

Principal Component Density Estimation for Scenario Generation Using Normalizing Flows

Eike Cramer^{a,b}, Alexander Mitsos^{c,a,d}, Raúl Tempone^{e,f}, Manuel Dahmen^{a,*}

^a Forschungszentrum Jülich GmbH, Institute of Energy and Climate Research, Energy Systems Engineering (IEK-10), Jülich 52425, Germany

^b RWTH Aachen University Aachen 52062, Germany

^c JARA-ENERGY, Jülich 52425, Germany

^d RWTH Aachen University, Process Systems Engineering (AVT.SVT), Aachen 52074, Germany

^e RWTH Aachen University, Institute of Uncertainty Quantification, Aachen 52062, Germany

^f King Abdullah University of Science and Technology (KAUST), Computer, Electrical, and Mathematical Sciences & Engineering Division (CEMSE), Saudi Arabia

Abstract: Neural networks-based learning of the distribution of non-dispatchable renewable electricity generation from sources such as photovoltaics (PV) and wind as well as load demands has recently gained attention. Normalizing flow density models are particularly well suited for this task due to the training through direct log-likelihood maximization. However, research from the field of image generation has shown that standard normalizing flows can only learn smeared-out versions of manifold distributions. Previous works on normalizing flow-based scenario generation do not address this issue, and the smeared-out distributions result in the sampling of noisy time series. In this paper, we exploit the isometry of the principal component analysis (PCA), which sets up the normalizing flow in a lower-dimensional space while maintaining the direct and computationally efficient likelihood maximization. We train the resulting principal component flow (PCF) on data of PV and wind power generation as well as load demand in Germany in the years 2013 to 2015. The results of this investigation show that the PCF preserves critical features of the original distributions, such as the probability density and frequency behavior of the time series. The application of the PCF is, however, not limited to renewable power generation but rather extends to any data set, time series, or otherwise, which can be efficiently reduced using PCA.

Keywords:

renewable energy; scenario generation; normalizing flows; dimensionality reduction; principal component analysis

1 Introduction

The renewable electricity generation technologies photovoltaics (PV) and wind depend on natural occurrences and are therefore non-dispatchable. Additionally, the realization of these renewable power generation outputs exhibits uncertain and volatile behavior, which poses new challenges for the design and operation of energy systems compared to dispatchable fossil power generation (Agora Energiewende and Sandbag, 2020; Mitsos et al., 2018). To account for the uncertainty in renewable electricity sources and other relevant energy system parameters like electricity demands system operators require information about possible realizations. This information can either be provided via short-term point-forecasting (Tascikaraoglu and Uzunoglu, 2014) or via consideration of multiple scenarios (Morales et al., 2013). Here, the term scenario refers to a possible realization of the uncertain and volatile parameter over a certain time span. A set of scenarios thus refers to a collection of time series of equal length. Decision-making based on scenarios covers the broad spectrum of possible realizations and is, therefore, highly relevant for design problems (Morales et al., 2013). Scenarios can be applied in stochastic programming formulations for the design and operation of energy systems as well as energy-related scheduling tasks (Kaut and Wallace, 2003; Birge and Louveaux, 2011). The distributions of time series intervals are often unknown and consist of many non-independent dimensions due to the correlation between time steps. These distributions typically do not follow standard distribution models like multivariate Gaussians. Thus, sampling these distributions for scenario generation remains an open research question.

*Manuel Dahmen, Forschungszentrum Jülich GmbH, Institute of Energy and Climate Research, Energy Systems Engineering (IEK-10), Jülich 52425, Germany
E-mail: m.dahmen@fz-juelich.de

There are many contributions to the literature on how to generate scenarios. In general, the published approaches can be classified as univariate modeling approaches, i.e., step-by-step models, and multivariate modeling approaches, where multiple time steps are modeled in parallel (Ziel and Veron, 2018). Multivariate modeling approaches are trained on sets of scenarios created from historical time series. The equidistant segments of univariate scenarios are viewed as multidimensional points, e.g., a scenario of one day with hourly recordings is viewed as a 24-dimensional data point. Contributions on univariate modeling approaches include the traditional Box-Jenkins approach (Box et al., 1967) for sampling stochastic processes (Sharma et al., 2013) and artificial neural network (ANN)-based autoregressive models (Vagropoulos et al., 2016). Examples of multivariate modeling approaches are Gaussian mixture models (Wang et al., 2018), Copula methods (Pinson et al., 2009; Kaut and Wallace, 2011), and moment matching techniques (Chopra and Selvamuthu, 2020).

With the increase of computational power and the development of advanced machine learning algorithms, it is now possible to train specialized ANNs, so-called deep generative models (DGMs), to learn high-dimensional probability distributions without any statistical assumptions about the data. Thus, DGMs are powerful tools for multivariate modeling. In 2018, Chen et al. (2018c) proposed the use of generative adversarial networks (GANs) (Goodfellow et al., 2014) for scenario generation. Since then, scenario generation using GANs and other DGMs has become a popular topic (Chen et al., 2018c; Jiang et al., 2018, 2019; Zhang et al., 2020; Wei et al., 2019; Zhang and Zhang, 2020). Chen et al. (2018c) later extended their original work by introducing a scenario selection procedure (Chen et al., 2018b) and by using Bayesian GANs (Chen et al., 2018a). Other examples of GANs for wind and PV power output scenarios are presented in (Jiang et al., 2018; Zhang et al., 2020; Jiang et al., 2019). For more consistent convergence in training, the authors in Chen et al. (2018c); Jiang et al. (2018); Zhang et al. (2020); Jiang et al. (2019) used Wasserstein GANs (Arjovsky et al., 2017) where they enforce a Lipschitz constraint on the critic network. Besides the generation of wind and PV scenarios, GAN-based scenario generation was also applied to residential load forecasts (Gu et al., 2019) and hydro-wind-solar hybrid systems (Wei et al., 2019). Schreiber et al. (2019) study different loss functions for GANs and found the Wasserstein distance to be superior to the binary-cross-entropy. Besides GANs, a popular type of DGMs are variational autoencoders (VAEs) (Kingma and Welling, 2014). Examples of VAEs for scenario generation include electric vehicle load demand (Pan et al., 2019), hydro wind-solar hybrid systems (Zhanga et al., 2018), as well as hydro concentrated solar power hybrid systems (Qi et al., 2020).

Despite their recent success in scenario generation for renewable power generation, both GANs and VAEs show inconsistencies in training. Specifically, the Nash Equilibria obtained through GAN training are reported to be unstable, and there is no guarantee for the generator to sample from the target distribution (Arjovsky et al., 2017). The VAE uses the Evidence Lower Bound (ELBO) loss function in training which gives no concrete measure on the actual quality of the fit (Kingma and Welling, 2014).

In contrast to GANs and VAEs, a third DGM structure, normalizing flows (Dinh et al., 2017; Papamakarios et al., 2019), can directly fit the probability density function (PDF) of the unknown distribution by log-likelihood maximization. Similar to the more established transport map approach (Marzouk et al., 2016), normalizing flows model the distribution of a data set as a deterministic transformation of a Gaussian. While transport maps typically use triangular maps (Marzouk et al., 2016) to set up convex fitting problems, normalizing flows are designed using invertible neural networks to build more flexible designs. In theory, using a sufficiently expressive normalizing flow network and enough training data, the trained distribution will converge to the true distribution. However, normalizing flows are not as well established as GANs and VAEs in scenario generation yet. To our knowledge, the only works using normalizing flows in the context of energy time series are Zhang and Zhang (2020) and Ge et al. (2020) focusing on demand time series and Dumas et al. (2021) generating PV and wind electricity time series.

Recently, Brehmer and Cranmer (2020) and Behrmann et al. (2021) showed that normalizing flows are by design unable to fit distributions that lie on lower-dimensional manifolds. Instead, fitting a full dimensional normalizing flow to a manifold distribution results in exploding likelihood functions from numerically singular Jacobians (Behrmann et al., 2021) and a smeared-out version of the true distribution (Brehmer and Cranmer, 2020). Sampling from the smeared-out fit then leads to the samples outside of the true distribution, e.g., noisy time series. In 2016, Gemicci et al. (2016) presented the necessary foundations to build normalizing flows on manifolds, i.e., to perform density estimation in a

lower-dimensional space. However, they do not further elaborate on such an approach and do not report on any numerical experiments. [Brehmer and Cranmer \(2020\)](#) build normalizing flows in lower-dimensional space by fixing some of the latent space dimensions to a constant.

We find manifolds to be frequently present in energy time series due to the temporal correlation between time steps. However, [Zhang and Zhang \(2020\)](#), [Ge et al. \(2020\)](#), and [Dumas et al. \(2021\)](#) employ standard normalizing flow structures to such data, i.e., real non volume preserving transformation (RealNVP) ([Dinh et al., 2017](#)) or nonlinear independent component estimation (NICE) ([Dinh et al., 2015](#)). As a result, their works suffer from complications resulting from data manifolds. For instance, the training of [Ge et al. \(2020\)](#) results in extremely high density values ($\log p(x) \approx 4000$), which indicates that the model is trying to describe an infinite density. The validation loss shown by the authors indicates strong overfitting. However, the authors do not declare the exploding likelihood and the overfitting as issues and present the spurious results without further discussion. [Zhang and Zhang \(2020\)](#) acknowledge that their scenarios exhibit noisy behavior that does not match the expected results. However, they attribute this noise to their conditional training approach. [Dumas et al. \(2021\)](#) draw a comparison between normalizing flows, GANs, and VAEs at the example of PV, wind, and load time series. The authors find that their normalizing flow models are unable to recover the autocorrelation within the time series and instead exhibit very high fluctuations. However, they still conclude that scenarios generated from normalizing flows show good results despite the unrealistic characteristics. Based on our observation of the frequent complications resulting from data manifolds, we argue that normalizing flows should be set up with a lower-dimensional latent space to avoid generating unrealistic energy time series scenarios.

As part of this paper, we provide the following contributions: We elicit the contradiction between manifolds and the diffeomorphic transformations required for normalizing flows. Further, we use simple two-dimensional examples to highlight how data manifolds lead to the generation of unrealistic and out-of-distribution data. In addition, we show how the principal component analysis (PCA) can be used to reduce the dimensionality of energy time series effectively and show theoretically that the resulting principal component flow (PCF) does not affect the density estimation procedure, which avoids the need for balancing reconstruction and likelihood maximization losses like in [Brehmer and Cranmer \(2020\)](#). In numerical experiments, we train the PCF on data of PV and wind power generation as well as load demand in Germany in the years 2013 to 2015. For reference, we compare our results with scenarios generated using the alternative methods Copulas ([Pinson et al., 2009](#)) and Wasserstein GANs ([Chen et al., 2018c](#)). The results show that the PCF learns the distributions better than the full space normalizing flow (FSNF) and also maintains the frequency behavior of the original time series. The PCF-generated scenarios perform equally good or better compared to the Copula and W-GAN scenarios in all considered metrics.

The remainder of this paper is organized as follows: In Section 2, we introduce the general concept of normalizing flows and review the RealNVP affine coupling layer ([Dinh et al., 2017](#)) which will serve as the underlying flow structure in this paper. In Section 3, we continue the discussion started by [Brehmer and Cranmer \(2020\)](#) and [Behrmann et al. \(2021\)](#) about the effects of manifolds on normalizing flows through a clarifying toy example and show theoretically that PCA does not affect the density estimation in lower-dimensional latent space. In Section 4, we present the results of simulation studies on data of PV and wind power generation as well as load demand in Germany ([Open power systems data, 2019](#)). Finally, in Section 5 we conclude our work.

2 Density Estimation using Normalizing Flows

Normalizing flows are invertible transformations $\mathbf{f}$ between a complex target distribution and a well-described base distribution, e.g., a multivariate Gaussian ([Papamakarios et al., 2019](#)). Analog to the inverse transport map methodology ([Marzouk et al., 2016](#)), normalizing flows aim to model a complex distribution as a transformation of a simple one instead of manually deriving a complex model that fits the data. Thus, the sampling takes place in the known base distribution, and the transformation does not have to consider the randomness in the data.

A normalizing flow transformation must be set up as a diffeomorphism, i.e., both the forward and the inverse transformation must be continuously differentiable ([Papamakarios et al., 2019](#)). In their

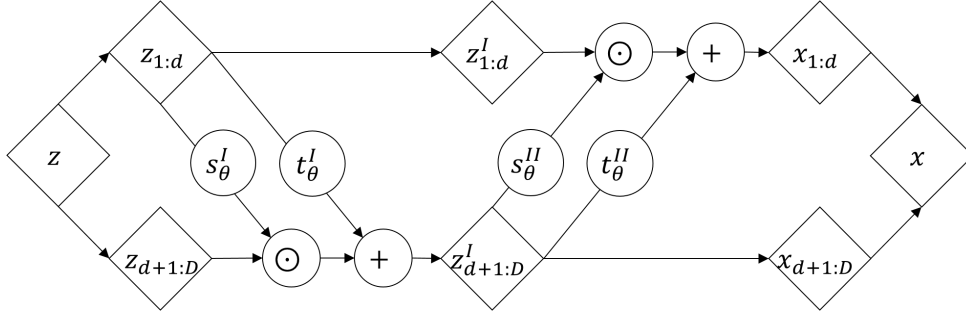


Fig. 1. Real non-volume preserving transformation (RealNVP) (Dinh et al., 2017) with two coupling layers with alternating identity and affine transformations. Arrows point in generative direction. Compositions may include more than two coupling layers. Functions $s_{\theta}^I(\mathbf{z}_{1:d})$, $t_{\theta}^I(\mathbf{z}_{1:d})$, $s_{\theta}^{II}(\mathbf{z}_{d+1:D}^I)$, and $t_{\theta}^{II}(\mathbf{z}_{d+1:D}^I)$ are trainable ANNs with parameters θ .

standard form, normalizing flows require equal dimensionality of base and target distributions, in which case the density of the target distribution is well described using the change of variables formula (CVF) (Papamakarios et al., 2019)

$$p_X(\mathbf{x}) = p_Z(\mathbf{f}^{-1}(\mathbf{x})) |\det \mathbf{J}_{\mathbf{f}}(\mathbf{f}^{-1}(\mathbf{x}))|^{-1}, \quad (1)$$

where $p_X(\mathbf{x})$ and $p_Z(\mathbf{z})$ are the densities of the samples $\mathbf{x}$ and $\mathbf{z}$ of the target distribution X and the base distribution Z , respectively, and $\mathbf{J}_{\mathbf{f}^{-1}}(\mathbf{x})$ is the Jacobian of the inverse transformation $\mathbf{f}^{-1}$. To fit complex distributions, the diffeomorphic transformation $\mathbf{f}$ needs to be flexible and expressive. However, expressive and yet easily invertible functions with tractable Jacobian determinants are often difficult to engineer. Fortunately, diffeomorphisms are composable, and therefore normalizing flow models can be built using compositions of simple transformations, i.e.,

$$\mathbf{f} = \mathbf{f}_K \circ \mathbf{f}_{K-1} \circ \dots \circ \mathbf{f}_2 \circ \mathbf{f}_1, \quad (2)$$

where $\mathbf{f}_1$ to $\mathbf{f}_K$ are simple diffeomorphisms and the operator $\circ$ denotes function composition. In logarithmic form, the CVF of compositions is given by

$$\log p_X(\mathbf{x}) = \log p_Z(\mathbf{f}^{-1}(\mathbf{x})) + \sum_{k=1}^K \log |\det \mathbf{J}_{\mathbf{f}_k}(\mathbf{f}_k^{-1}(\mathbf{x}_k))|^{-1}, \quad (3)$$

where $\mathbf{x}_k$ is an intermediate variable $\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1})$. In practice, the transformation is often set up as a trainable function $\mathbf{f}_{\theta}$ with parameters θ . By using the CVF, the transformation $\mathbf{f}_{\theta}$ can be trained via direct likelihood maximization and for numerical reasons, the log form in Equation (3) is maximized in training:

$$\max_{\theta} \log p_X(\mathbf{x}; \theta) = \log p_Z(\mathbf{f}_{\theta}^{-1}(\mathbf{x})) + \log |\det \mathbf{J}_{\mathbf{f}_{\theta}}(\mathbf{f}_{\theta}^{-1}(\mathbf{x}))|^{-1} \quad (4)$$

Here, the likelihood $p_X(\mathbf{x}; \theta)$ is parameterized by the trainable parameters θ of the transformation $\mathbf{f}_{\theta}$ and the historical samples of the target distribution $\mathbf{x}$ take the role of training data.

In the last six years, many flow construction methods have been proposed, e.g., Dinh et al. (2015) and Dinh et al. (2017). A prominent normalizing flow model is RealNVP (Dinh et al., 2017) which uses an affine coupling layer and has shown promising results in a prior application to time series data (Zhang and Zhang, 2020). The idea of the RealNVP affine coupling layer is to split the full input vector $\mathbf{z} = \mathbf{z}_{1:D}$ of dimension D and apply an affine transformation to one part of the input vector $\mathbf{z}_{d+1:D}$ conditioned on the remaining part of the input vector $\mathbf{z}_{1:d}$ that is kept constant. Here, d is usually set to $d = D/2$ to allow for maximal interaction between dimensions but can take other values $1 < d < D$, e.g., if D is

uneven. The standard forward transformation $\mathbf{f}_{CL} : \mathbf{z} \rightarrow \mathbf{x}$ is given by

$$\mathbf{x}_{1:d} = \mathbf{z}_{1:d}, \quad (5)$$

$$\mathbf{x}_{d+1:D} = \exp(\mathbf{s}_{\theta}(\mathbf{z}_{1:d})) \odot \mathbf{z}_{d+1:D} + \mathbf{t}_{\theta}(\mathbf{z}_{1:d}), \quad (6)$$

where the functions $\mathbf{s}_{\theta}(\mathbf{z}_{1:d})$ and $\mathbf{t}_{\theta}(\mathbf{z}_{1:d})$ are feed-forward ANNs called conditioner networks with parameters θ and input and output dimensions d and $D-d$, respectively. The $\odot$ operator denotes element-wise multiplication. Note that when applied as a composition in alternating form RealNVP can build flexible and easily invertible transformations with tractable Jacobian determinants. A visual description of a composition of two affine coupling layers is presented in Figure 1.

The affine coupling layer in Equations (5) and (6) has the advantage that the Jacobian of the transformation $\mathbf{f}_{CL}$ is a lower triangular matrix, i.e.,

$$\mathbf{J}_{\mathbf{f}_{CL}}(\mathbf{z}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \frac{\partial \mathbf{x}_{d+1:D}}{\partial \mathbf{z}_{1:d}} & \text{diag}[\exp(\mathbf{s}_{\theta}(\mathbf{z}_{1:d}))] \end{bmatrix}, \quad (7)$$

which means that the log of the absolute value of its determinant is simply given by

$$\log |\det(\mathbf{J}_{\mathbf{f}_{CL}}(\mathbf{z}))| = \sum_{i=d+1}^D \mathbf{s}_{\theta,i}(\mathbf{z}_{1:d}), \quad (8)$$

and the Jacobian of the inverse transformation $\mathbf{f}_{CL}^{-1} : \mathbf{x} \rightarrow \mathbf{z}$ satisfies

$$\log |\det(\mathbf{J}_{\mathbf{f}_{CL}^{-1}}(\mathbf{x}))| = - \sum_{i=d+1}^D \mathbf{s}_{\theta,i}(\mathbf{x}_{1:d}), \quad (9)$$

according to the inverse function theorem (Papamakarios et al., 2019). With Equations (8) and (9), the log-Jacobian determinant is computed from a simple evaluation of the forward or inverse transformation.

For a more detailed introduction and a review of other normalizing flow designs, the interested reader is referred to the original RealNVP paper by Dinh et al. (2017) and the review articles by Papamakarios et al. (2019) and Kobyzev et al. (2020).

3 Principal Component Density Estimation

In this section, we first discuss the effects of manifolds on the Jacobians and the contradiction between manifolds and diffeomorphic transformations and visualize how distributions get smeared out at the example of a one-dimensional distribution embedded in two-dimensional space. Secondly, we show how unlike many other dimensionality reduction techniques PCA can be used in combination with standard normalizing flows for manifold density estimation with tractable and direct likelihood computation.

3.1 Normalizing Flows and Manifolds

Normalizing flows are transformations between the space of observable variables (target distribution) and a latent space with independent variables, i.e., a space with zero covariance (base distribution) (Papamakarios et al., 2019). As a consequence, normalizing flows disentangle the information contained in the observable variables, e.g., the power generation over a given period, to the set of independent latent variables, i.e., the normalizing flow transformation eliminates the correlation between the dimensions (Papamakarios et al., 2019; Kobyzev et al., 2020). For the disentanglement to function properly, the normalizing flow transformation $\mathbf{f}$ and its inverse $\mathbf{f}^{-1}$ must be continuously differentiable, i.e., the Jacobian of $\mathbf{f}$ must be non-singular for $\mathbf{f}^{-1}$ to be differentiable and vice versa (Papamakarios et al., 2019). The distributions of many data sets do not occupy the entire space of observable variables, i.e., the distributions lie on lower-dimensional manifolds (Fefferman et al., 2016). A transformation from a latent space Gaussian with equal dimensionality to the space of observable variables must compress the

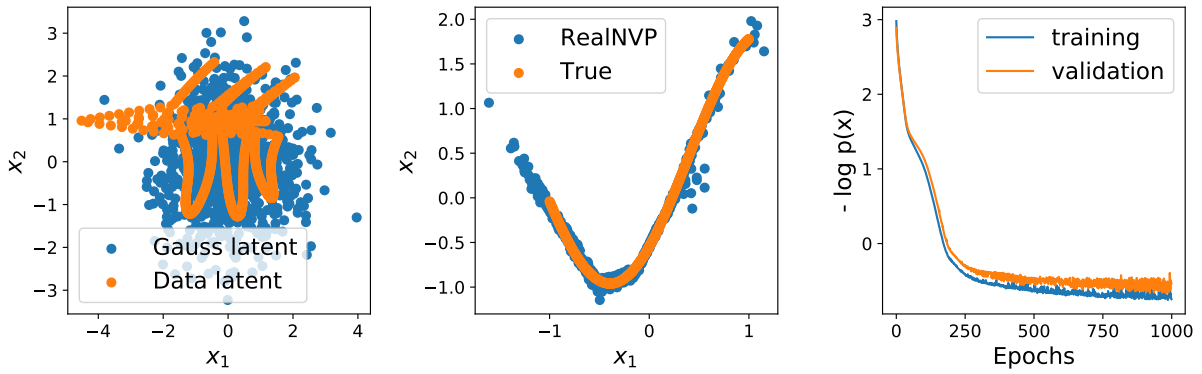


Fig. 2. RealNVP (Dinh et al., 2017) trained on 1D manifold in 2D space $e(x_1, x_2)$. Left: samples of 2D Gaussian (blue) and training data after transformation to Gaussian (orange). Center: Samples from trained RealNVP (blue) and true (orange) data distribution. Right: training and validation loss over number of epochs.

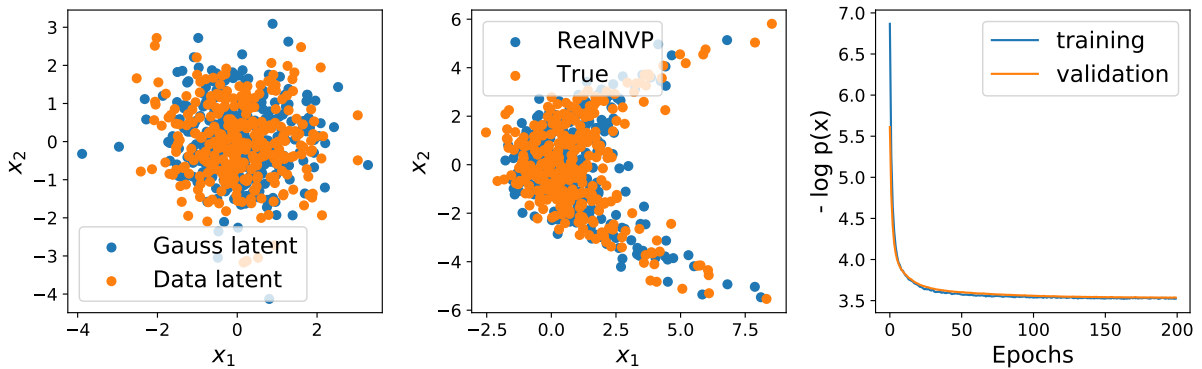


Fig. 3. RealNVP (Dinh et al., 2017) trained on 2D kite-shaped distribution in 2D space (x_1, x_2) . Left: samples of 2D Gaussian (blue) and training data after transformation to Gaussian (orange). Center: Samples from trained RealNVP (blue) and true (orange) data distribution. Right: training and validation loss over number of epochs.

Gaussian samples to the lower-dimensional space and, therefore, have a singular Jacobian (Hyvärinen and Pajunen, 1999; Behrmann et al., 2021). Consequently, a Gaussian with equal dimensionality cannot be matched to a manifold distribution using a diffeomorphic transformation.

However, normalizing flow transformations like RealNVP (Dinh et al., 2017) are designed to be diffeomorphic between spaces of equal dimensionality. When applied to manifold data, they have to include all latent space variables in the transformation with a non-zero gradient even if the data does not show any related variance. This leads to a contradiction between a problem that cannot be solved using a diffeomorphism and a strictly diffeomorphic normalizing flow. In practice, fitting a normalizing flow to manifolds leads to numerically singular Jacobians (Behrmann et al., 2021) as the training attempts to compress the Gaussian to the lower-dimensional space. For sampling, the learned distribution is then smeared out around the true distribution (Brehmer and Cranmer, 2020), causing skewed distribution densities and generation of out-of-distribution data.

To illustrate this fundamental problem, we train two RealNVP models with identical architecture on samples of a one-dimensional, uniform distribution embedded in a two-dimensional curve and a two-dimensional distribution. Figure 2 and Figure 3 show the encoded training data (Data latent) and samples of a two-dimensional Gaussian (Gauss latent) on the left, the training data (True), and samples generated using the RealNVP (RealNVP) in the center, and training and validation loss on the right

for the two different distributions, respectively. The example in Figure 2 highlights that the points from the one-dimensional curve remain on a one-dimensional curve after the transformation with the diffeomorphism. Accordingly, any point in the latent space outside of this curve has no counterpart in the true data distribution. Thus, the samples drawn in the Gaussian latent space (left) are transformed to out-of-distribution data, as can be seen by the RealNVP samples outside of the curve in the center of Figure 2. Nevertheless, the generated distributions resemble the shape of the true distributions and the out-of-distribution samples may not be noticed for higher-dimensional distributions, e.g., time series, that cannot be viewed in their ambient space.

The absolute values of the loss functions are not comparable as they describe different distributions. However, while the validation loss of the one-dimensional example in Figure 2 indicates overfitting and unstable training, the validation loss of the two-dimensional example in Figure 3 indicates good generalization, stable training, and convergence in fewer epochs. Besides the loss functions, the data transformed to the latent space in the left of Figure 3 and the RealNVP generated data in the center of Figure 3 show good matches of the Gaussian and true data distribution, respectively.

3.2 Principal Component Flow Layer

Given an available injective map $\psi : \mathbb{R}^M \rightarrow \mathbb{R}^D$, $\mathbf{x} = \psi(\tilde{\mathbf{x}})$ with $D > M$ transforming data points from the lower-dimensional latent space $\tilde{\mathbf{x}}$ to their full space embedding $\mathbf{x}$, a flow on a Riemannian manifold can be built (Gemici et al., 2016). Since the injective map expands the dimensionality, the typical CVF-relation (Equation (3)) of infinitesimal volumes using the Jacobian determinant does no longer apply. Instead, the relation between the two infinitesimal volumes is given by

$$dX = \sqrt{\det(\mathbf{J}_\psi^T \mathbf{J}_\psi)} dZ, \quad (10)$$

with Jacobian $\mathbf{J}_\psi$ of the mapping ψ . The generalized form of the CVF for the injective flow then is:

$$p_X(\mathbf{x}) = p_Z(\psi(\mathbf{x})^{-1}) \left[\det(\mathbf{J}_\psi(\psi^{-1}(\mathbf{x}))^T \mathbf{J}_\psi(\psi^{-1}(\mathbf{x}))) \right]^{-0.5} \quad (11)$$

For the injective map ψ to be applicable for the use in combination with normalizing flows, its inverse ψ^{-1} must be available and easy to compute, and the Jacobian determinant term in Equation (11) must be computationally tractable during training. Brehmer and Cranmer (2020) propose to build an injective map by fixing some of the latent space dimensions of a normalizing flow to a constant. However, their approach still results in expensive Jacobian computations and requires balancing a reconstruction loss with the likelihood maximization. To avoid expensive Jacobian computations and achieve easily invertible dimensionality reduction, we exploit the isometric, i.e., distance preserving, characteristic of the affine PCA (Pearson, 1901). PCA is based on the singular value decomposition of the sample covariance matrix $\mathbf{K}_{X,X}$ of the data distribution X , i.e.,

$$\mathbf{K}_{X,X} = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu}_X)(\mathbf{x}_i - \boldsymbol{\mu}_X)^T = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^{-1}, \quad (12)$$

where N is the number of data points, $\mathbf{x}$ are the data points, and $\boldsymbol{\mu}_X$ is the empirical mean vector of the distribution. The covariance matrix $\mathbf{K}_{X,X}$ is decomposed into a diagonal matrix of singular values $\boldsymbol{\Sigma}$ and two unitary matrices $\mathbf{U}$ and $\mathbf{V}$ of left and right singular vectors, respectively. The columns of the matrix of right singular vectors $\mathbf{V}$ corresponding to the largest singular values are called the principal components of the data distribution X . By truncating the columns with small or zero singular values, we can use the resulting semi-orthogonal matrix $\mathbf{V}_P$ for the affine embedding function:

$$\mathbf{x} = \psi_{\text{PCA}}(\tilde{\mathbf{x}}) = \mathbf{V}_P \tilde{\mathbf{x}} + \boldsymbol{\mu}_X \quad (13)$$

with $\mathbf{V}_P$ as the truncated matrix of right singular vectors and $\boldsymbol{\mu}_X$ as the mean value of the distribution (Pearson, 1901).

In general, isometries are defined via:

$$\mathbf{J}_\psi(\tilde{\mathbf{x}})^T \mathbf{J}_\psi(\tilde{\mathbf{x}}) = \mathbf{I}_M, \quad \forall \tilde{\mathbf{x}} \in \mathbb{R}^M \quad (14)$$

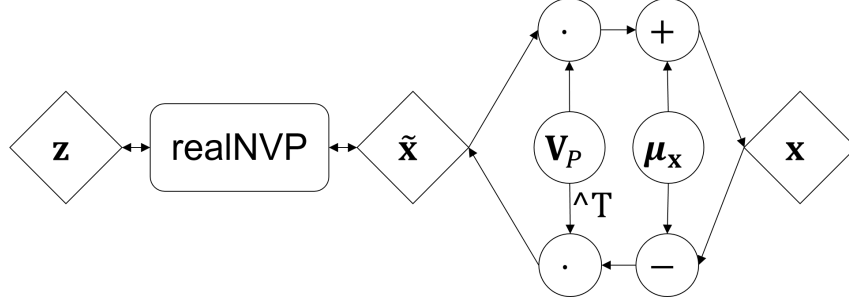


Fig. 4. Principal component flow (PCF) structure with PCA layer as last layer in generative direction and RealNVP (see Figure 1) as trainable normalizing flow in lower-dimensional space.

Here, $\mathbf{I}_M$ is the $M \times M$ identity matrix. For PCA, Equation (14) holds as the Jacobian of ψ_{PCA} is $\mathbf{J}_{\psi_{\text{PCA}}}(\tilde{\mathbf{x}}) = \mathbf{V}_P$, where $\mathbf{V}_P$ is semi-orthogonal. Thus:

$$\mathbf{J}_{\psi_{\text{PCA}}}(\tilde{\mathbf{x}})^T \mathbf{J}_{\psi_{\text{PCA}}}(\tilde{\mathbf{x}}) = \mathbf{V}_P^T \mathbf{V}_P = \mathbf{I}_M, \quad \forall \tilde{\mathbf{x}} \in \mathbb{R}^M \quad (15)$$

The isometric property of PCA results in two major advantages for the combination with normalizing flows. First, the PCA (pseudo-) inverse can be computed using the transpose of $\mathbf{V}_P$:

$$\tilde{\mathbf{x}} = \psi_{\text{PCA}}^{-1}(\mathbf{x}) = \mathbf{V}_P^T (\mathbf{x} - \boldsymbol{\mu}_X), \quad (16)$$

which makes the inverse explicit and computationally efficient. Second, the PDF described by the CVF is invariant to the PCA dimensionality reduction, as the expression in Equation (15) enters in the Jacobian determinant term of Equation (11), i.e.,

$$\left[|\det(\mathbf{J}_{\psi_{\text{PCA}}}(\tilde{\mathbf{x}})^T \mathbf{J}_{\psi_{\text{PCA}}}(\tilde{\mathbf{x}}))| \right]^{-0.5} = \left[\left| \det \left(\underbrace{\mathbf{V}_P^T \mathbf{V}_P}_{=\mathbf{I}_M} \right) \right| \right]^{-0.5} = 1, \quad \forall \tilde{\mathbf{x}} \in \mathbb{R}^M. \quad (17)$$

When we build a composition of a standard normalizing flow and the PCA following Equation (3) it becomes apparent that the PCA transformation does not influence the PDF described by the CVF:

$$\begin{aligned} \log p_X(\mathbf{x}) &= \log p_Z(\mathbf{f}^{-1} \circ \psi_{\text{PCA}}^{-1}(\mathbf{x})) + \log |\det(\mathbf{J}_{\mathbf{f}^{-1}}(\psi_{\text{PCA}}^{-1}(\mathbf{x})))| + \underbrace{\log [|\det(\mathbf{V}_P^T \mathbf{V}_P)|]}_{=0}^{-0.5} \\ &= \log p_Z(\mathbf{f}^{-1}(\tilde{\mathbf{x}})) + \log |\det(\mathbf{J}_{\mathbf{f}^{-1}}(\tilde{\mathbf{x}}))| \end{aligned} \quad (18)$$

In Equation (18), $\tilde{\mathbf{x}} = \psi_{\text{PCA}}^{-1}(\mathbf{x})$ is the lower dimensional representation (Equation (13)) and $\mathbf{f}$ is a standard normalizing flow model, e.g., RealNVP (Dinh et al., 2017). Because PCA does not influence the CVF, it can be solved prior to fitting the normalizing flow and the PCF layer does not include trainable variables for the log-likelihood maximization. Hence, we avoid composite loss functions that balance reconstruction loss and likelihood maximization as in Brehmer and Cranmer (2020). The available inverse and separation of PCA fit and normalizing flow training make the use of PCA computationally very efficient, as neither the inverse nor the Jacobian have to be computed at any point. Note that the efficient inverse and the omittable Jacobian determinant do not transfer to most other dimensionality reduction techniques, e.g., local linear embeddings (Roweis and Saul, 2000) or diffusion maps (Coifman and Lafon, 2006). A graphical description of the combination of the PCA and RealNVP is shown in Figure 4.

4 Numerical Experiments

In this section, we train the PCF on real-world time series data. The three different data sets considered contain data of PV power generation, wind power generation, and load demand, where each represents

Tab. 1. Number of principal components for cumulative explained variance (CEV). Data with 15 min resolution (96 dimensions) from [Open power systems data \(2019\)](#).

Explained variance	PV	wind	load demand
$\geq 99.00\%$	3	6	5
$\geq 99.90\%$	6	10	16
$\geq 99.99\%$	16	44	63
100.00%	62	96	96

the total values in Germany in the years 2013 to 2015 ([Open power systems data, 2019](#)). In the analysis below, we treat each of the time series independently. Prior to any processing, we clean the data of any days with missing values. To our knowledge, there are no curtailment effects in the data. PV and wind scenarios are scaled by the installed capacity at each time step. Thus, the networks are trained on the so-called capacity factor. The load demand data is scaled to the $[0, 1]$ interval resulting in what we refer to as the demand factor in the following. Learning the capacity factor distribution avoids skewness of the distribution due to the addition of further generation capacity in the given time frame. We have not observed long-term trends in the load demand data of the selected period. To be processed by the normalizing flow, the univariate time series are cut into intervals of one day each. With sampling intervals of 15 min, this leads to a set of 96-dimensional time series fragments that will serve as training data, i.e., a set of 96-dimensional vectors, which is a typical approach in the field of DGM scenario generation ([Chen et al., 2018c](#); [Zhang and Zhang, 2020](#)). This approach is known as multivariate prediction ([Ziel and Weron, 2018](#)) as multiple time steps are predicted at the same time.

For a comparison with more established scenario generation approaches, we also generate scenarios using Copulas and Wasserstein GANs (W-GANs). In contrast to autoregressive models ([Sharma et al., 2013](#)) and similar to normalizing flows, Copulas, and W-GANs describe the full distribution independent of conditional inputs. We use the linear interpolation of the inverse cumulative distribution function to fit the Copula to the data ([Pinson et al., 2009](#)), and for the W-GANs, we follow the approach by [Chen et al. \(2018c\)](#). Note that both Copulas and W-GANs use the same idea of multivariate scenario generation, i.e., treating scenarios as a vector rather than a sequence of values.

First, we investigate the manifold dimensionality of the data sets by looking at the explained variance ratio of the principal components. The explained variance ratio is the value of the singular values of the covariance matrix scaled by their sum, i.e., the relative amount of variance in the data, which is described by the right singular vector (the principal component) corresponding to the given singular value ([Tipping and Bishop, 1999](#)). As an indicator of the manifold dimensionality, we look at the cumulative explained variance (CEV) which describes how much of the information in terms of variance is maintained when the data is compressed to the space of principal components. Table 1 lists some relevant CEV values and the corresponding latent space dimensionality for our data sets. The results show that a significantly reduced dimensionality is sufficient to maintain close to all of the variance information in the data. The 100 % threshold is reached when the latent dimensionality is equal to the numerical rank of the covariance matrix. The PV data reaches this threshold at a latent dimensionality of 62. Note that the 100% CEV appears with fewer principal components than dimensions of the data, if either one or more of the dimensions has zero variance or if there is an exact linear dependency of one of the dimensions on one or more of the other dimensions. Another important thing to consider is that the PCA is fitted using the *empirical* covariance matrix, i.e., with a finite number of samples. Hence, the numerical rank might be different from the rank of the covariance matrix of the process and the low variance components identified by the PCA may not be fully representative of the underlying process. Hence, the low variance and high-frequency components identified by the PCA may not be desired.

We train full-space normalizing flow (FSNF) and PCF models on each of the three different data sets and compare the generated to the historical data (target). For each data set, we select two latent dimensionalities, namely, 16 and 62 for PV to represent 99.99% and 100% of the variance, and 6 and 10 for wind as well as 5 and 16 for demand to represent 99% and 99.9% of the variance, respectively. In each training, 20% of the 1096 scenarios are set aside as validation sets for hyperparameter tuning and to avoid overfitting. All of the following analysis is performed using the full historical scenario sets in

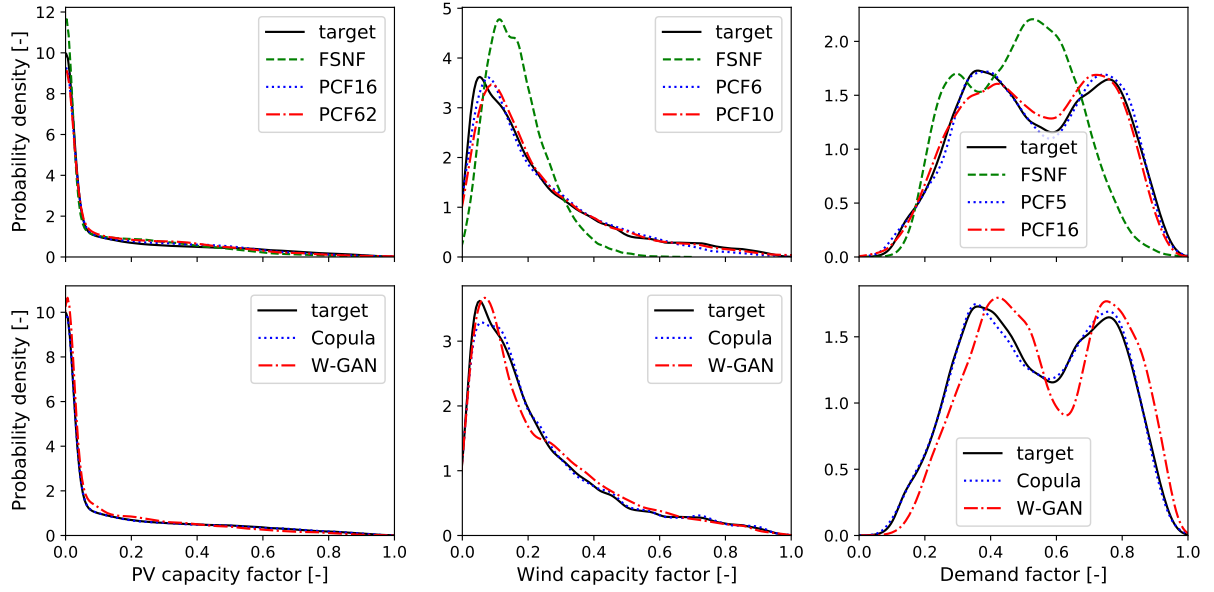


Fig. 5. Comparison of the probability density function (PDF) from kernel density estimation (Parzen, 1962). Top: historical data (target, solid lines), generated data from full space normalizing flow (FSNF, dashed lines), and generated data from PCF (PCF, dotted line and dash-dotted lines) Bottom: historical data (target, solid lines), generated data from the Copula (Copula, dotted lines), and generated data from W-GAN (W-GAN, dash-dotted lines). Left: PV capacity factor, center: wind capacity factor, right: demand factor.

comparison to sets of generated scenarios, to avoid random errors in the evaluation as a result of too few data points.

We find that, for all networks, five RealNVP affine coupling layers with fully connected conditioner networks with two hidden layers of the same size as the input dimension each is sufficient and additional complexity does not improve the representation. All normalizing flow models are implemented using the Python-based machine learning libraries Tensorflow version 2.4.0 (Abadi and Agarwal, 2015) and Tensorflow-Probability version 0.12.1 (Dillon et al., 2017). We use the PCA routine from the open-source scikit-learn Python library version 0.24.0 (Pedregosa et al., 2011) for dimensionality reduction.

We estimate the probability density function (PDF) of the data using kernel density estimation (KDE) with Gaussian kernels (Parzen, 1962) as shown in Figure 5. The top line of Figure 5 shows the PDF plots of the historical scenarios, FSNF-generated scenarios, and scenarios generated using the proposed PCF for each of the three data sets. For reference, Figure 5 compares the Copula and W-GAN-generated scenario PDFs with the historical scenarios, in the second line.

For the PV data in the left of Figure 5, the data from all three normalizing flow structures appear to match the distribution. The FSNF data exhibits a higher density than the target for capacity factors smaller than 0.05, between 0.1 and 0.45, and a lower density above a certain capacity factor of 0.45. The PCF62 data shows an overestimation between a capacity factor of 0.1 and 0.4, however, the PCF62 density is closer to the density of the target distribution than the FSNF density, in particular, for capacity factors below 0.1. Finally, the PCF16 data distribution is closest to the target distribution with good fits for capacity factors smaller than 0.2 and similar deviations from the target as observed with the FSNF and the PCF62.

For the wind capacity factor (center in Figure 5) and the demand factor (right in Figure 5), we observe a more drastic overestimation of the PDF for the highest densities and an underestimation of the tails by the FSNF. As for the PV scenarios, the density estimates for wind and load demand scenarios show an improved approximation of the distribution through dimensionality reduction with PCA, although the covariance matrices are not rank deficient. It appears that the FSNF tends to ignore rare events during the

Tab. 2. P-values (p-value $\in [0, 1]$) of Kolmogorov–Smirnov test (Hodges, 1958): Statistical comparison of historical and generated data. High p-values (≥ 0.1) indicate high probability that generated scenarios have same distribution as the historical scenarios. Results indicating a good match of the distributions are marked with an asterisk *.

	FSNF	W-GAN	Copula	PCF5	PCF6	PCF10	PCF16	PCF62
Solar	2.67×10^{-22}	1.21×10^{-04}	0.4911*				0.2483*	2.07×10^{-5}
Wind	6.69×10^{-20}	0.268*	0.4782*	0.9502*		0.0537		
Demand	3.85×10^{-81}	6.35×10^{-21}	0.7727*		0.2380*		0.0361	

likelihood maximization and thus overestimates the areas of high density. With a lower dimensionality, rare events occupy a relatively larger space and are therefore more likely to be considered by the PCF likelihood maximization. Furthermore, the PDF fits of both wind and demand appear to improve by considering fewer principal components. By adding more principal components to get from 99% to 99.9% CEV, the training problem becomes less well-conditioned as the additional dimensions are only narrowly occupied by data, i.e., the distribution comes closer to a lower-dimensional manifold.

The bottom line of Figure 5 shows good matches of PV and wind capacity factor PDFs for both Copulas and W-GANs. The Copula scenarios also show a matching demand factor PDF, while the W-GAN results in a skewed PDF. A comparison of the top and bottom lines in Figure 5 highlights that PDF representations by the PCF are as good or better compared to the literature approaches.

Note that an exact evaluation of the KDE plot can be misleading since the KDE results are not exact, in particular, at the tails of the distribution and in low-density parts (Wied and Weißbach, 2012). However, the trends for all three data sets show improved fits of the PCF setups compared to the FSNF, which indicates better distribution fits due to the dimensionality reduction.

To support the results from the visual comparison of KDE plots, we run a Kolmogorov–Smirnov test (KS-test) (Hodges, 1958). The KS-test is used to judge whether two distinct sets of samples stem from the same distribution. Table 2 lists the p-values (p-value $\in [0, 1]$) of the tests. In general, high p-values (≥ 0.1) can be interpreted as a match of distributions, and low p-values indicate that the two sample sets stem from different distributions (Hodges, 1958). For all three data sets, the p-values of the FSNF indicate that the historical and the generated data do not follow the same distribution. In contrast, the p-values for PCF16 for PV, PCF6 for wind, and PCF5 for load demand are ≥ 0.1 , indicating a good representation of the historical data distribution. The PCF10 for wind and the PCF16 for demand show p-values ≥ 0.05 and ≥ 0.01 which still indicate good matches of the distribution and significantly better matches compared to the FSNF. The PCF62 for PV gives a lower p-value and hence lower confidence that the data stems from the historical data distribution. This result matches the observations in Figure 5, where we observed a slightly worse fit from PCF62 compared to PCF16. Still, there is a significant improvement compared to the FSNF. Furthermore, the KS-test confirms our observation of increasing quality of fit for the 99% CEV compared to the 99.9% CEV.

The KS-test of the reference models, returns high p-values for all datasets for the Copula-generated scenarios, while only the wind capacity factor shows a high p-value among the W-GAN-generated scenarios. In general, the KS-test indicates good matching distributions only for Copulas and for the PCFs with low numbers of principal components.

In conclusion, the PCF returns significantly better fits of the distribution compared to the FSNF. Furthermore, the PDF fits by the PCF are at least as good or better compared to the literature methods. Considering fewer principal components appears to enhance the learning of the distributions, and accepting CEV values around 99% does not compromise the overall match of the distribution.

We use the power spectral density (PSD) based on the Welch transform (Welch, 1967) to investigate in the frequency domain whether the FSNF and the PCF can reproduce the fluctuational behavior of the original time series. Figures 6 shows the PSD of the target distribution, the FSNF, and the PCF generated scenarios for the three considered data sets, respectively.

For the PV data, the target distribution shows high amplitudes ($> 10^{-4}$) for low frequencies up to $\frac{1}{3.5h}$ which are well-matched by all three DGMs. For higher frequencies ($> \frac{1}{3.5h}$), the target amplitude declines faster than the FSNF amplitudes, indicating the generation of noisy behavior over shorter periods

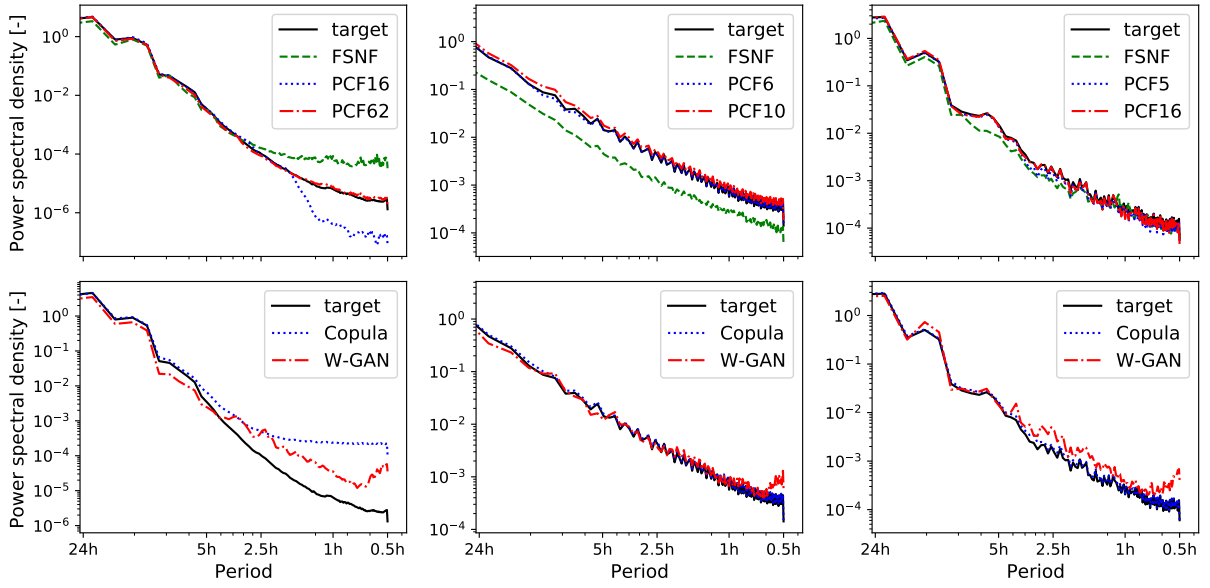


Fig. 6. Power spectral density with Welch transform (Welch, 1967). Top: historical data (target, solid lines), generated data from full space normalizing flow (FSNF, dashed lines), and generated data from PCF (PCF, dotted line and dash-dotted lines). Bottom: historical data (target, solid lines), generated data from the Copula (Copula, dotted lines), and generated data from W-GAN (W-GAN, dash-dotted lines). Left: PV capacity factor, center: wind capacity factor, right: demand factor.

by the FSNF. In contrast, PCF16 produces lower amplitudes than the target distribution in the regime of frequencies $> \frac{1}{2h}$, which indicates the filtering of some of the low amplitude fluctuation in the historical data set. The PCF62 shows a good fit for the true behavior of the target up to the highest frequencies.

The PSD analysis for the wind data shows a good match of the frequency behavior for the data generated from the PCF6 and PCF10. The FSNF generated scenarios show a general underestimation of the amplitudes. Note that the wind data exhibits higher fluctuation on all frequencies compared to the PV data. Therefore, the fluctuations at high-frequencies are represented in the principal components, and no filter effect can be observed. As the overall PSD shape is matched, the FSNF appears to find the right fluctuation behavior only with consistently lower amplitudes. Here, the narrower distribution described by the FSNF (see Figure 5) leads to a smaller range of scenarios and, therefore, lower amplitudes.

As for the wind scenarios, the PSD analysis of the load demand data in Figure 6 shows a good match of the PCF-generated scenarios to the historical data. The FSNF matches the overall PSD, but there are lower amplitudes between frequencies of $> \frac{1}{8h}$ and $> \frac{1}{2h}$.

The PSDs of the W-GAN-generated scenarios show an increase in amplitude for frequencies between $\frac{1}{1h}$ and $\frac{1}{0.5h}$ for all three datasets, indicating noisy scenarios. Furthermore, the PSDs of the PV capacity factor and the demand factor of the W-GAN-generated scenarios are higher compared to the historical scenario PSDs, i.e., the W-GANs generate scenarios with unrealistically high fluctuations. The Copula-generated scenarios reproduce the PSDs of the wind capacity factor and the demand factor well. However, the Copula-generated PV capacity factor scenarios show high amplitudes for frequencies $> \frac{1}{5h}$.

In conclusion, the PSD analysis of the PCF-generated scenarios clearly shows improved results for all data sets due to the dimensionality reduction compared to the FSNF. Some effects are specific to the PCA dimensionality reduction of the given data, e.g., the filter effect observed for the PV data and the noise in the case of the load demand data. Among all considered scenario generation approaches, the PCF is the only approach that does not result in noisy behavior or otherwise higher fluctuations for all three datasets.

Finally, we investigate whether the PCF recovers the dimensions with zero variance that occur during the nighttime hours of the PV scenarios. Figure 7 shows the time between midnight and 4am of five

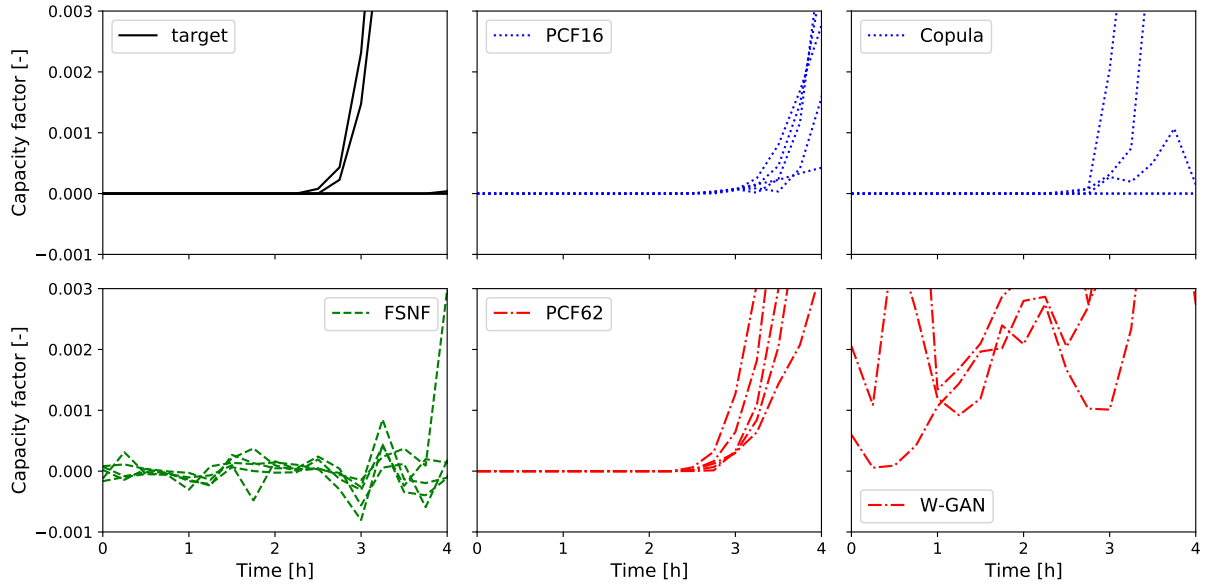


Fig. 7. Five PV capacity factor scenarios from target, FSNF, PCF16, PCF62, Copula, and W-GAN respectively. Time frame between midnight and 4 am.

randomly selected scenarios from the historical data, the normalizing flow models, and Copula and W-GAN-generated scenarios. The FSNF-generated scenarios clearly show noisy behavior. On the other hand, PCF16 and PCF62 do not show any noise and thus preserve the zero variance feature of the respective dimensions. The W-GAN scenarios show high fluctuations without any structure, while the Copula scenarios appear to identify the zero variance of the nighttime hours correctly. In addition to Figure 7, Table 3 lists the mean and variance for the time steps between midnight and 4 am. The mean and variance values for PCF16 and PCF62 confirm the observations from Figure 7. Figure 7 and Table 3 highlight the ‘smeared-out’ characteristic of the learned distribution as described in Section 3.1 and by Brehmer and Cranmer (2020). The results show that the FSNF is unable to detect dimensions of zero variance and does not accurately fit the manifold distribution, while PCF detects the manifold and correctly reproduces its features. In contrast to Figure 7, the mean and variance values of the Copula-generated scenarios highlight that the Copula does not identify the zero variance characteristic of the nighttime hours. It appears that the Copula reproduces the zero variance characteristic for most but not

Tab. 3. Marginal mean and variance values for PV scenarios between midnight and 4 am. Historical scenarios in comparison to FSNF, PCF16, and PCF62-generated scenarios as well as Copula and W-GAN generated scenarios.

	Mean						Variance					
	Historical	FSNF	PCF16	PCF62	Copula	W-GAN	Historical	FSNF	PCF16	PCF62	Copula	W-GAN
0:00	0	-3.94×10^{-5}	0	0	9.88×10^{-5}	0.0046949	0	-3.94×10^{-5}	0	0	9.88×10^{-5}	0.00469491
0:15	0	4.91×10^{-5}	0	0	0.0018714	0.0030314	0	4.91×10^{-5}	0	0	0.00187143	0.00303145
0:30	0	3.89×10^{-5}	0	0	0.000323	0.0048844	0	3.89×10^{-5}	0	0	0.00032296	0.00488435
0:45	0	-7.77×10^{-5}	0	0	0.0007804	0.0029591	0	-7.77×10^{-5}	0	0	0.00078039	0.0029591
1:00	0	-0.0001856	0	0	0.0005853	0.0030573	0	-0.00018558	0	0	0.00058527	0.00305727
1:15	0	-2.28×10^{-5}	0	0	0	0.0028363	0	-2.28×10^{-5}	0	0	-1.00×10^{-10}	0.00283627
1:30	0	0.0001475	0	0	0	0.0028889	0	0.00014755	0	0	-8.00×10^{-10}	0.00288887
1:45	0	0.0001189	0	0	0.0005246	0.0032642	0	0.0001189	0	0	0.00052463	0.00326423
2:00	0	-1.14×10^{-5}	0	0	0	0.0030402	0	-1.14×10^{-5}	0	0	6.00×10^{-10}	0.00304017
2:15	0	-1.81×10^{-5}	0	0	0	0.0031055	0	-1.81×10^{-5}	0	0	6.00×10^{-10}	0.00310545
2:30	4.00×10^{-6}	9.91×10^{-5}	3.10×10^{-6}	1.64×10^{-5}	0.0008303	0.0028191	4.04×10^{-6}	9.91×10^{-5}	3.05×10^{-6}	1.64×10^{-5}	0.00083027	0.00281913
2:45	4.21×10^{-5}	-0.0001062	3.31×10^{-5}	0.0001311	0.0005581	0.0029713	4.21×10^{-5}	-0.00010616	3.31×10^{-5}	0.00013112	0.00055814	0.00297128
3:00	0.0002261	-0.0002776	0.0001789	0.0004552	0.0002479	0.0029983	0.00022609	-0.00027757	0.00017895	0.00045525	0.00024788	0.00299829
3:15	0.0006365	0.000326	0.0005092	0.0010372	0.0013232	0.0037313	0.00063655	0.00032604	0.00050917	0.00103717	0.00132324	0.00373134
3:30	0.0016171	0.000278	0.0013039	0.0022794	0.0026894	0.0047879	0.00161713	0.00027802	0.00130388	0.00227943	0.0026894	0.00478792
3:45	0.0032558	0.0006409	0.0026619	0.0040074	0.0040934	0.0062684	0.00325582	0.00064086	0.00266187	0.00400742	0.00409343	0.00626838
4:00	0.0058694	0.0021747	0.0049369	0.0066868	0.0068678	0.009123	0.0058694	0.00217466	0.00493695	0.00668681	0.0068678	0.00912296

all of the scenarios.

In conclusion, the PCF is the only method among the considered approaches that consistently identifies the zero variance characteristic of the PV scenarios during nighttime.

5 Conclusion

In this paper, we illustrate the issues associated with learning the probability distribution of electricity time series scenarios using normalizing flow density models. We find that these distributions lie on lower-dimensional manifolds due to the high autocorrelation between time steps. We show that standard normalizing flows are ill-suited to learn these manifold distributions and, instead, induce ill-conditioned or singular Jacobians that interfere with the normalizing flow training. However, this problem was previously not addressed in the literature on scenario generation, and we argue that this has led to some spurious results. To mitigate the problems arising from data manifolds, we use dimensionality reduction. Here, we exploit the isometric characteristic of the principal component analysis (PCA). We show theoretically that the PCA does not influence the density estimation since its log-Jacobian determinant is by design always zero. Thus, we avoid having to balance reconstruction loss and likelihood maximization. Furthermore, no expensive Jacobian or matrix inverse computations are necessary, which makes PCA very computationally efficient. Critically, neither the explicit inverse nor the omissible Jacobian determinant apply to most other dimensionality reduction techniques.

We apply the combined principal component flow (PCF) next to a standard full space normalizing flow (FSNF) to learn the distribution of PV capacity factors, wind capacity factors, and load demand scenarios from Germany from 2013 to 2015. The results show that the PCF can generate realistic PV, wind, and load demand scenarios despite a significant dimensionality reduction, while the FSNF overestimates the probability density function (PDF) in areas of high density and underestimates the tails. Moreover, the PDF fits improve with 99% cumulative explained variance (CEV) as opposed to 99.9% CEV, as the training problem becomes better conditioned. Besides the PDF, the power spectral density (PSD) analysis reveals that PCF recovers the general periodic behavior of the time series. As observed for the PV scenarios, PCA can introduce a potentially desired filter effect. On the other hand, there may be a trade-off between accurate distribution fits and inclusion of all the details in the data set. Therefore, it is critical to test how PCA-based dimensionality reduction affects the data and the final selection of principal components should always be made considering the desired application of the scenarios.

In addition to our analysis of the PCF-generated scenarios, we run a comparison to two more established methods, namely Copulas and Wasserstein-GANs. Our results highlight that the PCF performs either similarly good or better in all considered metrics compared to the literature methods. In fact, the PCF is the only method that shows good matches in the PDFs, scores high p-values in the KS-test, does not have any unexpected fluctuation, and recovers the zero variance characteristic of the photovoltaic time series during the nighttime.

In conclusion, our results highlight the importance of considering the inherent latent dimensionality of the data when using normalizing flow generative models, and that the combination of PCA dimensionality reduction and normalizing flows can build powerful distribution models for energy time series scenarios.

Acknowledgements

This work was performed as part of the Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE) and received funding from the Helmholtz Association of German Research Centres. This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-2019-CRG8-4033, and the Alexander von Humboldt Foundation.

We thank the anonymous reviewers for their valuable feedback which greatly helped us to improve our manuscript.

Nomenclature

Acronyms

ANN	Artificial neural network
CEV	Cumulative explained variance
CVF	Change of variables formula
DGM	Deep generative model
FSNF	Full space normalizing flow
GAN	Generative adversarial network
KDE	Kernel density estimation
NICE	Nonlinear independent component estimation
PV	Photovoltaic
PSD	Power spectral density
PCA	Principal component analysis
PCF	Principal component flow
PDF	Probability density function
realNVP	Real non-volume preserving transformation
VAE	Variational autoencoder

Greek letters

θ	Trainable parameters of conditioner networks s_θ and t_θ
μ_X	Mean value of distribution X
ψ	Injective transformation

Latin letters

d	Dimension
D	Dimensionality of target distribution
f	Transformation
f_{CL}	Coupling layer transformation
$\mathbf{I}$	Identity matrix
$\mathbf{J}$	Jacobian
K	Number of transformations in composition
M	Dimensionality $< D$
p_X/p_Y	Probability density of X and Y
s_θ	Conditioner network
t_θ	Conditioner network
$\mathbf{V}$	Matrix of right singular vectors
$\mathbf{V}_P$	Truncated matrix of right singular vectors
$\mathbf{x}$	Target distribution sample
$\tilde{\mathbf{x}}$	Compressed target distribution sample
X	Target distribution
$\mathbf{z}$	Base distribution sample
Z	Base distribution

Bibliography

- Abadi, M. and Agarwal, A. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Agora Energiewende and Sandbag (2020). The European power sector in 2019: Up-to-date analysis on the electricity transition. *Agora Energiewende and Sandbag*, page 49.

- Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.
- Behrmann, J., Vicol, P., Wang, K.-C., Grosse, R., and Jacobsen, J.-H. (2021). Understanding and mitigating exploding inverses in invertible neural networks. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1792–1800. PMLR.
- Birge, J. R. and Louveau, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.
- Box, G. E., Jenkins, G. M., and Bacon, D. W. (1967). Models for forecasting seasonal and non-seasonal time series. Technical report, Wisconsin Univ. Madison Department of Statistics.
- Brehmer, J. and Cranmer, K. (2020). Flows for simultaneous manifold learning and density estimation. *arXiv preprint arXiv:2003.13913*.
- Chen, Y., Li, P., and Zhang, B. (2018a). Bayesian renewables scenario generation via deep generative networks. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE.
- Chen, Y., Wang, X., and Zhang, B. (2018b). An unsupervised deep learning approach for scenario forecasts. In *2018 Power Systems Computation Conference (PSCC); Dublin, Ireland; 11-15 June 2018*, pages 1–7. IEEE.
- Chen, Y., Wang, Y., Kirschen, D., and Zhang, B. (2018c). Model-free renewable scenario generation using generative adversarial networks. *IEEE Transactions on Power Systems*, 33(3):3265–3275.
- Chopra, I. and Selvamuthu, D. (2020). Scenario generation in stochastic programming using principal component analysis based on moment-matching approach. *OPSEARCH*, 57(1):190–201.
- Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30.
- Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. (2017). Tensorflow distributions. *arXiv preprint arXiv:1711.10604*.
- Dinh, L., Krueger, D., and Bengio, Y. (2015). NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.
- Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*.
- Dumas, J., Lanaspeze, A. W. D., Cornélusse, B., and Sutera, A. (2021). A deep generative model for probabilistic energy forecasting in power systems: Normalizing flows. *arXiv preprint arXiv:2106.09370*.
- Fefferman, C., Mitter, S., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049.
- Ge, L., Liao, W., Wang, S., Bak-Jensen, B., and Pillai, J. R. (2020). Modeling daily load profiles of distribution network for scenario generation using flow-based generative network. *IEEE Access*, 8:77587–77597.
- Gemici, M. C., Rezende, D., and Mohamed, S. (2016). Normalizing flows on Riemannian manifolds. *arXiv preprint arXiv:1611.02304*.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680. MIT Press, Cambridge, MA, USA.

- Gu, Y., Chen, Q., Liu, K., Xie, L., and Kang, C. (2019). GAN-based model for residential load generation considering typical consumption patterns. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE.
- Hodges, J. L. (1958). The significance probability of the Smirnov two-sample test. *Arkiv för Matematik*, 3:469–486.
- Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439.
- Jiang, C., Chen, Y., Mao, Y., Chai, Y., and Yu, M. (2019). Forecasting spatio-temporal renewable scenarios: A deep generative approach. *arXiv preprint arXiv:1903.05274*.
- Jiang, C., Mao, Y., Chai, Y., Yu, M., and Tao, S. (2018). Scenario generation for wind power using improved generative adversarial networks. *IEEE Access*, 6:62193–62203.
- Kaut, M. and Wallace, S. W. (2003). *Evaluation of scenario-generation methods for stochastic programming*. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik.
- Kaut, M. and Wallace, S. W. (2011). Shape-based scenario generation using copulas. *Computational Management Science*, 8(1-2):181–199.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kobyzev, I., Prince, S. J. D., and Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *arXiv preprint arXiv:1908.09257*.
- Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. (2016). Sampling via measure transport: An introduction. *Handbook of Uncertainty Quantification*, page 1–41.
- Mitsos, A., Asprion, N., Floudas, C. A., Bortz, M., Baldea, M., Bonvin, D., Caspari, A., and Schäfer, P. (2018). Challenges in process optimization for new feedstocks and energy sources. *Computers & Chemical Engineering*, 113:209–221.
- Morales, J. M., Conejo, A. J., Madsen, H., Pinson, P., and Zugno, M. (2013). *Integrating Renewables in Electricity Markets: Operational Problems*, volume 205. Springer Science & Business Media, 1 edition.
- Open power systems data (2019). Time series. https://doi.org/10.25832/time_series/2019-06-05. Accessed: 2020-03-30.
- Pan, Z., Wang, J., Liao, W., Chen, H., Yuan, D., Zhu, W., Fang, X., and Zhu, Z. (2019). Data-driven EV load profiles generation using a variational auto-encoder. *Energies*, 12(5).
- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2019). Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*.
- Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Pinson, P., Madsen, H., Nielsen, H. A., Papaefthymiou, G., and Klöckl, B. (2009). From probabilistic forecasts to statistical scenarios of short-term wind power production. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(1):51–62.

- Qi, Y., Hu, W., Dong, Y., Fan, Y., Dong, L., and Xiao, M. (2020). Optimal configuration of concentrating solar power in multienergy power systems with an improved variational autoencoder. *Applied Energy*, 274:115124.
- Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.
- Schreiber, J., Jessulat, M., and Sick, B. (2019). Generative adversarial networks for operational scenario planning of renewable energy farms: A study on wind and photovoltaic. In *International Conference on Artificial Neural Networks*, pages 550–564. Springer.
- Sharma, K. C., Jain, P., and Bhakar, R. (2013). Wind power scenario generation and reduction in stochastic programming framework. *Electric Power Components and Systems*, 41(3):271–285.
- Tascikaraoglu, A. and Uzunoglu, M. (2014). A review of combined approaches for prediction of short-term wind speed and power. *Renewable and Sustainable Energy Reviews*, 34:243–254.
- Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.
- Vagropoulos, S. I., Kardakos, E. G., Simoglou, C. K., Bakirtzis, A. G., and Catalao, J. P. (2016). ANN-based scenario generation methodology for stochastic variables of electric power systems. *Electric Power Systems Research*, 134:9–18.
- Wang, Z., Shen, C., and Liu, F. (2018). A conditional model of wind power forecast errors and its application in scenario generation. *Applied Energy*, 212:771–785.
- Wei, H., Hongxuan, Z., Yu, D., Yiting, W., Ling, D., and Ming, X. (2019). Short-term optimal operation of hydro-wind-solar hybrid system with improved generative adversarial networks. *Applied Energy*, 250:389–403.
- Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73.
- Wied, D. and Weißbach, R. (2012). Consistency of the kernel density estimator: a survey. *Statistical Papers*, 53(1):1–21.
- Zhang, L. and Zhang, B. (2020). Scenario forecasting of residential load profiles. *IEEE Journal on Selected Areas in Communications*, 38(1):84–95.
- Zhang, Y., Ai, Q., Xiao, F., Hao, R., and Lu, T. (2020). Typical wind power scenario generation for multiple wind farms using conditional improved Wasserstein generative adversarial network. *International Journal of Electrical Power & Energy Systems*, 114:105388.
- Zhanga, H., Hua, W., Yub, R., Tangb, M., and Dingc, L. (2018). Optimized operation of cascade reservoirs considering complementary characteristics between wind and photovoltaic based on variational auto-encoder. In *MATEC Web of Conferences*, volume 246, page 01077. EDP Sciences.
- Ziel, F. and Weron, R. (2018). Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. *Energy Economics*, 70:396–420.