A Bayesian Approach for Estimating Uncertainty in Stochastic Economic Dispatch Considering Wind Power Penetration

Zhixiong Hu, Yijun Xu, *Member, IEEE*, Mert Korkali, *Senior Member, IEEE*, Xiao Chen, Lamine Mili, *Life Fellow, IEEE*, and Jaber Valinejad, *Student Member, IEEE*

***Abstract*—The increasing penetration of renewable energy resources in power systems, represented as random processes, converts the traditional deterministic economic dispatch problem into a stochastic one. To estimate the uncertainty in the stochastic economic dispatch (SED) problem for the purpose of forecasting, the conventional Monte-Carlo (MC) method is prohibitively time-consuming for practical applications. To overcome this problem, we propose a novel Gaussian-process-emulator (GPE)-based approach to quantify the uncertainty in SED considering wind power penetration. Facing high-dimensional real-world data representing the correlated uncertainties from wind generation, a manifold-learning-based Isomap algorithm is proposed to efficiently represent the low-dimensional hidden probabilistic structure of the data. In this low-dimensional latent space, with Latin hypercube sampling (LHS) as the computer experimental design, a GPE is used, for the first time, to serve as a nonparametric, surrogate model for the original complicated SED model. This reduced-order representative allows us to evaluate the economic dispatch solver at sampled values with a negligible computational cost while maintaining a desirable accuracy. Simulation results conducted on the IEEE 118-bus test system reveal the impressive performance of the proposed method.**

***Index Terms*—Stochastic economic dispatch, reduced-order modeling, manifold learning, uncertainty estimation, renewable energy.**

Manuscript received April 14, 2020; revised June 9, 2020 and July 18, 2020; accepted August 5, 2020. Date of publication August 10, 2020; date of current version December 16, 2020. This work was supported, in part, by the U.S. National Science Foundation under Grant 1917308 and by the United States Department of Energy Office of Electricity Advanced Grid Modeling Program, and performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Document released as LLNL-JRNL-791972. Paper no. TSTE-00398-2020. *(Corresponding author: Yijun Xu.)*

Zhixiong Hu is with the Statistical Department, University of California-Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: zhu95@ucsc.edu).

Yijun Xu, Lamine Mili, and Jaber Valinejad are with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Northern Virginia Center, Falls Church, VA 24061 USA (e-mail: xuyijun2012app@gmail.com; lmili@vt.edu; jabervalinejad@vt.edu).

Mert Korkali is with the Computational Engineering Division, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA (e-mail: korkali1@llnl.gov).

Xiao Chen is with the Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550 USA (e-mail: chen73@llnl.gov).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSTE.2020.3015353

I. Introduction

POWER systems are inherently stochastic. Sources of stochasticity include time-varying loads, renewable energy intermittencies, and random outages of generating units, lines, and transformers, to cite a few. These stochasticities translate into uncertainties in the power system models. To address this problem, research activities have focused on estimating uncertainty in power system planning, monitoring, and control [1]–[7]. Among them, the topic of stochastic economic dispatch (SED) has recently attracted considerable academic attention due to the increasing penetration of renewable energy resources, which brings randomness in the economic dispatch model that are, therefore, threatening the flexibility in the next-day operations [8]–[12].

To account for these uncertainties and provide a more reliable day-ahead forecast, some researchers propose to adopt a scenario-based optimization approach. However, this approach only considers a finite set of sampling realizations, which is obviously an oversimplification of the numerous cases that may occur in reality [10], [11]. By contrast, other researchers propose to make use of uncertainty-quantification (UQ) techniques to better describe the inherent stochastic properties of the system response. However, the traditional Monte-Carlo (MC)-sampling-based methods are prohibitively time-consuming when accurate estimation of uncertain model outputs is needed [13]. This problem calls for the development of new computationally efficient and accurate uncertainty modeling techniques for power system applications [8], [12].

To address this need, some researchers propose to adopt the surrogate-based approaches to reduce the computation time required by the MC method. Safta *et al.* [8] were the first to apply polynomial chaos expansions as an SED surrogate model. Thereafter, Li *et al.* [12] further improved this method by using the compressed samples to reform polynomial chaos expansion. However, polynomial chaos is a typical parametric method that suffers from "the curse of dimensionality" [14], and therefore calls for some dimensionality-reduction techniques such as the Karhunen-Loève expansion (KLE). However, this linear-transformation-based method is inadequate to preserve information in data in a nonlinear space. Its normality assumption for the latent variables cannot be guaranteed in practice either [8], [12].

To overcome the aforementioned weakness, this paper proposes a new framework of UQ in the SED problem based on the Gaussian process emulator (GPE) and Isomap that demonstrate significant improvements of the existing method. The contributions of the paper consist of the following:

- To reduce the computational burden in the MC method and avoid using a parametric surrogate, this paper proposes to utilize a nonparametric surrogate, i.e., the GPE [15], to conduct UQ in the SED problem for the first time. Note that unlike some other applications, such as wind power forecast, that mainly utilizes GPE as a statistical regression tool [16], our SED mainly utilizes the advantages of the GPE method as a reduced-order modeling tool for the purpose of computational efficiency.
- To reduce the dimension for the GPE-based surrogate, this paper proposes, for the first time, to adopt a manifold-learning-based Isomap method [17], [18]. This nonlinear dimensionality-reduction technique allows us to better preserve the low-dimensional embedding of the data compared with the linear-transformation-based KLE method. Furthermore, to abandon the normality assumption for the latent variables that are adopted in the KLE, a kernel density estimator (KDE) is utilized to obtain the closed-form probability of the density functions.
- Since Isomap is unable to conduct a nonlinear dimension recovery after the dimension reduction while the latent variables cannot be evaluated in SED-based physical model directly, we further merge extra GPEs into the nonlinear dimension-recovery procedure. This finally enables the closed-loop combination of these advanced machine-learning tools applicable to this important practical application.
- One advantage of the proposed dimension-reduction and dimension-recovery framework is that it enables the nonlinear-transformation-based dimension-reduction technique to be compatible with a general surrogate-based UQ problem.

All the above contributions enable us to finally obtain an excellent performance the proposed method over the traditional method on the IEEE 118-bus test system using the real-world wind farm data. The remainder of this paper is structured as follows: in Section II, the problem formulation is presented. In Section III, the mathematical background for the proposed framework is presented. Section IV describes the proposed method. Section V presents the case study. Conclusions and future work are provided in Section VI.

II. Problem Formulation

Traditionally, under some physical and economic constraints, power system economic dispatch is known as a deterministic optimization problem. It aims to identify an optimal set of power outputs of a fixed set of online thermal generating units that yield a minimum cost, denoted by $Q(\mathbf{g})$. The latter is generally thought to be nonrandom since the traditional thermal generating units $\mathbf{u}$ can be optimized and set equal to some deterministic optimal values. However, in the face of an increasing penetration of renewable energy resources, the abovementioned statement cannot hold true. Due to the intrinsic randomness of the renewable generation, represented (using random fields) as functions of a vector of random variables, $\boldsymbol{\omega}$, denoted by $\mathbf{p}(\boldsymbol{\omega})$, the deterministic economic problem for finding $Q(\mathbf{g}) = \arg\min_{\mathbf{g}}\{f_{\text{SED}}(\mathbf{g})\}$ is extended to an SED problem described by

$$Q(\mathbf{g}, \mathbf{p}(\boldsymbol{\omega})) = \arg\min_{\mathbf{g}}\{f_{\text{SED}}(\mathbf{g}, \mathbf{p}(\boldsymbol{\omega}))\}. \tag{1}$$

Here, f_{SED} represents the objective function. For this problem, the randomness brought by $\mathbf{p}(\boldsymbol{\omega})$ will lead to different optimized values of $\mathbf{g}$, which will inevitably change the deterministic cost, $Q(\mathbf{g})$, into a random cost, $Q(\mathbf{g}, \mathbf{p}(\boldsymbol{\omega}))$. In this paper, we consider the randomness brought by the wind farms as a spatiotemporal random field, which is denoted by $\mathbf{p}_i(\boldsymbol{\omega}, \mathbf{t})$ for the power generation of the ith wind farm. Here, the time $t \in \mathbb{T}$ and $\mathbb{T}$ is a finite integer set representing hours in a day, namely, $\mathbb{T} = \{1, 2, \ldots, 24\}$. Suppose that we conduct a day-ahead SED problem over 24 hours of a power system with three farms. Then, we have an input of three random fields, $\{\mathbf{p}_1(\boldsymbol{\omega}, \mathbf{t})$, $\{\mathbf{p}_2(\boldsymbol{\omega}, \mathbf{t})$, $\{\mathbf{p}_3(\boldsymbol{\omega}, \mathbf{t})\}$, consisting of 72 random variables in total. Our UQ goal is to quantify the statistical moments of $Q(\mathbf{g}, \mathbf{p}(\boldsymbol{\omega}))$, such as the mean and variance for a day-ahead forecast.

Remark 1: Note that since we focus on the quantification of uncertainties in the SED problem instead of their modeling, the detailed description of "f_{SED}" as well as all the equality and inequality constraints of the SED directly follow from [8]. Also, note that the proposed UQ framework is general in that it is not limited to this SED problem. The day-ahead SED forecast problem addressed here does need this framework since their dimension is very high due to the incorporation of a time-series-based SED model with $\mathbb{T} = \{1, 2, \ldots, 24\}$. If we deal with a hour-ahead forecast problem (i.e., $\mathbb{T} = \{1\}$), then the total dimension is only 3, instead of 72 as mentioned above. Then, the uncertainty in this hour-ahead SED problem can be quantified directly without any need to use of the proposed dimension-reduction-based UQ framework.

III. Theoretical Background

A. Gaussian Process Emulator

1) Fundamentals: Let us first assume a nonlinear model denoted as $f(\cdot)$. Its corresponding vector-valued random input of p dimensions is denoted by $\mathbf{x}$. Due to the randomness of $\mathbf{x}$, we may observe n samples as a finite collection of the model inputs, which are described by $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. Accordingly, its model output, $f(\mathbf{x})$, also becomes random, and has its corresponding n realizations, denoted by $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}$. Let us assume that the model output is a realization of a Gaussian process; then the finite collection, $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}$, of the random variables, $f(\mathbf{x})$, will follow a joint multivariate normal probability distribution given by

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & \cdots & k(\mathbf{x}_1,\mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n,\mathbf{x}_1) & \cdots & k(\mathbf{x}_n,\mathbf{x}_n) \end{bmatrix} \right). \tag{2}$$

Here, let us denote the mean function by $\boldsymbol{m}(\cdot)$ and a kernel function that represents the covariance function by $\boldsymbol{k}(\cdot,\cdot)$. Then, (2) can be simplified as

$$\boldsymbol{f}(\mathbf{X})\,|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{X}), \boldsymbol{k}(\mathbf{X},\mathbf{X})\right), \tag{3}$$

where $\mathbf{X}$ is an $n \times p$ matrix, denoted by $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\intercal$; $\boldsymbol{f}(\mathbf{X})$ stands for $[f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)]^\intercal$; and $\boldsymbol{m}(\mathbf{X})$ represents $[m(\mathbf{x}_1), m(\mathbf{x}_2), \ldots, m(\mathbf{x}_n)]^\intercal$.

Now, if an independent, identically and normally distributed (i.i.d.) noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_n)$ (where $\mathbf{I}_n$ and σ^2 are an n-dimensional identity matrix and the variance, respectively) is considered on the system output, $\boldsymbol{f}(\mathbf{X})$, the observations $\mathbf{Y}$ will be expressed as

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{X}), \boldsymbol{k}(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}_n\right). \tag{4}$$

Note that $\boldsymbol{\varepsilon}$ is also called a "nugget". If $\sigma^2 = 0$, then $f(x)$ is observed without noise. However, in practice, the nugget is always added for the sake of numerical stability.

Next, we present the way to use the abovementioned finite collection of n samples, $(\mathbf{Y}, \mathbf{X})$, to infer the unknown system output, $\mathbf{y}(\mathbf{x})$, on the sample space of $\mathbf{x} \in \mathbb{R}^p$ in a Bayesian-inference framework.

2) Bayesian Inference: It is well-known that a Bayesian posterior distribution of the unknown system output can be inferred from a Bayesian prior distribution of $\mathbf{y}(\mathbf{x})$ and the likelihoods obtained from the observations. Let us first assume a Bayesian prior distribution of $\mathbf{y}(\mathbf{x})|\mathbf{x}$ given by

$$\mathbf{y}(\mathbf{x})|\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{x}), \boldsymbol{k}(\mathbf{x},\mathbf{x}) + \sigma^2\mathbf{I}_{n_x}\right). \tag{5}$$

Combined with the observations provided by the finite collection of samples $\{\mathbf{Y}, \mathbf{X}\}$, we can express the joint distribution of $\mathbf{Y}$ and $\mathbf{y}(\mathbf{x})|\mathbf{x}$ as

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{y}(\mathbf{x})|\mathbf{x} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}(\mathbf{X}) \\ \boldsymbol{m}(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}\right), \tag{6}$$

where $\mathbf{K}_{11} = \boldsymbol{k}(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}_n$; $\mathbf{K}_{12} = \boldsymbol{k}(\mathbf{X},\mathbf{x})$; $\mathbf{K}_{21} = \boldsymbol{k}(\mathbf{x},\mathbf{X})$; and $\mathbf{K}_{22} = \boldsymbol{k}(\mathbf{x},\mathbf{x}) + \sigma^2\mathbf{I}_{n_x}$.

Now, using the rules of the conditional Gaussian distribution (a.k.a. Gaussian conditioning or statistical linearization) [19], we express the Bayesian posterior distribution of the system output $\mathbf{y}(\mathbf{x})|\mathbf{x}, \mathbf{Y}, \mathbf{X}$ conditioned upon the observations $(\mathbf{Y}, \mathbf{X})$ as a Gaussian distribution given by $\mathbf{y}(\mathbf{x})|\mathbf{x}, \mathbf{Y}, \mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\Sigma}(\mathbf{x}))$. Here, we have

$$\boldsymbol{\mu}(\mathbf{x}) = \boldsymbol{m}(\mathbf{x}) + \mathbf{K}_{21}\mathbf{K}_{11}^{-1}(\mathbf{Y} - \boldsymbol{m}(\mathbf{X})), \tag{7}$$

$$\boldsymbol{\Sigma}(\mathbf{x}) = \mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12}. \tag{8}$$

To this point, the form of the GPE has been derived. On the one hand, (7) can be used as a surrogate model (a.k.a. the *response surface* or *reduced-order* model) to very closely capture the behavior of the nonlinear SED model while keeping the computational cost low. On the other hand, (8) can be used to quantify the uncertainty of the surrogate itself. In this paper, we only need to use (7) as a surrogate model to conduct efficient UQ.

3) Mean and Covariance Functions: To further define the GPE, we need to select the forms of the mean function $\boldsymbol{m}(\cdot)$ and the covariance function represented via the kernel $\boldsymbol{k}(\cdot,\cdot)$.

The mean function models the prior belief about the existence of a systematic trend expressed as

$$\boldsymbol{m}(\mathbf{x}, \boldsymbol{\beta}) = \mathbf{H}(\mathbf{x})\boldsymbol{\beta}. \tag{9}$$

Here, $\mathbf{H}(\mathbf{x})$ can be any set of basis functions. For example, let $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]$ indicate the ith sample, $i = 1, 2, \ldots, n$ and x_{ik} represents its kth element, $k = 1, 2, \ldots, p$. For instance, $\mathbf{H}(\mathbf{x}_i) = 1$ is a constant basis; $\mathbf{H}(\mathbf{x}_i) = [1, x_{i1}, \ldots, x_{ip}]$ is a linear basis; $\mathbf{H}(\mathbf{x}_i) = [1, x_{i1}, \ldots, x_{ip}, x_{i1}^2, \ldots, x_{ip}^2]$ is a pure quadratic basis; and $\boldsymbol{\beta}$ is a vector of hyperparameters.

Since the covariance function is represented by a kernel function, choosing the latter is a must. A popular choice is the square exponential kernel, expressed as $k_{\text{SE}} = \tau^2 \exp(-\sum_{k=1}^{p} \frac{r_k^2}{2\ell_k^2})$, where $r_k = |x_{ik} - x_{jk}|$. Other popular choices include exponential, rational quadratic, and Martin kernels [15].

As for the parameters of a kernel function, they are defined as follows: τ and ℓ_k are the hyperparameters defined in the positive real line; σ^2 and ℓ_k correspond to the order of magnitude and the speed of variation in the kth input dimension, respectively. Let $\boldsymbol{\theta} = [\tau, \ell_1, \ldots, \ell_p]$ contains the hyperparameters of the covariance function, i.e.,

$$k(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta}) = \text{Cov}(\mathbf{x}_i, \mathbf{x}_j|\boldsymbol{\theta}). \tag{10}$$

Until now, the model structure of the GPE has been fully defined. For simplicity, we write $\boldsymbol{\eta} = (\sigma^2, \boldsymbol{\beta}, \boldsymbol{\theta})$ to represent all the hyperparameters in the GPE model.

4) Training Samples Generation: In order to obtain the GPE-based surrogate described in (7), we need to obtain the observation sets contained in $(\mathbf{Y}, \mathbf{X})$. To obtain the system realization $\mathbf{Y}$, we must generate n samples, $\mathbf{X}$, that will be evaluated through the original power system model. To avoid long training time of the GPE, n should be small. Therefore, Latin hypercube sampling (LHS) serves as a popular choice to generate these small number of samples. Unlike the MC sampling that generates totally random samples, LHS generates near-random samples that follow a standard uniform distribution $\mathcal{U}(0, 1)$ based on an equal-interval segmentation. For a nonuniform distribution, the inverse transformation of the cumulative distribution function (cdf) is applied to map the uniformly distributed samples into the targeted distribution [20]. Note that in this paper, we only use LHS to generate mutually independent samples. When the dependence is considered, the copula theory can be adopted [21].

5) GPE Construction: Using $(\mathbf{Y}, \mathbf{X})$, we estimate the hyperparameters $\boldsymbol{\eta}$ in the GPE. Following Gelman *et al.* [22], we choose to adopt the Gaussian maximum likelihood estimator (MLE) since it is the most efficient estimator under the Gaussian distribution which is followed by the residuals and it is easy to compute. First, to indicate the hyperparameters, let us rewrite (4) as

$$\mathbf{Y}|\mathbf{X}, \boldsymbol{\eta} \sim \mathcal{N}\left(\boldsymbol{m}(\mathbf{X}), \boldsymbol{k}(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}_n\right). \tag{11}$$

Then, using MLE, we obtain

$$\widehat{\boldsymbol{\eta}} = \left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right) = \underset{\boldsymbol{\beta},\boldsymbol{\theta},\sigma^2}{\arg\max} \log P\left(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2\right). \quad (12)$$

Using (9)–(11) and simplifying $\mathbf{H}(\mathbf{x})$ into $\mathbf{H}$, the marginal log-likelihood can be expressed as

$$\begin{aligned}\log P\left(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2\right) &= -\frac{1}{2}(\mathbf{Y} - \mathbf{H}\boldsymbol{\beta})^{\intercal} \\ &\times \left[\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right]^{-1} (\mathbf{Y} - \mathbf{H}\boldsymbol{\beta}) \\ &- \frac{n}{2}\log 2\pi - \frac{1}{2}\log\left|\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right|, \quad (13)\end{aligned}$$

which implies that the MLE of $\boldsymbol{\beta}$ conditioned upon $\boldsymbol{\theta}$ and σ^2 is a weighted least-squares estimate given by

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}, \sigma^2) = \left[\mathbf{H}^{\intercal}\left[\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right]^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^{\intercal}\left[\boldsymbol{k}(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) + \sigma^2 \mathbf{I}_n\right]^{-1}\mathbf{Y}. \quad (14)$$

Since $\widehat{\boldsymbol{\beta}}$ is a function of $(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2)$, let us insert (14) into (13) to reduce the number of hyperparameters. Then, (12) is also simplified as

$$\left(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2\right) = \underset{\boldsymbol{\theta},\sigma^2}{\arg\max} \log P\left(\mathbf{Y}|\mathbf{X}, \hat{\boldsymbol{\beta}}\left(\boldsymbol{\theta}, \sigma^2\right), \boldsymbol{\theta}, \sigma^2\right). \quad (15)$$

Now, we only need to estimate the hyperparameters $(\widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2)$ instead of $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}}, \widehat{\sigma}^2)$. Then, we utilize a gradient-based optimizer to achieve this optimization as described in [15]. Once $\widehat{\boldsymbol{\eta}}$ is obtained, the GPE model is fully constructed. More details can be found in [22].

B. Isomap-Based Dimensionality Reduction

In practice, the dimension for the random fields representing wind-farm generation may be so high (as mentioned in Section II) that the GPE cannot be constructed efficiently. Therefore, an efficient dimensionality reduction becomes a prerequisite. Here, we conduct the dimensionality reduction based on Isomap, which stands for isometric mapping. Similar to the KLE method, Isomap is a spectral method that utilizes an eigendecomposition to learn the low-dimensional representation of the original data. However, unlike the linear-transformation-based KLE, Isomap is a nonlinear dimensionality-reduction technique that tries to preserve the low-dimensional embedding of the data set through the geodesic distances, instead of the Euclidean distances adopted in the KLE. Let us describe this approach in the sequel.

First, let us denote the raw data before dimensionality reduction as an $n \times p_{\text{raw}}$ matrix expressed as $\mathbf{W}$, where $p_{\text{raw}} > p$. Our final goal is to use Isomap to transform this $\mathbf{W}$ into the aforementioned $n \times p$ matrix $\mathbf{X}$ that can be effectively adopted to construct the GPE. Now, we elaborate on the three major steps, including creating neighborhood graph, obtaining geodistance and eigendecomposition, of Isomap, respectively.

1) Neighborhood Graph: A weighted graph, $\mathcal{G}$, is, in general, expressed as $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{L})$. Here, $v_i \in \mathcal{V}$ is the set of vertices, which are connected with edges $e(v_i, v_j) \in \mathcal{E}$ and characterized by the corresponding lengths (weights) $l(v_i, v_j) \in \mathcal{L}$.

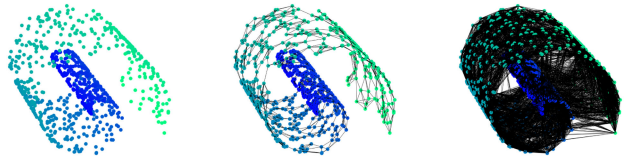


Fig. 1. Neighborhood graphs of 800 samples from a 3-dimensional Swiss roll manifold (left) with $k = 5$ (middle) and $k = 50$ (right). Vertices are distinguished by color. A small k captures neighborhood relationships that should be preserved, while a large k assumes strong global connections.

To construct the graph $\mathcal{G}$ from the raw data $\mathbf{W}$, let us first use the n observations $\{\mathbf{w}_i\}_{i=1}^n$ to represent n vertices. Between any two of them, $\mathbf{w}_i$ and $\mathbf{w}_j$, their pairwise Euclidean distance is denoted by $\delta_{i,j} = \|\mathbf{w}_i - \mathbf{w}_j\|$. Using $\delta_{i,j}$, $\forall i, j = 1, \ldots, n$, we can form an $n \times n$ matrix, $\boldsymbol{\Delta}$, to represent the Euclidean distance information between all vertices. Now, we can assign edges between vertices based on $\boldsymbol{\Delta}$. For any vertex, $\mathbf{w}_i$, we only assign an edge between $\mathbf{w}_i$ and $\mathbf{w}_j$ if $\mathbf{w}_j$ is the k-nearest neighbor of $\mathbf{w}_i$. Using these vertices and edges, the topology of a graph is roughly represented by $\mathbf{W}$ and $\boldsymbol{E}$. Figure 1 shows how the neighborhood graphs look like in the Swiss roll example. In general, a properly small k is quite helpful in classifying different groups of samples. A large k, on the other hand, provides more concentrated embedding. To be conservative in our case, we suggest using a relatively large k (e.g., ~ 30) to ensure concentration.

2) Geodistance Calculation: Now, with the aforementioned graphs and the Euclidean distances $\delta_{i,j}$, we seek to find the geodesic distance (shortest path) between ith and jth vertices denoted by $d_{i,j}$. Similarly, these pairwise geodesic distance elements $d_{i,j}$ will provide the $n \times n$ geodesic distance matrix $\mathbf{D}$ that preserves the nonlinear manifold information of the data sets. Here, we use the Dijkstra's algorithm to obtain $d_{i,j}$ since it is a popular method for solving the shortest-path problem [23]. Based on $\mathcal{G}(\mathbf{W}, \boldsymbol{E}, \boldsymbol{\Delta})$,[1] Algorithm 1 describes the details of how we can use Dijkstra's algorithm to find the shortest path between $\mathbf{w}_i$, the source vertex, and all the other vertices, $\mathbf{w}_j$, where $j = 1, \ldots, n$.

Till now, the graph is characterized by $\mathcal{G}(\mathbf{W}, \boldsymbol{E}, \mathbf{D})$, which can better preserve the manifold information using geodistances, $\mathbf{D}$.

3) Eigendecomposition: Similar to the KLE method that applies eigendecomposition to extract the low-dimensional representation of the raw data, $\mathbf{X}$, the proposed Isomap method can also directly apply eigendecomposition on the geodistance-based matrix, $\mathbf{D}$, to extract $\mathbf{X}$.

Let us first define the Isomap objective function, U, as

$$U(\mathbf{X}) = \min_{\mathbf{X}} \|\tau(\mathbf{D}) - \tau(\mathbf{D}_{\mathbf{X}})\|_F. \quad (17)$$

Here, $\tau(\mathbf{D})$ stands for the shortest-path inner product matrix; $\mathbf{D}_{\mathbf{X}}$ is a symmetric, hollow matrix of the Euclidean distance, whose element, $d_{\mathbf{X}}(x_i, x_j)$ holds for $\{d_{\mathbf{X}}(x_i, x_j) =$

[1]Here, we convert the set notation $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{L})$ into a matrix form.

Algorithm 1: Dijkstra's Shortest-Path Algorithm.

1: For the source vertex, $\mathbf{w}_i$, initialize its geodistance with respect to the other vertex $\mathbf{w}_j$ by setting

$$d_{i,j} = \begin{cases} 0 & \text{if } i = j, \\ \infty & \text{if } i \neq j. \end{cases} \tag{16}$$

2: Initially, choose the source vertex, $\mathbf{w}_i$, as the current vertex, $\mathbf{w}_c$, whose current geodistance, d_c, is, therefore, 0.
3: Mark the source vertex, $\mathbf{w}_i$, as a visited vertex;
4: **while** there is a non-visited neighborhood vertex; **do**
5: Starting from the current vertex, $\mathbf{w}_c$, visit its non-visited adjacent vertices;
6: For the adjacent vertex, $\mathbf{w}_s, s \in j$, update its current geodistance as $d_{i,s} = \min\{d_{i,s}, d_c + \delta_{c,s}\}$;
7: Pick the adjacent vertex with the smallest geodistance as the new current vertex, $\mathbf{w}_c$, whose current geodistance value is assigned to d_c;
8: Mark the new current vertex, $\mathbf{w}_c$, as a visited vertex.
9: **end while**

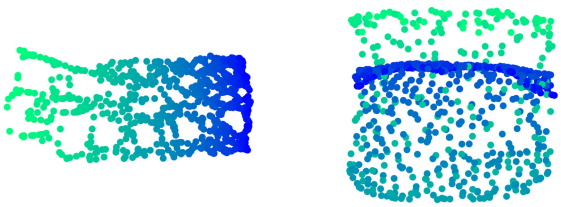


Fig. 2. Isomap-learned, 2-dimensional embedding of the Swiss roll when $k = 5$ (left frame) and $k = 50$ (right frame).

$\|x_i - x_j\|\}$; $\tau(\mathbf{D}_\mathbf{X})$ denotes the Euclidean inner product matrix; and $\|\cdot\|_F$ is the Frobenius matrix norm.

From [24], we know that the global minimum of (17) can be obtained by the largest p eigenvectors of $\tau(\mathbf{D})$. Then, let us reform $\tau(\mathbf{D})$ as $\tau(\mathbf{D}) = -\frac{1}{2}\mathbf{J}\mathbf{D}^{(2)}\mathbf{J}$. Here, $\mathbf{D}^{(2)} = \{d_{i,j}^2\}$ and $\mathbf{J} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\intercal$, where $\mathbf{I}$ is an identity matrix and $\mathbf{1}$ is a vector of all ones. Now, the embedding, $\mathbf{X}$, is obtained as $\mathbf{X} = (\lambda_1^{1/2}\mathbf{v}_1, \ldots, \lambda_p^{1/2}\mathbf{v}_p)$, where $\{\lambda_s\}_{s=1}^p$ denotes the first p eigenvalues of $\tau(\mathbf{D})$ and $\{\mathbf{v}_s\}_{s=1}^p$ are the corresponding eigenfunctions. More details on mathematical explanation are included in [24]. Figure 2 presents the calculated embedding from graphs in Fig. 1 using Isomap.

Remark 2: Note that although the Isomap method takes more computing time to explore the data structure than the traditional KLE method, it does not influence the computational efficiency of this UQ application. This is because this data-processing step can be conducted offline. This makes sense since the manifold of the historical wind data can be viewed as unchanged for a certain time period, and therefore does not need to be updated repeatedly. For an online UQ of the SED application, once the current system topology as well as the control and operating states are updated, we can directly apply the offline-trained, low-dimensional data structure on the online application.

IV. Proposed Method

In this section, we summarize the proposed method to solve the SED problem using the aforementioned theories. In general, the proposed method includes five steps that are elaborated below:

4) Dimensionality Reduction: Starting from the high-dimensional raw data, included in a p_{raw}-dimensional matrix, $\mathbf{W}$, which represents the random process of the wind power penetration, we need to project $\mathbf{W}$ matrix into the low-dimensional latent space, represented in a p-dimensional matrix, $\mathbf{X}$, with the aforementioned Isomap method.

5) Closed-Form Probability Density Function (pdf) Estimation: Although some dimensionality-reduction methods, such as KLE, assume that every latent variable in the low-dimensional space simply follows a Gaussian distribution; however, this assumption cannot be guaranteed in practice. Furthermore, an accurate pdf description for the input samples is a key factor for the success of the UQ task. Therefore, to avoid the weakness in Gaussian assumption, we adopt a well-known kernel-based pdf estimation method (presented in Appendix A in detail) to obtain the closed-form expressions for the pdf of every latent variable. With these pdf expressions, using an inverse cdf mapping in the aforementioned LHS [20], we can generate the input samples following any type of pdfs.

6) Nonlinear Dimension Recovery: Although we have obtained the low-dimensional embedding for the raw data, we are unable to directly inject the samples of these latent variables into our physics-based economic dispatch model, which requires an exact dimension for the input as p_{raw}. Therefore, the samples generated for the latent variables should be mapped back to the p_{raw}-dimensional space to calculate the realizations of the SED model. Since Isomap is unable to provide the inverse transformation from $\mathbf{x}$ to $\mathbf{W}$, we propose to use the GPE to achieve this nonlinear inverse mapping.

First, let us formulate this problem. To obtain the closed-form expression between $\mathbf{x}$ and $\mathbf{w}$, we can take $\mathbf{x}$ as the inputs of the unknown model and $\mathbf{w}$ as the corresponding output. Based on the raw data $\mathbf{W}$ and the data for latent variables $\mathbf{X}$ obtained through Isomap, we get an observation set $\{\mathbf{X}, \mathbf{W}\}$. This observation set will serve as the aforementioned finite collection of samples that enable us to construct the GPE-based surrogate model between model input $\mathbf{x}$ and the model output $\mathbf{w}$.

Let us further elaborate on this problem by taking a hourly-based raw data matrix $\mathbf{W}$, which, therefore, has 24 dimensions, as an example. First, for the observed raw data $\mathbf{W}^{\text{ob}} = [\mathbf{W}_1^{\text{ob}}, \ldots, \mathbf{W}_{24}^{\text{ob}}]$, let us assume that given $\mathbf{x}$, $\mathbf{W}_1, \ldots, \mathbf{W}_{24}$ are conditionally independent. Then, the conditioned pdf of $\mathbf{W}$ is given by $f_{\mathbf{W}|\mathbf{x}}(\cdot) = \prod_{t=1}^{24} f_{W_t|\mathbf{x}}(\cdot)$, where $f_{W_t|\mathbf{x}}$ denotes the conditioned pdf of W at time t. The marginal density of $\mathbf{W}$ can be further obtained as $f_{\mathbf{W}}(\cdot) = \int_{x_1} \cdots \int_{x_p} f_{\mathbf{W}|\mathbf{x}}(\cdot) f_{\mathbf{x}}(\cdot)\, \mathrm{d}x_1 \cdots \mathrm{d}x_p$, where $f_{\mathbf{x}}(\cdot)$ denotes the

joint density of $x_1, \ldots, x_p$. If $x_1, \ldots, x_p$ are believed to be independent, $f_{\mathbf{x}}(\cdot) = \prod_{s=1}^{p} f_{x_s}(\cdot)$. Now, we are able to estimate the $f_{W_t|\mathbf{x}}(\cdot)$ for $t = 1, \ldots, 24$. Following the Bayesian-inference procedure, described in (5)–(8), to construct a GPE for $f_{W_t|\mathbf{x}}(\cdot)$, let us first assume the prior distribution as

$$W_t(\mathbf{x})|\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{m}_t(\mathbf{x}), \boldsymbol{k}\left(\mathbf{x}, \mathbf{x}\right) + \sigma^2 \mathbf{I}_{n_x}\right). \tag{18}$$

Given observations $\{\mathbf{X}^{\text{ob}}, \mathbf{W}_t^{\text{ob}}\}$, the consequent posterior distribution is obtained as

$$W_t(\mathbf{x})|\mathbf{x}, \mathbf{X}^{\text{ob}}, \mathbf{W}_t^{\text{ob}} \sim \mathcal{N}\left(\boldsymbol{\mu}_t(\mathbf{x}), \boldsymbol{\Sigma}_t(\mathbf{x})\right). \tag{19}$$

Now, we have trained 24 GPEs that provide closed-form relationships between the latent variable $\mathbf{x}$ and $\mathbf{w}$. Note that the 24 GPEs can be trained in parallel.

7) Training a GPE-Based SED Model: Till now, we should further construct a GPE-based surrogate model to replace the original SED model. Since the aforementioned nonlinear dimension-recovery procedure can map the latent variable $\mathbf{x}$ back to $\mathbf{w}$, we can obtain n samples of $\mathbf{W}$ as the SED model inputs. By evaluating these samples in f_{SED}, we can get a finite collection of the model output, $\{Q_i\}_{i=1}^n$, accordingly. After choosing the proper Bayesian prior mean function (e.g., pure quadratic and kernel functions), we can easily construct the GPE-based surrogate model based on the observations provided from the model input, $\mathbf{X}$, and the model output, $\{Q_i\}_{i=1}^n$.

8) Uncertainty Quantification (UQ) for SED: Since we have obtained the reduced-order model of the SED model represented as the GPE, we are now able to conduct UQ for the SED problem efficiently. By generating a large number of the samples for the latent variables using LHS, the GPE-based surrogate model can be evaluated repeatedly at almost no time cost. Then, the statistical moment, such as the mean and variance, of our quantity of interest, i.e., the cost, Q, can be obtained in a cost-effective way.

9) Summarization: Now, we can summarize the above steps in a flowchart as depicted in Fig. 3. In this flowchart, we have separated the computing steps into two parts: an offline preparation and an online UQ. In the former, all the computing steps are based on the historical data that can be assumed to be unchanged for a certain time period; hence, that part does not need to be updated repeatedly as done in the steps of the online application part. We also need to emphasize that the GPE training procedure based on the evaluations of the actual SED model is the most time-consuming step for the online computing stages. Luckily, we only need a small number, n, to train an accurate GPE model. This finally enables the GPE-based SED model to be cost-effective for the online applications. More details about the performance of the proposed method considering computing accuracy and efficiency are presented in the next section.

Remark 3: Note that our proposed method has both a dimension reduction and a dimension-recovery procedure as shown in Fig. 3. The purpose of dimension reduction is to have a low-dimensional input for the SED surrogate model that can be trained in a more computationally efficient manner while the purpose of dimension recovery is to evaluate the SED actual model since the samples in the latent space cannot be evaluated in an actual physical model, e.g., the SED model.

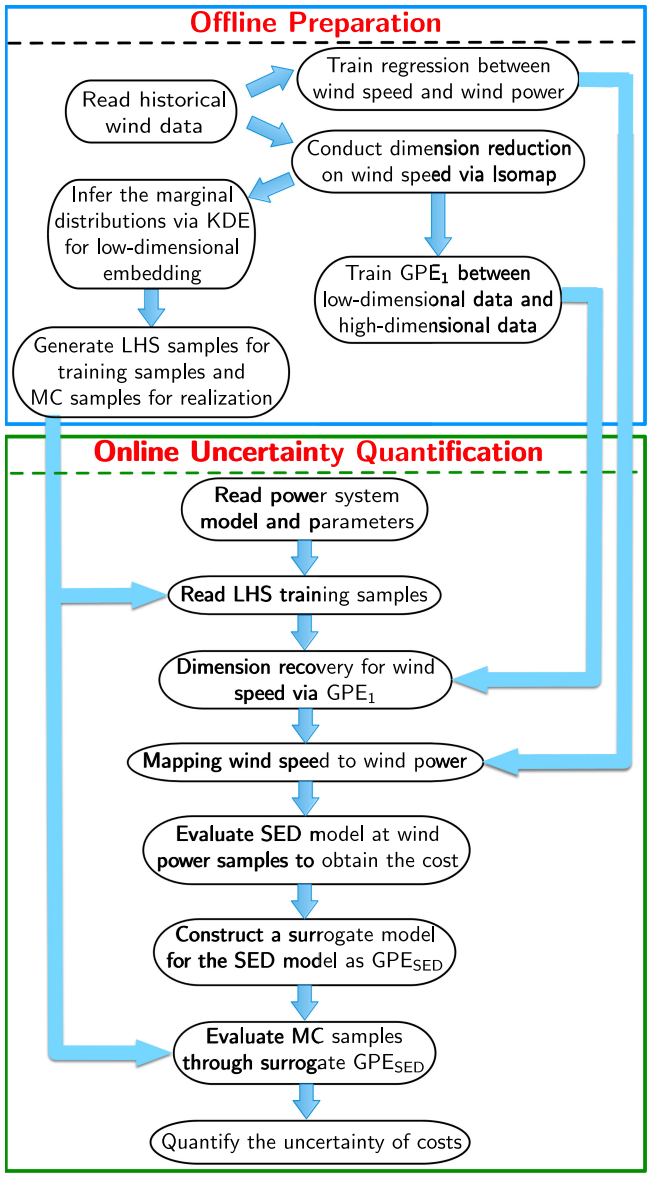


Fig. 3. Flowchart of the GPE-based UQ framework for SED.

V. Case Studies

In this section, we test our method on the IEEE 118-bus system [25] using NREL's Western Wind Data Set [26].

A. Prediction on Wind Power

1) Experimental Settings: The NREL's Western Wind Data Set provides wind speed and power data at 10-min intervals from the years 2004 through 2006 for all numbered wind turbines in the Western United States. For practical reasons, instead of focusing on single wind turbines, we are interested in the overall power generated from a wind farm, which is a group of wind turbines in the same location. In the experiment, we pick one farm in Livermore, CA (#LV) and two farms in Seattle, WA (#SE1, #SE2). For each farm, three turbines are extracted from the data set:

TABLE I
TEST OF ISOMAP UNDER DIFFERENT p-VALUES

p	1	2	3	4	5	6
#LV	0.863	0.962	0.973	0.980	0.983	0.984
#SE1	0.972	0.997	0.998	0.999	0.999	0.999
#SE2	0.937	0.987	0.994	0.996	0.996	0.996

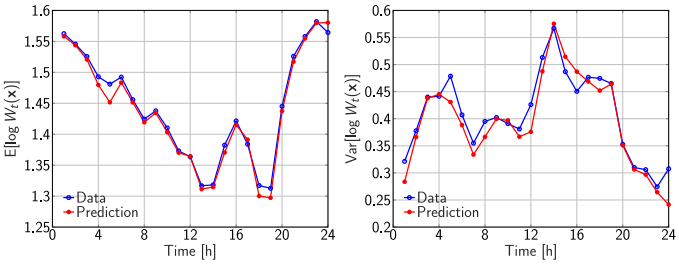


Fig. 4. The mean (left panel) and variance (right panel) of $\log W_t(\mathbf{x})$ varying by hour at #LV in January. Blue curves denote empirical trajectories. Red curves are predictions computed from GP-Isomap samples.

#9247, #9248, #9249 for #LV; #28914, #28928, #28959 for #SE1; and #29138, #29153, #29154 for #SE2. These three wind farms are added at Buses 16, 58, and 78, respectively, to introduce the randomness in the SED model.

Assuming that the turbines in the same farm have the same wind speed and wind power all the time, we calculate (at each time stamp) the averaged wind speed and wind power over three selected wind turbines and treat it as the prevailing wind speed and wind power of that farm. For each farm, we take hourly averages on the common wind speed and wind power to obtain the daily 24-dimensional wind speed, $\mathbf{W}$, and wind power, $\mathbf{P}$, for each day in January between 2004 and 2006, which leads to a total of 93 data points. Similarly, observations for any other months can also be collected.

2) Dimension Reduction via Isomap: For each farm, we apply the GPE-Isomap method on learning $\log \mathbf{W}(\mathbf{x})$, where $\mathbf{x}$ is the p-dimensional latent variables of $\log \mathbf{W}$. Here, the log transformation is to ensure the positivity of $\mathbf{W}$.

To make the Isomap algorithm more adaptive, we propose to use the information explaining factor (IEF) to indicate a proper selection of the tunable parameter p based on the aforementioned first p largest eigenvalues of $\tau(\mathbf{D})$, i.e., $\{\boldsymbol{\lambda}_s\}_{s=1}^{p}$. This IEF is expressed as $\text{IEF} = \sqrt{\frac{\sum_{s=1}^{p}\lambda_s^2}{\sum_{s=1}^{S}\lambda_s^2}}$. Obviously, this factor ranges within (0,1). The larger the IEF is, the more information the p-dimensional embedding retains.

Using the data for the three selected wind sites: #LV, #SE1 and #SE2, let us provide the IEF values under different p-values in the Table I. It is clear we can adaptively obtain a proper p-value by setting a threshold value for the IEF. For example, for a 95% threshold, $p = 2$ will be selected. Similarly, for a 97% threshold, $p = 3$ will be selected.

In our experiments, to obtain a good approximation accuracy, we set the threshold to 97%. Consequently, we find that $p = 3$ is enough for representing the 24-dimensional $\mathbf{W}$. Figure 4 represents the mean and variance trajectories of $\log W_t(\mathbf{x})$ throughout

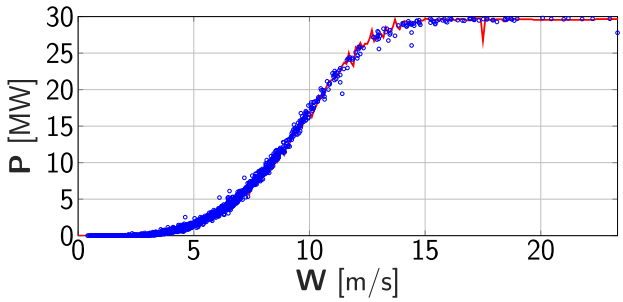


Fig. 5. Wind power ($\mathbf{P}$) versus wind speed ($\mathbf{W}$) at Farm #LV in January. Blue points denote real data points. Red curve represents predictions by decision tree regression.

the day in #LV. It turns out that using much-lower-dimensional latent space, our method is able to generate high-dimensional samples that successfully capture the trends of the data. It is also worth pointing out that the computing time of the Isomap algorithm is typically less than 5 s, which is fast enough for practical applications.

3) Wind Data Processing: To test the spatial dependency of $\mathbf{W}$ among the three farms, we calculate distance correlation factors [27] between $\mathbf{x}$ on different farms. The results are shown in Appendix B, where a way to deal with the dependency between #SE1 and #SE2 is also provided.

The relationship between wind speed and wind power is modeled by a decision tree regression model [28]. The result of wind power fitting in #LV is shown in Fig. 5. It shows that the predicted wind power samples have quite a good match with the real-world data. In the end, samples of $\mathbf{P}$ are proposed by taking GPE-Isomap samples of $\mathbf{W}(\mathbf{x})$ into a trained regression tree model.

Remark 4: Here, we do not directly apply Isomap on wind power for the reason that: for wind speed smaller than approximately 3 m/s or greater than approximately 28 m/s, the rated wind power is zero. Estimation of high-dimensional zero-inflated densities is difficult. The same strategy has been adopted in [8]. In practice, this is because that for the safety reasons considered in the current turbine technology, when the wind speed is larger than 25 m/s, also known as the cut-off wind speed, the wind power is set to zero.

B. Predictive Inferences on Minimum Production Cost

So far, in Section IV -A, we have prepared constructions of the input, the power generation $\mathbf{P}$, for the SED model in the IEEE 118-bus test system with $|T| = 24$ time periods. The three wind farms (#LV, #SE1, and #SE2) are used as three augmented renewable power generation buses in the system. In total, $\mathbf{P}$ is a 72-dimensional random vector (24 for each farm) represented by 9-dimensional latent variables (3 for each farm). In this section, the GPE surrogate with a pure quadratic mean function and a squared exponential kernel is constructed for the SED test system.

1) Performance Test: To test the performance of the GPE surrogate model, we compare the predictive inferences of Q

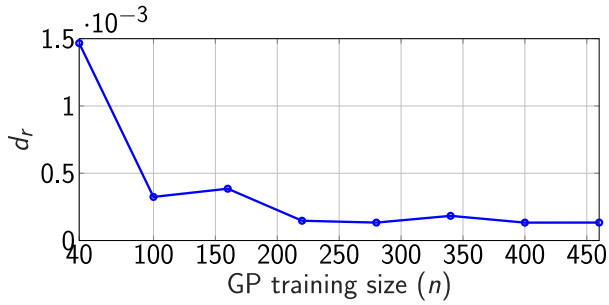


Fig. 6. Relative difference between $\mathbb{E}[Q_{\text{GP}}]$ and $\mathbb{E}[Q_{\text{MC}}]$ changing by GPE training size. The GPE surrogate presents very powerful fitting on the test system.

TABLE II
BREAKDOWN OF CPU TIMES OF THE PROPOSED METHOD UNDER DIFFERENT TRAINING SAMPLES

CPU Time	Step 1	Step 2	Step 3	Step 4	Total
GPE ($n = 100$)	≈ 7 s	≈ 45 s	≈ 1 s	≈ 4 s	≈ 58 s
GPE ($n = 200$)	≈ 7 s	≈ 90 s	≈ 2 s	≈ 4 s	≈ 105 s
MC	—	—	—	—	≈ 1 h

calculated from our GPE surrogate with those gained by the direct MC sampling from the original complicated SED model. Let Q_{GP} denote the estimation of Q from our GPE surrogate. Q_{MC} is calculated by direct MC realizations from the test system. For a fair performance comparison, 8,000 realizations are used to ensure the convergence of Q_{MC}. One of the most important results for the SED is the expected (mean) minimum cost $\mathbb{E}[Q]$. Here, we define the relative difference between $\mathbb{E}[Q_{\text{GP}}]$ and $\mathbb{E}[Q_{\text{MC}}]$ to quantify the relative error of the proposed method as $d_r = \frac{|\mathbb{E}[Q_{\text{GP}}]-\mathbb{E}[Q_{\text{MC}}]|}{\mathbb{E}[Q_{\text{MC}}]}$, where $\mathbb{E}[Q_{\text{MC}}]$ is treated as a baseline. $\mathbb{E}[Q_{\text{GP}}]$ varies with GPE training size n. The smaller the d_r is, the better the GPE surrogate fits its target system. Figure 6 depicts the trace plot of d_r over n. Notice that the approach achieves less than $10^{-3} d_r$ when $n = 100$. This means the GPE-based surrogate, trained with 100 LHS realizations, successfully models the response surface of the tested SED system under 8,000 scenarios. It can also be seen from d_r that the performance of modeling fitting is possible to fluctuate slightly. This is the inevitable randomness caused by the nature of the random-sampling-based method and this is typically true when training size, n, is not large. However, the overall accuracy of the GPE method is high enough for this practical application when n becomes larger than 100. Further increasing n brings negligible benefits but, in turn, sacrificing the computing efficiency as shown in Table II.

Also, as shown in the flowchart in Fig. 3, the online calculation process largely involves four steps that require spending time on

1) the dimension recovery,
2) the evaluations of actual model for training,
3) the construction of the GPE-based surrogate, and
4) the realizations of the MC samples through GPE.

Here, the computing times of all the above steps are listed in Table II.

It can be seen from Table II that using a CPE method with only 100 training samples, we can complete all the computing steps in less than 1 min. Considering that in practice, even the power system operators may have some small operation intervals, such as 5 or 10 min of operation, the proposed method is fast enough for online applications.

Here, we would like to mention that the offline preparation procedure in this algorithm is also computational efficient. As shown in Fig. 3, it consists of multiple steps, i.e., the sample generation, the regression training between the wind speed and the wind power, Isomap-based dimension reduction, and the training of the GPE_1 models between the low- and high-dimensional data. However, the offline preparation can be performed in less than half a minute. Thus, the proposed method will not only greatly accelerate the online UQ procedure, but enable a fast offline preparation as well.

TABLE III
MEAN, 95% CONFIDENCE INTERVAL (CI), AND STANDARD DEVIATION (STD. DEV.) OF THE MINIMUM PRODUCTION COST PREDICTIVE INFERENCES ($)

	Mean ($\times 10^6$)	95% CI ($\times 10^6$)	Std. Dev. ($\times 10^4$)
Q_{GP}	2.954	(2.863, 3.021)	4.029
Q_{MC}	2.955	(2.870, 3.018)	4.010
$Q_{\text{GP-K}}$	2.956	(2.869, 3.020)	4.078
$Q_{\text{MC-K}}$	2.955	(2.874, 3.018)	3.908

2) Comparison Studies: To justify the reliability of the GPE-Isomap method on constructing $\mathbf{P}$, we also employ KLE with 17-dimensional latent variables selected by 95% variance explanation criterion as suggested in [8] to compare the results with our approach. Correspondingly, $Q_{\text{GP-K}}$ denotes the estimation of Q from our GPE surrogate, yet with the KLE-constructed $\mathbf{P}$. Similarly, $Q_{\text{MC-K}}$ is calculated by direct MC sampling, yet by using KLE-constructed $\mathbf{P}$. Both GPE surrogates are trained with 100 realizations. Comparison among the estimation results obtained with the four methods is provided in Table III.

It can be seen that GPE surrogates with both the KLE method and Isomap offer quite comparable results. Both of them well approximate the test system. However, KLE requires a 17-dimensional latent space while our GPE-Isomap method embeds P into only 9 latent variables. With the ability to explore much-lower-dimensional representations, our method has a great potential to handle larger-scale systems.

Here, let us further conduct comparison studies with another technique, i.e., LHS, which is a stratified sampling technique that can be used to reduce the number of runs necessary for an MC simulation to achieve a reasonably accurate random distribution [29]. This has been further verified and applied in some power system applications such as probabilistic load margin and probabilistic power flow analyses [30]–[32]. Researchers advocate this method for its capability to provide a good statistical approximation of the power system states by using a small number of "near-random" samples, e.g., 200 and 500. Here, we choose different sample sizes, N_{LHS}, for LHS to make comparison studies with the proposed method, considering the computing accuracy and the computational efficiency. The simulation results are shown in Table IV. It can be seen that although LHS with 200 and 500 samples can provide a reasonably accurate approximation for the mean, it loses some

TABLE IV
COMPARISON STUDIES WITH THE LHS METHOD

	Mean ($\times 10^6$)	Std. Dev. ($\times 10^4$)	CPU Time
Q_{MC}	2.955	4.010	$\approx$ 1 h
Q_{GP}	2.954	4.029	$\approx$ 1 min
$Q_{\text{LHS}}(N_{\text{LHS}} = 200)$	2.960	4.517	$\approx$ 1.5 min
$Q_{\text{LHS}}(N_{\text{LHS}} = 500)$	2.957	4.501	$\approx$ 3.5 min

TABLE V
TESTS UNDER DIFFERENT MEAN FUNCTIONS

	MC	Quadratic	Linear	Constant
$\mathbb{E}[Q](\times 10^6)$	2.955	2.954	2.709	2.334

TABLE VI
RESULTS OF REPLICATING OBSERVED MINIMUM PRODUCTION COST ($)

	Mean ($\times 10^6$)	95% CI ($\times 10^6$)	Std. Dev. ($\times 10^4$)
Q_{obs}	2.943	(2.849, 3.017)	4.996
Q^r_{GP}	2.949	(2.844, 3.024)	5.203
$Q^r_{\text{GP-K}}$	2.949	(2.858, 3.025)	5.108

accuracy for the standard deviation. The proposed Isomap-GPE method with only 100 training samples outperforms LHS in both the computing accuracy and the computational efficiency based on the benchmark results of the MC method. Also note that the computing time of the Isomap-GPE method has included every step of the online calculations illustrated in Fig. 3. This comparison study further demonstrates the good performance of the proposed method.

3) Prior Trend Matters: It is well-known that a good Bayesian prior facilitates the success of Bayesian inference. With a good Bayesian prior, the Bayesian posterior can converge fast; thus, it can be inferred more effectively. For the Bayesian-framework-based GPE method, the mean function, $\boldsymbol{m}(\cdot)$, serves as a prior trend for the surrogate model. Here, we test the performance of the GPE-based method under different trends, e.g., constant, linear, and quadratic trends, under the same training sample, $n = 100$. The test results are shown in Table V. Compared to the benchmark result of the MC method, it is clear that a quadratic trend serves as an excellent Bayesian prior for this application.

4) Cross-Validation: Furthermore, in principle, a well-performing surrogate should also reasonably replicate true observations Q_{obs}, which are calculated by taking 93 true wind power generation samples into the test system. Correspondingly, we have its replications Q^r_{GP} from the GPE-Isomap surrogate and $Q^r_{\text{GP-K}}$ from the GPE-KLE surrogate. Results in Table VI indicate that our GPE surrogate is able to reproduce similar observational scenarios of minimum production costs generated by the test system. The Q-Q plot shown in Fig. 7 also validates the rationality of the proposed method using the MLE discussed in Section III.

5) Validation in a Different Season: It is well-known that the wind power generation might vary across months. Up to now, we have validated our method using the wind data collected in January, which is in winter. To further test the performance of the proposed method, we select another month in the summer

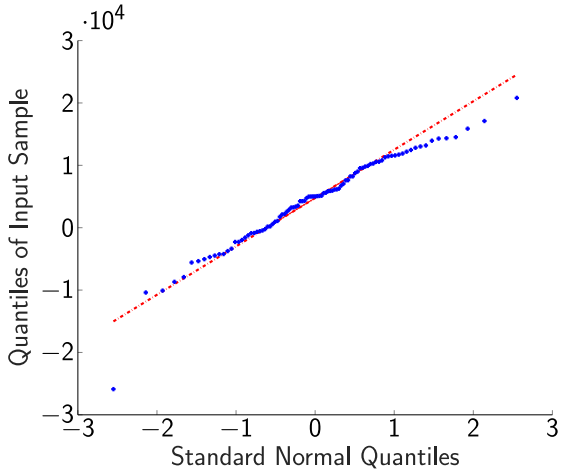


Fig. 7. The Q-Q plot of the residuals between SED realizations using the true wind generations and the estimated results using the proposed method.

TABLE VII
MEAN, 95% CONFIDENCE INTERVAL (CI), AND STANDARD DEVIATION (STD. DEV.) OF THE MINIMUM PRODUCTION COST PREDICTIVE INFERENCES ($) IN JULY

	Mean ($\times 10^6$)	95% CI ($\times 10^6$)	Std. Dev. ($\times 10^4$)
Q_{GP}	3.001	(2.977, 3.024)	1.279
Q_{MC}	3.000	(2.976, 3.017)	1.207

TABLE VIII
RESULTS OF REPLICATING OBSERVED MINIMUM PRODUCTION COST ($) IN JULY

	Mean ($\times 10^6$)	95% CI ($\times 10^6$)	Std. Dev. ($\times 10^4$)
Q_{obs}	2.998	(2.980, 3.021)	1.311
Q_{GP}	3.003	(2.971, 3.025)	1.363

(i.e., July) to validate our proposed method under the same experimental settings. The simulation results are displayed in Table VII. They reveal that the proposed method can still provide highly accurate UQ results in the summer.

Again, we further conduct an abovementioned replication test to cross-validate our method. The simulation results are shown in Table VIII. These results indicate that our proposed method is able to reproduce similar observational scenarios of the minimum production costs generated by the test system during the month of July as well. This further verifies the reliability of the proposed method.

VI. CONCLUSION

In this paper, we propose a GPE-based framework in quantifying uncertainty for the SED problem. The proposed framework utilizes the manifold-learning-based Isomap method to conduct an effective dimensionality reduction, which further accelerates the nonparametric GPE in the propagation of uncertainties. The simulation results on the modified IEEE 118-bus system show that the proposed method is significantly more computationally efficient than the traditional MC method while achieving the

TABLE IX
DISTANCE CORRELATIONS FOR $\mathbf{x}$ WITHIN FARMS

	$x_1 - x_2$	$x_1 - x_3$	$x_2 - x_3$
#LV	0.175	0.249	0.168
#SE1	0.295	0.259	0.200
#SE2	0.279	0.215	0.233

TABLE X
DISTANCE CORRELATIONS FOR $\mathbf{x}$ BETWEEN FARMS

	#LV – #SE1	#LV – #SE2	#SE1 – #SE2
x_1	0.141	0.119	0.596
x_2	0.223	0.170	0.302
x_3	0.149	0.245	0.253

desired simulation accuracy. Future work will include the following:

- We will improve of the performance of the proposed method on larger-scale test systems.
- We will make the proposed method more adaptive. More specifically, a more concrete way to select the parameter k for the neighborhood graph will be studied.

APPENDIX A
KERNEL DENSITY ESTIMATION (KDE)

Given n finite samples, $x_1, x_2, \ldots, x_n$, observed from a univariate random variable X with an unknown pdf $f_X(x)$, a KDE is suggested in [33] to obtain the closed-form expression for the pdf, $f_X(x)$, as

$$\hat{f}_X(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{20}$$

Here, $K(\cdot)$ is the kernel smoothing function, $h > 0$ is the bandwidth. It is worth pointing out that the choice of bandwidth and kernel $K(\cdot)$ will influence the estimation results of the KDE. In general, choosing a proper bandwidth is the key point. It is suggested to adopt a rule-of-thumb bandwidth estimator defined as $h = (4\hat{\sigma}^5/3n)^{1/5} \approx 1.06\hat{\sigma}n^{-1/5}$, where $\hat{\sigma}$ is sample standard deviation. As for the choice of $K(\cdot)$, popular choices include: Epanechnikov (optimal in mean square error sense), normal, triangular, etc. [33].

APPENDIX B
MODELING CORRELATIONS OF WIND SPEED BETWEEN WIND FARMS

The value of distance correlation is 0 if there is no dependency and 1 if the dependency is extremely strong. Table IX implies that $\mathbf{x}$ on the same farm can be treated independently. Table X shows that x_1 between #SE1 and #SE2 have a relatively high distance correlation. One can still treat them independently since the value is not too close to one. In addition, we provide the following ways to handle the dependency:

Considering that the Pearson correlation of x_1 between #SE1 and #SE2 is calculated to be 0.618, it is proper to assume that they are linearly dependent, which can be modeled using linear regression. If we use x_{11} and x_{12} to distinguish x_{i1} in #SE1 and #SE2, the joint samples of x_{11}, x_{12} can be gained by two steps:

1) Sample x_{11} from its KDE estimated density function
2) Sample $x_{12} \sim \mathcal{N}(\hat{\beta}_0 + x_{11}\hat{\beta}_1, \hat{\sigma}^2)$, where $\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2$ are estimated intercept, slope, and mean squared error from the results of linear regression between x_{12} and x_{11}, respectively.

A yet another way to deal with correlations between #SE1 and #SE2 is to simply concatenate two 24-dimensional wind data together and use our GPE-Isomap method on the 48-dimensional concatenated data.

REFERENCES

[1] M. Hedayati-Mehdiabadi, J. Zhang, and K. W. Hedman, "Wind power dispatch margin for flexible energy and reserve scheduling with increased wind generation," *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1543–1552, Oct. 2015.

[2] D. Villanueva, A. Feijóo, and J. L. Pazos, "Simulation of correlated wind speed data for economic dispatch evaluation," *IEEE Trans. Sustain. Energy*, vol. 3, no. 1, pp. 142–149, Jan. 2012.

[3] X. Liu and W. Xu, "Economic load dispatch constrained by wind power availability: A here-and-now approach," *IEEE Trans. Sustain. Energy*, vol. 1, no. 1, pp. 2–9, Apr. 2010.

[4] M. Xie, J. Xiong, S. Ke, and M. Liu, "Two-stage compensation algorithm for dynamic economic dispatching considering copula correlation of multiwind farms generation," *IEEE Trans. Sustain. Energy*, vol. 8, no. 2, pp. 763–771, Apr. 2017.

[5] Y. Xu, L. Mili, A. Sandu, M. R. von Spakovsky, and J. Zhao, "Propagating uncertainty in power system dynamic simulations using polynomial chaos," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 338–348, Jan. 2019.

[6] Q. Wang, C. B. Martinez-Anido, H. Wu, A. R. Florita, and B.-M. Hodge, "Quantifying the economic and grid reliability impacts of improved wind power forecasting," *IEEE Trans. Sustain. Energy*, vol. 7, no. 4, pp. 1525–1537, Oct. 2016.

[7] Y. Xu, M. Korkali, L. Mili, X. Chen, and L. Min, "Risk assessment of rare events in probabilistic power flow via hybrid multi-surrogate method," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1593–1603, Mar. 2020.

[8] C. Safta, R. L.-Y. Chen, H. N. Najm, A. Pinar, and J.-P. Watson, "Efficient uncertainty quantification in stochastic economic dispatch," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2535–2546, Jul. 2017.

[9] S. Roy, "Inclusion of short duration wind variations in economic load dispatch," *IEEE Trans. Sustain. Energy*, vol. 3, no. 2, pp. 265–273, Apr. 2012.

[10] P. A. Ruiz, C. R. Philbrick, E. Zak, K. W. Cheung, and P. W. Sauer, "Uncertainty management in the unit commitment problem," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 642–651, May 2009.

[11] S. Takriti, J. R. Birge, and E. Long, "A stochastic model for the unit commitment problem," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1497–1508, Aug. 1996.

[12] J. Li, N. Ou, G. Lin, and W. Wei, "Compressive sensing based stochastic economic dispatch with high penetration renewables," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1438–1449, Mar. 2019.

[13] A. S. Nair, P. Ranganathan, H. Salehfar, and N. Kaabouch, "Uncertainty quantification of wind penetration and integration into smart grid: A survey," in *Proc. 49th North Amer. Power Symp.*, 2017, pp. 1–6.

[14] Z. Lu, M. Liu, W. Lu, and Z. Deng, "Stochastic optimization of economic dispatch with wind and photovoltaic energy using the nested sparse grid-based stochastic collocation method," *IEEE Access*, vol. 7, pp. 91 827–91 837, 2019.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[16] S. Fang and H.-D. Chiang, "A high-accuracy wind power forecasting model," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1589–1590, Mar. 2017.

[17] K.-K. Xu, H.-X. Li, and Z. Liu, "ISOMAP-based spatiotemporal modeling for lithium-ion battery thermal process," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 569–577, Feb. 2018.

[18] M. Balasubramanian and E. L. Schwartz, "The Isomap algorithm and topological stability," *Sci.*, vol. 295, no. 5552, 2002, Art. no. 7.

[19] M. L. Eaton, *Multivariate Statistics: A Vector Space Approach*. New York, NY, USA: John Wiley & Sons, 1983.

[20] L. Devroye, "Sample-based non-uniform random variate generation," in *Proc. 18th ACM Conf. Winter Simul.*, 1986, pp. 260–265.
[21] R. B. Nelsen, *An Introduction to Copulas*. Berlin, Germany: Springer, 2007.
[22] A. Gelman *et al.*, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL, USA: Chapman & Hall, 2014.
[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.
[24] Z. Zhang, T. W. S. Chow, and M. Zhao, "M-Isomap: Orthogonal constrained marginal Isomap for nonlinear dimensionality reduction," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 180–191, Feb. 2013.
[25] University of Washington, Power Systems Test Case Archive, Oct. 2019. [Online]. Available: http://www.ee.washington.edu/research/pstca/
[26] NREL Western Wind Data Set, Sep. 2019. [Online]. Available: https://www.nrel.gov/grid/western-wind-data.html, Accessed on: Sep. 20, 2019.
[27] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *Ann. Stat.*, vol. 35, no. 6, pp. 2769–2794, 2007.
[28] L. Rokach and O. Z. Maimon, *Data Mining with Decision Trees: Theory and Applications*, 2nd ed. Singapore: World Scientific, 2008.
[29] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, 2nd ed. New York, NY, USA: Springer, 2018.
[30] J. Zhao, Y. Bao, and G. Chen, "Probabilistic voltage stability assessment considering stochastic load growth direction and renewable energy generation," in *Proc. IEEE Power Energy Soc. Gen. Meet.*, 2018, pp. 1–5.
[31] H. Yu, C. Chung, K. Wong, H. Lee, and J. Zhang, "Probabilistic load flow evaluation with hybrid Latin hypercube sampling and Cholesky decomposition," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 661–667, May 2009.
[32] Y. Chen, J. Wen, and S. Cheng, "Probabilistic load flow method based on Nataf transformation and Latin hypercube sampling," *IEEE Trans. Sustain. Energy*, vol. 4, no. 2, pp. 294–301, Apr. 2013.
[33] B. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall/CRC, 1998.

Zhixiong Hu is working towards his Ph.D. degree from the Department of Statistics at University of California, Santa Cruz, CA. He is currently a Summer Intern under NSF MSGI program at Berkeley Lab Computing Sciences, Berkeley, CA. He did the Internship at Lawrence Livermore National Laboratory, Livermore, CA in 2019. His research interests include multivariate time series analysis, scalable posterior inferences, uncertainty quantification, graph modeling, graph neural network architecture, and deep reinforcement learning.

Yijun Xu (Member, IEEE) received the Ph.D. degree from Bradley Department of Electrical and Computer Engineering at Virginia Tech, Falls Church, VA, on December, 2018. He is currently a Postdoc Associate at Virginia Tech-Northern Virginia Center, Falls Church, VA. He did the computation internship at Lawrence Livermore National Laboratory, Livermore, CA, and power engineer internship at ETAP – Operation Technology, Inc., Irvine, California, in 2018 and 2015, respectively. His research interests include power system uncertainty quantification, uncertainty inversion, and decision-making under uncertainty.

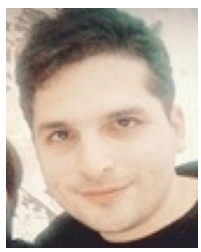
Mert Korkali (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Northeastern University, Boston, MA, in 2013. He is currently a Research Staff Member at Lawrence Livermore National Laboratory, Livermore, CA. From 2013 to 2014, he was a Postdoctoral Research Associate at the University of Vermont, Burlington, VT.

His current research interests lie at the broad interface of robust state estimation and fault location in power systems, extreme event modeling, cascading failures, uncertainty quantification, and probabilistic grid planning. He is the Co-Chair of the IEEE Task Force on Standard Test Cases for Power System State Estimation. Dr. Korkali is currently serving as an Editor of the IEEE OPEN ACCESS JOURNAL OF POWER AND ENERGY and of the IEEE POWER ENGINEERING LETTERS, and an Associate Editor of *Journal of Modern Power Systems and Clean Energy*.

Xiao Chen received his Ph.D. degree in Applied Mathematics from Florida State University, Tallahassee, FL in 2011. He is a Computational Scientist and a Project Leader in the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory, Livermore, CA. He works primarily on the development and application of advanced computational and statistical methods and techniques to power engineering, reservoir simulation, subsurface engineering, and seismic inversion. His research interests include uncertainty quantification, data assimilation, and machine learning.

Lamine Mili (Life Fellow, IEEE) received the Ph.D. degree from the University of Liège, Belgium, in 1987. He is a Professor of Electrical and Computer Engineering, Virginia Tech, Blacksburg. He has five years of industrial experience with the Tunisian electric utility, STEG. At STEG, he worked in the planning department from 1976 to 1979 and then at the Test and Meter Laboratory from 1979 till 1981. He was a Visiting Professor with the Swiss Federal Institute of Technology in Lausanne, the Grenoble Institute of Technology, the École Supérieure D'électricité in France and the École Polytechnique de Tunisie in Tunisia, and did consulting work for the French Power Transmission company, RTE.

His research has focused on power system planning for enhanced resiliency and sustainability, risk management of complex systems to catastrophic failures, robust estimation and control, nonlinear dynamics, and bifurcation theory. He is the Co-Founder and Co-Editor of the *International Journal of Critical Infrastructure*. He is the Chairman of the IEEE Working Group on State Estimation Algorithms. He is a recipient of several awards including the US National Science Foundation (NSF) Research Initiation Award and the NSF Young Investigation Award.



Jaber Valinejad (Student Member, IEEE) is currently working towards his Ph.D. degree at the Bradley Department of Electrical and Computer Engineering, Virginia Tech, USA. His current research interests include power systems, resilience and community resilience, cyber-physical social systems and social computing, artificial intelligence, and reinforcement learning.