IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 24, 2018, accepted October 5, 2018, date of publication October 15, 2018, date of current version November 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875936

Scenario Generation for Wind Power Using Improved Generative Adversarial Networks

CONGMEI JIANG[1], YONGFANG MAO[1], YI CHAI[1], MINGBIAO YU[2], AND SONGBING TAO[1]
[1]Key Laboratory of Complex System Safety and Control, Ministry of Education, College of Automation, Chongqing University, Chongqing 400044, China
[2]School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Corresponding author: Yongfang Mao (yfm@cqu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61633005 and Grant 61374135.

ABSTRACT Wind power scenarios have a significant impact on stochastic optimization problems for power systems in which wind power is a significant component. Generative adversarial networks (GANs) are a powerful class of generative models, and can generate realistic scenarios for renewable power sources without the need for any modeling assumptions. However, the performance of GANs in generating scenarios can further be improved by modifying the way in which a Lipschitz constraint on discriminator network is imposed. Another critical problem of applying deep neural networks is overfitting, a phenomenon especially prone to appear on small training sets. In this paper, we propose an improved GAN for the generation of wind power scenarios. To improve the training speed, we use a gradient penalty term to enforce the Lipschitz constraint based on the output and input of the discriminator network. To improve the scenario quality, we further use a consistency term in the training procedure. Besides, the overfitting problem can be effectively alleviated by the enforced Lipschitz continuity. The proposed method is applied to actual time series data from the NREL wind integration data set. The experimental results demonstrate that our method outperforms the existing methods.

INDEX TERMS Deep learning, generative adversarial networks, scenario generation, wind power.

I. INTRODUCTION

Modern power systems are undergoing a transition into smart grids to meet environmental targets (renewable power production, energy storage, and price-responsive demand, as well as providing power for plug-in hybrid electric vehicles), which poses a great challenge to the power industry [1]. Wind energy accounts for the largest component of the growth in renewable energy and is therefore playing an increasingly important role in the operation and planning of power systems. However, the uncertainty and variability of wind power production mean that additional power production sources must be available to provide sufficient ramping capability, posing further technological and economic challenges, especially in the context of large-scale integration of wind power. Therefore, accurate modeling of wind power output is key to reducing system-wide costs and enforcing reliability criteria to allow decision-making under the uncertainty faced by power system operators and planners.

As an intermittent energy source, wind power has inherent stochastic characteristics, which may or may not act as suggested by forecasting. In wind power forecasting, there are three different representations of wind power uncertainty, namely, the probabilistic forecasting, risk index, and scenario approaches [2]. The question of which of these is the best has not been the topic of much academic research, and the choice of the most suitable representation has generally been guided by specific situations. In the scenario representation, the spatiotemporal correlations of uncertainty information can be considered as generating a variety of time trajectories of wind power outputs. By using a series of possible wind power scenarios, system operators and participants are able to investigate the influence of wind power on cost and reliability to allow decision-making in an uncertain environment, such as operational planning with volatile wind power production [3], storage portfolio optimization of wind-storage systems [4], and strategic bidding in electricity markets [5]. It is therefore necessary to investigate scenario generation approaches to wind power for integrated wind power systems.

Many techniques have been proposed for wind power scenario modeling. Numerical weather prediction (NWP) [6] uses weather information to construct a physical model. The scenario set can be generated using wind power forecasts, by modeling the relationship between wind power output and extracted wind speed [7], [8]. In [9], a moment matching

technique is presented for generating synthetic scenarios consistent with specified values of the marginal moments and correlations. Cholesky decomposition and cubic transformation are applied to construct the discrete joint distribution to obtain the specified correlations. In [10], an autoregressive moving average (ARMA) method is proposed to produce a plausible scenario set for multiple wind sites, and a similar method is adopted in [11]. In [12], the vector autoregressive formulations are extended by incorporating moving average terms to construct a joint state space model. A state space form of the wind speed is then proposed to generate wind power scenarios. In [13], a Gaussian copula for stochastic dependence modeling is presented and is applied to the uncertainty analysis of large-scale wind power integration. Inspired by the copula theory, an exponential Gaussian copula approach is proposed in [14] for the generation of wind power scenarios from probabilistic forecasts of wind power, and the exponential covariance structure is further estimated from the probability distribution of wind power variations in [15].

These model-based methods are usually based on statistical assumptions (e.g., Gaussian), which makes them incapable in practice of accurately capturing the data distribution of wind power profiles [16]. Scenario samplings (e.g., inverse transform and Monte Carlo) from multivariate distributions also affect the quality of the generated scenarios. Methods like copula and ARMA rely on probabilistic forecasts, which are directly affected by the forecast accuracy of wind sites. Therefore, it difficult to use model-based methods to capture the full diversity of renewable resources, especially when considering spatiotemporal correlations among multiple sites.

A number of machine learning algorithms to generate wind scenarios have also been proposed recently. Artificial neural networks (ANNs) are among the most popular intelligent methods. In [17], an ANN is combined with historical time series values, one or more exogenous variables (e.g., ambient temperature and wind speed), and appropriate time indices to create more representative scenarios. In [18], a probabilistic wind power prediction model is proposed to generate scenarios based on radial basis function neural networks (RBFNNs). In [19], to reduce the forecasting error in generated scenarios, an ensemble of scenarios is generated in various ways, including support vector machine (SVM), multi-layer perceptron (MLP), regularized linear regression, and random forest methods. These intelligent algorithms should be suitable for capturing the uncertainty in renewable power production owing to their excellent ability to map the nonlinear relationship between input and output. However, their performance will be strongly affected by feature selection and tuning of the problem.

With the development of machine learning, generative adversarial networks (GANs) [20] have received great attention and have been used in various situations [21]– [26]. In [16], a data-driven method is presented for generating renewable scenarios using GANs. This method can generate scenarios for spatiotemporally correlated multiple sites by learning the distribution of historical data; such scenarios can also be generated based on conditioned information, such as mean values, ramp events, and forecast errors in wind power. In [27], Bayesian GANs are constructed and trained to produce a set of renewable power scenarios with the same pattern as historical data. Even if wind and solar data are deliberately blended, the method can simultaneously distinguish and generate different scenarios, allowing better representation of renewable power production processes. These methods of applying GANs for scenario generation are mainly related to generate scenarios which reflect the dynamic patterns of renewable resources. In [28], an unsupervised approach based on GANs for scenario forecast is proposed to generate a group of future realizations. The generated scenarios can capture the reliability and sharpness features and reflect both forecast information and dynamic patterns of wind power production [29]. Deep generative models can learn to capture uncertainty in renewable power production with a full diversity of behavior, and can be trained through the use of differentiable networks without the need for any additional tuning. Because of their use of unsupervised learning, another important advantage of GANs is that they do not require manual labeling of data, which is usually impossible for large datasets.

The contributions of the present paper can be briefly summarized as follows. Based on the work described in [30] and [31], we propose an improved GAN to generate wind power scenarios, using an alternative technique for enforcing the Lipschitz continuity through a gradient penalty and a consistency term in the training procedure. Our proposed method can avoid the problems of exploding or vanishing gradients and achieve more stable training for wind power scenario generation. We show that our method can generate high quality scenarios for wind power and is superior to existing methods in capturing the data distribution of real historical observations. The proposed method can also achieve faster convergence, thus reducing the training time for the adversarial networks. In addition, our method has better generalization capability and is less prone to overfitting, which is more suitable than existing methods for cases where there is an insufficient amount of training data or where wind farms have not been long in power production.

The rest of this paper is organized as follows. Section II presents the mathematical formulations of the tasks of scenario generation. The GAN and improved GAN are described in more detail in Section III. The model structure of the improved GAN is presented in Section IV, and the algorithm given there is then used to generate scenarios. The experimental results are illustrated and compared in Section V. The results are discussed and our final conclusions are presented in Section VI.

II. PROBLEM FORMULATION

In this section, we present the scenario generation procedure for two tasks of interest. To improve the performance

of GANs in generating scenarios, we discuss some of the motivations for the use of improved GANs.

A. SCENARIO GENERATION FOR A SINGLE SITE

The wind power output of a site can be modeled as a stochastic process. We assume that historical data x_t, $t = 1, \ldots, T$, on wind power are available for a certain generation site. Our objective is to construct and train a GAN model to generate scenarios representing uncertainties for a single site by using these historical wind power data $\{x_t\}$ as the training set. The generated wind power scenarios should capture various patterns representing all possible behaviors of the wind power production process.

B. SPATIOTEMPORAL SCENARIO GENERATION

It is well known that for a large system, the wind varies among different areas. The scale of these changes is affected by distance. Thus, what we focus on here is not a single wind farm or a single region. Rather, our task is to generate wind power scenarios in different places at the same time. Consider a set of historical observations for multiple wind energy sources at N sites. We denote by $x_{t,s}$, $t = 1, \ldots, T$, $s = 1, \ldots, N$, a data matrix of historical wind power data. Our objective is to construct and train a GAN model to generate scenarios for multiple wind generation sites by using the historical data $\{x_{t,s}\}$ as the training set. The generated wind power scenarios should capture the spatiotemporal correlations among multiple sites.

GANs are a popular class of generative models, and have been widely studied and used owing to their excellent ability to learn from experience. However, optimization of GANs is difficult and may result in low-quality samples or failure of convergence due to the application of weight clipping for enforcing a Lipschitz constraint in the training process. In addition, deep learning on small datasets is very prone to overfitting. To deal with these problems,we introduce an improved GAN, as described in the next section.

III. METHOD

We first present the GAN and how to enforce Lipschitz continuity in the neural network setting. We then describe the application of our improved method to wind power scenario generation.

A. WASSERSTEIN GAN

A GAN is among the most powerful frameworks arising in the context of generative models and is modeled as a competing game between two interconnected neural networks. The general architecture of a GAN is shown in Fig. 1. A generator network attempts to generate realistic-looking samples given some noise source, while a discriminator network tries to distinguish between generated samples and true samples. Since their invention, GANs have become widely used for unsupervised learning of data distributions of unlabeled data. The Wasserstein GAN (WGAN) [32] is a development that is considered to be an effective alternative to traditional GAN training. Through its use of the Wasserstein distance, its performance is better than that of a traditional GAN.

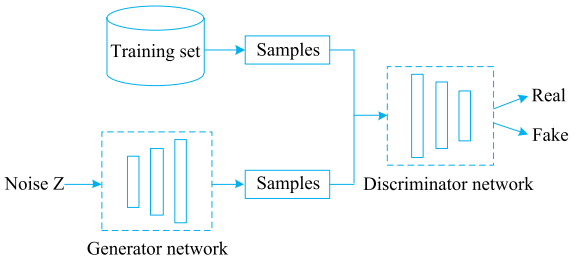


FIGURE 1. General architecture of a GAN.

For convenience, we use x to denote the historical data for a single site or multiple sites. We denote the true distribution of the historical data x by P_r and assume that the random noise variable input z is sampled from a simple known distribution P_z (e.g., a Gaussian). Our goal is to translate a sample z drawn from P_z into a desired sample that follows the distribution P_r. This is accomplished by constructing and training adversarial networks: a generator network and a discriminator network. These two networks are trained alternately until the training converges.

The generator network is trained to fool the discriminator network by taking upsampling operations of several fully connected layers and deconvolutional layers to output plausible samples. To achieve this task, we need to define a loss function L_G to update the parameters for the generator network. During the training process, a random variable z sampled from the distribution P_z is fed into the generator. Then, the output $G(z)$ of the generator is a new random variable, whose distribution is denoted by P_G. A small value of L_G indicates that the samples generated from the distribution P_G are almost the same as the historical samples from the perspective of the discriminator. We use the loss function L_G defined in [32]:

$$L_G = -\mathbb{E}_{z\sim P_z}[D(G(z))], \tag{1}$$

where $\mathbb{E}$ denotes the empirical means of the batch updates.

The discriminator network is trained alternately with the generator network. It is fed with input samples coming either from the generator network or from the training set. By taking a series of downsampling operations of several fully connected layers and convolutional layers, the discriminator network outputs a continuous value to measure the input samples. Depending on the sources of different samples, the output value of the discriminator network can be expressed as follows:

$$P_{\text{real}} = D(x), \tag{2a}$$
$$P_{\text{fake}} = D(G(z)). \tag{2b}$$

The discriminator network is trained to distinguish between P_r and P_G, i.e., to maximize the difference between

$\mathbb{E}[D(\cdot)]$ and $\mathbb{E}[D(G(\cdot))]$. Similarly, we need to define a loss function L_D to update the parameters of the discriminator network. When the discriminator network is good at differentiating between generated samples and real samples, the value of L_D should be small. Therefore, the loss function L_D can be written as

$$L_D = -\mathbb{E}_{x\sim P_r}[D(x)] + \mathbb{E}_{z\sim P_z}[D(G(z))]. \quad (3)$$

For a given discriminator network, maximizing the output L_D means minimizing $-\mathbb{E}[D(G(\cdot))]$, resulting in the expression (1). For a given generator network, the discriminator network should maximize $\mathbb{E}[D(\cdot)]$ (real samples) and at the same time minimize $\mathbb{E}[D(G(\cdot))]$ (generated samples). This gives the expression (3). Note that the adversarial networks are parametrized by the weights that are leveraged to remember learning experiences in the training process. With the loss functions defined, we can then formulate the value function of the two-player game. Based on Kantorovich–Rubinstein duality [33], the game between the two interconnected neural networks is the minimax objective

$$\min_G \max_D V(G, D) = \mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{z\sim P_z}[D(G(z))]. \quad (4)$$

The Wasserstein distance, also known as the earth-mover distance, was introduced into GANs in [32]. In terms of model training, this distance has better properties than other metrics (e.g., the Jensen–Shannon divergence and the Kullback–Leibler divergence) because it directly measures the difference between the probability distributions P_G and P_r. This metric is given by

$$W(P_r, P_G) = \sup_\omega \mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{z\sim P_z}[D(G(z))], \quad (5)$$

where ω is the parameter of the discriminator network.

Since the Wasserstein metric $W(P_r, P_G)$ directly calculates data distributions of samples from different sources, it is a more valuable cost function and provides an index for training of GANs. Not only is this metric related to the quality of generated scenarios, but it is also very convenient for debugging and hyperparameter searching.

B. IMPROVED GAN

The WGAN uses the Wasserstein metric to measure different data distributions of disjoint parts, which solves the problem of training instability caused by the data distribution not being appropriately measured by the cross-entropy (Jensen–Shannon divergence). The use of this metric requires that the weights of the discriminator network lie within a compact space to satisfy a Lipschitz constraint. Since the capacity of the network is limited and there could be gradient exploding or vanishing problems in the training, the weight clipping may lead to optimization difficulties. An improved strategy for enforcing Lipschitz continuity is proposed in [30] based on optimal transport theory [33]. Inspired by the optimal discriminator network that has unit gradient norm almost everywhere under P_G and P_r, this alternative strategy adds a gradient penalty based on the output and input of the discriminator network. This gradient penalty is given by

$$GP|_{\hat{x}} = \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \quad (6)$$

where $\hat{x} = tx + (1-t)G(z)$ for $t \sim U[0, 1]$.

Although it is difficult to enforce the unit gradient norm constraint everywhere, there is an effective alternative for model training. With the gradient penalty GP explicitly defined, the loss function of the discriminator is

$$L_{GP} = \mathbb{E}_{z\sim P_z}[D(G(z))] - \mathbb{E}_{x\sim P_r}[D(x)] + \lambda GP|_{\hat{x}}. \quad (7)$$

The gradient penalty term GP performs better than the weight clipping for enforcing the Lipschitz continuity, and can achieve more stable training on a wide variety of GAN architectures (e.g., the DCGAN architecture and the 101-layer ResNet) and can generate higher-quality samples on different datasets (e.g., the CIFAR-10 and LSUN bedrooms). Since the gradient term can only be punished at sampled data points in the training process, a large number of data points will not be sampled at all. In addition, the output of the generator network is significantly different from the actual data point at the start of the training. The Lipschitz continuity is not enforced until the data distribution P_G of the generated data points and the real distribution P_r of the training data points are sufficiently close to each other. Therefore, the performance of the gradient penalty method can be further improved.

In [31], an additional Lipschitz continuity condition is proposed to improve the training of GANs. Instead of focusing on particular data points sampled between the real and generated points, a region around the real data manifold is considered. In particular, two perturbed data points x' and x'' near the observed real data point x are used to check the continuity condition. The two virtual points are found by applying the stochastic dropout to the hidden layers of the discriminator. The performance can be improved slightly by further controlling the second-to-last layer of the discriminator. Therefore, there is an additional consistency term in the loss function (7) of the following form:

$$CT|_{x',x''} = \mathbb{E}_{x\sim P_r}[\max(0, d(D(x'), D(x'')) + 0.1d(D_-(x'), D_-(x'')) - M')], \quad (8)$$

where M' is a bounded constant, $D_-(\cdot)$ is the second-to-last layer of the discriminator, and d is the ℓ_2 metric on the input space.

The gradient penalty term GP, (6), enforces the continuity at specific data points, while the consistency term CT, (8), complements the continuity over the data manifold and its surrounding regions. Therefore, these two terms can be used together to improve the training of a GAN. Putting them together, the loss function of the discriminator can then be expressed as follows:

$$L_{CT} = \mathbb{E}_{z\sim P_z}[D(G(z))] - \mathbb{E}_{x\sim P_r}[D(x)] + \lambda_1 GP|_{\hat{x}} + \lambda_2 CT|_{x',x''}. \quad (9)$$

In order to verify the effect of GP and CT on model training for wind power scenario generation, we plot the histograms of the discriminator's weights in Fig. 2 after we train it using weight clipping, GP, and GP and CT, respectively. Fig. 2(a) shows that the weights are pushed to two values, which are the extreme values of the clipping range. It suggests that the network capacity of the discriminator trained using weight clipping is underused in the training. Besides, the gradient may change exponentially or not significantly depending on the value of the clipping threshold. Therefore, the optimization process is difficult and there could be gradient exploding or vanishing problems in the training process. In contrast, as Fig. 2(b) shows, the GAN trained using GP does not suffer from these behaviors. The more stable gradients enable the discriminator network's capacity to be more fully utilized. We also compare the weights of the discriminator trained using GP with those of the discriminator trained using GP and CT. Fig. 2(b) and Fig. 2(c) show that both methods can control the weights within a symmetric distribution. And it is interesting to see the weights range [0.2251, −0.2269] in Fig. 2(c) is within a smaller and more symmetric distribution compared with that [0.2764, −0.2387] in Fig. 2(b). In order to further verify the role of the CT term, we plot the maximum Euclidean norm of the gradients for each iteration in Fig.3. As stated previously, the optimal WGAN discriminator has unit gradient norm almost everywhere under P_G and P_r. The closer to 1 the norms are, the better the 1-Lipschitz continuity

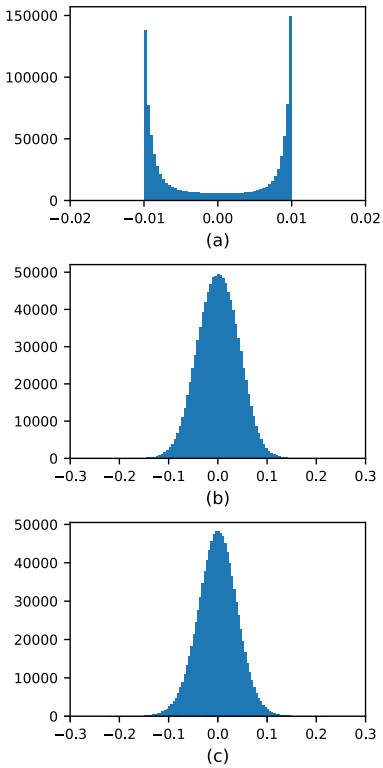


FIGURE 2. Histograms of the weights of the discriminator trained for wind power scenario generation using (a) weight clipping, (b) GP, and (c) GP and CT, respectively.

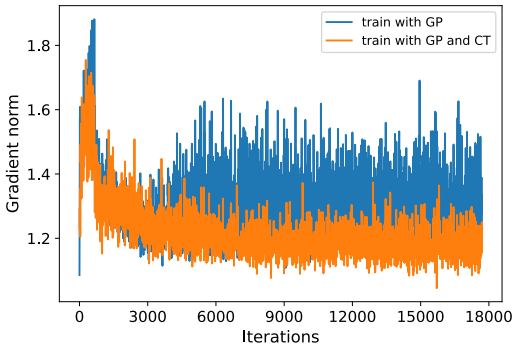


FIGURE 3. Training evolution of the gradient norm of the discriminator with respect to its input on a wind dataset.

is preserved. It is obvious that the GAN trained using GP and CT can better enforce the Lipschitz continuity in the training process.

With the redefined value function L_{CT}, we then can use the GAN to implement the generation of wind power scenarios by training two interconnected adversarial networks. In the initial stage of the training process, L_{CT} is small and L_G is large, because the generator network has not yet learned the data distribution of the wind power profiles. In this case, the generator network generates wind scenarios totally different from real scenarios, and the discriminator network can easily distinguish between them. The generator network gradually learns various patterns in historical wind data. The adversarial networks are continuously updated and alternately trained. As the training tends to convergence, the generator network is able to generate realistic wind power scenarios with small L_G, while L_{CT} is large and the discriminator network can hardly distinguish between generated scenarios and real scenarios. Eventually, the output wind power scenarios of the generator network are able to represent the stochastic processes of wind power production.

IV. MODEL STRUCTURE AND TRAINING ALGORITHM

GANs have a flexible network structure. The framework for a GAN is to formulate the generative modeling problem as an adversarial process consisting of two interconnected deep neural networks. In this section, the model structure and the algorithm are described.

A. MODEL STRUCTURE

The model structure of our GAN is based on the structures used in [30], [31], and [34]. To make a fair comparison between our method and existing ones, we use the same network structure. Table 1 lists the details of the generator and discriminator network structures. In this table, "MLP" denotes the multi-layer perceptron, "Conv" and "Deconv" denote the convolutional and deconvolutional layers, respectively, "DIM" is the dimension of the filters, "D_" denotes the second-to-last layer of the discriminator, and "Sigmoid" is an activation function used to limit the output range in the interval [0, 1].

TABLE 1. Model structure of the GAN for scenario generation.

Generator network	Discriminator network
Input: Noise z, 128	Input: 24*24*1
MLP, 3*3*4*DIM	Conv, DIM
Deconv, 2*DIM	Conv, 2*DIM
Deconv, DIM	Conv, 4*DIM
Deconv, 24*24*1	Reshape, 3*3*4*DIM (D_)
Sigmoid	MLP, 1 (D)

The discriminator network has the reverse structure to the generator network. The latter includes three deconvolutional layers to upsample the input noise z to generate wind power scenarios, whereas the discriminator network includes three convolutional layers to downsample data from historical and generated samples. ReLU activation and LeakyReLU activation are used in the hidden layers of the generator network and the discriminator network, respectively. Dropout is only applied in the output of each hidden layer of the discriminator network. Batch normalization has been used in most previous GAN structures [34], [35]. Since the penalty terms in the loss function of the discriminator network are handled independently for a single input, not for the entire batch [35], the batch normalization can be omitted or replaced by layer normalization [36] in our model structure.

Algorithm 1 Proposed GAN for Generating Scenarios. We Use Default Values of $m = 64$, $\lambda_1 = 10$, $\lambda_2 = 2$, $N_{\text{iter}} = 4 \times 10^4$, $n_{\text{critic}} = 5$, $\gamma = 0.0003$, $\beta_1 = 0.5$, $\beta_2 = 0.9$

Require: m, the batch size; λ_1 and λ_2, weights; N_{iter}, the number of iterations; n_{critic}, the number of discriminator iterations per generator iteration. γ, β_1, β_2, Adam hyperparameters.
Require: initial parameters ω for discriminator and θ for generator
for N_{iter} training iterations **do**
  for n_{critic} iterations **do**
    # Update parameters for discriminator network
    for $i = 1, \ldots, m$, **do**
      Sample data $x \sim P_r$, variable $z \sim P_z$, a number $\varepsilon \sim U[0, 1]$
      $\tilde{x} \leftarrow G(z)$
      $\hat{x} \leftarrow \varepsilon x + (1 - \varepsilon)\tilde{x}$
      $L^{(i)} \leftarrow D(\tilde{x}) - D(x) + \lambda_1 \, GP|_{\hat{x}} + \lambda_2 \, CT|_{x', x''}$
    end for
    $\omega \leftarrow Adam\left(\nabla_\omega \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, \omega, \gamma, \beta_1, \beta_2\right)$
  end for
  # Update parameters for generator network
  Sample a batch of variables $\{z^{(i)}\}_{i=1}^{m} \sim P_z$
  $\theta \leftarrow Adam\left(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D(G(z^{(i)})), \theta, \gamma, \beta_1, \beta_2\right)$
end for

B. ALGORITHM

Our proposed method for generating wind power scenarios can be trained using Algorithm 1. We use the *Adam* optimizer to update the parameters of the discriminator network and the generator network [37]. Our adversarial networks learn the data distribution of historical data in a batch updating style. In our experiment, we use $\lambda_1 = 10$ from [30] and $\lambda_2 = 2$ from [31] for the setting of the improved GAN. Another hyperparameter M' from the consistency term CT can be set to zero. In all experiments, n_{critic} is set to 5, so that there are five discriminator iterations per generator iteration in the alternating training of the adversarial networks. Once the model has been trained to convergence using this algorithm, the generator network is able to generate wind power scenarios that preserve the same data distribution as historical data. Note that both the generator and discriminator networks are trained through deep learning, which can be implemented through various deep learning frameworks. All our experiments for training GANs are programmed using Python 3.6 with an open source software library TensorFlow [38].

V. EXPERIMENTS

In this section, we describe our experiments on a historical wind power dataset and their results. We first show the scenario generation for two tasks of interest, then we compare our method with existing methods for scenario generation though several aspects. These experimental results indicate that our method should provide a more efficient and flexible fashion for the generation of wind power scenarios.

A. DATA DESCRIPTION

The wind power data we use is from the paper [16]. The data is collected from the NREL Wind Integration Dataset [39]. This dataset is designed to support the next generation of integration studies and provides wind power data for more than 126,000 sites in the United States. Historical data with a temporal resolution of 5 minutes is provided. The wind farms in Washington State are selected to use as the input dataset. For different scenario generation tasks, the input samples are divided into a training set and a validation set. In general, we can randomly select 20% of the input samples as the validation set. Since we want to generate wind power scenarios for a single site and for multiple sites, we need to prepare different datasets for the tasks.

B. SCENARIO GENERATION AND SPATIOTEMPORAL CORRELATIONS

For different scenario generation tasks, we can use the same GAN model. The framework for using the same GAN for scenario generation is illustrated in Fig. 4. For a single site of interest, wind farms in geographical proximity are collected as input samples to represent the stochastic generation dynamics. For multiple sites, instead of inputting historical

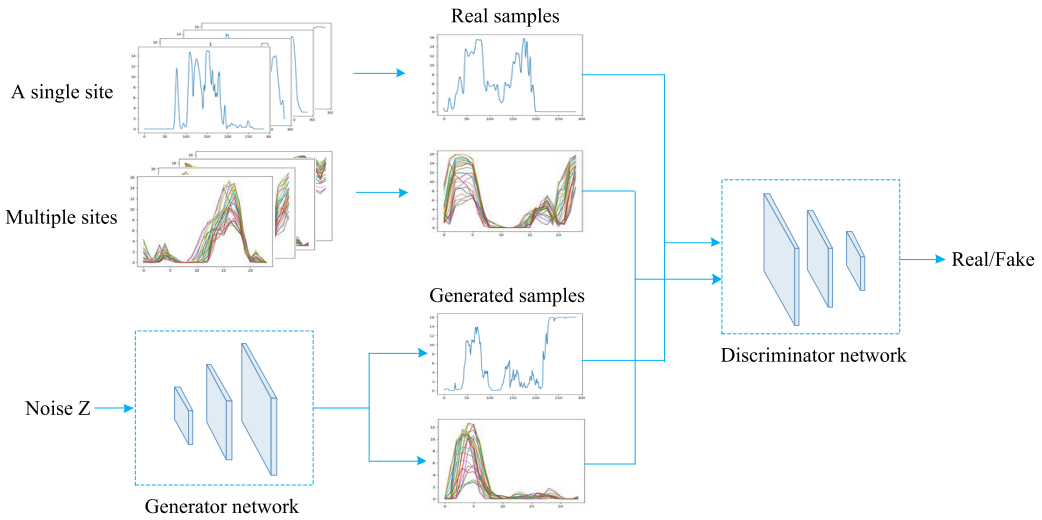


FIGURE 4. Illustration of the framework for using the same GAN for different tasks.

data $x^{(i)}$ for a single site, here we input the model with a data matrix $\{x^{(i)}\}$ that is composed of 24 wind farm data points with a resolution of one hour as the historical observations.

1) TEMPORAL CORRELATION

For scenario generation for a single site, our generating model is repeatedly inputted with the historical samples until the discriminator loss converges. Fig. 5 compares the scenarios generated by WGAN and our method. These samples are selected by visual inspection. We compare the generated samples with the real samples and select the samples with the same patterns. We can see that the samples generated by both methods can correctly capture the hallmark features (e.g., large peak values, daily variations, and ramp events of large fluctuations) of the wind power profiles from the validation set, which is used to assess whether the generating model is well trained.

To evaluate if the generated samples have the same statistical properties as the real samples, we calculate the autocorrelation coefficient $R(h)$ for them by

$$R(h) = \sum_{i=1}^{n-h} \frac{(S_i - \mu)(S_{i+h} - \mu)}{\sum_{i=1}^{n} (S_i - \mu)^2}, \tag{10}$$

where h is the look-ahead time and S represents generated samples or realizations with mean μ.

The autocorrelation coefficient measures the degree of correlation of a time series between two different periods. The bottom rows of Fig. 5 show that the scenarios generated by the two methods have very similar temporal correlations with the historical data.

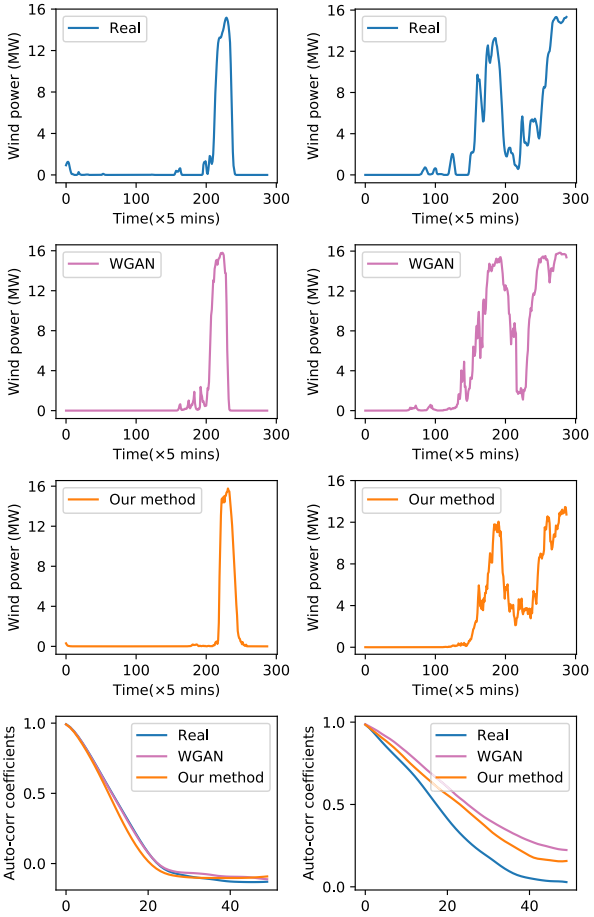


FIGURE 5. Comparison of scenario generation for a single site.

2) SPATIAL CORRELATION

For scenario generation for multiple sites, we examine whether our generating model can capture the spatial correlation of wind power outputs. Fig. 6 shows a comparison of the wind power scenarios generated by WGAN and our method. Their dynamic behaviors are similar. By visual inspection we find the spatial and temporal correlations in

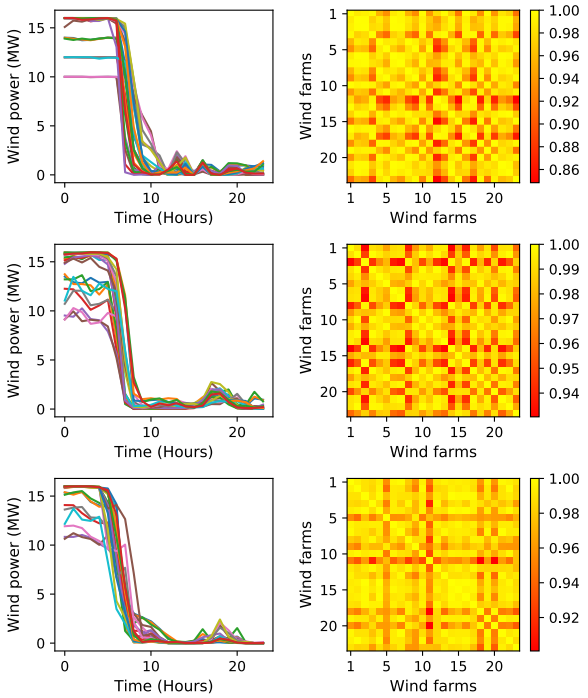


FIGURE 6. Wind power scenarios and spatial correlation coefficient colormaps for multiple sites: (top) historical data; (middle) sample generated by WGAN; (bottom) sample generated by our method.

the real data are correctly preserved by these two methods. Owing to the resemblance of the generated scenarios and the historical observations in terms of, for example, peak values, mean values, and ramp events, the trained discriminator can hardly distinguish input samples with the same pattern.

To further examine the spatial correlation of generated scenarios, we make use of the Pearson correlation coefficient $\rho_{i,j}$ for different wind farms. The Pearson correlation coefficient is a statistic that reflects the degree of similarity between two variables. Given the set of time series S, each term $\rho_{i,j}$ is calculated by

$$\rho_{i,j} = \frac{Cov(S_i, S_j)}{\sigma_{S_i}\sigma_{S_j}}, \tag{11}$$

where σ is the standard deviation and $Cov(S_i, S_j)$ is the covariance of S_i and S_j.

We compute the spatial correlation coefficients of generated samples and historical data, and visualize them in Fig. 6. The multi-dimensional data of the samples is shuffled, which makes the spatial correlation more complex to learn. We can see that all the patches of these three sets of colormaps have relatively large values. Therefore, our generating model can provide us with a useful tool to capture the stochastic behavior of wind power for correlated multiple sites.

C. MEANINGFUL EVALUATION INDEX AND IMPROVED PERFORMANCE

One advantage of the WGAN is that the Wasserstein distance is continuously approximated by training the discriminator to optimality, which provides a useful convergence index for research on adversarial networks [32]. This index is not only useful for debugging and hyperparameter searching, but also has a direct relation to the quality of generated samples. Convergence of the index to a small value should correspond to high quality of the generated wind power scenarios. The proposed method also has the desirable property that the value function is correlated with the quality of the generated samples, as can be seen in Figs. 5 and 6. To compare the quality of generated scenarios, we train the GAN using three different methods (i.e., weight clipping, GP, and GP and CT) on the training set and plot the training curves in Fig. 7. For any of these methods, the discriminator losses of real and generated samples exhibit large difference in the early stage of training. This is because the generator has not yet learned the data distribution of the real data at this stage. Then, the adversarial networks start training and force the discriminator losses to gradually converge. When the model is trained to converge, the Wasserstein distance in Fig. 7(b) is closer to zero than that in Fig. 7(a). It suggests the data distribution of the historical observations can be better captured by using GP in the training procedure. Fig. 7(b) and Fig. 7(c) show that we can further use the CT term to improve the quality of generated samples. These experimental results indicate the GP and CT can be used together to improve the quality of generated wind power scenarios.

An obvious benefit of our method over weight clipping for wind power scenario generation is that it has a high training speed. Owing to the use of weight clipping on discriminator network, the GAN is relatively slow to train as shown in Fig. 7(a). When the GAN is trained using GP, as shown in Fig. 7(b), it can make better use of the capacity of discriminator network and achieve a significantly faster training speed. Therefore, the Wasserstein distance can reach a minimum at a very early stage of the model training. Fig. 7(b) and Fig. 7(c) indicate that the CT term has no obvious improvement for training speed. As can be seen from the experimental results, our method starts to converge toward a minimum after about 10,000 iterations [see Fig. 7(c)], while the GAN trained using weight clipping takes nearly 30,000 iterations [see Fig. 7(a)] to reach the same level. Clearly, our method can achieve faster convergence than the existing method for scenario generation.

D. OVERFITTING PROBLEM AND EFFECTIVE IMPROVEMENT

Deep learning has become a key research topic in machine learning, but one of the biggest problems of deep neural networks is overfitting. This problem is prone to occur especially in very small training sets. This can be due to the presence of noise or to a small number of samples, which leads to insufficient representation of the predetermined rules. In general, the performance of the training model is evaluated by decreasing the quantity of data and by using a validation set to verify the training results and check the training model to see how well it performs.

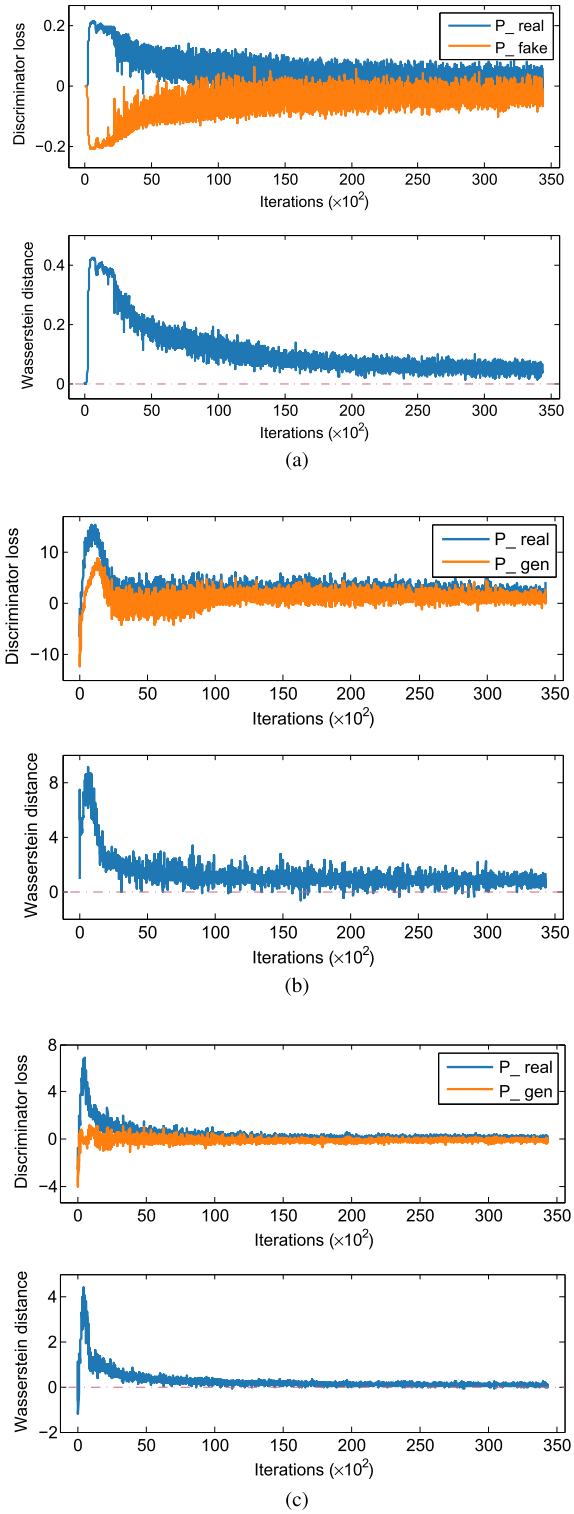


FIGURE 7. Comparison of training curves of GAN trained on a wind dataset using (a) weight clipping, (b) GP, and (c) GP and CT.

GANs, like all deep neural network models, are prone to overfitting when trained on limited data [40]. To explore the behavior that occurs when adversarial networks overfit the training set, we train the GAN on a subset of the dataset for

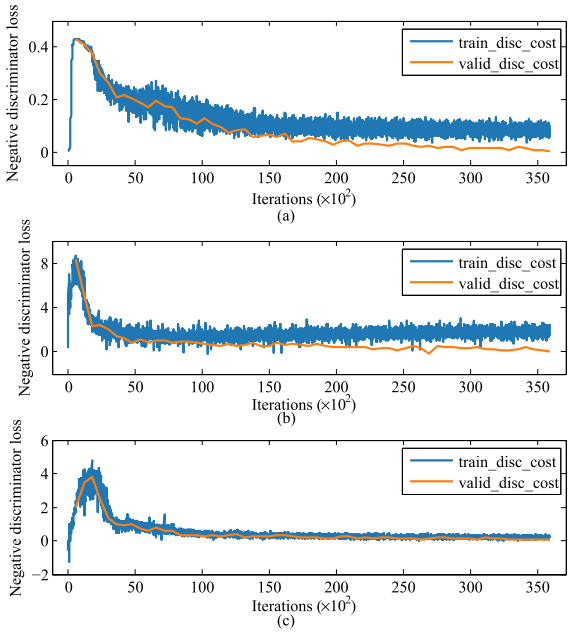


FIGURE 8. Training and validation losses of GAN trained on a small training set using (a) weight clipping, (b) GP, and (c) GP and CT, respectively.

multiple sites using three different methods, namely, weight clipping, GP, and GP and CT. The subset consists of 500 samples and is randomly selected from the dataset. Fig. 8 presents the negative discriminator losses for these three methods on the training set (blue curves) and the validation set (orange curves). The validation set error is checked every 100 generator iterations. When overfitting occurs, the model is perfectly able to match the training data, but cannot adapt to the validation data. It can be seen from Fig. 8(a) and Fig. 8(b) that the two losses diverge, suggesting that the adversarial networks are subject to obvious overfitting and provide an inaccurate estimate of $W(P_r, P_G)$, at which point all bets are directly related to the quality of the generated scenarios. In contrast, from Fig. 8(c), we can see that the discriminator losses consistently decrease with the almost same trend for both the training and validation sets, which demonstrates that our method is less prone to overfitting than the other two methods when used for scenario generation.

VI. DISCUSSION AND CONCLUSIONS

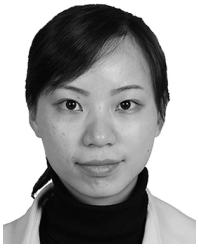
We have presented a new method using an improved GAN for scenario generation for wind power production. This method is data-driven and does not require any statistical assumptions. We use an alternative technique for enforcing a Lipschitz constraint to train the adversarial networks. A gradient penalty term is used to improve the training speed. The data distribution of wind power profiles can be better captured by our enforced Lipschitz continuity. Experimental results based on an NREL wind dataset show that our method is more efficient and of greater practical value than existing methods for generating wind power scenarios.

Since our method can implement scenario generation tasks both for a single site and for correlated multiple sites without any changes to the model structure, different datasets can be trained with great flexibility. In addition, as our method does not rely on any sampling techniques, it can be used to directly generate a large number of scenarios and can provide a useful tool for uncertainty modeling in integrated wind power systems. In future work, we will extend this work to explore probabilistic forecasting problems over various forecast horizons and prediction intervals related to the integration of wind power production.

REFERENCES

[1] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power Energy Mag.*, vol. 7, no. 2, pp. 52–62, Mar./Apr. 2009.
[2] Y. Zhang, J. Wang, and X. Wang, "Review on probabilistic forecasting of wind power generation," *Renew. Sustain. Energy Rev.*, vol. 32, pp. 255–270, Apr. 2014.
[3] J. Wang, M. Shahidehpour, and Z. Li, "Security-constrained unit commitment with volatile wind power generation," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1319–1327, Aug. 2008.
[4] S. Wogrin and D. F. Gayme, "Optimizing storage siting, sizing, and technology portfolios in transmission-constrained networks," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3304–3313, Nov. 2015.
[5] Y. Wang, Y. Dvorkin, R. Fernández-Blanco, B. Xu, T. Qiu, and D. S. Kirschen, "Look-ahead bidding strategy for energy storage," *IEEE Trans. Sustain. Energy*, vol. 8, no. 3, pp. 1106–1117, Jul. 2017.
[6] C. Monteiro *et al.*, "Wind power forecasting: State-of-the-art 2009," Decis. Inf. Sci. Division, Argonne Nat. Lab., Lemont, IL, USA, Tech. Rep. ANL/DIS-10-1, 2009, pp. 124–130, vol. 32, no. 2.
[7] E. M. Constantinescu, V. M. Zavala, M. Rocklin, S. Lee, and M. Anitescu, "A computational framework for uncertainty quantification and stochastic optimization in unit commitment with wind power generation," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 431–441, Feb. 2011.
[8] N. Chen, Z. Qian, I. T. Nabney, and X. Meng, "Wind power forecasts using Gaussian processes and numerical weather prediction," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 656–665, Mar. 2014.
[9] K. Hoyland, M. Kaut, and S. W. Wallace, "A heuristic for moment-matching scenario generation," *Comput. Optim. Appl.*, vol. 24, nos. 2–3, pp. 169–185, 2003.
[10] J. M. Morales and R. Mínguez, and A. J. Conejo, "A methodology to generate statistically dependent wind speed scenarios," *Appl. Energy*, vol. 87, no. 3, pp. 843–855, 2010.
[11] A. Papavasiliou and S. S. Oren, "Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network," *Oper. Res.*, vol. 61, no. 3, pp. 578–592, 2013.
[12] G. Díaz, J. Gómez-Aleixandre, and J. Coto, "Wind power scenario generation through state-space specifications for uncertainty analysis of wind power plants," *Appl. Energy*, vol. 162, no. 1, pp. 21–30, 2016.
[13] G. Papaefthymiou and D. Kurowicka, "Using copulas for modeling stochastic dependence in power system uncertainty analysis," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 40–49, Feb. 2009.
[14] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, "From probabilistic forecasts to statistical scenarios of short-term wind power production," *Wind Energy*, vol. 12, no. 1, pp. 51–62, 2009.
[15] X.-Y. Ma, Y.-Z. Sun, and H.-L. Fang, "Scenario generation of wind power based on statistical uncertainty and variability," *IEEE Trans. Sustain. Energy*, vol. 4, no. 4, pp. 894–904, Oct. 2013.
[16] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3265–3275, May 2018.
[17] S. I. Vagropoulos, E. G. Kardakos, C. K. Simoglou, A. G. Bakirtzis, and J. P. S. Catalao, "ANN-based scenario generation methodology for stochastic variables of electric power systems," *Electr. Power Syst. Res.*, vol. 134, pp. 9–18, May 2016.
[18] G. Sideratos and N. D. Hatziargyriou, "Probabilistic wind power forecasting using radial basis function neural networks," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 1788–1796, Nov. 2012.
[19] H. Pandžić, Y. Dvorkin, T. Qiu, Y. Wang, and D. S. Kirschen, "Toward cost-efficient and reliable unit commitment under uncertainty," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 970–982, Mar. 2016.
[20] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3. 2014, pp. 2672–2680.
[21] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. (2017). "Are GANs created equal? A large-scale study." [Online]. Available: https://arxiv.org/abs/1711.10337
[22] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 588–598, Sep. 2017.
[23] C. Li, K. Xu, J. Zhu, and B. Zhang. (2017). "Triple generative adversarial nets." [Online]. Available: https://arxiv.org/abs/1703.02291
[24] C. Li *et al.*, "ALICE: Towards understanding adversarial learning for joint distribution matching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5495–5503.
[25] H.-T. Zheng, W. Wang, W. Chen, and A. K. Sangaiah, "Automatic generation of news comments based on gated attention neural networks," *IEEE Access*, vol. 6, pp. 702–710, Nov. 2017.
[26] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018.
[27] Y. Chen, P. Li, and B. Zhang, "Bayesian renewables scenario generation via deep generative networks," in *Proc. 52nd Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, USA, Mar. 2018, pp. 1–6.
[28] Y. Chen, X. Wang, and B. Zhang. (2017). "An unsupervised deep learning approach for scenario forecasts." [Online]. Available: https://arxiv.org/abs/1711.02247
[29] P. Pinson and R. Girard, "Evaluating the quality of scenarios of short-term wind power generation," *Appl. Energy*, vol. 96, pp. 12–20, Aug. 2012.
[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. (2017). "Improved training of Wasserstein GANs." [Online]. Available: https://arxiv.org/abs/1704.00028
[31] X. Wei, B. Gong, Z. Liu, W. Lu, and L. Wang. (2018). "Improving the improved training of Wasserstein GANs: A consistency term and its dual effect." [Online]. Available: https://arxiv.org/abs/1803.01541
[32] M. Arjovsky, S. Chintala, and L. Bottou. (2017). "Wasserstein GAN." [Online]. Available: https://arxiv.org/abs/1701.07875
[33] C. Villani, *Optimal Transport: Old and New*. Berlin, Germany: Springer, 2008.
[34] A. Radford, L. Metz, and S. Chintala. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434
[35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. (2016). "Improved techniques for training GANs." [Online]. Available: https://arxiv.org/abs/1606.03498
[36] J. L. Ba, J. R. Kiros, and G. E. Hinton. (2016). "Layer normalization." [Online]. Available: https://arxiv.org/abs/1607.06450
[37] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980
[38] M. Abadi *et al.* (2016). "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1603.04467
[39] C. Draxl, A. Clifton, B.-M. Hodge, and J. McCaa, "The wind integration national dataset (WIND) toolkit," *Appl. Energy*, vol. 151, pp. 355–366, Aug. 2015.
[40] Y. Li and L. Shen, "cC-GAN: A robust transfer-learning framework for HEp-2 specimen image segmentation," *IEEE Access*, vol. 6, pp. 14048–14058, 2018.

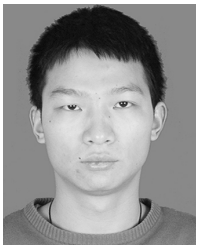
CONGMEI JIANG received the B.S. degree in automation from the University of South China, Hengyang, China, in 2010. He is currently pursuing the Ph.D. degree with the School of Automation, Chongqing University. His research is on optimization, control and machine learning, with applications in wind-integrated power systems.



YONGFANG MAO received the Ph.D. degree from Chongqing University, Chongqing, China, in 2008. She is currently an Associate Professor with the School of Automation, Chongqing University. Her research interests include nonstationary signal processing, mechanical fault diagnosis, prognosis and health management, and process monitoring.

YI CHAI received the Ph.D. degree from Chongqing University, Chongqing, China, in 2001. He is currently a Professor with the School of Automation, Chongqing University. His research interests include information processing, integration and control, and computer network and system control.

MINGBIAO YU received the B.S. degree in automation from the University of South China in 2010, and the M.S. degree from Jiangsu University in 2013. He is currently pursuing the Ph.D. degree with the School of Instrument Science and Engineering, Southeast University. His research interests include signal processing, system modeling, inertial sensor technology, and airborne gravity gradiometry.



SONGBING TAO received the B.S. degree from Chongqing University, Chongqing, China, in 2014, where he is currently pursuing the Ph.D. degree. His current research interests include signal processing-based fault diagnosis for wind power generation system, multichannel signal analysis and filtering, and PCA-based incipient fault diagnosis.