This article has been accepted for publication in IEEE Transactions on Sustainable Energy. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2022.3223764

Deep-quantile-regression-based surrogate model for joint chance-constrained optimal power flow with renewable generation

Ge Chen, *Graduate Student Member, IEEE,* Hongcai Zhang, *Member, IEEE,* Hongxun Hui, *Member, IEEE,* and Yonghua Song, *Fellow, IEEE*

***Abstract*—Joint chance-constrained optimal power flow (JCC-OPF) is a promising tool for managing distributed renewable generation uncertainties. However, existing works are usually based on power flow equations, which require accurate network parameters that may be unobservable in many distribution systems. To address this issue, this paper proposes a learning-based surrogate model for JCC-OPF with renewable generation. This model equivalently converts joint chance constraints into quantile-based forms. Two multi-layer perceptrons are trained based on special loss functions to predict the quantile of constraint violations and expected power loss. By reformulating these two MLPs into mixed-integer linear constraints, we can replicate the JCC-OPF without network parameters. Two preprocessing steps, i.e., data augmentation and calibration, are further developed to improve its performance. The former trains a simulator to generate more training samples for enhancing the prediction accuracy of MLPs. The latter designs a positive parameter based on empirical prediction errors to calibrate the outputs of MLPs so that feasibility can be guaranteed. Numerical experiments based on the IEEE 33- and 123-bus systems validate that the proposed model can achieve desirable feasibility and optimality simultaneously with no need for network parameters.**

***Index Terms*—Optimal power flow, joint chance constraints, deep quantile regression, distribution network, distributed renewable generation.**

NOMENCLATURE

Sets

$\mathcal{B}$	Index set of branches.
$\mathcal{L}$, $\mathcal{L}_{\text{pl}}$	Index set of hidden layers in the quantile-MLP and loss-MLP.
$\mathcal{N}$	Index set of historical samples.

Parameters

$\boldsymbol{G}^{\text{DG}}$	Available distributed generation (MW).
$\overline{\boldsymbol{G}}^{\text{DG}}$	Forecast of available distributed generation (MW).
$\hat{h}$	Labels generated by the XGBoost simulator.
$\boldsymbol{I}^{\max}$	Upper bounds of magnitudes of branch current (kA).
N^l	Neuron number in the l-th hidden layer of the quantile-MLP.
$\boldsymbol{p}^{\text{d}}$, $\boldsymbol{q}^{\text{d}}$	Active and reactive power demands (MW).
r_{ij}, x_{ij}	Resistance and reactance of branch (i,j) (p.u.).
$\boldsymbol{V}^{\min}$, $\boldsymbol{V}^{\max}$	Lower and upper bounds of magnitudes of bus voltages (p.u.).
$\boldsymbol{W}$, $\boldsymbol{b}$	Weights and bias of quantile-MLP.
$\boldsymbol{W}^{\text{loss}}$, $\boldsymbol{b}^{\text{loss}}$	Weights and bias of loss-MLP.
α, β	Probability parameters.
δ	Empirical prediction error.
ϵ	Risk parameter in chance constraints.
κ^{Beta}, κ^{Weibull}	Scaling factors for generating uncertainty samples.
ρ	Calibration parameter.
$\boldsymbol{\phi}$	Ratios of actual active outputs of distributed generators to their reactive outputs.

Uncertainties

$\boldsymbol{\omega}$	Uncertain level of distributed generation.

Variables

G	Energy purchasing from the upper-level grid (MW).
h	Maximum violation of power flow constraints.
$\boldsymbol{I}$	Magnitudes of branch currents (kA).
$\boldsymbol{p}$, $\boldsymbol{q}$	Nodal active and reactive power injections (MW).
$\boldsymbol{p}^{\text{DG}}$	Actual used active distributed generation (MW).
p^{loss}	Total power loss (MW).
$\hat{p}^{\text{loss}}$	Prediction of the expected total power loss given by the loss-MLP (MW).
$\boldsymbol{P}$, $\boldsymbol{Q}$	Active and reactive power flows (MW).
$\boldsymbol{r}$, $\boldsymbol{\mu}$	Auxiliary variables.
$\boldsymbol{V}$, $\boldsymbol{I}$	Magnitudes of voltages and currents (p.u).
$\boldsymbol{x}$	Collection of nodal active and reactive power injections (MW).
$\boldsymbol{z}_l$, $\boldsymbol{s}_l$	Outputs of linear mapping and ReLU in neurons.
$\boldsymbol{\lambda}$	Actual utilization rates of distributed generation on each bus.
$\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h)$	$1-\epsilon$ quantile of the maximum violation of power flow constraints.
$\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x})$	Prediction of $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h)$.

Operators and functions

$\mathbb{I}$	Indicator function.
$\mathbb{P}^{\boldsymbol{\omega}}(\cdot)$	Probability under the effects of uncertainty $\boldsymbol{\omega}$.
$\mathbb{E}^{\boldsymbol{\omega}}(\cdot)$	Expectation under the effects of uncertainty $\boldsymbol{\omega}$.

This paper is funded in part by the Science and Technology Development Fund, Macau SAR (File no. SKL-IOTSC(UM)-2021-2023, File no. 0003/2020/AKP, and File no. 0011/2021/AGJ). (Corresponding author: *Hongcai Zhang*.)

G. Chen, H. Zhang, H. Hui, and Y. Song are with the State Key Laboratory of Internet of Things for Smart City and Department of Electrical and Computer Engineering, University of Macau, Macao, 999078 China (email: hczhang@um.edu.mo).

I. INTRODUCTION

OPTIMAL power flow (OPF) plays a critical role in the operation of distribution networks [1]. By solving OPF, network operators can find the most economical dispatch strategy while ensuring operational security. However, since distributed generation (DG) has been increasingly integrated into distribution networks [2], considerable uncertainties are introduced, which dramatically increases the difficulty of solving OPF [3]. Traditionally, the uncertainties in OPF are usually managed by robust optimization [4] or stochastic programming [5]. However, robust optimization usually

derives overly conservative solutions because it requires that constraints should be satisfied even in extreme conditions. Stochastic programming may be time-consuming since it introduces many additional variables and constraints to describe different scenarios. Chance constrained programming (CCP) is an alternative method to account for the uncertainties from DG in OPF [6]. It allows constraint violations with a small probability so that operators can effectively balance robustness and optimality based on their preferences. Moreover, once we find tractable reformulations of chance constraints, CCP can achieve great optimality and computational efficiency simultaneously. Many recent efforts have also been made to apply CCP to OPF. Reference [7] combined CCP to the linearized DistFlow model to coordinate uncertain DG with flexible resources in distribution networks. Reference [8] applied CCP to a linearized AC OPF model to schedule wind generation. However, the above papers used individual chance constraints to control the violation probability of every critical constraint. This individual manner may not guarantee the joint satisfaction probability of all critical constraints [9].

Joint chance-constrained optimal power flow (JCC-OPF) directly restricts the joint satisfaction probability of all critical constraints [10]. Hence, it is preferable for ensuring system-level security. For instance, references [11], [12] employed JCC-OPF to restrict the joint probability of critical constraint violations, where Bonferroni approximation was used to convert joint chance constraints into solvable individual ones. Reference [13] proposed a joint chance-constrained linearized DistFlow model to schedule the reactive power compensation for distribution networks. References [14], [15] combined JCC-OPF with a scenario approach to reformulate the probability constraints into tractable deterministic forms. However, most existing works still face two challenges:

1) Most published works, including [10]–[15], are based on power flow equations, which require accurate power network parameters (e.g., network topology and branch impedance). However, these parameters may be unknown in distribution networks due to unaware topology changes or inaccurate data maintenance [16].
2) Existing papers usually introduce approximations (e.g., the linearized DistFlow [13]) or relaxations (e.g., the semi-definite relaxation of the DistFlow [14], [15]) so that the impacts of uncertainties are convenient to describe. However, these approximation or relaxation models may affect the optimality or feasibility of solutions. For instance, reverse power flows may occur in radial distribution networks with high DG penetration. In that case, the semi-definite relaxation is not exact and may not ensure feasibility [17].

Due to the widespread use of smart meters, collecting operational data (e.g., power injections, bus voltages, and branch currents) in distribution systems has become easier and cheaper nowadays. Since this data contains some knowledge of distribution network parameters, learning-based methods may directly utilize this data to solve OPF, which may overcome the above two challenges [18]. The published learning-based methods for OPF problems can be generally divided into the following four categories.

1) Learning-assisted methods: Learning-assisted methods usually leverage machine learning techniques to help model-based methods to make chance constraints tractable. For instance, references [19], [20] employed the Gaussian mixture model to approximate non-Gaussian uncertainties in OPF with Gaussian uncertainties. Then, chance constraints can be easily reformulated into tractable forms. Reference [21] combined the scenario approach with a learning-based sampling method to reduce the computational burden. However, they are model-based methods, in which network parameters are still required.

2) OPF-then-learn methods: OPF-then-learn methods first generate a training dataset by solving multiple OPF instances, where the input features are usually the system's operation conditions (e.g., power demands and/or uncontrollable renewable generation) and the output labels are OPF solutions (e.g., power schedules), respectively. Then, supervised learning models are trained to predict optimal solutions of OPF based on given inputs directly. For example, references [22], [23] trained multi-layer perceptrons (MLPs) to predict solutions of DC and AC OPF problems. These methods were further combined with the Lagrangian dual approach [24] or reconstruction steps [25] to enhance their feasibility. Reference [26] replaced MLPs with graph neural networks to improve prediction accuracy. Reference [27] extended these methods to probabilistic OPF by replacing MLPs with Gaussian process regression. Generally speaking, these methods can reduce the solving time of OPF because the solving process is replaced by the inference of learning models. However, they require OPF solutions as training labels. In order to generate these labels, they still need to solve physical model-based OPF problems.

3) Reinforcement learning: Reinforcement learning (RL) trains agents in a specific environment to maximize the cumulative reward (e.g., the opposite of energy purchasing from the upper-level grid). In reference [28], RL was combined with the Lagrangian dual approach to solve OPF with guaranteed feasibility. In reference [29], behavior cloning was combined with RL to generate a desirable initial start so that the training process can be accelerated. Besides, RL has also been applied to many other scheduling problems, such as volt-VAR optimization [30] and battery controls [31]. Since RL can be model-free, it has the potential to solve OPF without network parameters. However, RL needs lots of "trial-and-error" based on interactions with the existing distribution systems to learn a policy. The "trial-and-error" can be risky for real distribution systems and is unacceptable in practice.

4) Constraint learning methods: Constraint learning methods usually train neural networks to learn OPF constraints. For instance, references [32], [33] trained binary classifiers to judge the feasibility of a given decision. Then, the trained classifiers were equivalently reformulated as mixed-integer linear constraints so that the OPF problems could be replicated. Reference [34] replaced the binary classifiers with a regression neural network to improve the feasibility of solutions. Constraint learning only requires operational data instead of OPF solutions as training labels, so the requirement of network parameters can be bypassed. However, it is challenging for constraint learning to handle joint chance constraints. Specif-

ically, if we wish to learn joint chance constraints, then our training sets must contain enough samples of statistical results (e.g., the quantile of constraint violations). However, these samples are difficult to collect because only realizations of variables can be observed in practice.

To overcome the aforementioned challenges, this paper proposes a data-driven surrogate model for JCC-OPF with renewable generation. The specific contributions are twofold:

1) We extend the conventional constraint learning methods to joint chance constraints and develop a surrogate model for JCC-OPF. Two MLPs are trained to predict the quantile of the maximum constraint violation and expectation of the power loss. Inspired by deep quantile regression techniques, we introduce special loss functions so that these MLPs can be trained based on only historical realizations. Then, by reformulating these two MLPs into mixed-integer linear constraints, JCC-OPF can be replicated without network parameters.
2) We design two pre-processing steps, i.e., data augmentation and calibration, to improve the performance of the proposed surrogate model. The former trains a simulator based on historical data to generate more samples for improving the accuracy of MLPs. The latter calibrates the outputs of MLPs based on empirical prediction errors to help enhance the feasibility of the solutions.

The remaining parts are organized as follows. Section II describes the formulation of the JCC-OPF problem. Section III introduces the proposed learning-based surrogate model in detail. Section IV demonstrates simulation results, and Section V concludes this paper.

II. Formulation of JCC-OPF

In this section, we first introduce the traditional model-based formulation of JCC-OPF.

1) Power injections: By using $i \in \mathcal{V}$ to index buses, the active and reactive power injections on each bus, i.e., $\boldsymbol{p} \in \mathbb{R}^{|\mathcal{V}|}$ and $\boldsymbol{q} \in \mathbb{R}^{|\mathcal{V}|}$, can be expressed as:

$$\boldsymbol{p} = -\boldsymbol{p}^{\mathrm{d}} + \boldsymbol{p}^{\mathrm{DG}}, \quad \boldsymbol{q} = -\boldsymbol{q}^{\mathrm{d}} + \boldsymbol{\phi} * \boldsymbol{p}^{\mathrm{DG}}, \tag{1}$$

where $\boldsymbol{p}^{\mathrm{d}}$ and $\boldsymbol{q}^{\mathrm{d}}$ represent the active and reactive power demands on each bus, respectively; $\boldsymbol{p}^{\mathrm{DG}}$ is the actual used active power from DG; $\boldsymbol{\phi}$ is a ratio of the actual active power of DG to its reactive power; $*$ denotes the element-wise multiplication. The actual used active power $\boldsymbol{p}^{\mathrm{DG}}$ can be expressed by:

$$\boldsymbol{p}^{\mathrm{DG}} = \boldsymbol{\lambda} * \boldsymbol{G}^{\mathrm{DG}}, \tag{2}$$

where $\boldsymbol{\lambda}$ and G^{DG} are the actual utilization rate and maximum available value of DG, respectively. In practice, the value of $\boldsymbol{G}^{\mathrm{DG}}$ is uncertain, which can be expressed as follows:

$$\boldsymbol{G}^{\mathrm{DG}} = \overline{\boldsymbol{G}}^{\mathrm{DG}} * (\mathbf{1} + \boldsymbol{\omega}), \tag{3}$$

where $\overline{\boldsymbol{G}}^{\mathrm{DG}}$ represents the nominal available DG obtained by predictions and $\boldsymbol{\omega}$ is the corresponding uncertain level.[1]

2) Power flow model: The power flow model of a radial network can be expressed by the DistFlow model [35]:

$$\begin{cases} \sum_{k\in\mathcal{C}_j} P_{jk} = p_j + P_{ij} - r_{ij} I_{ij}^2, \\ \sum_{k\in\mathcal{C}_j} Q_{jk} = q_j + Q_{ij} - x_{ij} I_{ij}^2, \\ V_j^2 = V_i^2 - 2(r_{ij}P_{ij} + x_{ij}Q_{ij}) \\ \qquad\quad + (r_{ij}^2 + x_{ij}^2) I_{ij}^2, \\ I_{ij}^2 = \frac{P_{ij}^2 + Q_{ij}^2}{V_i^2}, \end{cases} \quad \forall (i,j) \in \mathcal{B}, \tag{4}$$

where P_{ij} and Q_{ij} are the active and reactive power flows on branch (i,j), respectively; V_i and I_{ij} are the magnitudes of the voltage at bus i and current on branch (i,j), respectively; r_{ij} and x_{ij} denotes the resistance and reactance of branch (i,j), respectively. Set $\mathcal{C}_j$ contains the child bus indexes of bus j. Set $\mathcal{B}$ represents the index set of branches in this network.

3) Security constraints: To ensure operation security, the magnitudes of all bus voltages and branch currents shall maintain in the corresponding allowable ranges. According to (1)-(4), the uncertainties from DG also affect the bus voltages and branch currents. To better balance the optimality and feasibility of solutions, a joint chance constraint is employed to describe the voltage and current limitations:

$$\mathbb{P}^{\boldsymbol{\omega}} \left(\boldsymbol{V}^{\min} \le \boldsymbol{V} \le \boldsymbol{V}^{\max},\ \boldsymbol{I} \le \boldsymbol{I}^{\max} \right) \ge 1 - \epsilon, \tag{5}$$

where $\mathbb{P}^{\boldsymbol{\omega}}(\cdot)$ denotes the probability of constraints satisfaction under the influence of uncertainty $\boldsymbol{\omega}$; $\boldsymbol{V}$ and $\boldsymbol{I}$ are the vector forms of V_i and I_{ij}; ϵ is the risk parameter.

4) Energy purchasing: The energy purchasing from the upper-level grid, i.e., G, can be calculated based on the network-level power balance:

$$G = \mathbf{1}^{\intercal} \boldsymbol{p}^{\mathrm{d}} + p^{\mathrm{loss}} - \boldsymbol{\lambda}^{\intercal} \boldsymbol{G}^{\mathrm{DG}}, \tag{6}$$

where p^{loss} is the total power loss and can be calculated by:

$$p^{\mathrm{loss}} = \sum_{(i,j)\in\mathcal{B}} r_{ij} I_{ij}^2. \tag{7}$$

Finally, the JCC-OPF is formulated as:

$$\min_{\boldsymbol{\lambda}, G} \quad \mathbb{E}^{\boldsymbol{\omega}}(G), \quad \text{s.t.: Eqs. (1)-(7).} \tag{**P1**}$$

where $\mathbb{E}^{\boldsymbol{\omega}}(\cdot)$ is the expected energy purchasing from the upper-level grid under the effects of uncertainty $\boldsymbol{\omega}$.

III. Solution Methodology

As mentioned in Section I, formulating **P1** may be challenging because some network parameters are often unknown. Therefore, we propose a learning-based surrogate model to address the previous challenge. This model first introduces deep quantile regression to replicate the joint chance constraint (5). Meanwhile, another neural network is trained to predict

[1] In practice, the available outputs of different distributed generators may have potential correlations. Since these correlations are already contained in the operational data, they can be learned by neural networks. Thus, the proposed learning-based model can also consider these correlations.

the expected power loss. Then, by reformulating the trained neural networks into mixed-integer linear constraints, **P1** can be replicated without network parameters.

A. Deep quantile regression to replicate chance constraints

1) Motivation: The joint chance constraint (5) can be equivalently reformulated into quantile-based deterministic forms. Specifically, we define $\boldsymbol{x}$ as the nominal active and reactive power injections on each bus (except the root bus):

$$\boldsymbol{x} = [\boldsymbol{p},\ \boldsymbol{q}]. \tag{8}$$

We also define a new variable h as the maximum violation of OPF constraints:

$$h(\boldsymbol{x},\boldsymbol{\omega}) = \max\{\boldsymbol{V}^{\min} - \boldsymbol{V}(\boldsymbol{x},\boldsymbol{\omega}), \boldsymbol{V}(\boldsymbol{x},\boldsymbol{\omega}) - \boldsymbol{V}^{\max}, \boldsymbol{I}(\boldsymbol{x},\boldsymbol{\omega}) - \boldsymbol{I}^{\max}\}. \tag{9}$$

Note the impacts of the uncertainty from DG, i.e., $\boldsymbol{\omega}$, have been implied in the samples of $h(\boldsymbol{x},\boldsymbol{\omega})$ because both the voltage $\boldsymbol{V}$ and current $\boldsymbol{I}$ are affected by $\boldsymbol{\omega}$. Based on (9), the joint chance constraint (5) can be expressed as:

$$\mathbb{P}^{\boldsymbol{\omega}}\left(h(\boldsymbol{x},\boldsymbol{\omega}) \le 0\right) \ge 1-\epsilon, \tag{10}$$

which can be further equivalently reformulated into the following quantile-based form:

$$\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega})) \le 0, \tag{11}$$

where $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$ is the $1-\epsilon$ quantile of h at a given $\boldsymbol{x}$:

$$\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega})) = \inf\{y : \mathbb{P}^{\boldsymbol{\omega}}\left(h(\boldsymbol{x},\boldsymbol{\omega}) \le y\right) \ge 1-\epsilon\}. \tag{12}$$

According to (11), if the mapping from $\boldsymbol{x}$ to $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$ can be accurately described by simple relations, then the the joint chance constraint can be tractable. This motivates us to introduce a powerful deep learning technique, deep quantile regression, to predict $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$.

2) Introduction of deep quantile regression: Traditional regression is a process to model the relationship between dependent output variables and independent input variables. For example, based on the dataset $\{(\boldsymbol{x}_n,\boldsymbol{\omega}_n,h_n)\}_{n\in\mathcal{N}}$ ($\mathcal{N}$ is the index set of samples), we can train a regression model $\hat{h}(\boldsymbol{x},\boldsymbol{\omega})$ to predict h with a given $\boldsymbol{x}$ and $\boldsymbol{\omega}$. The mean squared error is usually employed as the loss function to train traditional regression models:

$$\text{Loss}^{\text{R}} = (h(\boldsymbol{x},\boldsymbol{\omega}) - \hat{h}(\boldsymbol{x},\boldsymbol{\omega}))^2. \tag{13}$$

When traditional regression models are employed to predict the quantile $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$, they require samples of $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$ as the training labels. However, historical operational data may not contain sufficient quantile samples.

Conversely, deep quantile regression can predict the quantile only based on realizations of h [36]. This advantage results from a special loss function, as follows:

$$\text{Loss}^{\text{QR}} = \psi\cdot(1-\epsilon-\mathbb{I}(\psi\le 0)). \tag{14}$$

Here $\psi = h - \hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x})$, where $\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x})$ is the prediction of $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$ given by the quantile regression. Symbol $\mathbb{I}(\cdot)$ denotes the indicator function. The following **Proposition** proves that we can predict the quantile without quantile samples based on the loss function (14).

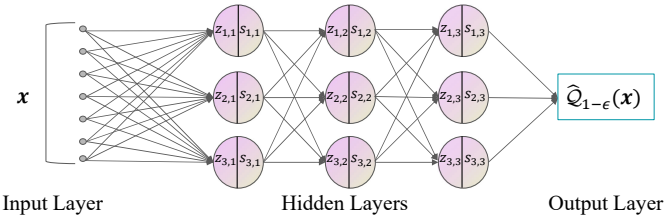


Fig. 1. Structure of an example MLP with 3 hidden layers, where $z_{n,l}$ and $s_{n,l}$ denote the outputs of the linear mapping and ReLU of the n-th neroun in layer l, respectively.

Proposition 1. *The quantile $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$ can be obtained by minimizing the expectation of (14), as follows [36]:*

$$\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega})) = \underset{\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x})}{\operatorname{argmin}}\ \mathbb{E}(\text{Loss}^{\text{QR}}). \tag{15}$$

Proof: See Appendix A.

As a result, we can train a quantile regression neural network based on the historical realizations, i.e., $\{(\boldsymbol{x}_n,h_n)\}_{n\in\mathcal{N}}$ to represent the mapping from $\boldsymbol{x}$ to the quantile $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$. Note the label h_n is noisy due to the impacts of $\boldsymbol{\omega}$.

3) Replication of joint chance constraints: With the noisy dataset $\{(\boldsymbol{x}_n,h_n)\}_{n\in\mathcal{N}}$, we can train a quantile regression network $\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x})$ to predict the target quantile $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$. Then, Eq. (11) is replaced by:

$$\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x}) \le 0. \tag{16}$$

This paper chooses an MLP with ReLU activation as the quantile regression model. For convenience, this MLP is called "quantile-MLP." A typical MLP is composed of one input layer, $|\mathcal{L}|$ hidden layers, and one output layer, as shown in Fig. 1. Each neuron is made up of a linear mapping and a nonlinear ReLU function. By using symbol l to index the hidden layers ($l\in\mathcal{L}$), the target quantile can be estimated by the forward propagation of the trained quantile-MLP, as follows:

$$\boldsymbol{s}_0 = \boldsymbol{x}, \tag{17}$$

$$\boldsymbol{z}_l = \boldsymbol{W}_l\boldsymbol{s}_{l-1} + \boldsymbol{b}_l, \forall l\in\mathcal{L}, \tag{18}$$

$$\boldsymbol{s}_l = \max(\boldsymbol{z}_l, 0), \quad \forall l\in\mathcal{L}, \tag{19}$$

$$\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x}) = \boldsymbol{W}_{|\mathcal{L}|+1}\boldsymbol{s}_{|\mathcal{L}|} + \boldsymbol{b}_{|\mathcal{L}|+1}. \tag{20}$$

Eq. (17) defines the input layer; Eqs. (18) and (19) represent the linear mapping and ReLU in hidden layers, respectively; Eq. (20) defines the output layer. Vector $\boldsymbol{z}_l$ and $\boldsymbol{s}_l$ are the outputs of the linear mapping and activation function in hidden layer l; Matrix $\boldsymbol{W}_l$ and vector $\boldsymbol{b}_l$ are the weights and bias of layer l, which are parameters to be learned; $\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x})$ is the estimation of $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x},\boldsymbol{\omega}))$ given by the quantile-MLP.

Remark 1. *We can also let the quantile-MLP output multiple quantile values at once so that operators can adjust their operational strategies based on their preferences. To realize this, we only need to change its forward propagation into:*

$$\begin{cases} \textit{Eqs. (17)-(19)}, \\ \left[\hat{\mathcal{Q}}_{1-\epsilon_i}(\boldsymbol{x}), \forall i\in\mathcal{I}\right]^{\intercal} = \boldsymbol{W}_{|\mathcal{L}|+1}\boldsymbol{s}_{|\mathcal{L}|} + \boldsymbol{b}_{|\mathcal{L}|+1}, \end{cases} \tag{21}$$

where $\mathcal{I}$ is the index set of risk parameters. Then, even if multiple quantile values are required, only one single MLP needs to be trained.

B. Replication of the objective

Based on (6), the objective of **P1** can be calculated by:

$$\begin{aligned}\mathbb{E}^{\boldsymbol{\omega}}(G) &= \mathbb{E}^{\boldsymbol{\omega}}\left(\mathbf{1}^{\intercal}\boldsymbol{p}^{\mathrm{d}} + p^{\mathrm{loss}} - \boldsymbol{\lambda}^{\intercal}\boldsymbol{G}^{\mathrm{DG}}\right), \\ &= \mathbf{1}^{\intercal}\boldsymbol{p}^{\mathrm{d}} + \mathbb{E}^{\boldsymbol{\omega}}(p^{\mathrm{loss}}) - \boldsymbol{\lambda}^{\intercal}\boldsymbol{G}^{\mathrm{DG}}. \end{aligned} \tag{22}$$

Eq. (22) indicates that the expected power loss is necessary. According to (7), the power loss calculation needs branch currents, so the power flow model is still required. To bypass this requirement, another MLP (we call it "loss-MLP"), is trained to predict the expected power loss. The samples of nominal power injection $\boldsymbol{x}$ and power loss p^{loss} are treated as the features and noisy labels (p^{loss} is affected by the uncertainty $\boldsymbol{\omega}$), respectively. The mean squared error is employed as the loss function, as follows:

$$\mathrm{Loss}^{\mathrm{loss}} = (p^{\mathrm{loss}} - \hat{p}^{\mathrm{loss}}(\boldsymbol{x}))^2, \tag{23}$$

where $\hat{p}^{\mathrm{loss}}(\boldsymbol{x})$ is the prediction given by the loss-MLP.

Proposition 2. *The expected power loss can be obtained by minimizing the expectation of (23), as follows:*

$$\mathbb{E}^{\boldsymbol{\omega}}(p^{loss}) = \underset{\hat{p}^{loss}(\boldsymbol{x})}{\operatorname{argmin}}\, \mathbb{E}(\mathrm{Loss}^{loss}). \tag{24}$$

Proof: See Appendix B.

After training, the expected power loss can be also predicted based on the forward propagation of the loss-MLP:

$$\boldsymbol{s}_0^{\mathrm{loss}} = \boldsymbol{x}, \tag{25}$$

$$\boldsymbol{z}_l^{\mathrm{loss}} = \boldsymbol{W}_l^{\mathrm{loss}}\boldsymbol{s}_{l-1}^{\mathrm{loss}} + \boldsymbol{b}_l^{\mathrm{loss}}, \forall l \in \mathcal{L}^{\mathrm{loss}}, \tag{26}$$

$$\boldsymbol{s}_l^{\mathrm{loss}} = \max(\boldsymbol{z}_l^{\mathrm{loss}}, 0), \quad \forall l \in \mathcal{L}^{\mathrm{loss}}, \tag{27}$$

$$\mathbb{E}^{\boldsymbol{\omega}}(p^{\mathrm{loss}}) = \boldsymbol{W}_{|\mathcal{L}^{\mathrm{loss}}|+1}^{\mathrm{loss}}\boldsymbol{s}_{|\mathcal{L}^{\mathrm{loss}}|} + \boldsymbol{b}_{|\mathcal{L}^{\mathrm{loss}}|+1}^{\mathrm{loss}}, \tag{28}$$

where the subscript "loss" is used to mark those variables belonging to the loss-MLP.

C. Mixed-integer linear replication of JCC-OPF

Once the two MLPs are trained, the quantile of the maximum constraint violation and expected power loss can be predicted by (17)-(20) and (25)-(28) with no need for building any power flow model. However, Eqs. (19) and (27) are intractable due to the maximum operator. To address this, we employ the Big-M method used in [32]–[34] to convert these intractable constraints into mixed-integer linear forms. Specifically, by introducing auxiliary variables $\boldsymbol{r}_l$ and $\boldsymbol{\mu}_l$ for each hidden layer, Eqs. (18)-(19) can be reformulated as:

$$\begin{cases} \boldsymbol{s}_l - \boldsymbol{r}_l = \boldsymbol{W}_l\boldsymbol{s}_{l-1} + \boldsymbol{b}_l, \\ 0 \le \boldsymbol{s}_l \le M \cdot \boldsymbol{\mu}_l, \\ 0 \le \boldsymbol{r}_l \le M \cdot (1 - \boldsymbol{\mu}_l), \\ \boldsymbol{\mu}_l \in \{0,1\}^{N_l}, \end{cases} \tag{29}$$

where M is a big enough real number; N_l denotes the neuron number in the l-th hidden layer of the quantile-MLP. Here each element of vector $\boldsymbol{\mu}_l$ also represents the activation state of every neuron. If the n-th element $\mu_{n,l}$ equals one, then the n-th neuron in hidden layer l is "active" (i.e., its ReLU's input is non-negative). If $\mu_{n,l} = 0$, then this neuron is "inactive" (i.e., its ReLU's input is negative). Note that many published works have confirmed the exactness of the transformation from (18)-(19) to (29) [37], [38]. Thus, this transformation does not introduce any additional approximation error.

Similarly, Eqs. (26)-(27) can be equivalently converted into the same form of (29), which is recorded as follows:

$$\{\text{Eq. (29)}\}^{\mathrm{loss}}. \tag{30}$$

Then, the original JCC-OPF problem, **P1**, can be replicated by the following learning-based surrogate model:

$$\begin{aligned} \min_{\boldsymbol{\lambda}, G} \quad & \mathbb{E}^{\boldsymbol{\omega}}(G), \\ \text{s.t.} \quad & \text{Eqs. (8), (16), (17), (20), (22), (25), and (28)-(30).} \end{aligned} \tag{**P2**}$$

Obviously, the auxiliary binary variable number in **P2** is the same as the total neuron numbers of the two MLPs.

Remark 2. *The proposed surrogate model only requires historical samples to train MLPs but does not need to build an exact power flow model. As a result, even if some network parameters are unobservable so that an OPF model can not be formulated mathematically based on physical laws, we can still find the solutions for the JCC-OPF by solving the proposed data-driven surrogate model.*

D. Pre-processing to improve performance

1) Motivation: If the quantile-MLP is directly trained based on the historical dataset $\{\boldsymbol{x}_n, h_n\}_{n\in\mathcal{N}}$, its prediction accuracy may be undesirable. Specifically, the historical dataset usually only contains one sample $\{\boldsymbol{x}_n, h_n\}$ at $\boldsymbol{x} = \boldsymbol{x}_n$ (other samples usually have different $\boldsymbol{x}$). Thus, the quantile-MLP may not learn the true distribution of h at $\boldsymbol{x} = \boldsymbol{x}_n$ well. Moreover, even if we have sufficient samples to train the quantile-MLP, prediction errors are still inevitable, which may harm the feasibility of solutions. Thus, two pre-processing steps, i.e., data augmentation and calibration, are designed to improve the performance of the proposed model.

2) Data augmentation: We design a data augmentation step to construct an ideal training set for the quantile-MLP. Its key idea is very straightforward: Train a simulator based on the historical data and use this simulator to generate more training samples. The detailed procedure is summarized in Table 1. Then, at a given $\boldsymbol{x} = \boldsymbol{x}^{(k)}$, multiple labels, i.e., $\{h_n^{(k)}\}_{n=1}^{N_\omega}$ can be generated. As a result, the distribution of h can be explicitly described. Here the XGBoost regressor is used as our simulator due to its great prediction accuracy [39].

3) Calibration: Although the XGBoost-based simulator can generate sufficient training data, the generated data may not present the true distribution. In other words, the prediction errors of the simulator may be significant. To guarantee the feasibility of solutions, we further design a calibration step after data augmentation. Its key idea is to calibrate the foretasted quantile based on the largest empirical prediction

This article has been accepted for publication in IEEE Transactions on Sustainable Energy. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2022.3223764

Algorithm 1 Data augmentation

01 **Simulator training:** Train a regressor as our simulator based on the historical dataset $\{\boldsymbol{x}_n, \boldsymbol{\omega}_n, h_n\}_{n\in\mathcal{N}}$. Its input and output are $(\boldsymbol{x}, \boldsymbol{\omega})$ and h, respectively;
02 **For** $k \in \mathcal{K} = [1, 2, \cdots, K]$
03 Randomly select one $\boldsymbol{x}$ and multiple $\boldsymbol{\omega}$ from the historical dataset to construct different pairs, and record them as $\{(\boldsymbol{x}^{(k)}, \boldsymbol{\omega}_n^{(k)})\}_{n=1}^{N_\omega}$ (N_ω is the number of $\boldsymbol{\omega}$ we chosen);
04 Give the above pairs as inputs to the simulator to predict h, and record the predictions as $\{\hat{h}_n^{(k)}\}_{n=1}^{N_\omega}$;
05 **End for**
06 **Data collection:** Collect the generated pairs and predictions to construct a new dataset, i.e., $\{\boldsymbol{x}^{(k)}, \boldsymbol{\omega}_n^{(k)}, \hat{h}_n^{(k)}\}_{n=1}^{N_\omega}, \forall k \in \mathcal{K}$. By removing $\boldsymbol{\omega}_n^{(k)}$, we can get the ideal training set of the quantile-MLP, i.e., $\{\boldsymbol{x}^{(k)}, \hat{h}_n^{(k)}\}_{n=1}^{N_\omega}, \forall k \in \mathcal{K}$.

error of the simulator. Specifically, these empirical prediction errors can be expressed as:

$$\delta_n = h_n - \hat{h}_n, \quad \forall n \in \mathcal{N}, \tag{31}$$

where h_n is the actual sample of h; $\hat{h}_n$ is the prediction of the simulator; $\mathcal{N}$ is the index set of the historical sample. We use ρ to represent the largest empirical prediction error:

$$\rho = \max_{\forall n\in\mathcal{N}} \delta_n. \tag{32}$$

Meanwhile, by using $\hat{h}_n$ as training labels, we can train the quantile-MLP to predict the target quantile $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega}))$. We record this prediction as $\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x})$. Then, the joint chance constraint (11) can be approximated by:

$$\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x}) + \rho \leq 0. \tag{33}$$

Note that we can easily collect sufficient samples of $\hat{h}$ to train the quantile-MLP because $\hat{h}$ is generated by our simulator. Thus, we can accurately predict $\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x})$.

Proposition 3. *Suppose the sample number $|\mathcal{N}|$ satisfies:*

$$|\mathcal{N}| \geq \frac{1}{\alpha}\frac{e}{1-e}\left(2n^\delta - 1 + \ln\frac{1}{\beta}\right), \tag{34}$$

where $\alpha \in [0, 1]$ and $\beta \in [0, 1]$; n^δ is a parameter and here we have $n^\delta = 1$. Then, the following chance constraint holds with confidence at least $1 - \beta$:

$$\mathbb{P}^{\boldsymbol{\omega}}\left(\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x}) + \rho \geq \mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega}))\right) \geq 1 - \alpha, \tag{35}$$

where $\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x}) + \rho$ and $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega}))$ are the left-hand side terms of the proposed approximation (33) and original joint chance constraint (11), respectively.

Proof: See Appendix C.

Proposition 3 demonstrates that (33) can serve as a conservative approximation of (11) with probability $1-\alpha$. According to (34), the value of α can be kept at a small level when sufficient samples are used to calculate ρ. In this case, we can guarantee the feasibility of the proposed model even if there are prediction errors.

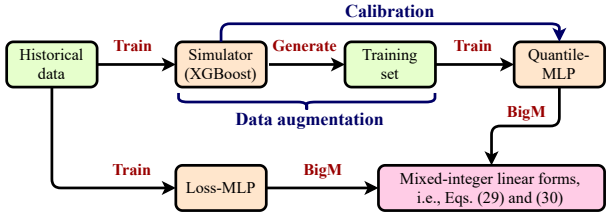


Fig. 2. The whole procedure to establish the proposed surrogate model.

E. Summary of the proposed surrogate model

1) Whole procedure: By applying the pre-processing steps, **P2** can be replaced by the following surrogate model **P3**:

$$\min_{\boldsymbol{\lambda}, G} \quad \mathbb{E}^{\boldsymbol{\omega}}(G), \tag{**P3**}$$
$$\text{s.t.:} \quad \text{Eqs. (8), (17), (20), (22), (25), (28)-(30), and (33).}$$

Fig. 2 illustrates the procedure for establishing the proposed learning-based surrogate model. Specifically, we first leverage historical data to train a simulator for data augmentation. Then, the quantile-MLP is trained to predict the target quantile $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega}))$. A calibration step is further applied to improve feasibility. Meanwhile, the loss-MLP is trained based on the historical data to predict the expected power loss. Finally, by reformulating these two MLPs into solvable mixed-integer forms, the proposed surrogate model of JCC-OPF can be established without network parameters.

2) Measurement requirement: The proposed surrogate model relies on data measurements from distribution networks. According to (2) and (8)-(9), the required historical samples include nodal power injections, bus voltages, and branch currents. In practice, power injections are usually measured because they determine the electricity bill of users. However, since the measurement redundancy on a distribution system is usually low, only parts of bus voltage and branch currents are monitored in practice. Nevertheless, we can still calculate the samples of the maximum constraint violation among the measured buses and branches based on (9). By using these samples to train MLPs, the proposed surrogate model can derive a strategy that can at least guarantee the operational security of the measured buses and branches. On the contrary, the model-based methods and other data-driven models (e.g., OPF-then-learning methods) can hardly derive a solution in that case without complete and accurate measurement of network parameters. As a result, the proposed model is still meaningful for systems with low measurement redundancy.

In practice, the topology of the distribution network may change because of network reconfiguration. Hence, the trained MLPs in one topology might show undesirable performance in another new topology. Nevertheless, in practice distribution networks, only a few switches participate in reconfiguration. Thus, the maximum possible number of topology scenarios is very limited [40]. Since the switching operations are usually controlled by operators, so that the switching states can be easily recorded. Therefore, the operators can train different MLPs for every topology scenario. As a result, the proposed model can be readily extended to consider the cases with topology changes.

3) Advantages of MLPs: Except for MLPs, many other state-of-art deep learning models, e.g., Convolutional Neural

This article has been accepted for publication in IEEE Transactions on Sustainable Energy. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2022.3223764

Networks (CNNs), Recurrent Neural Networks (RNNs), and Graph Neural Networks (GNNs), have been employed for solving OPF problems [26], [41]. Nevertheless, we choose MLPs instead of other models to serve as the surrogate model of JCC-OPF. MLPs have simple structures but desirable approximation capability [42]. Moreover, MLPs can be equivalently reformulated as tractable mixed-integer linear forms. Conversely, CNNs may not be suitable for JCC-OPF because they are invariant to translation [42].[2] RNNs usually use hyperbolic tangent activation functions, which are not piece-wise linear. Thus, it is hard to reformulate RNNs into tractable forms. GNNs can utilize the topology information to improve their prediction accuracy [44]. However, the topology may be unknown in many distribution systems so that GNN models may be hard to build.

Besides the above state-of-art learning models, many interpretable learning models have also been proposed for power system modeling, including decision trees [45] and ensemble models [46]. However, most of them only focused on modeling instead of scheduling problems like JCC-OPF. Moreover, an MLP can usually achieve better approximation than a single decision tree. Thus, we choose MLPs as the surrogate model of JCC-OPF. Some state-of-the-art techniques can also improve the interpretability of learning models including MLPs, such as SHapley Additive exPlanation [47], Local interpretable model-agnostic explanations [48], and Deep Learning Important FeaTures [49]. These techniques can tell the importance of every feature. However, it is hard for them to fulfill the need for power system modeling because they may not explicitly describe the true physics of systems. Enhancing the interpretability of the proposed model could be an important future research direction.

IV. Case Study

A. Simulation set up

We implement our case studies based on two test systems, i.e., the IEEE 33-bus and 123-bus systems. The slack bus voltages of these two systems, i.e., V_1, are 12.66 kV and 4.16 kV (phase-to-phase voltage), respectively. The feasible region of bus voltages in both systems are set as [0.9 p.u., 1.1 p.u.], while the maximum allowable branch currents are 0.421 kA and 0.7 kA (phase-to-phase current), respectively. The ratio of the DG's active power to its reactive power, i.e., parameter ϕ in (1), is set as 0 and 0.33 in these two systems, respectively. Note that the layout of DG units in a distribution network can be either concentrated [50] or dispersed [51], [52]. Since this layout may affect the power flow results significantly, we design two sets of case studies to better demonstrate the effectiveness of the proposed learning-based model: 1) "concentrated layout", i.e., all DG units are close to each other, and 2) "dispersed layout", i.e., different DG units are dispersed in the power networks. Fig. 3 shows the 33-bus and 123-bus test systems with the concentrated layout, where the nominal available generation capacities of DG units, $\overline{G}_i^{\mathrm{DG}}$, are set as 2 MW and 1.5 MW, respectively. Fig. 4 illustrates the two test systems with the dispersed layout, and the values of $\overline{G}_i^{\mathrm{DG}}$ are set as 0.6 MW in both systems.

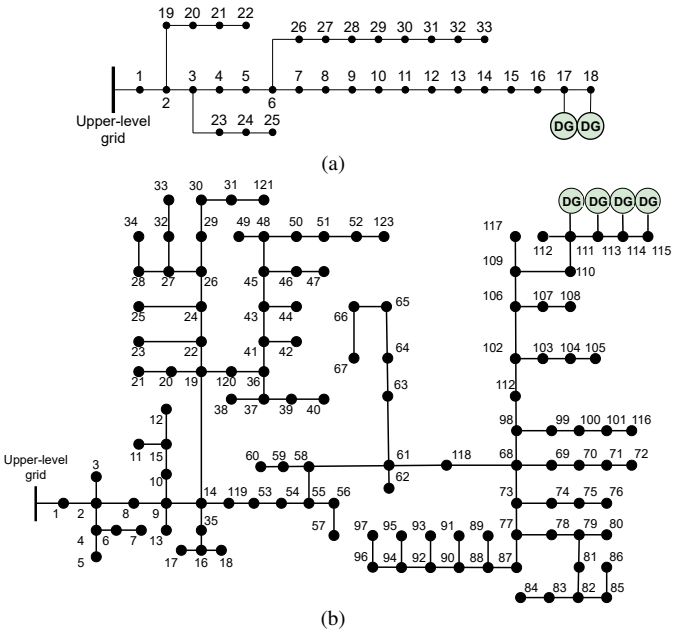


Fig. 3. Structures of the (a) 33-bus and (b) 123-bus test systems with the concentrated layout, i.e., all DG units are located close to each other. The nominal maximum available output of each DG unit in (a) is 2 MW, while it is 1.5 MW in (b).

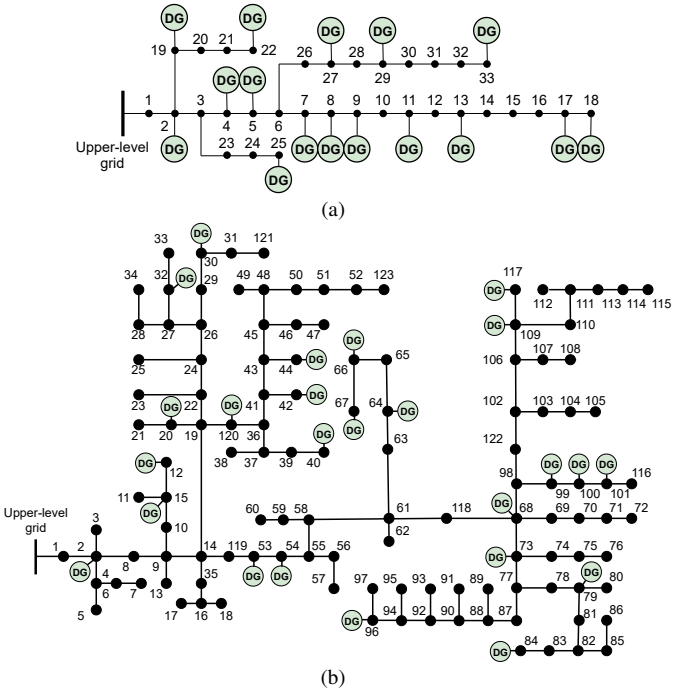
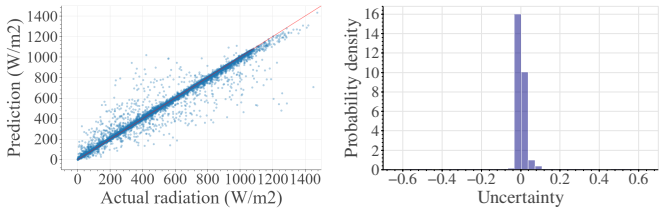


Fig. 4. Structures of the (a) 33-bus and (b) 123-bus test systems with the dispersed layout. Here, the 33-bus system has 16 DG units, while the 123-bus system contains 25 DG units. In both two systems, the nominal available output of each DG unit is set as 0.6 MW.

We conduct power flow simulations based on Pandapower, a power system simulation toolbox in Python environment [53], to generate the historical data. In Pandapower, the power flow

[2]Suppose we employ a CNN to recognize dog images. No matter where the dog is in this image, a CNN can successfully recognize that there is a dog. This characteristic is called "translation invariance". However, nodal power injection data may not be invariant to translation. When a specific power injection appears on different buses, the corresponding OPF solutions should also be different, but a CNN may give the same result [43].

This article has been accepted for publication in IEEE Transactions on Sustainable Energy. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2022.3223764



(a) (b)

Fig. 5. (a) Comparison of the actual and predicted solar radiation and (b) the probability density of uncertainty $\boldsymbol{\omega}$ calculated by forecasting errors.

calculation is based on the full AC power flow model. During the simulations, we first randomly generate 10,000 pairs of nominal bus power injections and uncertain levels of DG, i.e., $(\boldsymbol{x}, \boldsymbol{\omega})$. Here the nominal power injection $\boldsymbol{x}$ is generated by a uniform distribution between its minimum and maximum allowable values. Based on these pairs, we can calculate the actual power injections on each bus, and then the bus voltages $\boldsymbol{V}$ and branch currents $\boldsymbol{I}$ can be calculated by Pandapower. With $\boldsymbol{V}$ and $\boldsymbol{I}$, the power loss p^{loss} and maximum constraint violation $h(\boldsymbol{x}, \boldsymbol{\omega})$ can be obtained based on (7)-(9). Then, following **Algorithm** 1, we conduct the data augmentation to generate the training set for the quantile-MLP. The two parameters N_ω and K in **Algorithm** 1 are set as 1000 and 100, respectively.

To demonstrate the generalization ability of the proposed model, we use different distributions to simulate the samples of the uncertain level $\boldsymbol{\omega}$, as follows:

1) **Case 1**: The samples of $\boldsymbol{\omega}$ are simulated by a Gaussian distribution, i.e., $\boldsymbol{\omega} \sim \text{Gaussian}(0, 0.1)$;
2) **Case 2**: The samples of $\boldsymbol{\omega}$ are generated based on a Beta distributed uncertainty $\boldsymbol{\omega}^{'}$, i.e., $\boldsymbol{\omega} = \kappa^{\text{Beta}}(\boldsymbol{\omega}^{'} - \boldsymbol{\mu}^{\text{beta}})$, where $\boldsymbol{\omega}^{'} \sim \text{Beta}(2, 6)$;
3) **Case 3**: The samples of $\boldsymbol{\omega}$ are simulated based on a Weibull distributed uncertainty $\boldsymbol{\omega}^{'}$, i.e., $\boldsymbol{\omega} = \kappa^{\text{Weibull}}(\boldsymbol{\omega}^{'} - \boldsymbol{\mu}^{\text{Weibull}})$, where $\boldsymbol{\omega}^{'} \sim \text{Weibull}(1, 5)$.
4) **Case 4**: The samples of $\boldsymbol{\omega}$ are simulated based on real solar radiation data in Hawaii [54]. Fig. 5 illustrates the prediction results of the forecasting model and probability density of uncertainty $\boldsymbol{\omega}$.

The scaling factor $\kappa^{\text{Beta}}/\kappa^{\text{Weibull}}$ is designed to make the magnitudes of the generated $\boldsymbol{\omega}$ more realistic, while the constant $\boldsymbol{\mu}$ is set as the expectation of $\boldsymbol{\omega}^{'}$ to make the expectation of $\boldsymbol{\omega}$ keep at zero. All these samples have been uploaded to [55].

All numerical experiments are implemented on an Intel(R) 8700 3.20GHz CPU with 16 GB memory. The quantile-MLP and loss-MLP are established based on Pytorch. Problem **P3** is built by CVXPY and solved by GUROBI.

B. Benchmarks

To demonstrate the superiority of the proposed model, we introduce the following four model-based benchmarks:

1) **B1**: Linearized DistFlow model used in [7], [13] combined with the scenario approach;
2) **B2**: Second-order cone programming (SOCP) relaxation of AC OPF model used in [14], [15][3] combined with the scenario approach;
3) **B3**: Risk-neutral full AC OPF model, which is directly implemented in Pandapower.
4) **Baseline**: A line searching-based method that provides near optimal solutions, which is introduced in detail in the following paragraph.

Benchmarks **B1**-**B3** introduce approximations or relaxations, so their solutions may not be optimal. Thus, even if the proposed model shows better performance than **B1**-**B3**, we may be unable to justify its effectiveness. To address this issue, we further design a line searching method as our **Baseline**. In **Baseline**, we first initialize the decision variable $\boldsymbol{\lambda}$ with an all-one vector and regard $-\overline{\boldsymbol{G}}^{\text{DG}}$ as the searching direction to update $\boldsymbol{\lambda}$. The update stepsize α is set as 0.0025. Then, we combine the updated $\boldsymbol{\lambda}$ with different uncertainty sample $\{\boldsymbol{\omega}_n\}_{\forall n \in \mathcal{N}}$ to construct multiple pairs, i.e., $\{(\boldsymbol{\lambda}, \boldsymbol{\omega}_n)\}_{\forall n \in \mathcal{N}}$. Each pair is sent to Pandapower to calculate corresponding voltages and currents. By counting the number of $\boldsymbol{\omega}_n$ that makes the updated $\boldsymbol{\lambda}$ violate voltage and current limitations, we can check whether the updated $\boldsymbol{\lambda}$ can satisfy the joint chance constraint (5). Finally, the benchmark **Baseline** outputs the largest $\boldsymbol{\lambda}$ that can satisfy (5). Since **Baseline** needs to solve many OPF instances to check the feasibility of each updated $\boldsymbol{\lambda}$, it is very time-consuming. Nevertheless, it can achieve desirable optimality because the largest feasible $\boldsymbol{\lambda}$ can be found. As a result, we can justify the accuracy of the proposed model by comparing it to **Baseline**.

C. Case study with the concentrated layout

This section conducts the case study with the concentrated layout, i.e., all DG units are close to each other. Specifically, we compare the average utilization rates of DG, violation probabilities of the joint chance constraint (5), energy purchasing from the upper-level grid, and solving times of different models in both the 33-bus and 123-bus systems shown in Fig. 3. Appendix D introduces the calculation methods of the average utilization rates and violation probabilities in detail.

1) 33-bus test system: We first evaluate the prediction accuracy of the quantile-MLP and loss-MLP by comparing the true and predicted values of the quantile of the maximum constraint violation, i.e., $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega}))$, and expectation of power loss, i.e., $\mathbb{E}^{\boldsymbol{\omega}}(p^{\text{loss}})$. The neuron numbers of the quantile-MLP are set as (25, 25, 25), i.e., three hidden layers with 25 neurons in each layer, while the neuron numbers of the loss-MLP are set as (10, 10, 10). Note that the historical operational dataset only contains realizations but does not contain quantile or expectation samples. Nevertheless, with the realizations of h and p^{loss}, we can construct more samples with the same $\boldsymbol{x}$ and different $\boldsymbol{\omega}$, i.e., $\{(\boldsymbol{x}_i, \boldsymbol{\omega}_n, h_n, p^{\text{loss}}_n)\}_{\forall n \in \mathcal{N}}$. Then, the actual quantile and expectation can be calculated based on the sample set $\{(\boldsymbol{x}_i, \boldsymbol{\omega}_n, h_n, p^{\text{loss}}_n)\}_{\forall n \in \mathcal{N}}$. During the training of neural networks, 70% of historical samples are randomly selected as the training set, while the rest 30% of samples are used as the testing set. Fig. 6 compares the actual and predicted values of the 80% quantile and expected power loss on the testing set in Cases 1-3. Obviously, most samples are very close to the red lines, which demonstrates the accuracy of MLPs. However, in Figs. 6 (a), (c), and (e), the prediction

[3]Note that the SOCP relaxation is equivalent to the semi-definite relaxation of OPF in radial distribution networks [17].

This article has been accepted for publication in IEEE Transactions on Sustainable Energy. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2022.3223764

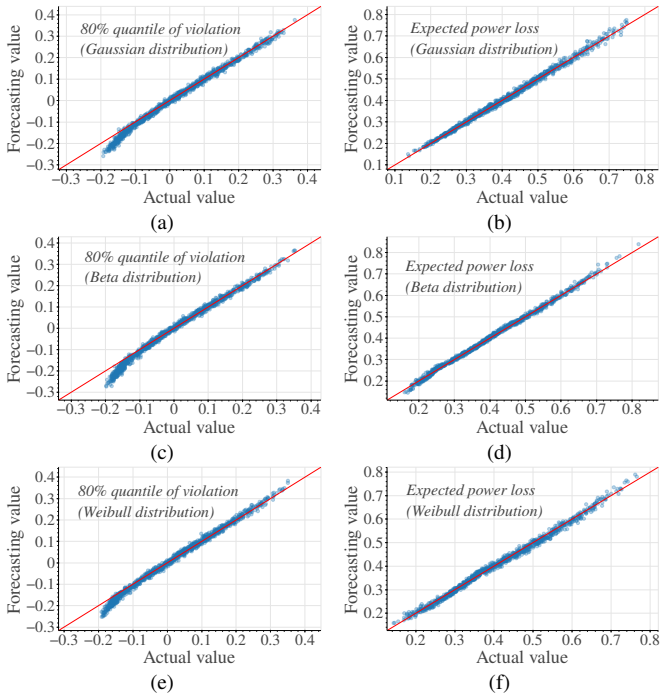


Fig. 6. Comparisons of actual and predicted values of (a) 80% quantile of constraint violation in Case 1 (Gaussian uncertainties), (b) expected power loss in Case 1, (c) 80% quantile of constraint violation in Case 2, (d) expected power loss in Case 2 (Beta uncertainties), (d) 80% quantile of constraint violation in Case 3 (Weibull uncertainties), (e) expected power loss in Case 3 . The red line represents the position where predictions equal actual values.

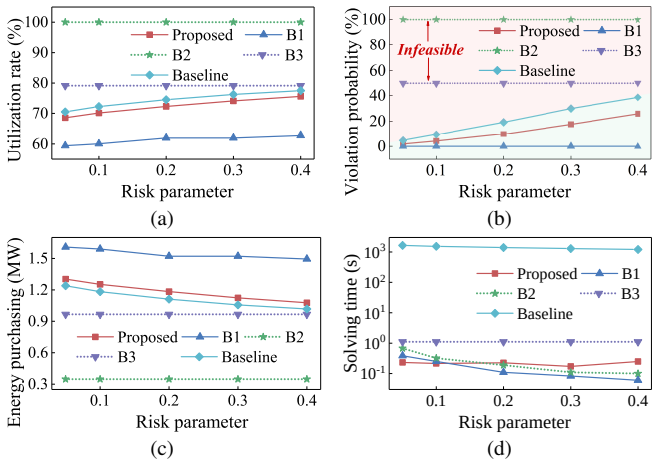


Fig. 7. Results of (a) average utilization rates of DG, (b) maximum violation probabilities of the joint chance constraint (5), (c) energy purchasing from the upper-level grid, and (d) solving times in Case 1 (the uncertainty $\boldsymbol{\omega}$ follows a Gaussian distribution). Dot lines represent infeasible results. In (b), the green and red areas denote the feasible and infeasible regions, respectively.

errors of the quantile-MLP become significant when the actual quantile $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}$ is around -0.2. Nevertheless, if the quantile is much smaller than zero, e.g., $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon} \leq -0.1$, the chance constraint (5) is inactive. In this case, the prediction error is almost always negative, i.e., the MLP's prediction is smaller than the corresponding actual value, so this prediction is also negative. As a result, the chance constraint (5) is still inactive in the surrogate model and does not affect the optimal solution. Hence, although there are significant prediction errors when $\mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}$ is far below zero, the proposed model can still achieve desirable accuracy.

After evaluating the accuracy of MLPs, we compares the performances of different models in **Cases 1-4** with various uncertainties. Fig. 7 compares the results of different models in **Case 1**. Among all models, the linearized DistFlow model, **B1**, derives the most conservative results, while its violation probability and utilization rate of DG are the lowest. In order to linearize the DistFlow, **B1** not only ignores the nonlinear constraint, i.e., the last equation of (4), but also removes all the terms related to current I_{ij}. Reference [56] pointed out that **B1** overestimated bus voltages. Considering that promoting the integration of DG will increase bus voltages, less DG can be utilized in **B1** because its overestimated bus voltages are still required to be smaller than the corresponding upper bound. Nevertheless, its solution is always feasible for the joint chance constraint (5). Conversely, the SOCP relaxation **B2** shows very poor feasibility (its violation probability approaches 100%, which is much higher than the risk parameter), although it achieves the highest utilization rate of DG and lowest energy purchasing. This is because reverse power flows occur in the system, which makes its SOCP relaxation inexact. The risk-neutral model **B3** also fails to meet the joint chance constraint (5) since it directly ignores the impacts of uncertainties. Thus, both **B2** and **B3** are not applicable to distribution systems with high DG penetration. Since **Baseline** exhaustively searches the domain of decision variables, it achieves the best optimality with guaranteed feasibility among all models. However, it needs to solve a huge number of OPF instances, so its computational efficiency is very poor. The average utilization rate of DG given by the proposed approach is only slightly lower than that of the **Baseline**, while it is much higher than **B1**. Meanwhile, the proposed model can always ensure the feasibility of solutions. Although the proposed model introduces some binary variables for reformulating MLPs, its computational efficiency is desirable. For example, the solving time of the proposed approach keeps around 0.3s and is three to four orders of magnitude smaller than that of the **Baseline**. As mentioned in III-C, the binary variables in the proposed model correspond to the activation states of neurons in MLPs. Once critical constraints are introduced to restrict the input $\boldsymbol{x}$, many neurons become stably active or inactive. Since the activation states of these neurons are fixed, the corresponding binary variables are also constant. Table I shows the numbers of stably active and inactive neurons in **Cases 1-3**. Over half of neurons have stable activation states, so the corresponding binary variables can be regarded as known parameters. As a result, the proposed model can still achieve excellent computational performance. Moreover, unlike **B1**-**B3** and **Baseline**, the proposed model only needs historical data to train MLPs but does not require the network parameters to build power flow models. These results demonstrate the desirable optimality, feasibility, and computational efficiency of the proposed model.

Fig. 8 compares the results of different models in **Case 2**, in which the uncertainty $\boldsymbol{\omega}$ follows Beta distribution. The results are very similar to those in **Case 1**: the energy-efficiency of **B1** is undesirable, while **B2** and **B3** can not guarantee the feasibility of solutions. In contrast, the proposed model can achieve desirable optimality and feasibility simultaneously with no need for network parameters. The solving time of the

This article has been accepted for publication in IEEE Transactions on Sustainable Energy. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2022.3223764

TABLE I
NUMBERS OF STABLY ACTIVE AND INACTIVE IN CASES 1-3

Neuron numbers		Total	Stably active	Stably inactive
Case 1	Quantile-MLP	75	24	17
	Loss-MLP	30	9	9
Case 2	Quantile-MLP	75	21	17
	Loss-MLP	30	10	12
Case 3	Quantile-MLP	75	22	16
	Loss-MLP	30	8	11

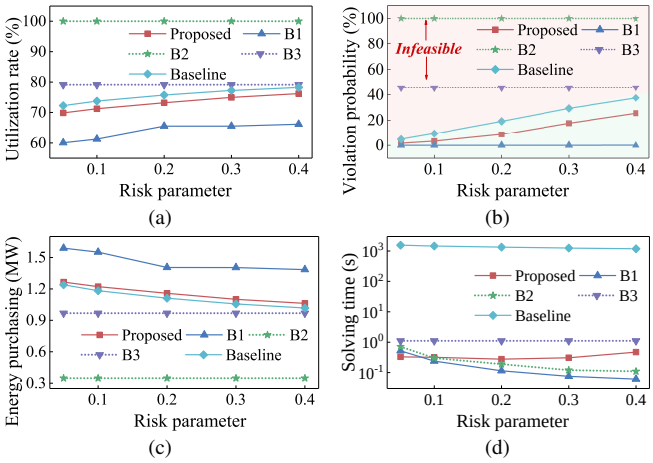


Fig. 8. Results of (a) average utilization rates of DG, (b) maximum violation probabilities, (c) energy purchasing, and (d) solving times in Case 2 (the uncertainty ω follows a Beta distribution).

proposed model still keeps around 0.3s, which is much smaller than that of **Baseline**.

Fig. 9 illustrates the results in **Case 3**, in which the uncertainty ω follows Weibull distribution. Similarly, the proposed model achieves better energy-efficiency compared to **B1**, and outperforms **B2** and **B3** on feasibility. Moreover, it shows much better computational efficiency than **Baseline**.

Fig. 10 shows the results of different models in **Case 4**. As mentioned in Section IV-E, the samples of uncertainty ω are constructed based on a real solar radiation dataset. Similar to Cases 1-3, the proposed model achieves better optimality than **B1**. Unlike **B2** and **B3**, which fail to satisfy the joint chance constraint (5), the proposed model can also always guarantee

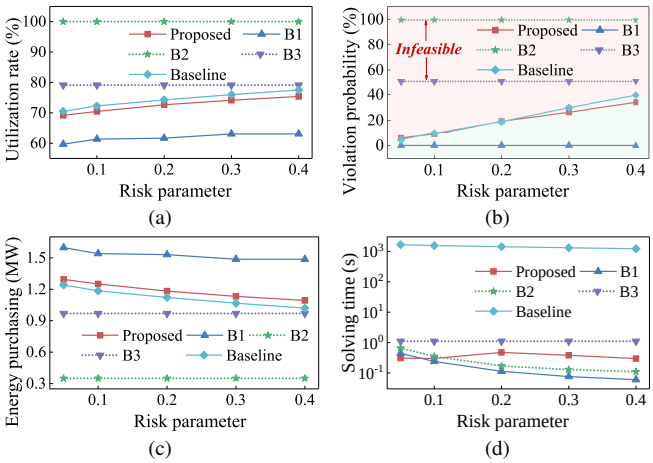


Fig. 9. Results of (a) average utilization rates of DG, (b) maximum violation probabilities, (c) energy purchasing, and (d) solving times in Case 3 (the uncertainty ω follows a Weibull distribution).

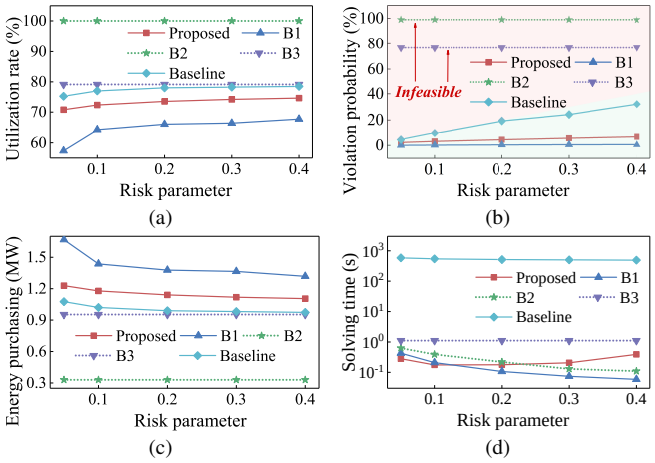


Fig. 10. Results of (a) average utilization rates of DG, (b) maximum violation probabilities, (c) energy purchasing, and (d) solving times in Case 4 (the samples of ω are constructed based on real data [54]).

feasibility. Moreover, although its optimality is slightly worse than the **Baseline**, the computational efficiency is much higher.

2) 123-bus test system: We further conduct a case study based on the 123-bus system to demonstrate the benefits of the proposed model. The neuron numbers of the quantile- and loss-MLPs are set as (30, 30, 30) and (10, 10, 10), respectively. The uncertainties are the same as those in **Case 3** (Weibull uncertainties). The results in Section IV-C show that **B1** is overly conservative. However, this conservativeness may be contributed by either the power flow approximation (linearized DistFlow) or the joint chance constraint reformulation (scenario approach). To highlight that the linearized DistFlow model introduces unnecessary conservativeness, we modify the benchmark **B1** as **B1-SAA**, in which the joint chance constraint is handled by sample average approximation (SAA). SAA is a promising way to handle joint chance constraints with excellent optimality, but it is also time-consuming because numerous binary variables need to be involved [6].

The results on the 123-bus system are shown in Fig. 11. Similar to the results on the 33-bus system, the SOCP relaxation **B2** can not always guarantee feasibility due to the existence of reverse power flows. The risk-neutral model **B3** also fails to satisfy the joint chance constraint because it ignores the impacts of uncertainties. For the linearized DistFlow **B1-SAA**, its energy purchasing amount is much higher than that of the proposed model. This result indicates that the linearized DistFlow introduces significant conservativeness. Moreover, since SAA needs to introduce a large number of binary variables, its computational efficiency is much worse than that of the proposed one. For example, at $\epsilon = 0.2$, the solving time of **B1-SAA** reaches 82.63s, while it is only 0.15s in the proposed model. These results further confirm the great performance of the proposed model.

D. Case study with the dispersed layout

1) 33-bus test system: This case is built upon the the IEEE 33-bus system with 16 dispersed DG units, as illustrated in Fig. 4(a). The uncertainty ω follows a Weibull distribution, which is the same as that in **Case 3**. The results of the

This article has been accepted for publication in IEEE Transactions on Sustainable Energy. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2022.3223764

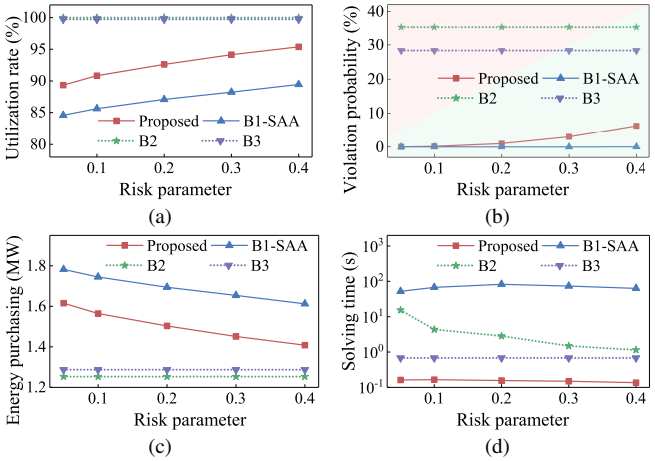


Fig. 11. Results of (a) average utilization rates of DG, (b) maximum violation probabilities, (c) energy purchasing, and (d) solving times in the case based on the IEEE 123-bus system. Here ω follows a Weibull distribution.

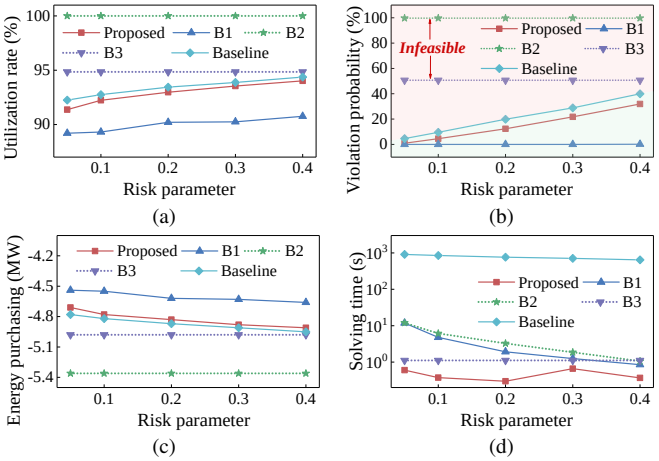


Fig. 12. Results of (a) average utilization rates of DG, (b) maximum violation probabilities of the joint chance constraint in JCC-OPF, (c) energy purchasing from the upper-level grid, and (d) solving times in the 33-bus test case with 16 DG units. The uncertainty ω follows a Weibull distribution.

new case study are shown in Fig. 12. Similar to **Cases 1-4**, the linearized DistFlow model **B1** obtains the lowest average utilization rate of DG and highest energy purchasing because it overestimates bus voltages. The SOCP relaxation **B2** shows very poor feasibility because it is inexact when there are reverse power flows. The risk-neutral model **B3** also fails to meet constraints since it directly ignores uncertainties. The searching method **Baseline** achieves the best optimality but is computationally expensive. The proposed model can always ensure the feasibility of solutions with desirable computational efficiency. Moreover, its optimality is quite close to that of **Baseline** and much better than that of **B1**.

2) 123-bus test system: This test case is based on the IEEE 123-bus system with 25 dispersed DG units. The system structure is illustrated in Fig. 4(b). The uncertainty ω is assumed to follow an unknown Weibull distribution, which is the same as that in the 123-bus test case in Section IV-C. We also introduce **B1**-**B3** and **Baseline** as our benchmarks. Fig. 13 shows the results of this test case. Similarly, since the total active power demand is only 4.885MW and much smaller than the total available renewable generation, the distribution network always sells extra DGs' outputs in all cases. Benchmarks **B1** and

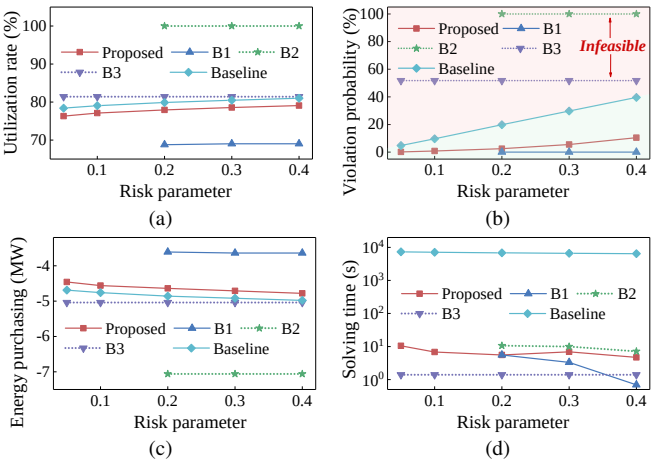


Fig. 13. Results of (a) average utilization rates of DG, (b) maximum violation probabilities of the joint chance constraint in JCC-OPF, (c) energy purchasing from the upper-level grid, and (d) solving times in the 123-bus test case with 25 DG units. The uncertainty ω follows a Weibull distribution.

B2 can be successfully executed only when the risk parameter ϵ is high (i.e., $\epsilon \geq 0.2$), while the out-of-memory issue occurs in the rest cases. Both **B1** and **B2** are based on the scenario approach, so their decisions should satisfy all constraints in a specific number of scenarios. This scenario number positively correlates with the uncertainty dimension but is inversely proportional to the risk parameter [14], [15]. Since this system contains many DG units, its scenario number is relatively high, especially when the risk parameter is small. Moreover, both **B1** and **B2** need to introduce multiple constraints for every scenario. As a result, their memory usages become enormous and may exceed the test platform with 16 GB memory. Besides the unacceptable memory consumption, **B1** also performs undesirable optimality because it overestimates bus voltages. Benchmark **B2** fails to satisfy the original joint chance constraint since the existence of reverse power flows makes the SOCP relaxation inexact. Benchmark **B3** also shows poor feasibility because it directly ignores the uncertainty's impacts. The searching method **Baseline** achieves the best optimality performance with guaranteed feasibility. However, its computational efficiency is significantly worse than those of others. Unlike **B2** and **B3**, the proposed model can consistently achieve desirable optimality and feasibility. Its solving time also maintains a low level and is much better than that of **Baseline**. In addition, the proposed model is fully data-driven and does not require network parameters. These results further confirm the effectiveness of the proposed model.

In summary, the above cases not only illustrate that the proposed model can achieve desirable performance without network parameters but also demonstrate its excellent generalization for arbitrary uncertainties. Moreover, the proposed model can perform well with both the concentrated and dispersed DG layouts.

E. Sensitivity analysis

In this section, we investigate the effects of MLPs with various structures (i.e., different numbers of neurons in MLPs) on the performance of the proposed surrogate model. The simulations are based on the IEEE 33-bus system. The hidden

This article has been accepted for publication in IEEE Transactions on Sustainable Energy. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2022.3223764

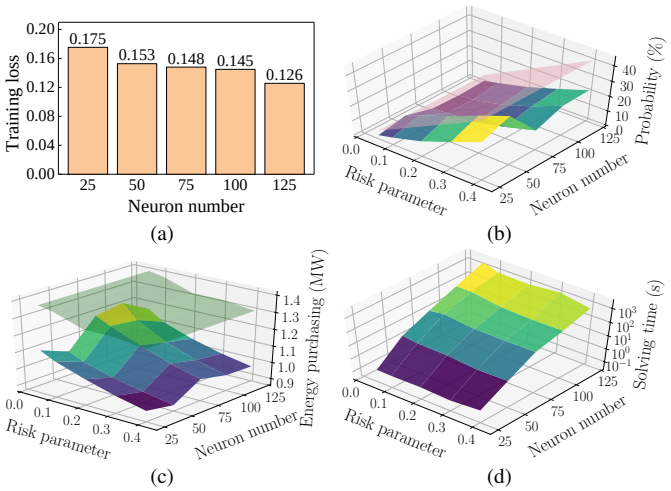


Fig. 14. Results of (a) loss function of the quantile-MLP, i.e., Eq. (14), (b) violation probability, (c) energy purchasing, and (d) solving times under different neuron numbers. In (b), the red surface represents the maximum allowable violation probability, i.e, the given risk parameter. In (c), the blue surface on the top refers to the energy purchasing of **B1**.

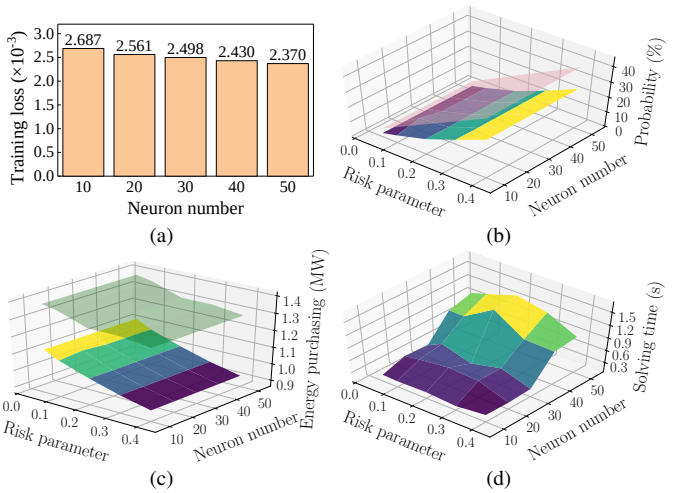


Fig. 15. Results of (a) loss function of the loss-MLP, (b) violation probability, (c) energy purchasing, and (d) solving times with different neuron numbers in the loss-MLP.

layer numbers of MLPs are fixed at three, and the used samples of $\boldsymbol{\omega}$ are the same as those in **Case 1**.

1) Neuron number of quantile-MLP: The results of the proposed model with different neuron numbers in the quantile-MLP are illustrated in Fig. 14, where "neuron number" refers to the neuron number in each hidden layer. The structure of the loss-MLP is fixed as (10, 10, 10). With the growth of the neuron number, the approximation ability of the quantile-MLP becomes stronger. Thus, the prediction loss decreases, as shown in Fig. 14(a). Since we use an inner approximation (33) to replace the original joint chance constraint (5) in the calibration step, the maximum violation probabilities are always lower than the risk parameter, i.e., the red surface in Fig. 14(b). With the growth of the neuron number, the prediction error of the quantile-MLP can be either negative or positive. As a result, both the violation probability and energy purchasing are not monotonous with respect to the neuron number. Nevertheless, the energy purchasing of the proposed model is always lower than that of **B1**, i.e., the green surface in Fig. 14(c). The solving time grows rapidly with the increase of the neuron number, as illustrated in Fig. 14(d). According to (29), the integer variable number introduced by reformulating the quantile-MLP is equal to the neuron number. Therefore, a larger neuron number leads to a higher computational burden. Nevertheless, with a small neuron number, the proposed model can already achieve desirable optimality and feasibility simultaneously in a short time, e.g., the solving time is around 0.3s when the neuron number is set as 25.

2) Neuron number of loss-MLP: We further investigate the effects of the loss-MLP's neuron number on the proposed model's performance, and the results are summarized in Fig. 15. The neuron number of the quantile-MLP is fixed as (25, 25, 25). Similarly, increasing the neuron number can reduce the training loss of the loss-MLP because this can enhance the prediction accuracy of the loss-MLP, as shown in Fig. 15(a). However, the power loss is usually much smaller than the summation of power demands. Therefore, even if we change the neuron number, the optimality and feasibility of the proposed model's solutions are nearly constant. Nevertheless, the violation probability is always lower than the required values, i.e., the red surface in Fig. 15(b), and the energy-efficiency is always better than that of **B1**, i.e., the green surface in Fig. 15(c). According to (29), the number of the auxiliary binary variables introduced by reformulating the loss-MLP equals its neuron number. Thus, a larger neuron number results in a higher computational burden, as shown in Fig. 15(d). Nevertheless, a small number of neurons is already enough for the proposed model because excellent optimality and feasibility can be accomplished.

V. Conclusions

This paper proposes a deep-quantile-regression-based surrogate model for the JCC-OPF problem. In the proposed model, two MLPs are trained to predict the $1-\epsilon$ quantile of the maximum constraint violation and expected power loss, respectively. By reformulating the forward propagation of the two MLPs into mixed-integer linear constraints, the JCC-OPF problem can be replicated. Two pre-processing steps, i.e., data augmentation and calibration, are further designed to enhance the performance of the proposed model. The data augmentation step trains an XGBoost-based regressor to generate more training samples so that the accuracy of the quantile regression can be improved. The calibration step designs a positive parameter to calibrate the deep quantile regression to improve the feasibility of solutions. Simulation results based on the IEEE 33- and 123-bus distribution systems confirm that the proposed model can successfully replicate the JCC-OPF problem without the network parameters. Moreover, its optimality is better than the widely used linearized DistFlow model under arbitrary uncertainties, while its feasibility is also much better than the SOCP relaxation of AC OPF.

Since the testing cases from the real world may also help us to improve the performance of the proposed surrogate model, we envision our future work to test it on a real system so that its benefits can be further validated. Meanwhile, since interpretability is important for power system applications, we wish to extend the proposed surrogate model based on

explainable learning techniques so that it can achieve desirable interpretability in the future.

APPENDIX A

*Proof of **Proposition 1***: The term on the right-hand side of (15) is equal to

$$\begin{aligned}\mathbb{E}(\text{Loss}^{\text{QR}}) = -\epsilon \int_{-\infty}^{\hat{\mathcal{Q}}^{1-\epsilon}} (h - \hat{\mathcal{Q}}^{1-\epsilon}) dF_H(h) \\ +(1-\epsilon) \int_{\hat{\mathcal{Q}}^{1-\epsilon}}^{\infty} (h - \hat{\mathcal{Q}}^{1-\epsilon}) dF_H(h),\end{aligned} \tag{36}$$

where $F_H(\cdot)$ denotes the cumulative distribution function of $h(\boldsymbol{x}, \boldsymbol{\omega})$ at $\boldsymbol{x}$ under the uncertainty $\boldsymbol{\omega}$. At the optimal solution that minimizes the expectation (36), the derivative of the expected loss should be zero:

$$\left. \frac{\partial \mathbb{E}(\text{Loss}^{\text{QR}})}{\partial \hat{\mathcal{Q}}^{1-\epsilon}} \right|_y = 0, \tag{37}$$

where y is the optimal solution of $\hat{\mathcal{Q}}^{1-\epsilon}$. Then, by substituting (36), Eq. (37) can be converted into the following form based on the Leibniz integral rule:

$$\epsilon \int_{-\infty}^{y} dF_H(h) - (1-\epsilon) \int_{y}^{\infty} dF_H(h) = 0. \tag{38}$$

By substituting $F_H(-\infty) = 0$ and $F_H(\infty) = 1$, Eq. (38) can be further reformulated as:

$$F_H(y) = 1 - \epsilon \Leftrightarrow y = \mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega})). \tag{39}$$

APPENDIX B

*Proof of **Proposition 2***: Based on (23), the expectation of $\text{Loss}^{\text{loss}}$ can be expressed as:

$$\begin{aligned}&\mathbb{E}(\text{Loss}^{\text{loss}}) \\ &= \mathbb{E}\left((p^{\text{loss}} - \hat{p}^{\text{loss}}(\boldsymbol{x}))^2\right) \\ &= \mathbb{E}\left((p^{\text{loss}} - \mathbb{E}^{\boldsymbol{\omega}}(p^{\text{loss}}))^2\right) + \left(\mathbb{E}^{\boldsymbol{\omega}}(p^{\text{loss}}) - \hat{p}^{\text{loss}}(\boldsymbol{x})\right)^2 \\ &= \text{Var}(p^{\text{loss}}) + \left(\mathbb{E}^{\boldsymbol{\omega}}(p^{\text{loss}}) - \hat{p}^{\text{loss}}(\boldsymbol{x})\right)^2,\end{aligned} \tag{40}$$

where $\text{Var}(p^{\text{loss}})$ is the variance of p^{loss}. By regarding $\mathbb{E}(\text{Loss}^{\text{loss}})$ as a function of $\hat{p}^{\text{loss}}(\boldsymbol{x})$, the minimum value of $\mathbb{E}(\text{Loss}^{\text{loss}})$ occurs at $\hat{p}^{\text{loss}}(\boldsymbol{x}) = \mathbb{E}^{\boldsymbol{\omega}}(p^{\text{loss}})$ according to (40).

APPENDIX C

*Proof of **Proposition 3***. The value of h can be expressed as the simulator's forecast $\hat{h}$ plus a prediction error δ:

$$h(\boldsymbol{x}, \boldsymbol{\omega}) = \hat{h}(\boldsymbol{x}, \boldsymbol{\omega}) + \delta. \tag{41}$$

If we treat δ as an uncertain parameter, then the empirical prediction errors $\{\delta_n\}_{\forall n \in \mathcal{N}}$ defined in (31) can be regarded as the randomly drawn samples of δ. Reference [57] pointed out that if the sample number $|\mathcal{N}|$ satisfies (34), the following chance constraint holds with confidence $1-\beta$:

$$\mathbb{P}^{\delta}(\rho \geq \delta) = \mathbb{P}^{\delta}\left(\hat{h}(\boldsymbol{x}, \boldsymbol{\omega}) + \rho \geq h(\boldsymbol{x}, \boldsymbol{\omega})\right) \geq 1 - \alpha, \tag{42}$$

where "=" holds because of (41). When both $\boldsymbol{x}$ and $\boldsymbol{\omega}$ are fixed, Eq. (31) only contains one uncertain parameter δ. Thus, here we have $n^{\delta} = 1$. Suppose we randomly draw one $\boldsymbol{x}$ and multiple $\boldsymbol{\omega}$. By combining this single $\boldsymbol{x}$ with different $\boldsymbol{\omega}$, we can construct many data pairs, i.e., $\{(\boldsymbol{x}, \boldsymbol{\omega}_n)\}_{\forall n \in \mathcal{N}^{\boldsymbol{\omega}}}$. These data pairs correspond to multiple realizations of $\hat{h}$ and h. We collect these realizations in two different sets, as follows:

$$\begin{cases}\hat{\mathcal{H}} = \{\hat{h}_{(n)} = \hat{h}(\boldsymbol{x}, \boldsymbol{\omega}_{(n)})\}_{\forall n \in \mathcal{N}^{\boldsymbol{\omega}}}, \\ \mathcal{H} = \{h_{(n)} = h(\boldsymbol{x}, \boldsymbol{\omega}_{(n)})\}_{\forall n \in \mathcal{N}^{\boldsymbol{\omega}}},\end{cases} \tag{43}$$

where $\forall n \in \mathcal{N}^{\boldsymbol{\omega}}$ is the index of realizations. Without loss of generality, we assume $\hat{h}_{(1)} \leq \hat{h}_{(2)} \leq \cdots \leq \hat{h}_{(|\mathcal{N}^{\boldsymbol{\omega}}|)}$ in $\hat{\mathcal{H}}$. Meanwhile, we can also rearrange the elements of $\mathcal{H}$ based on the order of $\omega_{(n)}$ in $\hat{\mathcal{H}}$. In other words, the n-th element of $\mathcal{H}$, i.e., $h_{(n)}$, has the same $\omega_{(n)}$ as $\hat{h}_{(n)}$ in $\hat{\mathcal{H}}$. Thus, the elements in $\mathcal{H}$ may not be monotonically increasing. Now, by denoting $N^{\mathcal{Q}} = \lceil |\mathcal{N}^{\boldsymbol{\omega}}| \cdot (1-\epsilon) \rceil$, we can obtain the $1-\epsilon$ quantile of $\hat{h}$ by finding the $N^{\mathcal{Q}}$-th element of set $\hat{\mathcal{H}}$, as follows:

$$\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x}) = \hat{h}_{(N^{\mathcal{Q}})}, \tag{44}$$

where $\lceil \cdot \rceil$ is the ceiling function. Since the elements in $\hat{\mathcal{H}}$ are ordered by their values, we have:

$$\hat{h}_{(N^{\mathcal{Q}})} \geq \hat{h}_{(n)}, \quad \forall n \in [1, 2, \cdots, N^{\mathcal{Q}}]. \tag{45}$$

Meanwhile, the maximum element among the first $N^{\mathcal{Q}}$ entries of $\mathcal{H}$ is recorded as:

$$h_{(n^*)} = \max_{\forall n \in [1,2,\cdots,N^{\mathcal{Q}}]} \{h_{(n)}\}. \tag{46}$$

Obviously, we have $1 \leq (n^*) \leq N^{\mathcal{Q}}$. According to (46), the value of $h_{(n^*)}$ is at least larger than $N^{\mathcal{Q}}$ elements of $\mathcal{H}$. Since the $1-\epsilon$ quantile of h can be regarded as the $N^{\mathcal{Q}}$-th largest element in $\mathcal{H}$, we must have:

$$h_{(n^*)} \geq \mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega})). \tag{47}$$

Based on the above discussion, we have:

$$\begin{aligned}&\mathbb{P}^{\boldsymbol{\omega}}\left(\hat{\mathcal{Q}}_{1-\epsilon}(\boldsymbol{x}) + \rho \geq \mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega}))\right) \\ &= \mathbb{P}^{\boldsymbol{\omega}}\left(\hat{h}_{(N^{\mathcal{Q}})} + \rho \geq \mathcal{Q}^{\boldsymbol{\omega}}_{1-\epsilon}(h(\boldsymbol{x}, \boldsymbol{\omega}))\right) \longleftarrow \text{Substitute (44)} \\ &\geq \mathbb{P}^{\boldsymbol{\omega}}\left(\hat{h}_{(N^{\mathcal{Q}})} + \rho \geq h_{(n^*)}\right) \longleftarrow \text{Substitute (47)} \\ &\geq \mathbb{P}^{\boldsymbol{\omega}}\left(\hat{h}_{(n^*)} + \rho \geq h_{(n^*)}\right) \longleftarrow \text{Substitute (45)} \\ &\geq 1 - \alpha \longleftarrow \text{Substitute (42).}\end{aligned} \tag{48}$$

This proves **Proposition 3**.

APPENDIX D

1) Average utilization rates: The utilization rate of the total available DG, i.e., λ^{Avg}, is defined as follows:

$$\lambda^{\text{Avg}} = \frac{\boldsymbol{\lambda}^{\intercal} \boldsymbol{G}^{\text{DG}}}{\mathbf{1}^{\intercal} \boldsymbol{G}^{\text{DG}}}, \tag{49}$$

where the numerator and denominator represent the total used DG and total available DG, respectively. Since the available DG, i.e., $\boldsymbol{G}^{\text{DG}}$ is uncertain according to (2), the value of λ^{Avg} is also uncertain. Thus, it is hard to evaluate the DG consumption level based on this uncertain λ^{Avg}. To address this issue, we use the expectation of λ^{Avg} to represent the DG consumption

level. This expectation is the average utilization rate of DG in Figs. 7-15. It can be calculated based on the historical samples of uncertainties:

$$\mathbb{E}(\lambda^{\text{Avg}}) = \frac{1}{|\mathcal{N}|} \sum_{\forall n \in \mathcal{N}} \frac{\boldsymbol{\lambda}^{\intercal} \widehat{\boldsymbol{G}}_n^{\text{DG}}}{\mathbf{1}^{\intercal} \widehat{\boldsymbol{G}}_n^{\text{DG}}}, \tag{50}$$

where $\widehat{\boldsymbol{G}}_n^{\text{DG}}$ is defined as:

$$\widehat{\boldsymbol{G}}_n^{\text{DG}} = \overline{\boldsymbol{G}}^{\text{DG}} * (\mathbf{1} + \boldsymbol{\omega}_n), \quad \forall n \in \mathcal{N}. \tag{51}$$

In our case study, once the JCC-OPF is solved, we can obtain the optimal $\boldsymbol{\lambda}$. By substituting this optimal $\boldsymbol{\lambda}$ to (50), the average utilization rate of DG, i.e., $\mathbb{E}(\lambda^{\text{Avg}})$ can be calculated.

2) Violation probabilities: The violation probabilities in our case study are calculated based on Monte Carlo simulations. Specifically, once the JCC-OPF is solved, we can get the optimal decision $\boldsymbol{x}$. Based on the optimal decision and historical samples of uncertainty $\boldsymbol{\omega}$, we can construct multiple pairs, i.e., $\{(\boldsymbol{x}, \boldsymbol{\omega}_n)\}_{\forall n \in \mathcal{N}}$. By giving these pairs to Pandapower [53], we can calculate the corresponding realizations of bus voltages and branch currents and judge whether the decision $\boldsymbol{x}$ violates constraints under different $\boldsymbol{\omega}_n$. Then, the violation probability (recorded as $\mathbb{P}^{\text{Vio}}$) can be calculated by:

$$\mathbb{P}^{\text{Vio}} = \frac{N^{\text{Vio}}(\boldsymbol{x})}{|\mathcal{N}|} \times 100\%, \tag{52}$$

where $N^{\text{Vio}}(\boldsymbol{x})$ is the number of $\boldsymbol{\omega}_n$ that makes the decision $\boldsymbol{x}$ violate constraints; $|\mathcal{N}|$ is the total number of $\boldsymbol{\omega}_n$.

REFERENCES

[1] H. Abdi, S. D. Beigvand, and M. La Scala, "A review of optimal power flow studies applied to smart grids and microgrids," *Renew. Sust. Energ. Rev.*, vol. 71, pp. 742–766, 2017.

[2] H. Hui, P. Siano, Y. Ding, P. Yu, Y. Song, H. Zhang, and N. Dai, "A transactive energy framework for inverter-based HVAC loads in a real-time local electricity market considering distributed energy resources," *IEEE Trans. Industr. Inform.*, early access, 2022.

[3] A. Zakaria, F. B. Ismail, M. H. Lipu, and M. A. Hannan, "Uncertainty models for stochastic optimization in renewable energy applications," *Renew. Energ.*, vol. 145, pp. 1543–1571, 2020.

[4] A. Lorca and X. A. Sun, "Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1702–1713, 2015.

[5] D. Phan and S. Ghosh, "Two-stage stochastic optimization for optimal power flow under renewable generation uncertainty," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 24, no. 1, pp. 1–22, 2014.

[6] X. Geng and L. Xie, "Data-driven decision making in power systems with probabilistic guarantees: Theory and applications of chance-constrained optimization," *Annu. Rev. Control*, vol. 47, pp. 341–363, 2019.

[7] A. Hassan, R. Mieth, M. Chertkov, D. Deka, and Y. Dvorkin, "Optimal load ensemble control in chance-constrained optimal power flow," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5186–5195, 2019.

[8] M. Lubin, Y. Dvorkin, and L. Roald, "Chance constraints for improving the security of ac optimal power flow," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 1908–1917, 2019.

[9] W. Xie and S. Ahmed, "Distributionally robust chance constrained optimal power flow with renewables: A conic reformulation," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1860–1867, 2018.

[10] A. Peña-Ordieres, D. K. Molzahn, L. A. Roald, and A. Wächter, "Dc optimal power flow with joint chance constraints," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 147–158, 2021.

[11] B. Odetayo, M. Kazemi, J. MacCormack, W. D. Rosehart, H. Zareipour, and A. R. Seifi, "A chance constrained programming approach to the integrated planning of electric power generation, natural gas network and storage," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6883–6893, 2018.

[12] K. Baker and A. Bernstein, "Joint chance constraints in ac optimal power flow: Improving bounds through learning," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6376–6385, 2019.

[13] P. Li, B. Jin, D. Wang, and B. Zhang, "Distribution system voltage control under uncertainties using tractable chance constraints," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 5208–5216, 2019.

[14] A. Venzke, L. Halilbasic, U. Markovic, G. Hug, and S. Chatzivasileiadis, "Convex relaxations of chance constrained ac optimal power flow," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2829–2841, 2018.

[15] A. Venzke and S. Chatzivasileiadis, "Convex relaxations of probabilistic ac optimal power flow for interconnected ac and hvdc grids," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 2706–2718, 2019.

[16] S. J. Pappu, N. Bhatt, R. Pasumarthy, and A. Rajeswaran, "Identifying topology of low voltage distribution networks based on smart meter data," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5113–5122, 2018.

[17] S. H. Low, "Convex relaxation of optimal power flow—part ii: Exactness," *IEEE Trans. Control. Netw. Syst.*, vol. 1, no. 2, pp. 177–189, 2014.

[18] G. Ruan, H. Zhong, G. Zhang, Y. He, X. Wang, and T. Pu, "Review of learning-assisted power system optimization," *CSEE J. Power Energy Syst.*, vol. 7, no. 2, pp. 221–231, 2021.

[19] W. Sun, M. Zamani, M. R. Hesamzadeh, and H.-T. Zhang, "Data-driven probabilistic optimal power flow with nonparametric bayesian modeling and inference," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1077–1090, 2020.

[20] G. Chen, H. Zhang, H. Hui, and Y. Song, "Chance-constrained regulation capacity offering for hvac systems under non-gaussian uncertainties with mixture-model-based convexification," *IEEE Trans. Smart Grid*, pp. 1–1, 2022.

[21] R. Hu and Q. Li, "Optimal operation of power systems with energy storage under uncertainty: A scenario-based method with strategic sampling," *IEEE Trans. Smart Grid*, vol. 13, no. 2, pp. 1249–1260, 2022.

[22] X. Pan, T. Zhao, M. Chen, and S. Zhang, "Deepopf: A deep neural network approach for security-constrained dc optimal power flow," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 1725–1735, 2021.

[23] W. Huang, X. Pan, M. Chen, and S. H. Low, "Deepopf-v: Solving ac-opf problems efficiently," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 800–803, 2022.

[24] A. Velloso and P. Van Hentenryck, "Combining deep learning and optimization for preventive security-constrained dc optimal power flow," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3618–3628, 2021.

[25] X. Pan, M. Chen, T. Zhao, and S. H. Low, "Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems," *arXiv preprint arXiv:2007.01002*, 2020.

[26] D. Owerko, F. Gama, and A. Ribeiro, "Optimal power flow using graph neural networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5930–5934, 2020.

[27] P. Pareek and H. D. Nguyen, "Gaussian process learning-based probabilistic optimal power flow," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 541–544, 2021.

[28] Z. Yan and Y. Xu, "Real-time optimal power flow: A lagrangian based deep reinforcement learning approach," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 3270–3273, 2020.

[29] Y. Zhou, B. Zhang, C. Xu, T. Lan, R. Diao, D. Shi, Z. Wang, and W.-J. Lee, "A data-driven method for fast ac optimal power flow solutions via deep reinforcement learning," *J. Mod. Power Syst. Clean Energy*, vol. 8, no. 6, pp. 1128–1139, 2020.

[30] Y. Zhang, X. Wang, J. Wang, and Y. Zhang, "Deep reinforcement learning based volt-var optimization in smart distribution systems," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 361–371, 2021.

[31] M. Al-Saffar and P. Musilek, "Reinforcement learning-based distributed bess management for mitigating overvoltage issues in systems with high pv penetration," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 2980–2994, 2020.

[32] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–7, 2020.

[33] A. Venzke, D. T. Viola, J. Mermet-Guyennet, G. S. Misyris, and S. Chatzivasileiadis, "Neural networks for encoding dynamic security-constrained optimal power flow to mixed-integer linear programs," *arXiv preprint arXiv:2003.07939*, 2020.

[34] G. Chen, H. Zhang, H. Hui, N. Dai, and Y. Song, "Scheduling thermostatically controlled loads to provide regulation capacity based on a

learning-based optimal power flow model," *IEEE Trans. Sustain. Energy*, vol. 12, no. 4, pp. 2459–2470, 2021.
[35] M. Baran and F. Wu, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Trans. Power Deliv.*, vol. 4, no. 1, pp. 735–743, 1989.
[36] L. Hao, D. Q. Naiman, and D. Q. Naiman, *Quantile regression*. No. 149, Sage, 2007.
[37] M. Fischetti and J. Jo, "Deep neural networks and mixed integer linear optimization," *Constraints*, vol. 23, no. 3, pp. 296–309, 2018.
[38] R. Anderson, J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma, "Strong mixed-integer programming formulations for trained neural networks," *Math Program*, vol. 183, no. 1, pp. 3–39, 2020.
[39] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.
[40] Z. Li, S. Jazebi, and F. de León, "Determination of the optimal switching frequency for distribution system reconfiguration," *IEEE Trans. Power Deliv.*, vol. 32, no. 4, pp. 2060–2069, 2017.
[41] Y. Jia, X. Bai, L. Zheng, Z. Weng, and Y. Li, "Convopf-dop: A data-driven method for solving ac-opf based on cnn considering different operation patterns," *IEEE Trans. Power Syst.*, pp. 1–1, 2022.
[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
[43] T. Wiatowski and H. Bölcskei, "A mathematical theory of deep convolutional neural networks for feature extraction," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1845–1866, 2018.
[44] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
[45] I. Genc, R. Diao, V. Vittal, S. Kolluri, and S. Mandal, "Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1611–1619, 2010.
[46] S. Zhang, D. Zhang, J. Qiao, X. Wang, and Z. Zhang, "Preventive control for power system transient security based on xgboost and dcopf with consideration of model interpretability," *CSEE J. Power Energy Syst.*, vol. 7, no. 2, pp. 279–294, 2021.
[47] K. Zhang, P. Xu, and J. Zhang, "Explainable ai in deep reinforcement learning models: A shap method applied in power system emergency control," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*, pp. 711–716, 2020.
[48] R. Machlev, M. Perl, J. Belikov, K. Y. Levy, and Y. Levron, "Measuring explainability and trustworthiness of power quality disturbances classifiers using xai—explainable artificial intelligence," *IEEE Trans. Industr. Inform.*, vol. 18, no. 8, pp. 5127–5137, 2022.
[49] Y. Lu, I. Murzakhanov, and S. Chatzivasileiadis, "Neural network interpretability for forecasting of aggregated renewable generation," in *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 282–288, 2021.
[50] F. Hedenus, N. Jakobsson, L. Reichenberg, and N. Mattsson, "Historical wind deployment and implications for energy system models," *Renew. Sust. Energ. Rev.*, vol. 168, p. 112813, 2022.
[51] S. Bahrami, M. H. Amini, M. Shafie-Khah, and J. P. S. Catalão, "A decentralized renewable generation management and demand response in power distribution networks," *IEEE Trans. Sustain. Energy*, vol. 9, no. 4, pp. 1783–1797, 2018.
[52] F. Sun, J. Ma, M. Yu, and W. Wei, "Optimized two-time scale robust dispatching method for the multi-terminal soft open point in unbalanced active distribution networks," *IEEE Trans. Sustain. Energy*, vol. 12, no. 1, pp. 587–598, 2021.
[53] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "Pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, 2018.
[54] "HI-SEAS Solar Radiation Prediction (September through December 2016)." [Online]. https://www.kaggle.com/code/runphilrun/hi-seas-solar-radiation-prediction/data.
[55] "DeepQuantileRegressionOPF." [Online]. https://github.com/lelouchsola/DeepQuantileRegressionOPF.
[56] "Lecture 11: DistFlow and LinDistFlow." [Online]. https://www.faculty.ece.vt.edu/kekatos/pdsa/Lecture11.pdf.
[57] K. Margellos, P. Goulart, and J. Lygeros, "On the road between robust optimization and the scenario approach for chance constrained optimization problems," *IEEE Trans. Automat. Contr.*, vol. 59, no. 8, pp. 2258–2263, 2014.