

A Novel Approach for State Estimation Using Generative Adversarial Network

Yi He, Songjian Chai, Zhao Xu, *Senior Member, IEEE*

Abstract— Accurate power system state estimation is essential for power system control, optimization, and security analysis. In this work, a model-free approach was proposed for power system static state estimation based on conditional Generative Adversarial Networks (GANs). Comparing with conventional state estimation approach, i.e., Weighted Least Square (WLS), any appropriate knowledge of system model is not required in the proposed method. Without knowing the specific model, the GANs can learn the inherent physics of underlying state variables purely relying on historic samples. Once the model has been well trained, it can generate the corresponding estimated system state given the system raw measurements. Particularly, the raw measurements are sometimes characterized by incompleteness and corruption, which gives rise to significant challenges for conventional analytic methods. The case study on IEEE 9-bus system validates the effectiveness of the proposed approach.

Keywords— State estimation, generative adversarial network, deep learning, conditional GAN

I. INTRODUCTION

In power systems, all the quantities (current, voltages, etc.) can be figured out from the minimum set of independent variables that are called power system state [1]. These states determine the power system operating conditions. Before the 1970s, the power system states used to be figured out by the load-flow calculation by using the current and voltages raw measurements [2]. However, there are some drawbacks involved in such load-flow calculation based on state estimation (SE). It is unlikely to collect the measurements without missing and contaminating in practice, as a consequence, the solution of load-flow could change completely due to incorrect measurements. In order to solve this issue, power system SE was introduced by Schweppe by integrating statistical estimation theory and load-flow [3-5]. The estimated state is the most likely estimate about the power system state generated via SE [6-7]. Since the estimated state can abide measurement losses, filter the measurement noise, detect bad measurement data [8-10] and determine the errors in network-model [11-13], the estimated state is more robust and reliable than the raw measurements.

SE plays an important role in modern energy management system, and the accurate SE is essential for power system control, optimization, and security analysis. Most of the existing SE approaches are based on Weighted Least Squares (WLS) due to its simplicity in fundamental and effectiveness in supervisory control and data acquisition (SCADA) system. In the early years, SE is performed widely based on SCADA measurements [14], [15], but recently, with increasing pene-

tration of phasor measurement units (PMU) in power system, it allows to directly measure voltage phasor relying on its synchronization provided by the global positioning satellite (GPS) system. The sampling rate of PMU can be up to 60 samples per second, which brings about huger amount of data as compared with SCADA system (5 samples/sec). As a result, the conventional SE methods are difficult to implement in real time due to the heavy computational burden. In addition, as the conventional methods require that the system is fully observable, they would be inapplicable in the situation of incomplete data or unobservable buses.

To overcome these issues, in this paper, a novel SE approach using conditional generative adversarial network (GAN) is proposed. GAN is the most promising framework in deep learning and have become a research front in the area of machine learning especially in computer vision because of its excellent performance in generating realistic images [16]. This inspires us to apply GAN in SE problem. The intuition of this process illustrating with images is that indistinct or incomplete images called "fake" images are given as input and then GANs can generate distinct and complete images called "real" images. In parallel, by feeding the raw system measurements into GAN, the estimated states can be directly generated. Through validating on IEEE 9-bus system, the proposed model proves to be effective in generating system states that are close to the true system states from the statistical perspectives.

By comparing with conventional approaches, it has three main advantages applied in SE: 1) data-driven and model-free. Proposed method does not need any knowledge of the system model; 2) high speed of processing. After learning with training data, the proposed method can quickly generate the output for a given input. 3) fault tolerant. Even with missing and contaminated data on bus measurements, the proposed method can be also skillful in estimating the states. On the worst case that part of measurement is missing and the system is unobservable, proposed method can also generate the true system states. To the best of our knowledge, this is the first work using deep learning models to power system SE processes.

The remainder of this paper is organized as follows: Section II presents the problem formulation. The proposed method in SE using GANs model is elaborated in Section III. Several tests are conducted, and their results are analyzed in Section IV. Finally, Section V concludes the paper.

II. PROBLEM FORMULATION

A. Conventional Power System State Estimation

Conventionally, in the SCADA system, the state variables are the magnitude and phase of bus voltage, and the measurements are real and reactive power. Thus, the measurement

The authors are with the Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China (email: 1994heyi@gmail.com; chaisongjian@gmail.com; zhao.xu@polyu.edu.hk).

function between the measurement and the state variables is nonlinear [7]. However, the measured quantities of PMU are voltage or current phasors, and the measurement function between the measurement and the state variables is linear. In this paper, we use only PMU measurements in our network; accordingly, the measurement model is linear comparing with nonlinear model in conventional measurements. Also, in both measurements and state variables, the phasors are in rectangular forms. In this paper, the state variables are the phasors of bus voltage in the system. In SE, the general measurement model can be expressed as:

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

where \mathbf{x} is the vector of state variables that is bus voltage phasors; \mathbf{y} is the vector of measurement obtained by PMU; $\mathbf{h}(\cdot)$ is the vector-valued measurement function, which is linear to the bus voltage phasors measurement; \mathbf{e} is measurement error vector that is assumed to white noise composed by zero mean with a covariance matrix \mathbf{R} , which is made up of the inverse of the measurement error variances.

B. State Estimation with Raw Measurement

In a system network, there are N buses at this system and vector of state variables \mathbf{x} is $[V_1 \dots V_N, \theta_1 \dots \theta_N]^T$. For bus j , $j = 1, \dots, N$, V_j and θ_j are the voltage magnitude and phase angle of buses respectively. The number of PMU is M and PMUs are deployed in the different buses to make the system is observable, which means that all the system buses voltage phasors can be measured. The buses which are installed PMUs can be measured directly via voltage phasors; the buses adjacent to PMU buses can be measured via current phasors; other buses can be measurement by injection measurement [17]. We assume one PMU is deployed at each bus, and the vector of raw measurement \mathbf{y} is $[y_1 \dots y_N]^T$. Each element of \mathbf{y} is $y_j = V_j \angle \theta_j$, $j = 1, \dots, N$ which denotes voltage phasor at bus that is installed PMU.

We use raw measurement and true system state pair $(\mathbf{y}, \tilde{\mathbf{x}})$ as training example. In this model, the input is raw measurement \mathbf{y} , and the output is the true system state $\tilde{\mathbf{x}}$. After collecting a large number of $(\mathbf{y}, \tilde{\mathbf{x}})$ pairs as the training set, the objective is to train a generative model based on GANs by using this training set. When the model is well trained, the generated system states $\hat{\mathbf{x}}$ is our estimated system state, which should have the capability to describe the same power system operating condition as the true system state $\tilde{\mathbf{x}}$.

C. State Estimation with Corrupt Raw Measurement

In power system, large errors can corrupt the PMU measurements and result in corrupt raw measurements. The cause of these large errors can be impulsive communication noise, the failures of instrument, cyber-attacks, etc. In SE, it is necessary to have the capability to handle this corrupt raw measurement. The set of corrupt raw measurement is C , and the number of corrupt raw measurement is n_c . The vector of corrupt raw measurement \mathbf{y}_c is $[y_1 \dots y_{n_c} \dots y_N]^T$. The model should detect the corrupt raw measurement, and generated system state should describe the real power system operating condition. By training with the training set which includes

plenty of $(\mathbf{y}_c, \tilde{\mathbf{x}})$ pairs, the SE with corrupt raw measurement can be applied easily. Our proposed method can generate estimated system state $\hat{\mathbf{x}}$ with corrupt raw measurement whatever the reason causes or however corrupt they are.

D. State Estimation with Incomplete Raw Measurement

Potential transformers (PTs) and current transformers (CTs) are the primary measurement equipment in substations that provide PMUs with the input signals and measure voltage and current phase values, respectively. And then, the Phasor Data Concentrator (PDC) gather the raw measurement data from PMUs by means of communication links and finally send them to the Control Center (CC). Thus, the situations that can cause losing PMU raw measurement are: impossibility of getting voltage or current measurements from PTs or CTs respectively, failure of the PDC, or failure of the local communication system.

There are two kind situations of incomplete raw measurement. One of them is that the system is still observable with losing PMU measurement, and in this case, the conventional methods are able to be applied to estimate system state while the other is that losing too many PMU measurement causes the system unobservable, and conventional methods are not applicable in this situation. n_L is the number of losing PMU measurement and \mathbf{y}_L is the vector of incomplete raw measurement, expressed as $[y_1 \dots y_{N-n_L}]^T$. To estimate system state, a large number of historical $(\mathbf{y}_L, \tilde{\mathbf{x}})$ pairs are added into the training set. The objective in SE with incomplete raw measurement is to train a generative model based on GANs by using this training set. The generated system state $\hat{\mathbf{x}}$ should be close to the true system state $\tilde{\mathbf{x}}$ in both situation of incomplete raw measurement that the system is observable or not.

III. PROPOSED METHOD

This section introduces the theory of GANs [18], how they are adapted to our problem in SE and the novel framework for SE. First, the method of GANs is reviewed and then the objectives and also the loss functions are formulated. Then, it introduces how to incorporate raw PMU measurement into the GANs model training procedure.

A. Wasserstein GANs

In this part, we focus on using GANs model to generate a system state regardless of raw PMU measurement which will be considered later in next part.

As defined before, $\tilde{\mathbf{x}} = [\tilde{x}_1 \dots \tilde{x}_N]^T$ is the true system state. Let the true system state examples denote as $\{\tilde{x}_j^{(i)}\}_{i=1}^m$, $j = 1, \dots, N$ and the distribution of true system state is denoted by $p_{data}(x)$. Suppose a group of noise vector input z with a known distribution $z \sim p_z(z)$ which is sampled from (e.g., uniform distribution or jointly Gaussian). GANs consist of two deep neural networks, called the generator and the discriminator. Let G express the generator written as $G(z; \theta^{(G)})$, and its function is parametrized by $\theta^{(G)}$; Let D express the discriminator written as $D(x; \theta^{(D)})$, and its function is para-

metrized by $\theta^{(D)}$. In two neural networks, the generator G and the discriminator D, $\theta^{(G)}$ and $\theta^{(D)}$ are their weights, respectively. And then, by training G and D simultaneously, the target estimated system state \hat{x} can be transformed from a sample z which is from the noise distribution $p_z(z)$.

Generator: When training the generator G, the inputs are from the noise distribution $p_z(z)$, undergoing a large number of up sampling operations, and the output is estimated system state. The training procedure can be expressed as a mapping:

$$G(z; \theta^{(G)}) : z \rightarrow p_G(z) \quad (1)$$

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 D should be trained with the generator simultaneously. The input of discriminator is the samples from either the generated distribution $p_G(z)$ or the true system state distribution $p_{data}(x)$. After plenty of down sampling operations, the output is a value p_{real} which is continuous and reflect what extent they are similar between the input and the true system state distribution $p_{data}(x)$. The training procedure of discriminator can be expressed as a mapping:

$$D(x; \theta^{(D)}) : x \rightarrow p_{real} \quad (2)$$

where x is the input from either $p_G(z)$ or $p_{data}(x)$. The aim of the discriminator is to try to distinguish between $p_G(z)$ and $p_{data}(x)$, and also to maximize the difference between these two distributions.

After defining the objectives of generator and discriminator, next step is training procedure. we train D to maximize its capacity of discernment between true system state distribution and generated distribution from G. We simultaneously train G to minimize the difference between these two distributions. The loss function of generator G and discriminator D are C_G and C_D respectively. They are expressed as:

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

$$C_D = -\mathbb{E}_{z \sim p_{data}(x)} [\log(D(x))] - \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (3)$$

Combining these two loss functions and formulate them as a two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log(D(x))] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (4)$$

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

According the Kantorovich-Rubinstein duality [19], The Wasserstein distance (Earth-Mover distance) is the dual of the minimax objective in (4). x and y are two random variables and $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ 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 expressed as:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|] \quad (5)$$

The Wasserstein distance can be viewed with the ‘‘cost’’ of the optimal plan that move all the ‘‘mass’’ $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ 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.

In GANs, our objective is to try to make the generated distribution $p_z(D(G(z)))$ close to the real distribution $p_{data}(D(x))$. Thus, the Wasserstein distance between them can be expressed as:

$$W(D(x), D(G(z))) = \sup_D \mathbb{E}_{x \sim p_{data}(x)} [D(x)] - \mathbb{E}_{z \sim p_z(z)} [D(G(z))] \quad (6)$$

When the Wasserstein distance is coverage, the optimal plan of moving ‘‘mass’’ is found. Therefore, the optimal generator G^* is also found. According to the literature of GANs, the JS divergence applied in original GAN [19] cannot reflect what extent two distribution are close when they are very different from each other. Hence, this characteristic result in that GAN is sensitive to the parameters and it may always generate the single pattern of estimated system state whose the probability is highest regardless of what is the input. However, Wasserstein distance in [20] can overcome this drawback and it can give the accurate distance between two distribution timely during the training procedure. Therefore, the generator can generate the optimal system state rather than the same one though the raw measurement is different.

B. Conditional GANs

In a general GANs model, the input is the noise vector z sampled from $p_z(z)$ where is no extra limitation for the generated output. GANs can be extended to a conditional mode with both the generator and discriminator being conditioned on some extra information y [20]. In condition GANs, the generator output should satisfy with this condition y .

Applying the conditional GANs in state estimation, the raw measurement is condition y in this model. The generated system state not only should close to true sample of system state in training set, but also should close to this raw measurement. The conditioning procedure can be performed by feeding y into both the generator and discriminator as additional input layer. Then the objective function of the two-player minimax game with Wasserstein distance would be:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [D(x|y)] - \mathbb{E}_{z \sim p_z(z)} [D(G(z|y))] \quad (7)$$

In Fig. 1, it shows the architecture of GANs that we use. The algorithm used in our proposed method is described in Algorithm 1.

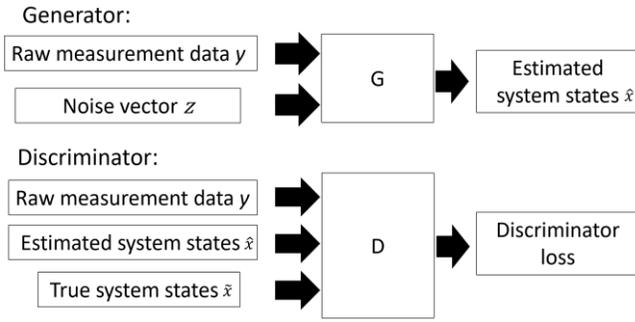


Fig. 1. The architecture of GANs that we use, including the input and output of the generator and discriminator, respectively.

Algorithm 1 Conditional GANs with Wasserstein Distance for State Estimation

Require: α , 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 $t = 0, \dots, k_{dis}$ **do**

- Sample batch of m noise samples $\{(z^{(i)}, y^{(i)})\}_{i=1}^m$ 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

In our GANs model, generator and discriminator are both differentiable functions whose neural network layers consist of multilayer perceptron (MLP), convolution, normalization, mx-pooling and Rectified Linear Units (ReLU). During training iterations, we train them in a batch and use gradient ascend algorithm in discriminator's training and gradient descend in generator's training. Besides, *RMSProp* algorithm is applied in both generator and discriminator to allow the learning rate to be self-adjustable. In the discriminator training, clipping is applied to meet specific conditions and avoid gra-

dent explosion. In section IV, there are detailed GANs model structure and the process of training.

IV. NUMERICAL RESULTS

A. Data Description

In this paper, we collect the true system state data by implementing the Monte Carlo power flow calculations on IEEE 9-bus system. To achieve distinct system states, we assume the load satisfies the Gaussian distribution with zero mean and standard deviation of 0.1. Load samples are then draw and fed into power flow computations to derive true system states. Fig. 2. shows IEEE 9-bus test system topology graph. Since in this system, Bus 1 is slack bus and Bus 2, Bus 3 are PV bus. These three buses' voltage magnitude are constant. Therefore, in our case study, we only implement experiment on PQ Bus 4 – 9.

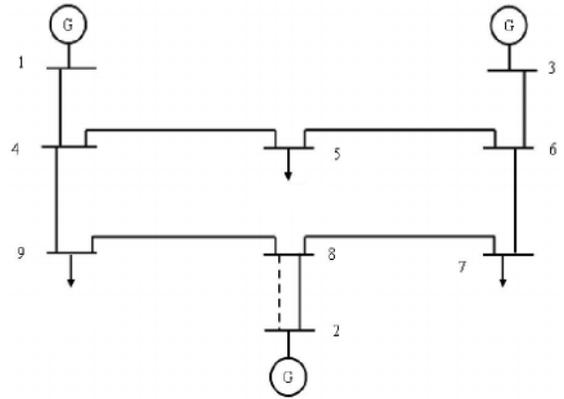


Fig. 2. IEEE 9-bus test system topology graph.

Then, the system raw measurements are created based on the true system states samples. Here, we consider three situations 1) measurements are contaminated with false data; 2) measurements are missing at several buses; 3) measurements are contaminated and missing. For the first situation, the raw measurements are contaminated with 70% errors on a randomly selected bus. Missing data examples are obtained by that randomly choose 3 buses from 6 PQ buses and set their voltage magnitude as 0 representing missing measurement.

B. State Estimation with GANs

Firstly, we test the model on the context of corrupt measurements. The proposed model is trained by the corrupt raw measurement and true system state pairs. 10, 000 example pairs are used and 7, 000 pairs are used in the training phase and the remaining are used for the testing. The noise samples sampled from Gaussian distribution is a 100-dimensional vector, the batch size is 32. Then the generator input is the noise vector z with the raw measurement as the condition y . For one batch, the shape of input is (32, 100+6), through two MLP layers and two de-convolutional layers and finally the output should be the system state and the shape is (32, 6). Our discriminator's input is generated system state \hat{x} or true system state \tilde{x} with condition y . The structure of discriminator consists of two MLP layers and two convolutional layers in sequence. The output of discriminator is the discriminator loss

(Wasserstein distance). Our program for proposed method is completed by Python on PyCharm IDE with NVIDIA Quadro P2000 GPU.

The discriminator loss and Mean Absolute Percentage Error (MAPE) between generated system states and true system states in the training process are depicted in Fig. 3 and Fig. 4, respectively. As can be noted in Fig. 3, the Wasserstein distance converges fast, indicating that the distance between the generated distribution $p_z(z)$ and the true system state distribution $p_{data}(x)$ quickly reaches its minimal value. It can be also seen from the Fig. 4 that the MAPE drop quickly with the decrease of Wasserstein distance, meaning that the generated system states are closer to the true system states.

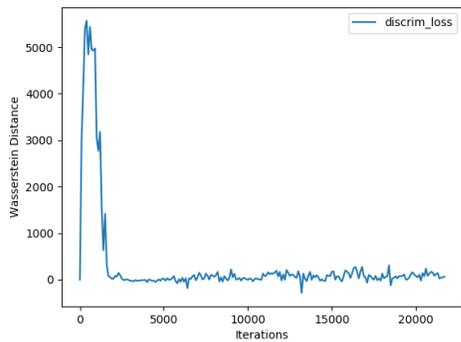


Fig. 3. Discriminator loss with iterations.

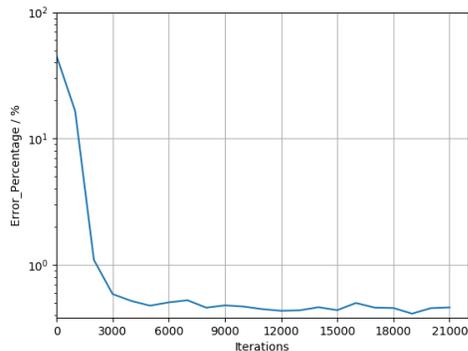


Fig. 4. MAPE with iterations.

Fig. 5. shows the comparison between two probabilistic density function (PDF) profiles fitted by true system states (black solid lines) and generated system states (red dashed lines) via GANs on all considered buses. As can be clearly seen from all buses, the generated states basically follow the distribution of true states, which means the proposed GAN model can well handle with the corrupt raw measurements in SE.

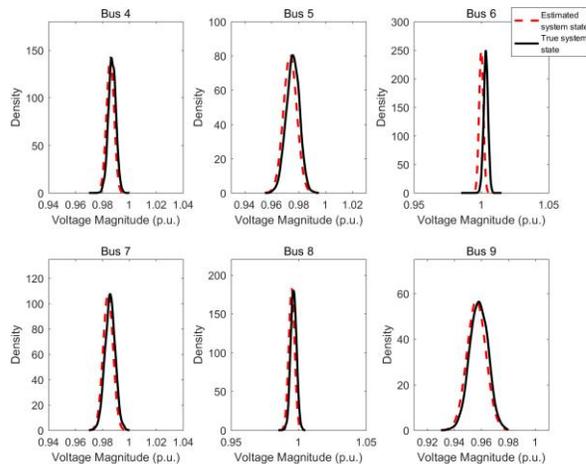


Fig. 5. PDF profiles with corrupt raw measurement input.

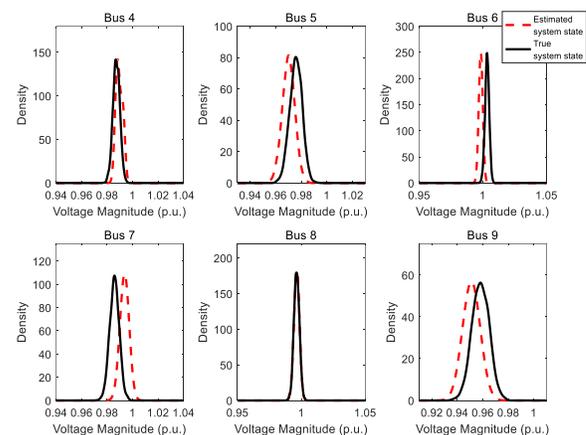


Fig. 6. PDF profiles with incomplete raw measurement input.

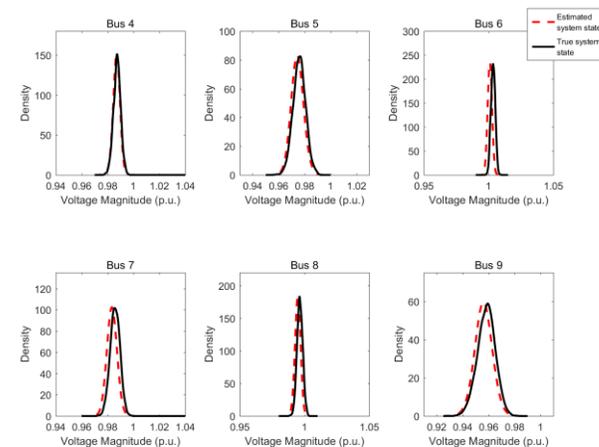


Fig. 7. PDF profiles with both corrupt and incomplete raw measurement input.

ACKNOWLEDGMENT

The work presented in this paper is supported by a Grant from the Research Grants Council (RGC) of the Hong Kong Special Administrative Region (HKSAR), China, under Theme based Research Scheme through Project No. T23-407/13-N and T23-701/14N.

REFERENCES

- [1] L. Zanni, "Power-system state estimation based on PMUs static and dynamic approaches—From theory to real implementation," Ph.D. dissertation, Inst. Elect. Eng., Polytechnique Fed' erale de Lausanne, Lausanne, Switzerland, 2017.
- [2] J. J. Grainger, W. D. Stevenson, and W. D. Stevenson, *Power system analysis*. 2003.
- [3] F. C. Schweppe and J. Wildes, "Power system static-state estimation, Part I: Exact model," *IEEE Transactions on Power Apparatus and systems*, no. 1, pp. 120-125, 1970.
- [4] F. C. Schweppe and D. B. Rom, "Power system static-state estimation, Part II: Approximate model," *IEEE Transactions on Power Apparatus and Systems*, no. 1, pp. 125-130, 1970.
- [5] F. C. Schweppe, "Power system static-state estimation, Part III: Implementation," *IEEE Transactions on Power Apparatus and systems*, no. 1, pp. 130-135, 1970.
- [6] A. Monticelli, *State estimation in electric power systems: a generalized approach*. Springer Science & Business Media, 2012.
- [7] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.
- [8] A. Monticelli and A. Garcia, "Reliable bad data processing for real-time state estimation," *IEEE Transactions on Power Apparatus and Systems*, no. 5, pp. 1126-1139, 1983.
- [9] T. Van Cutsem, M. Ribbens-Pavella, and L. Mili, "Hypothesis testing identification: A new method for bad data analysis in power system state estimation," *IEEE Transactions on Power Apparatus and Systems*, no. 11, pp. 3239-3252, 1984.
- [10] K. Clements and P. Davis, "Multiple bad data detectability and identifiability: a geometric approach," *IEEE Transactions on Power Delivery*, vol. 1, no. 3, pp. 355-360, 1986.
- [11] F. F. Wu and W.-H. Liu, "Detection of topology errors by state estimation (power systems)," *IEEE Transactions on Power Systems*, vol. 4, no. 1, pp. 176-183, 1989.
- [12] W.-H. Liu, F. F. Wu, and S.-M. Lun, "Estimation of parameter errors from measurement residuals in state estimation (power systems)," *IEEE Transactions on power systems*, vol. 7, no. 1, pp. 81-89, 1992.
- [13] J. Zhu and A. Abur, "Identification of network parameter errors using phasor measurements," in *2009 IEEE Power & Energy Society General Meeting*, 2009: IEEE, pp. 1-5.
- [14] A. Gómez-Expósito, A. de la Villa Jaén, C. Gómez-Quiles, P. Rousseaux, and T. Van Cutsem, "A taxonomy of multi-area state estimation methods," *Electric Power Systems Research*, vol. 81, no. 4, pp. 1060-1069, 2011.
- [15] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1617-1626, 2012.
- [16] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3265-3275, 2018.
- [17] B. Gou, "Optimal placement of PMUs by integer linear programming," *IEEE Transactions on power systems*, vol. 23, no. 3, pp. 1525-1526, 2008.
- [18] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672-2680.
- [19] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008.
- [20] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

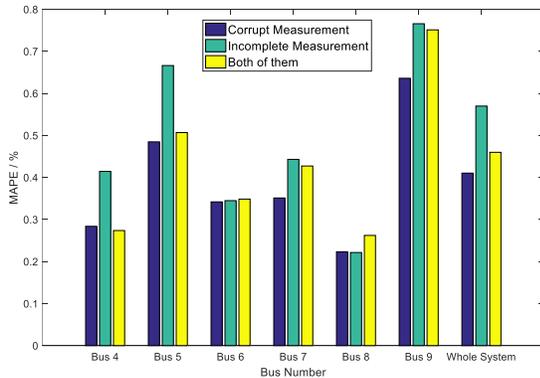


Fig. 8. MAPE on each bus and whole system in three situations.

Followed by the same model setting and steps in the case of corrupt data, we perform the experiments on the remaining two scenarios. The obtained PDF profiles for the incomplete raw measurement against that of system true state is depicted in Fig. 6. Comparing with the result of corrupt raw measurement, the distribution deviations between the generated system state and the true system state are larger, especially at Bus 5, Bus 7 and Bus 9. However, in this test, it is assumed that three measurements are randomly missing at 6 buses, resulting in 20 combinations in total. Except two combinations (missing measurements occur at Bus 4, 6, 8 and Bus 5, 7, 9), the rest 18 combinations represent the unobservable situations. Nevertheless, the proposed GAN-based model can still give the most likely state distribution in these unobservable situations, even though with deviations in certain buses, while the conventional methods fail to be applied to this case at all.

By considering corrupted and incomplete raw measurement simultaneously (one measurement is missing and one measurement is corrupted), the derived PDF profiles against that of true states are given in Fig. 7. In this case, as the unobservable bus number is curtailed, the results are slightly better than that of situation 2.

Fig. 8. shows the MAPE at each bus in three situations. The result is consistent with the previous comparative results of statistical distribution in Fig. 5-7. Largest MAPE occurs in situation 2, where bus 5 and 7 give the worst MAPE among all buses, which confirms the observations in Fig. 6.

V. CONCLUSION

In this paper, a model-free SE framework is proposed by considering the raw system measurements are contaminated with false data and deficiency. The model is based on conditional GAN, where Wasserstein distance is applied to improve its performance. Through testing on IEEE 9-bus system, the proposed GAN model is demonstrated to well mimic the statistical properties of true system states. The experiments on larger system and real system operation data will be conducted in future work.