234e-03** |
| | Mixed | 10% | 8.036e-02 | 1.338e-02 | 8.245e-04 | **5.526e-04** | 9.029e-02 | 1.537e-02 | 1.163e-02 | **1.922e-03** |
| | | 30% | 2.072e-01 | 7.824e-02 | 8.432e-04 | **5.777e-04** | 2.220e-01 | 7.981e-02 | 1.823e-02 | **2.619e-03** |
| | | 50% | 3.299e-01 | 1.940e-01 | 8.795e-04 | **5.876e-04** | 3.387e-01 | 2.253e-01 | 2.916e-02 | **3.962e-03** |
| | | 70% | 4.615e-01 | 4.309e-01 | 8.978e-04 | **6.000e-04** | 4.723e-01 | 5.435e-01 | 4.450e-02 | **5.108e-03** |
| | | 90% | 6.663e-01 | 6.813e-01 | 9.111e-04 | **6.387e-04** | 6.798e-01 | 6.993e-01 | 6.770e-02 | **5.344e-03** |

**Table 4** Performance comparison with benchmarks on 2746-bus Polish system.

| | Contamination Type | Contamination Level *r*% | 2746-bus Polish System | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Voltage Magnitude (p.u.) | | | | Phase Angle (rad) | | | |
| | | | LWLS-SE | WLAV-SE | DCNN-SE | CGAN-SE | LWLS-SE | WLAV-SE | DCNN-SE | CGAN-SE |
| MAE | Noise only | 0% | 1.020e-03 | 1.290e-03 | 8.343e-04 | **6.677e-04** | 1.070e-03 | 1.238e-03 | 4.900e-03 | **5.989e-04** |
| | Corrupted | 10% | 5.487e-02 | 9.812e-03 | 1.305e-03 | **7.339e-04** | 6.385e-02 | 1.176e-02 | 1.228e-02 | **1.113e-03** |
| | | 30% | 1.404e-01 | 5.191e-02 | 1.320e-03 | **7.609e-04** | 1.503e-01 | 5.479e-02 | 1.402e-02 | **1.222e-03** |
| | | 50% | 2.366e-01 | 1.263e-01 | 1.339e-03 | **8.679e-04** | 2.420e-01 | 1.418e-01 | 1.542e-02 | **1.425e-03** |
| | | 70% | 3.328e-01 | 2.909e-01 | 1.355e-03 | **9.105e-04** | 3.473e-01 | 3.248e-01 | 1.715e-02 | **1.997e-03** |
| | | 90% | 4.612e-01 | 4.827e-01 | 1.409e-03 | **9.459e-04** | 4.723e-01 | 4.944e-01 | 1.996e-02 | **2.281e-03** |
| | Incomplete | 10% | 1.013e-01 | 1.515e-02 | 1.334e-03 | **6.231e-04** | 1.336e-01 | 1.634e-02 | 1.533e-02 | **1.055e-03** |
| | | 30% | 2.819e-01 | 9.428e-02 | 1.368e-03 | **6.662e-04** | 2.972e-01 | 9.560e-02 | 1.707e-02 | **1.181e-03** |
| | | 50% | 4.712e-01 | 2.244e-01 | 1.381e-03 | **7.305e-04** | 4.836e-01 | 2.319e-01 | 2.022e-02 | **1.262e-03** |
| | | 70% | 6.106e-01 | 5.652e-01 | 1.447e-03 | **7.715e-04** | 6.254e-01 | 5.773e-01 | 2.458e-02 | **1.609e-03** |
| | | 90% | 8.921e-01 | 8.845e-01 | 1.475e-03 | **8.051e-04** | 8.908e-01 | 8.924e-01 | 3.670e-02 | **1.847e-03** |
| | Mixed | 10% | 7.967e-02 | 1.282e-02 | 1.468e-03 | **8.328e-04** | 8.214e-02 | 1.335e-02 | 3.131e-02 | **1.270e-03** |
| | | 30% | 2.294e-01 | 7.869e-02 | 1.480e-03 | **8.619e-04** | 2.423e-01 | 7.919e-02 | 3.927e-02 | **1.570e-03** |
| | | 50% | 3.493e-01 | 1.831e-01 | 1.506e-03 | **9.158e-04** | 3.575e-01 | 1.925e-01 | 6.762e-02 | **1.620e-03** |
| | | 70% | 4.786e-01 | 4.495e-01 | 1.512e-03 | **9.980e-04** | 4.979e-01 | 4.561e-01 | 1.456e-01 | **2.225e-03** |
| | | 90% | 6.815e-01 | 7.003e-01 | 1.522e-03 | **1.171e-03** | 7.001e-01 | 7.147e-01 | 2.153e-01 | **2.460e-03** |

Besides, the MAE of DCNN-SE grows rapidly, especially for mixed contamination measurement. The phenomenon indicates that the method is unable to estimate system states accurately. By contrast, the proposed method is not significantly influenced, and still can maintain the error within the acceptable range. The major reason for these results is that the GAN in the proposed method consists of a generator and a discriminator, while DCNN only has a generator and the discriminator can enhance the performance of the generator by providing the feedback (Wasserstein distance) between the true system states and the generated system states during the training process.

In 2746-bus Polish system, the MAE in DCNN-SE method increases sharply, especially for phase angle and in mixed contamination cases. The MAE of these cases is larger than 1.0e-2 which does not satisfy the requirement of SE. In contrast, CGAN-SE can maintain a high degree of accuracy for both of voltage magnitude and phase angle.

In summary, the MAE of voltage magnitude in all types and levels of scenarios range in [5.1e-4, 6.4e-4] p.u. and [6.7e-4, 1.2e-3] p.u. for the IEEE 118-bus system and 2746-bus Polish system, respectively. Also, the MAE of phase angle in all types and levels of scenarios range in[1.6e-3, 5.3e-3] rad and [6e-4, 2.5e-3] rad for the IEEE 118-bus system and 2746-bus Polish system, respectively.

## 5. Conclusion

In this paper, a model-free and data-driven method is proposed for SE of a power system. This method is based on conditional WGAN, where the Wasserstein distance is applied to improve training performance. With the corrupted or incomplete measurement of different contamination levels, this method can perform better comparing than conventional LWLS-SE method as well as robust W LAV-SE method and DCNN-SE method. The proposed method CGAN-SE not only can estimate the system states with high accuracy but can also capture the statistical properties of the system measurements either from the probability distribution of system states or spatial correlation of buses. The effectiveness of the proposed CGAN-SE is validated through testing on the IEEE 118-bus system, as well as a large system with 2746 buses and the mean absolute error is less than 1.2e-3 p.u. and 5.3e-3 rad for voltage magnitude and phase angle, respectively.

## 6. References


1. Qi, J., Sun, K., and Kang, W., 'Optimal Pmu Placement for Power System Dynamic State Estimation by Using Empirical Observability Gramian', *IEEE Transactions on Power Systems*, 2014, 30, (4), pp. 2041-2054.
2. Zhang, C., Jia, Y., Xu, Z., Lai, L.L., and Wong, K.P., 'Optimal Pmu Placement Considering State Estimation Uncertainty and Voltage Controllability', *IET Generation, Transmission & Distribution*, 2017, 11, (18), pp. 4465-4475.
3. Zhu, J. and Abur, A., 'Identification of Network Parameter Errors Using Phasor Measurements', in, *2009 IEEE Power & Energy Society General Meeting*, (IEEE, 2009)
4. Gómez-Expósito, A., de la Villa Jaén, A., Gómez-Quiles, C., Rousseaux, P., and Van Cutsem, T., 'A Taxonomy of Multi-Area State Estimation Methods', *Electric Power Systems Research*, 2011, 81, (4), pp. 1060-1069.
5. Kekatos, V. and Giannakis, G.B., 'Distributed Robust Power System State Estimation', *IEEE Transactions on Power Systems*, 2012, 28, (2), pp. 1617-1626.
6. Zhang, L., Wang, G., and Giannakis, G.B., 'Real-Time Power System State Estimation and Forecasting Via Deep Unrolled Neural Networks', *IEEE Transactions on Signal Processing*, 2019, 67, (15), pp. 4069-4077.

7. Kekatos, V., Giannakis, G.B., and Wollenberg, B., 'Optimal Placement of Phasor Measurement Units Via Convex Relaxation', *IEEE Transactions on Power Systems*, 2012, 27, (3), pp. 1521-1530.
8. Zanni, L., Power-System State Estimation Based on Pmus', (EPFL, 2017)
9. Huang, Y.-F., Werner, S., Huang, J., Kashyap, N., and Gupta, V., 'State Estimation in Electric Power Grids: Meeting New Challenges Presented by the Requirements of the Future Grid', *IEEE Signal Processing Magazine*, 2012, 29, (5), pp. 33-43.
10. Enshaee, A., Hooshmand, R.A., and Fesharaki, F.H., 'A New Method for Optimal Placement of Phasor Measurement Units to Maintain Full Network Observability under Various Contingencies', *Electric Power Systems Research*, 2012, 89, pp. 1-10.
11. Huang, L., Sun, Y., Xu, J., Gao, W., Zhang, J., and Wu, Z., 'Optimal Pmu Placement Considering Controlled Islanding of Power System', *IEEE Transactions on Power Systems*, 2013, 29, (2), pp. 742-755.
12. Korres, G.N. and Manousakis, N.M., 'State Estimation and Bad Data Processing for Systems Including Pmu and Scada Measurements', *Electric Power Systems Research*, 2011, 81, (7), pp. 1514-1524.
13. Hui, X., Qing-quan, J., Ning, W., Zhi-qian, B., Hai-tang, W., and Hong-xia, M., 'A Dynamic State Estimation Method with Pmu and Scada Measurement for Power Systems', in, *2007 International Power Engineering Conference (IPEC 2007)*, (2007)
14. Fang, C., Xueshan, H., Zhiyuan, P., and Li, H., 'State Estimation Model and Algorithm Including PMU', in, *2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, (2008)
15. Chakrabarti, S., Kyriakides, E., Ledwich, G., and Ghosh, A., 'Inclusion of Pmu Current Phasor Measurements in a Power System State Estimator', *IET Generation, Transmission & Distribution*, 2010, 4, (10), pp. 1104-1115.
16. Massignan, J.A.D., London, J.B.A., Maciel, C.D., Bessani, M., and Miranda, V., 'PMUs and SCADA Measurements in Power System State Estimation through Bayesian Inference', in, *2019 IEEE Milan PowerTech*, (2019)
17. Costa, A.S., Albuquerque, A., and Bez, D., 'An Estimation Fusion Method for Including Phasor Measurements into Power System Real-Time Modeling', *IEEE Transactions on Power Systems*, 2013, 28, (2), pp. 1910-1920.
18. Tavares, B., Freitas, V., Miranda, V., and Costa, A.S., 'Merging Conventional and Phasor Measurements in State Estimation: A Multi-Criteria Perspective', in, *2017 19th International Conference on Intelligent System Application to Power Systems (ISAP)*, (2017)
19. Zhao, J. and Mili, L., 'A Framework for Robust Hybrid State Estimation with Unknown Measurement Noise Statistics', *IEEE Transactions on Industrial Informatics*, 2018, 14, (5), pp. 1866-1875.
20. Göl, M. and Abur, A., 'A Hybrid State Estimator for Systems with Limited Number of Pmus', *IEEE Transactions on Power Systems*, 2015, 30, (3), pp. 1511-1517.
21. Mahmood, F.: 'Synchrophasor Based Steady State Model Synthesis of Active Distribution Networks', KTH Royal Institute of Technology, 2018
22. Jones, K.D., Pal, A., and Thorp, J.S., 'Methodology for Performing Synchrophasor Data Conditioning and Validation', *IEEE Transactions on Power Systems*, 2015, 30, (3), pp. 1121-1130.
23. Tate, J.E. and Overbye, T.J., 'Extracting Steady State Values from Phasor Measurement Unit Data Using Fir and Median Filters', in, *2009 IEEE/PES Power Systems Conference and Exposition*, (IEEE, 2009)
24. Zhou, M., Centeno, V.A., Thorp, J.S., and Phadke, A.G., 'An Alternative for Including Phasor Measurements in State Estimators', *IEEE Transactions on Power Systems*, 2006, 21, (4), pp. 1930-1937.
25. Sarri, S., Zanni, L., Popovic, M., Boudec, J.L., and Paolone, M., 'Performance Assessment of Linear State Estimators Using Synchrophasor Measurements', *IEEE Transactions on Instrumentation and Measurement*, 2016, 65, (3), pp. 535-548.
26. Zhu, J. and Abur, A., 'Effect of Phasor Measurements on the Choice of Reference Bus for State Estimation', in, *2007 IEEE Power Engineering Society General Meeting*, (IEEE, 2007)
27. Aminifar, F., Fotuhi-Firuzabad, M., Shahidehpour, M., and Khodaei, A., 'Observability Enhancement by Optimal Pmu Placement Considering Random Power System Outages', *Energy Systems*, 2011, 2, (1), pp. 45-65.
28. Monticelli, A. and Wu, F.F., 'Network Observability: Identification of Observable Islands and Measurement Placement', *IEEE Transactions on Power Apparatus and systems*, 1985, PAS-104, (5), pp. 1035-1041.
29. Zhao, J., Zhang, G., Das, K., Korres, G.N., Manousakis, N.M., Sinha, A.K., and He, Z., 'Power System Real-Time Monitoring by Using Pmu-Based Robust State Estimation Method', *IEEE Transactions on Smart Grid*, 2016, 7, (1), pp. 300-309.
30. Mestav, K.R., Luengo-Rozas, J., and Tong, L., 'Bayesian State Estimation for Unobservable Distribution Systems Via Deep Learning', *IEEE Transactions on Power Systems*, 2019, 34, (6), pp. 4910-4920.
31. Chen, Y., Wang, Y., Kirschen, D., and Zhang, B., 'Model-Free Renewable Scenario Generation Using Generative Adversarial Networks', *IEEE Transactions on Power Systems*, 2018, 33, (3), pp. 3265-3275.
32. He, Y., Chai, S., and Xu, Z., 'A Novel Approach for State Estimation Using Generative Adversarial Network', in, *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, (IEEE, 2019)
33. Abur, A. and Exposito, A.G., *Power System State Estimation: Theory and Implementation*, (CRC press, 2004)
34. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 'Generative Adversarial Nets', in, *Advances in neural information processing systems*, (2014)
35. Arjovsky, M., Chintala, S., and Bottou, L., 'Wasserstein Gan', *arXiv preprint arXiv:1701.07875*, 2017.
36. Villani, C., *Optimal Transport: Old and New*, (Springer Science & Business Media, 2008)
37. Mirza, M. and Osindero, S., 'Conditional Generative Adversarial Nets', *arXiv preprint arXiv:1411.1784*, 2014.
38. Zimmerman, R.D., Murillo-Sánchez, C.E., and Thomas, R.J., 'MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education', *IEEE Transactions on Power Systems*, 2011, 26, (1), pp. 12-19.

39. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., and Devin, M., 'Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems', *arXiv preprint arXiv:1603.04467*, 2016.
40. Krizhevsky, A., Sutskever, I., and Hinton, G.E., 'Imagenet Classification with Deep Convolutional Neural Networks', in, *Advances in neural information processing systems*, (2012)