Constraint Learning-based Optimal Power Dispatch for Active Distribution Networks with Extremely Imbalanced Data

Yonghua Song, *Fellow*, *IEEE*, *Fellow*, *CSEE*, Ge Chen, *Member*, *IEEE*, and Hongcai Zhang, *Senior Member*, *IEEE*, *Member*, *CSEE*

***Abstract*—Transition towards carbon-neutral power systems has necessitated optimization of power dispatch in active distribution networks (ADNs) to facilitate integration of distributed renewable generation. Due to unavailability of network topology and line impedance in many distribution networks, physical model-based methods may not be applicable to their operations. To tackle this challenge, some studies have proposed constraint learning, which replicates physical models by training a neural network to evaluate feasibility of a decision (i.e., whether a decision satisfies all critical constraints or not). To ensure accuracy of this trained neural network, training set should contain sufficient feasible and infeasible samples. However, since ADNs are mostly operated in a normal status, only very few historical samples are infeasible. Thus, the historical dataset is highly imbalanced, which poses a significant obstacle to neural network training. To address this issue, we propose an enhanced constraint learning method. First, it leverages constraint learning to train a neural network as surrogate of ADN's model. Then, it introduces Synthetic Minority Oversampling Technique to generate infeasible samples to mitigate imbalance of historical dataset. By incorporating historical and synthetic samples into the training set, we can significantly improve accuracy of neural network. Furthermore, we establish a trust region to constrain and thereafter enhance reliability of the solution. Simulations confirm the benefits of the proposed method in achieving desirable optimality and feasibility while maintaining low computational complexity.**

***Index Terms*—Deep learning, demand response, distribution networks, imbalanced data, optimal power flow.**

I. Introduction

DRIVEN by the objective of achieving carbon neutrality, incorporation of distributed renewable generation (DRG) into distribution networks is gaining momentum [1]. This trend towards higher DRG penetration is paving the way for converting traditionally passive distribution networks into active distribution networks (ADNs). Such a transformation unlocks the potential for demand-side flexibility [2].

A critical challenge for ADNs is effective scheduling of flexible demand-side resources, including distributed storage systems [3], electric vehicles [4], or heating, ventilation, and air conditioning (HVAC) systems [5], to enhance DRG integration. Traditionally, coordination of these resources has been controlled based on physical models of ADNs, i.e., optimal power flow (OPF) models. For instance, in reference [6], a bilevel optimization framework was developed based on the OPF model to coordinate mobile energy storage systems with DRG in distribution networks. Reference [7] constructed an OPF model to depict operation of distribution networks and then utilized generalized Benders decomposition method to control voltages. Reference [8] introduced a tight convex relaxation of original OPF model to represent dispatchable region of an ADN. While these model-based methods can effectively operate ADNs, they typically require knowing accurate network topology and line impedance. Unfortunately, such information is often unavailable in many distribution networks because of limited measurements [9]. Reference [10] also mentioned topology information may be unknown in practice. Consequently, these model-based methods may not be applicable in such cases.

Widespread adoption of smart meters has made collecting operational data of ADNs more cost-effective [11]. Therefore, deep learning-based methods can be used to learn from this data and build surrogates to operate ADNs [12], as this data often contains network modeling information. Generally, these methods fall into three categories, as shown in Fig. 1: optimize-then-learn approach, reinforcement learning (RL), and constraint learning, summarized as follows.

1) Optimize-then-learn Approach

Optimize-then-learn approach aims to train learning models to approximate unknown mapping from operating conditions, such as power demands and available renewable generation, to optimal dispatch decisions. Once this mapping is learned, trained model can directly predict decisions for new operating conditions without requiring a full OPF model. For example, in reference [13], a graph neural network was trained to predict optimal solutions for OPF problems based on provided power demands. Reference [14] designed a physics-informed neural

Manuscript received July 16, 2023; revised October 24, 2023; accepted November 30, 2023. Date of online publication December 28, 2023; date of current version January 5, 2024. This work is supported in part by the Science and Technology Development Fund, Macau SAR, China (File no. SKL-IOTSC(UM)-2021-2023, File no. 0003/2020/AKP, and File no. 0011/2021/AGJ).

Y. H. Song and H. C. Zhang 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.

G. Chen (corresponding author, email: chen4911@purdue.edu; ORCID: https://orcid.org/0000-0003-3376-9839) is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA.

DOI: 10.17775/CSEEJPES.2023.05970

2096-0042 © 2023 CSEE. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

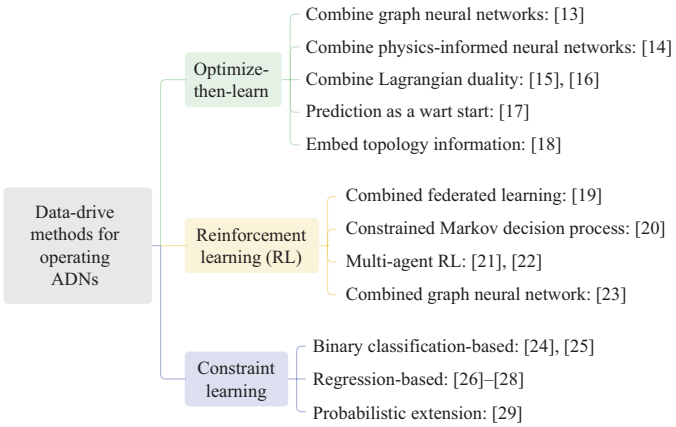


Fig. 1. Summary of the existing data-driven methods for operating ADNs.

network to act as a surrogate for DC OPF, embedding Karush-Kuhn-Tucker conditions in the loss function to enhance sample efficiency. Acknowledging that predicted decisions may be infeasible, reference [15] integrated Lagrangian duality during neural network training to improve solution feasibility. Reference [16] extended the method from [15] by training another neural network to predict instance-specific Lagrangian multipliers, which were found to enhance optimal performance in simulations. Rather than directly using the neural network's prediction as the final solution, reference [17] treated this prediction as a warm start for conventional OPF solvers, ensuring feasibility. Reference [18] went a step further by encoding topology information as a new feature, making this approach applicable to cases with flexible topologies.

A. Reinforcement Learning

RL is a technique for training agents based on interactions with environments to maximize future cumulative rewards. In [19], a RL model was devised using the actor-critic method to guide demand responses in ADNs. Federated learning was also implemented to enable agent training without sharing private information. Reference [20] modeled optimal operation of ADNs as a constrained Markov decision process and introduced a safe RL method to achieve voltage regulation and minimize energy costs. In [21], multi-agent RL was employed for voltage regulation, effectively reducing communication costs associated with network training. Similarly, reference [22] developed an RL model based on multi-agent deep deterministic policy gradient algorithm to schedule energy storage systems for voltage regulation. In [23], a combination of graph neural network and RL was used to incorporate topology information into ADNs. This integration enhanced optimality of RL model. Generally, RL can keep updating its policy based on continuous interactions. It has the potential to perform real-time operations of ADNs since it can immediately infer good actions based on current state information.

1) Constraint Learning

The core concept of constraint learning method is to replicate the OPF model using a trained neural network. When historical accidents and repair records of ADNs are available, one can obtain feasibility of historical samples. A historical sample is considered "feasible" if no related accident or repair record exists, indicating that it can satisfy all critical constraints. Otherwise, it is deemed "infeasible" and may fail to satisfy some constraints. Then, following the methodology described in references [24], a binary classification neural network can be trained to assess feasibility of a decision. After training, this neural network is reformulated as tractable mixed-integer linear constraints and serves as a surrogate for the OPF model. Reference [25] goes a step further by evaluating maximum possible constraint violation of this method based on mixed-integer linear reformulations of neural networks. If historical data of voltages and branch flows are available, as mentioned in references [26], the binary classification neural network can be replaced with a regression model to improve feasibility. In this case, output represents maximum constraint violation rather than feasibility of the decision. Reference [27] introduced a neural network compression step after neural network training to enhance computational efficiency. Additionally, reference [28] proposed a piecewise linearization-based interpretation for this method. To address probabilistic OPF problems, reference [29] extended this method by replacing conventional regression model with a deep quantile regression neural network. This approach also provided a mathematical proof demonstrating quantile regression neural network has the ability to predict the quantile of maximum constraint violation. In general, this method requires only operational data, bypassing need for network topology and line impedance. As such, it has been applied to various optimization problems, like voltage regulation [30] and carbon-electricity coordinated optimization [31]. Furthermore, this method can achieve desirable optimality because the Branch-and-Bound (B&B) algorithm can find global optima of mixed-integer linear surrogate [32].

Although previous research has demonstrated effectiveness of learning-based methods in operations of ADNs, these methods still face challenges. Specifically, optimize-then-learn approaches substitute the OPF problem by training regression models to approximate mapping from operating conditions to optimal decisions. However, these approaches necessitate optimal decisions as training labels, which are obtained through solving numerous OPF instances. Therefore, availability of topology and line impedance information is still essential for creating these instances, as referenced in [13]–[18]. RL-based methods update their policies by interacting with an environment [19]–[23]. This environment can be either a real ADN or an accurate simulator. However, interacting with a real system is almost impossible due to security concerns, while simulating ADNs accurately still requires accurate topology and line impedance information. In contrast to optimize-then-learn and RL-based approaches, constraint learning trains neural networks as surrogates and embeds these surrogates into optimization models to replicate power flow constraints. It only relies on historical operational data and does not necessitate exact topology and line impedance information, as indicated in references [24]–[29]. However, they face significant challenges in terms of data quality. On one hand, since distribution networks are mostly operated in a normal status [33] (e.g., over 99% of the time), there are commonly only few historical infeasible samples. Thus, the historical

dataset is highly imbalanced. Neural networks trained on such an imbalanced dataset may be unable to identify infeasible samples accurately. On the other hand, studies such as [34] have noted constraint learning may extrapolate aggressively from training data, leading to a solution in an area where the trained neural network has not learned. This may make prediction of the neural network inaccurate, leading to an unreliable solution.

In practical distribution networks, accurate knowledge of network topology and line impedance may not be available [9], [10], rendering use of optimize-then-learn and RL-based approaches challenging. Moreover, although power injections at all buses are typically measured, not all bus voltages and branch flows are monitored due to limited measurements. As mentioned earlier, regression-based constraint learning requires historical data for all bus voltages and branch flows to obtain training labels, making it less effective in such cases. Nevertheless, in situations where accidents occur, operators can label corresponding operational samples as "infeasible". By additionally marking samples without accidents as "feasible", it becomes possible to collect feasibility information for all historical samples. This information can be used to replicate the OPF model using classification-based constraint learning, where a binary classification neural network is trained as a surrogate to predict feasibility of a given decision. Nevertheless, the historical dataset's extreme imbalance may lead to undesirable model performance. To address this issue, this paper introduces an improved constraint learning-based method for optimizing power dispatch in ADNs. The approach consists of several key components: first, it employs classification-based constraint learning to replicate the OPF model. Second, it incorporates imbalanced learning techniques to mitigate effects of the imbalanced historical dataset. Third, a trust region is established to constrain the solution, ensuring reliability.

Compared to existing methods, this paper provides two significant contributions:

1) We introduce Synthetic Minority Oversampling Technique (SMOTE) to enhance classification performance of the neural network under the extremely imbalanced dataset. SMOTE creates synthetic samples by interpolating between existing infeasible samples and adds them to the training set during training. Then, the binary classification neural network can identify infeasible samples with higher accuracy, even when original dataset is highly imbalanced.
2) We design trust-region constraints to enhance the reliability of the proposed method's solution. Specifically, we train an one-class support vector clustering (OC-SVC) model to construct an approximate convex hull of all feasible samples as our trust region. By constraining the solution within this trust region, we can ensure the solution lies in the area the trained neural network has well learned. Then, unreliable solutions can be avoided. Moreover, this trust region does not introduce many additional constraints, so it will not introduce significant extra computational burden.

The remaining parts are organized as follows. Section II formulates the optimal power dispatch problem for ADNs. Section III introduces the proposed constraint learning-based method in detail. Section IV demonstrates our case study, and Section V concludes this paper.

II. Problem Formulation

Primary operational objective of an ADN is to minimize overall cost by optimizing scheduling of flexible resources, while simultaneously satisfying all critical constraints. In this paper, HVAC loads and distributed renewable generators are employed as representative examples of these flexible resources within ADNs. Due to the thermal inertia of buildings, the profiles of HVAC loads can be adjusted with imperceptible thermal discomforts [5], [35]. Thus, HVAC loads hold great potential as flexible resources for promoting the DRG integration in ADNs. It should be emphasized inclusion of other types of flexible resources is possible by adjusting formulation of power injections on buses.

A. Modeling of Different Components

1) HVAC Systems

By indexing different HVAC systems with $i \in \mathcal{I}$ and time slots with $t \in \mathcal{T}$, indoor temperature can be mathematically described using energy conservation equation [35]:

$$\theta_{i,t}^{\mathrm{in}} = a_i^{\mathrm{in}}\theta_{i,t-1}^{\mathrm{in}} + a_i^{\mathrm{out}}\theta_{t-1}^{\mathrm{out}} + a_i^{\mathrm{h}}(q_{i,t-1}^{\mathrm{heat}} - q_{i,t-1}^{\mathrm{cool}}),\ \forall i \in \mathcal{I},\ \forall t \in \mathcal{T}. \tag{1}$$

Here, symbols $\theta_{i,t}^{\mathrm{in}}$ and θ_t^{out} denote temperatures of indoor and outdoor environments, respectively. Symbol $q_{i,t-1}^{\mathrm{heat}}$ and $q_{i,t-1}^{\mathrm{cool}}$ represent indoor heat loads and cooling power provided by HVAC systems, respectively. Symbol a_i^{in}, a_i^{out}, and a_i^{h} are building parameters, which are calculated by [35]:

$$a_i^{\mathrm{in}} = e^{-\frac{g_i}{C_i}\Delta t},\ a_i^{\mathrm{out}} = 1 - a_i^{\mathrm{in}},\ a_i^{\mathrm{h}} = a_i^{\mathrm{out}}/g_i,\ \forall i \in \mathcal{I} \tag{2}$$

where C_i and g_i are building heat capacity and heat transfer coefficient between indoor and outdoor environments, respectively; Δt is length of a time interval. To avoid thermal discomforts, it is necessary to ensure all indoor temperatures are bounded by maximum and minimum allowable temperatures, i.e., $\boldsymbol{\theta}^{\min}$ and $\boldsymbol{\theta}^{\min}$, as follows:

$$\boldsymbol{\theta}^{\min} \le \boldsymbol{\theta}_t^{\mathrm{in}} \le \boldsymbol{\theta}^{\max},\ \forall t \in \mathcal{T} \tag{3}$$

Active and reactive power consumption of each HVAC system, i.e., $p_{i,t}^{\mathrm{HV}}$ and $q_{i,t}^{\mathrm{HV}}$, can be calculated based on the corresponding coefficient of performance COP_i and power factor ϕ_i, as follows:

$$p_{i,t}^{\mathrm{HV}} = \frac{q_{i,t}^{\mathrm{cool}}}{\mathrm{COP}_i},\ q_{i,t}^{\mathrm{HV}} = \frac{\sqrt{1-\phi_i^2}}{\phi_i} p_{i,t}^{\mathrm{HV}},\ \forall i \in \mathcal{I},\ \forall t \in \mathcal{T} \tag{4}$$

Meanwhile, we also constrain the above active power within allowable ranges imposed by device limitations:

$$\boldsymbol{p}_t^{\mathrm{HV}} \le \boldsymbol{p}_t^{\mathrm{HV,max}},\ \forall t \in \mathcal{T}. \tag{5}$$

2) Distributed Renewable Generators

By using $\boldsymbol{\lambda}_t$ to represent curtailment rates, actual outputs of distributed renewable generators, i.e., $\boldsymbol{p}_t^{\mathrm{DG}}$ can be calculated by:

$$\boldsymbol{p}_t^{\mathrm{DG}} = \boldsymbol{G}_t^{\mathrm{DG}}(\mathbf{1} - \boldsymbol{\lambda}_t),\ \mathbf{0} \le \boldsymbol{\lambda}_t \le \mathbf{1},\ \forall t \in \mathcal{T} \tag{6}$$

where available active power outputs of all distributed renewable generators are represented by $\boldsymbol{G}_t^{\mathrm{DG}}$. Here, symbol $*$ is used to denote element-wise multiplication.

3) Power Injections on Buses

Active and reactive power injected into every bus, i.e., $\boldsymbol{p}_t$ and $\boldsymbol{q}_t$, can be calculated based on energy conservation:

$$\boldsymbol{p}_t = -\boldsymbol{p}_t^{\mathrm{HV}} - \boldsymbol{p}_t^{\mathrm{base}} + \boldsymbol{p}_t^{\mathrm{DG}},\ \boldsymbol{q}_t = -\boldsymbol{q}_t^{\mathrm{HV}} - \boldsymbol{q}_t^{\mathrm{base}},\ \forall t \in \mathcal{T} \tag{7}$$

where $\boldsymbol{p}_t^{\mathrm{base}}$ and $\boldsymbol{q}_t^{\mathrm{base}}$ represent base active and reactive power demands, i.e., loads of users excluding HVAC systems, respectively.

4) Power Flow Model

DistFlow proposed in [36] can be employed to compute bus voltages and branch flows of a radial-type ADN:

$$\begin{cases} \sum_{k\in\mathcal{C}_j} P_{jk,t} = p_{j,t} + P_{ij,t} - r_{ij} I_{ij,t}^2 \\ \sum_{k\in\mathcal{C}_j} Q_{jk,t} = q_{j,t} + Q_{ij,t} - x_{ij} I_{ij,t}^2 \\ V_{j,t}^2 = V_{i,t}^2 - 2(r_{ij}P_{ij,t} + x_{ij}Q_{ij,t}) + (r_{ij}^2 + x_{ij}^2) I_{ij,t}^2 \\ I_{ij,t}^2 = \frac{P_{ij,t}^2 + Q_{ij,t}^2}{V_{i,t}^2} \\ \forall (i,j) \in \mathcal{B},\ \forall t \in \mathcal{T} \end{cases} \tag{8}$$

Here, variables $P_{ij,t}$ and $Q_{ij,t}$ represent active and reactive power flows, respectively, on branch (i,j). Variables $p_{j,t}$ and $q_{j,t}$ denote active and reactive power injection at bus j. Variables $V_{i,t}$ and $I_{ij,t}$ correspond to voltage and current magnitudes at bus i and branch (i,j), respectively. Parameters r_{ij} and x_{ij} signify resistance and reactance of branch (i,j). Symbol $(i,j) \in \mathcal{B}$ denotes index of each branch, while set $\mathcal{C}_j$ contains indexes of child buses connected to bus j.

In order to ensure system security, it is necessary to maintain specific ranges for bus voltages and branch power flows:

$$\underline{\boldsymbol{V}} \le \boldsymbol{V}_t \le \overline{\boldsymbol{V}},\ \sqrt{\boldsymbol{P}_t^2 + \boldsymbol{Q}_t^2} = \boldsymbol{S}_t \le \overline{\boldsymbol{S}},\ \forall t \in \mathcal{T} \tag{9}$$

where $\underline{\boldsymbol{V}}$ and $\overline{\boldsymbol{V}}$ denote minimum and maximum allowable voltage magnitudes; $\boldsymbol{S}_t$ and $\overline{\boldsymbol{S}}$ represent actual and upper bound of apparent power flows on different branches. Net active power at root node, i.e., G_t^{root}, can be obtained by applying network-level power balance:

$$G_t^{\mathrm{root}} = \mathbf{1}^{\mathrm{T}} \boldsymbol{p}_t + p_t^{\mathrm{loss}},\ \forall t \in \mathcal{T} \tag{10}$$

and power loss p_t^{loss} is computed by:

$$p_t^{\mathrm{loss}} = \sum_{(i,j)\in\mathcal{B}} I_{ij,t}^2 r_{ij},\ \forall t \in \mathcal{T} \tag{11}$$

B. Formulation of the Optimal Power Dispatch

Optimal dispatch of an ADN can be formulated as:

$$\min_{(\boldsymbol{p}_t^{\mathrm{HV}}, \boldsymbol{\lambda}_t)_{\forall t\in\mathcal{T}}} \sum_{t\in\mathcal{T}} \left(\eta^{\mathrm{buy}} G_t^{\mathrm{buy}} - \eta^{\mathrm{sell}} G_t^{\mathrm{sell}}\right) \Delta t \tag{P1}$$

$$\text{s.t.: } G_t^{\mathrm{buy}} - G_t^{\mathrm{sell}} = G_t^{\mathrm{root}},\ G_t^{\mathrm{buy}} \ge 0,\ G_t^{\mathrm{sell}} \ge 0,$$
$$\forall t \in \mathcal{T},\ (1)\text{–}(11) \tag{12}$$

The objective of **P1** is to minimize daily electricity purchasing cost of ADN. Parameters η^{buy} and η^{sell} are prices for purchasing and selling a unit of electricity, respectively, with condition $\eta^{\mathrm{buy}} \ge \eta^{\mathrm{sell}}$. Auxiliary variables, G_t^{buy} and G_t^{sell}, and constraint (**P1**) are introduced here to linearly represent cost: When G_t^{root} is positive, ADN buys electricity from upper-level grid. In this case, value of G_t^{buy} is positive, while G_t^{sell} equals zero. If G_t^{root} is negative, value of G_t^{buy} will equal zero, and G_t^{sell} becomes positive. Parameter Δt denotes length of a time slot. Decision variables contain power scheduling of HVAC systems, i.e., $\boldsymbol{p}_t^{\mathrm{HV}}$ and curtailment rates, i.e., $\boldsymbol{\lambda}_t$.

Establishing problem **P1** may be challenging because both power flow model (9) and power loss calculation (11) need information of network topology and line impedance that may be unavailable in practice. Data-driven methods, such as constraint learning, can bypass this requirement by training neural networks to replicate **P1**. However, as aforementioned, distribution networks are under normal operation status most of the time, making it difficult to collect sufficient infeasible samples for neural network training. Additionally, references [34] mentioned constraint learning's solution might be unreliable because it may lie in an area that neural network has not learned.

III. Solution Methodology

This paper presents a method that combines SMOTE, an imbalanced learning technique, and trust-region constraints with constraint learning method to address challenges mentioned earlier. Fig. 2 illustrates the whole procedure of the proposed method. Here, feasible samples are referred to as "*majority class instances*", while infeasible samples are referred to as "*minority class instances*". First, SMOTE is utilized to generate more minority class instances. Second, a binary

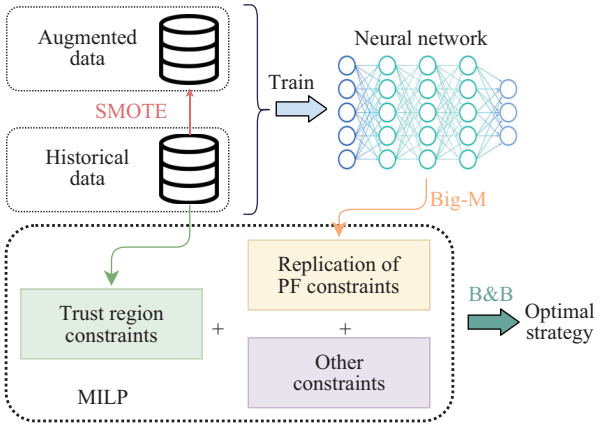


Fig. 2. The whole procedure of the proposed method.

classification neural network is trained based on historical samples and synthetic instances to determine feasibility of a specific decision. Then, the equivalent mixed-integer linear reformulation of the neural network is developed as a surrogate to replicate power flow constraints. Third, a trust region is established to constrain the solution, which can guarantee accuracy of the neural network. Finally, the surrogate, trust-region constraints, and other constraints are gathered to replicate **P1**. Since this replication is an mixed-integer linear problem, it can be efficiently solved by the Branch-and-Bound (B&B) algorithm with guaranteed optimality. In this section, we will introduce each step of the proposed method in detail.

A. Constraint Learning

We first introduce how to leverage constraint learning to replicate the OPF model without network topology and line impedance. As described in references [24], [25], a binary classification neural network is trained to judge feasibility of a decision. The input to the neural network, denoted as $\boldsymbol{x}_t$, comprises active/reactive power demands and actual output of distributed renewable generators at each bus. It can be expressed as follows:

$$\boldsymbol{x}_t = \big(\underbrace{\boldsymbol{p}_t^{\mathrm{HV}} + \boldsymbol{p}_t^{\mathrm{base}}}_{\text{active demand}}, \underbrace{\boldsymbol{q}_t^{\mathrm{HV}} + \boldsymbol{q}_t^{\mathrm{base}}}_{\text{reactive demand}}, \underbrace{\boldsymbol{p}_t^{\mathrm{DG}}}_{\text{used DRG}}\big),\ \forall t \in \mathcal{T} \quad (13)$$

The output of this neural network, i.e., $\boldsymbol{y}_t$, is the probabilities of given decision $\boldsymbol{x}_t$ that belongs to each category. Since this neural network is used to judge feasibility of $\boldsymbol{x}_t$, so we have the following two categories: "feasible" and "infeasible". Hence, $\boldsymbol{y}_t$ is a two-dimensional vector. If the first entry of $\boldsymbol{y}_t$ is no smaller than the second one, i.e., $y_{1,t} \geq y_{2,t}$, then the given decision $\boldsymbol{x}_t$ satisfies all critical constraints and belongs to the "feasible" category. Otherwise, it violates at least one constraint and belongs to the "infeasible" category.

When the Rectified Linear Unit (ReLU) is introduced as activation function, mapping from $\boldsymbol{x}_t$ to $\boldsymbol{y}_t$ can be approximately represented by forward propagation of this binary classification neural network, as follows:

$$\boldsymbol{s}_t^0 = \boldsymbol{x}_t,\ \forall t \in \mathcal{T} \quad (14)$$

$$\boldsymbol{s}_t^l = \boldsymbol{W}^l \boldsymbol{s}_t^{l-1} + \boldsymbol{b}^l,\ \forall l \in \mathcal{L},\ \forall t \in \mathcal{T} \quad (15)$$

$$\boldsymbol{h}_t^l = \max(\boldsymbol{s}_t^l, 0),\ \forall l \in \mathcal{L},\ \forall t \in \mathcal{T} \quad (16)$$

$$\boldsymbol{y}_t = (\boldsymbol{w}^{|\mathcal{L}|+1})^{\mathrm{T}} \boldsymbol{h}_t^{|\mathcal{L}|} + b^{|\mathcal{L}|+1},\ \forall t \in \mathcal{T} \quad (17)$$

where $\boldsymbol{s}^l$ and $\boldsymbol{h}_t^l$ represent outputs of linear mapping and nonlinear ReLU in each hidden layer l, respectively; $l \in \mathcal{L}$ denotes index of each hidden layer; $(\boldsymbol{W}^l, \boldsymbol{b}^l)_{\forall l \in \mathcal{L}}$ and $(\boldsymbol{w}^{|\mathcal{L}|+1}, b^{|\mathcal{L}|+1})$ are neural network's weights and bias. Since optimal solution should satisfy voltage and branch flow constraints, we need to restrict variable $\boldsymbol{x}_t$ belonging to "feasible" category. In other words, the following constraint should hold:

$$y_{1,t} \geq y_{2,t},\ \forall t \in \mathcal{T} \quad (18)$$

Obviously, the surrogate of power flow constraints (14)–(18) are nonconvex due to maximum operator in (16). If this surrogate is directly embedded as constraints, then it becomes challenging for off-the-shelf solvers to handle our optimization problem. To overcome this challenge, we follow references [24], [25] and leverage Big-M method to convert constraints (15)–(16) into a mixed-integer linear form:

$$\begin{cases} \boldsymbol{h}_t^l - \boldsymbol{r}_t^l = \boldsymbol{W}^l \boldsymbol{h}_t^{l-1} + \boldsymbol{b}^l \\ 0 \leq \boldsymbol{h}_t^l \leq M \boldsymbol{\mu}_t^l \\ 0 \leq \boldsymbol{r}_t^l \leq M(1 - \boldsymbol{\mu}_t^l) \\ \boldsymbol{\mu}_t^l \in \{0,1\}^{N_l} \end{cases} \quad \forall l \in \mathcal{L},\ \forall t \in \mathcal{T} \quad (19)$$

where N_l represents number of neurons in corresponding hidden layer. Then, even if this surrogate is embedded to replicate power flow constraints, our optimization problem can still be solved by commercial solvers like Gurobi with guaranteed optimality.

According to (10), G_t^{root} is summation of net power demands and power loss p_t^{loss}. Calculating the latter also needs line impedance information according to (11). Nevertheless, since p_t^{loss} is typically significantly smaller than total active power demands, we can approximate G_t^{root} by:

$$G_t^{\mathrm{root}} \approx \mathbf{1}^{\mathrm{T}} \boldsymbol{p}_t,\ \forall t \in \mathcal{T} \quad (20)$$

Then, requirement of the topology and line impedance can be bypassed. Note simulations in Section IV-C2 validate the proposed method can achieve desirable optimality with the above approximation.

B. Synthetic Minority Oversampling Technique

1) Oversampling

Synthetic Minority Oversampling Technique (SMOTE) is a popular oversampling method that balances class distribution by generating minority class instances [37]. To create new minority instances, SMOTE generates synthetic examples along line segments that connect k nearest neighbors of each minority class instance [37]. In this work, we use an improved version of SMOTE called "support vector machine-based SMOTE", which has been shown to outperform the original method [38]. This improved version first trains an OC-SVC model to find a minimum sphere in high-dimensional feature space to enclose most minority class instances. Based on this sphere, it identifies minority class instances that are difficult to classify, such as those close to majority class instances. Then, it generates synthetic samples in the vicinity of these minority class instances, specifically on line segments that connect nearest neighbors of each, as shown by purple diamonds in Fig. 3. Compared to original SMOTE, synthetic samples generated by the improved version are more effective because they are generated near classification decision boundary [38].

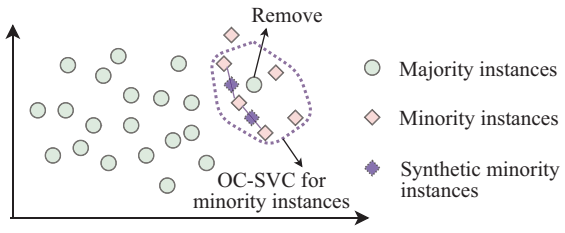


Fig. 3. Schematic diagram of the SOMTE-based oversampling and undersampling used in the proposed method.

Adding synthetic minority instances may enlarge domain of minority class and increase conservativeness. Nevertheless, the OC-SVC-based SMOTE ensures all generated minority instances are close to original ones. As a result, optimality of the proposed method can still maintain at a desirable level.

2) Undersampling

To further enhance performance of our proposed method, we apply an undersampling process to majority class instances. The key idea is to remove those majority class instances similar to minority ones. According to reference [39], the sphere built by the previous OC-SVC model can be used to judge whether a given instance is similar to minority class instances: If this instance falls within the sphere built by the OC-SVC model, such as green dot depicted in Fig. 3, then it is identified as one that is similar to minority class instances. To reduce difficulty associated with identifying minority class instances, we remove majority class instances within this sphere prior to neural network training. Then, we can significantly enhance classification performance of neural network.

C. Trust Region

As discussed in Section I, constraint learning may yield an unreliable solution that significantly deviates from historical samples because there is no specific constraint to prevent overly aggressive exploration of constraint learning. Therefore, it is crucial to define a trust region to constrain the solution. In this paper, we establish a trust region based on convex hull of all feasible samples in historical dataset. It should be noted feasible samples removed in the previous undersampling process are not involved in this step. Let $\boldsymbol{x}_n^{\text{feasible}}, \forall n \in \mathcal{N}^{\text{feasible}}$ denote feasible instances. Then, the trust region can be formed by the following linear constraints:

$$\begin{cases} \boldsymbol{x}_t = \sum\limits_{n\in\mathcal{N}^{\text{feasible}}} \omega_{n,t} x_n^{\text{feasible}} \\ \sum\limits_{n\in\mathcal{N}^{\text{feasible}}} \omega_{n,t} = 1 \end{cases}, \ \forall t \in \mathcal{T} \tag{21}$$

$$\omega_{n,t} \geq 0, \ \forall n \in \mathcal{N}^{\text{feasible}}, \ \forall t \in \mathcal{T} \tag{22}$$

By bounding the solution within this trust region, we can ensure the solution is in an area the neural network shows desirable accuracy. However, according to (21)–(22), the large number of feasible samples bring numerous variables and constraints, leading to a huge computational burden.

In order to reduce computational complexity, we train another OC-SVC model to approximate the original trust region. Fig. 4 shows how we construct this approximated trust region. This new OC-SVC model is trained based on all feasible samples. Hence, it can find a sphere that covers most feasible samples, which is shown as blue dashed lines in Fig. 4. After training, support vectors, i.e., samples on or outside the sphere (marked as red dots), can be identified. Then, we can approximately represent the original trust region with the convex hull of all support vectors, so (21)–(22) become:

$$\boldsymbol{x}_t = \sum_{n\in\mathcal{N}^{\text{SV}}} \omega_{n,t} x_n^{\text{SV}}, \ \sum_{n\in\mathcal{N}^{\text{SV}}} \omega_{n,t} = 1, \ \forall t \in \mathcal{T} \tag{23}$$

$$\omega_{n,t} \geq 0, \ \forall n \in \mathcal{N}^{\text{SV}}, \ \forall t \in \mathcal{T} \tag{24}$$

where x_n^{SV} is n-th support vector and $\mathcal{N}^{\text{SV}}$ is corresponding index set. Since support vectors are usually much fewer than majority class instances, computational complexity can be significantly reduced. Meanwhile, the following Proposition indicates feasibility can still be guaranteed even if we use approximate trust region to replace the original one.

Proposition 1. The proposed trust region $\mathcal{R}$ is a subset of the original one $\mathcal{R}^{\text{ori}}$:

$$\mathcal{R} = \{\boldsymbol{x}_t | (23)\text{–}(24)\} \subseteq \mathcal{R}^{\text{ori}} = \{\boldsymbol{x}_t | (21)\text{–}(22)\}$$

Proof: All support vectors used in (23)–(24) belong to "feasible" category. Thus, the convex hull formed by them, i.e., $\mathcal{R}$, a subset of the convex hull of all feasible samples, i.e., $\mathcal{R}^{\text{ori}}$. In other words, a feasible solution of (23)–(24) must also be feasible for (23)–(24).

D. Summary of the Proposed Method

By gathering the aforementioned constraints of HVAC systems, surrogate of the OPF model, and trust-region constraints, we can replicate **P1** as:

$$\min_{(\boldsymbol{p}_t^{\text{HV}}, \boldsymbol{\lambda}_t)_{\forall t\in\mathcal{T}}} \sum_{t\in\mathcal{T}} EC_t \tag{**P2**}$$

$$\text{s.t.} \ \underbrace{(1)\text{–}(5)}_{\text{HVAC constraints}}, \ \underbrace{(6)\text{–}(7), \ (13)\text{–}(14), \ (17)\text{–}(19)}_{\text{Surrogate of the OPF model}}$$

$$\underbrace{(\textbf{P1}), \ (20)}_{\text{Approximated net power}}, \ \underbrace{(23)\text{–}(24)}_{\text{Trust region constraints}}$$

P2 does not need network topology and line impedance, but rather only requires historical samples for neural network training. Furthermore, our method incorporates SMOTE and trust-region constraints so its performance can be guaranteed even if the historical dataset is extremely imbalanced.

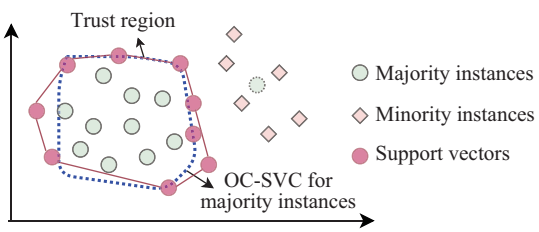


Fig. 4. Schematic diagram of the trust region.

IV. Case Study

A. Setting Up of the Test System

A case study is conducted based on the IEEE 33-bus and 123-bus test systems to verify benefits of the proposed method. Structures of the two systems are shown in Fig. 5, which comprise 16 and 25 different distributed renewable generators, respectively. Voltage levels for the two systems are 12.66 kV and 4.16 kV, respectively. Base voltage and power are set as root bus voltage and 1 MVA for both systems, respectively. Bus voltages are restricted within the range [0.9 p.u., 1.1 p.u.], and maximum allowable branch flows $\boldsymbol{S}^{\max}$ are set at 6

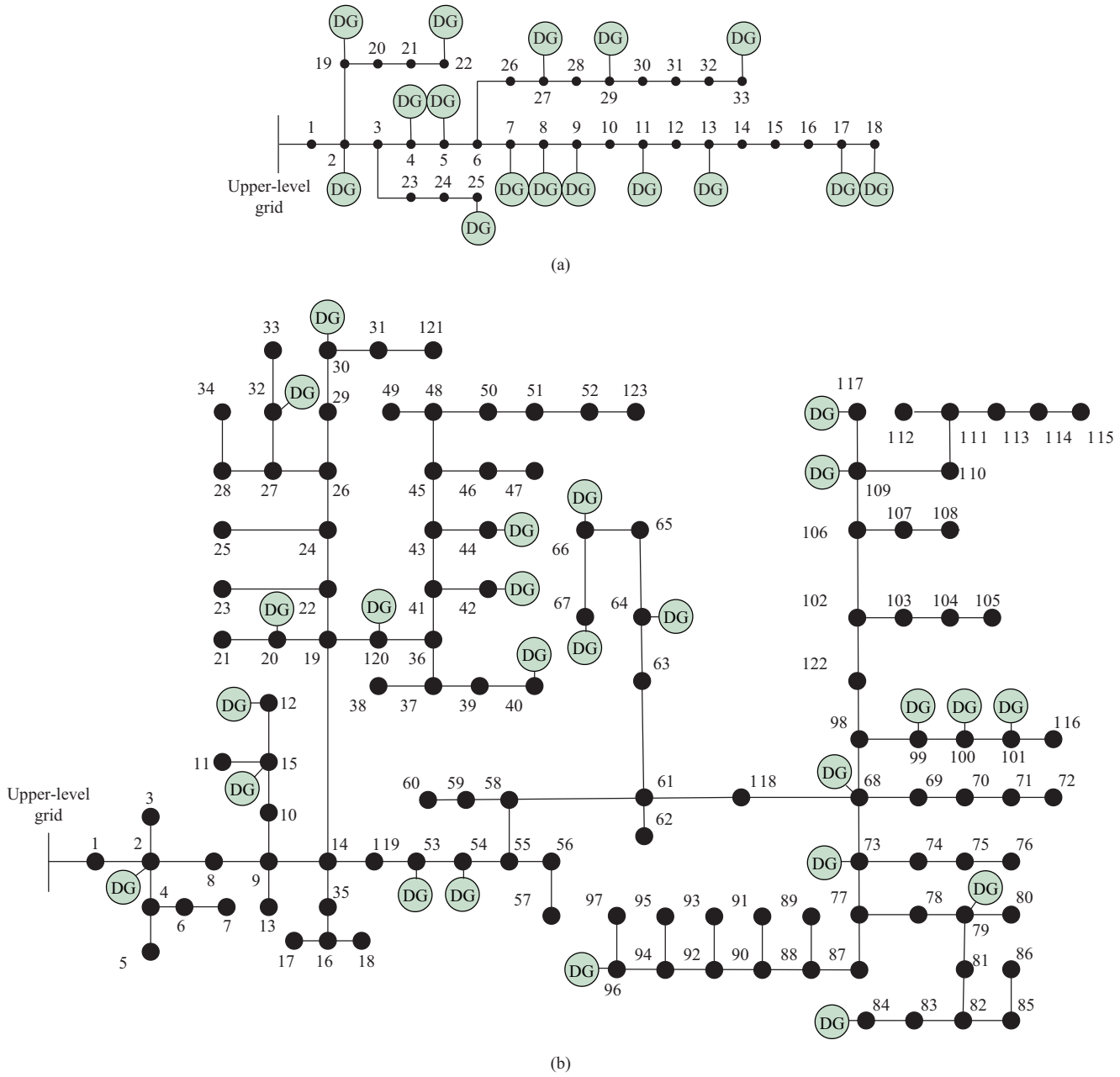


Fig. 5. Structures of (a) the 33-bus test system and (b) the 123-bus test system used in our case study. They contain 16 and 25 distributed generators (marked by "DG" in this figure), respectively.

MVA. Optimization horizon and time stepsize are 24 hours and 1 hour, respectively.

Historical data of the test ADN is generated using "Pandapower", a Python-based power system simulation tool [40]. To create the dataset, we first randomly generate multiple instances of $\boldsymbol{x_t}$ using a uniform distribution. These instances are then provided to Pandapower, which calculates corresponding bus voltages and branch flows. If an instance meets voltage and branch flow constraints outlined in (9), we label it as "feasible"; otherwise, it is labeled as "infeasible". From these instances, we randomly select 9,900 "feasible" instances and 100 "infeasible" instances to simulate the historical dataset. It is important to note this dataset is extremely imbalanced, with only 1% of instances labeled as "infeasible", which accurately reflects real-world scenario [33]. During training, 70% of instances are allocated as training set, while the remaining 30% are designated as testing set. With these samples, we can also establish trust-region constraints based on (23)–(24). Originally, the trust region should constrain all components of $\boldsymbol{x}_t$, i.e., power injections and use of DRG. Nevertheless, as Fig. 6 indicates, the available DRG has a larger fluctuation range than power demands, thus having a greater impact on bus voltages and branch flows. Consequently, we use the trust region to only restrict the use of DRG, i.e., $\boldsymbol{p}_t^{\mathrm{DG}}$ in (13) for simplification. Note the later simulations in Section IV-C3 demonstrate the proposed method can still achieve desirable performance after applying this simplification.

B. Benchmarks

To demonstrate benefits of the proposed method, three benchmarks are introduced for comparison:

1) **B1**: Conventional constraint learning used in [24], [25]. It does not involve SMOTE and trust-region constraints (termed "Without SMOTE & TRC").
2) **B2**: Conventional constraint learning combined with only SMOTE (termed "With SMOTE"). It does not involve trust-region constraints.
3) **B3**: SOCP relaxation of DistFlow [41].
4) **Baseline**: Original nonconvex DistFlow model [36]. It is solved by a nonlinear solver IPOPT.

Note the proposed method, **B1**, and **B2** are fully data-driven. They do not need network topology and line impedance information. On the contrary, both **B3** and **Baseline** are based on power flow equations, so they require this information.

C. Results of the 33-bus Test Case (Time step = 1 h)

We first test performance of different methods in six scenarios with various penetration levels of DRG in the 33-bus test system. Fig. 6 illustrates daily electricity prices, base power demands, indoor heat loads, outdoor temperatures, and available DRG. Parameters of buildings and HVAC systems are summarized in Table I.

1) Prediction Performance of Neural Networks

Constraint learning-based methods, including the proposed method, benchmarks **B1** and **B2**, necessitate training of neural networks. In our case study, we have trained two neural networks with identical structures. One network is used in the proposed method and **B2**, which incorporates SMOTE and is referred to as "w/ SMOTE". The other network is employed in **B1** without use of SMOTE and is termed "w/o SMOTE". Both networks consist of three hidden layers, each containing five neurons. Fig. 7 presents key metrics, including accuracy, true positive rate (TPR), and false positive rate (FPR), Receiver Operating Characteristic (ROC) curves, and Area Under the Curve (AUC) for both neural networks on the testing dataset. In line with discussion in Section IV-A, dataset comprises 99% "feasible" samples and 1% "infeasible" samples. In absence of SMOTE, the trained neural network consistently predicts "feasible" outcomes, achieving a 100% TPR but also a 100% FPR. This results in an AUC of only 0.5. Such performance could compromise operational security when replicating the OPF model. Nevertheless, with application of SMOTE, the trained neural network not only achieves high accuracy and TPR but also a low FPR. In other words, it correctly classifies most "infeasible" samples, leading to an AUC of 0.991, which is significantly higher than the case without SMOTE. As a result, it can serve as a reliable surrogate for the OPF model.

TABLE I
PARAMETERS OF BUILDINGS AND HVAC SYSTEMS

Parameters	Value	Parameter	Value
a_i^{in}	0.961	$\boldsymbol{\theta}^{\max}$	28°C
a_i^{out}	0.039	$\boldsymbol{p}_t^{\text{HV,max}}$	0.1 p.u.
a_i^{h}	1.961°C/p.u.	COP_i	6
$\boldsymbol{\theta}^{\min}$	24°C	ϕ_i	0.98

2) Feasibility, Optimality, and Computational Efficiency

Figure 8 compares voltage and branch flow violations across different methods in six scenarios with varying DRG penetration. Benchmark **Baseline** is directly solved by IPOPT without any relaxation or approximation, so both its voltage and branch flow violations are zeros in all scenarios. Benchmark **B3** exhibits excellent feasibility in scenarios S1, S2, and S3, while its voltage violations become significant in the rest of scenarios with high DRG penetration. For example, in S6, voltage violation of **B3** reaches 0.03 p.u. **B3** is a convex relaxation of the OPF model, and its exactness has been discussed in reference [41]. Specifically, this reference proved this relaxation is exact only if one of the following four

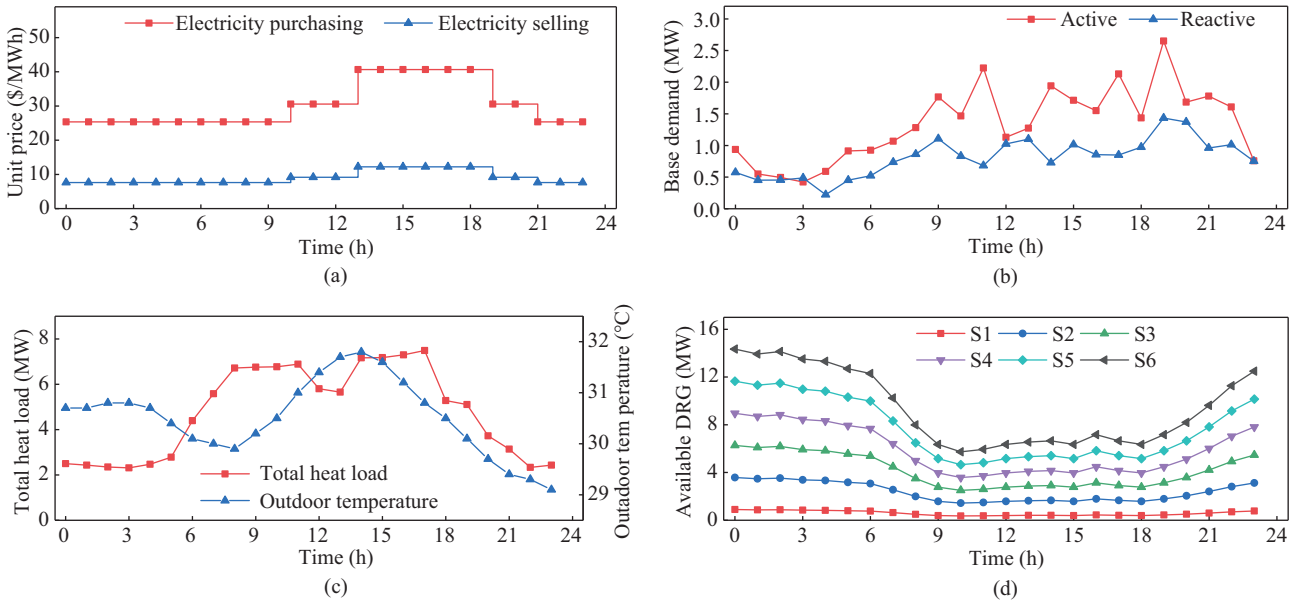


Fig. 6. Parameters used in our case study. (a) Prices for purchasing and selling a unit of electricity. (b) Base power demands (excluding HVAC loads). (c) Indoor heat loads and outdoor temperature. (d) Six scenarios with variant penetration levels of DRG.

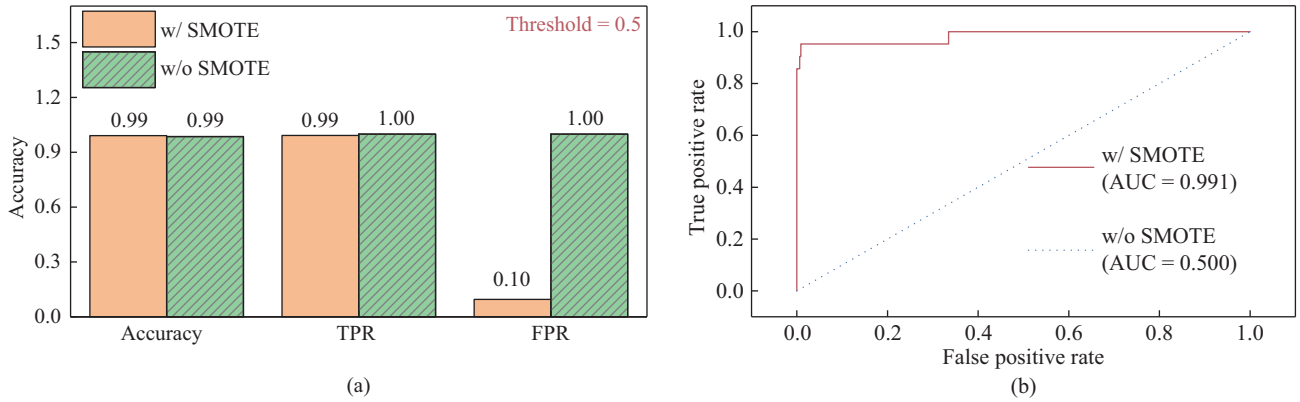


Fig. 7. Performance of the trained neural networks with and without the employment of SMOTE. (a) Prediction accuracy, TPR, and FPR. (b) Corresponding receiver operating characteristic (ROC) curves. For convenience, we use "w/" and "w/o" to denote "with" and "without", respectively.

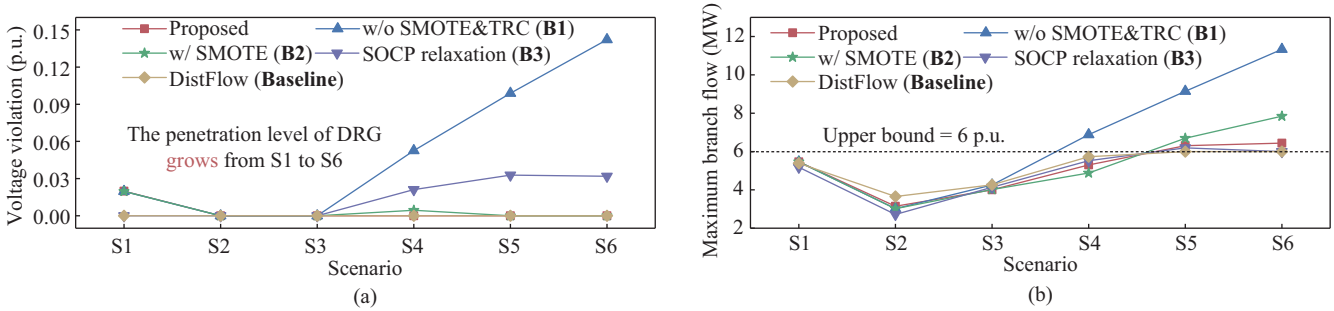


Fig. 8. Results of (a) maximum voltage violations and (b) maximum branch flows obtained by different methods in the six scenarios.

conditions is satisfied:

- There are no reverse power flows in the network.
- The r/x ratios on all branches are equal.
- The r/x ratios increase in downstream direction from substation (node 0) to leaves, and there are no reverse real power flows.
- The r/x ratios decrease in downstream direction, and there are no reverse reactive power flows.

In the 33-bus test system, the r/x ratios of branches do not consistently meet these conditions, meaning **B3** can ensure exactness only when no reverse power flow occurs. In S1-S3 with low DRG penetration, SOCP relaxation is exact and does not lead to constraint violations because there is no reverse power flow. However, in the remaining scenarios with high DRG penetration, SOCP relaxation becomes inexact, resulting in noticeable constraint violations. **B1** does not employ SMOTE to generate minority class instances, rendering its neural network incapable of identifying "infeasible" samples. Furthermore, it lacks trust-region constraints to restrict its solution. Thus, it shows poor feasibility and exhibits very large violations, e.g., maximum voltage and branch flow violations reach 0.14 p.u. and 5.33 p.u., respectively. In contrast, **B2** utilizes SMOTE to enhance its neural network performance, resulting in smaller constraint violations compared to **B1**. However, it also lacks incorporation of trust region constraints, which may result in a solution in an area the neural network has not learned. Thus, its constraint violations may be still significant sometimes, e.g., branch flow violation of **B2** reaches 1.84 p.u. in scenario S6. On the contrary, the proposed method incorporates SMOTE to improve prediction accuracy of its neural network and integrates trust region constraints to avoid unreliable solutions. Consequently, it demonstrates superior feasibility performance compared to **B1** and **B2**. Its maximum voltage and branch flow violations are only 0.01 p.u. and 0.43 p.u., respectively, which are significantly lower. These results highlight the excellent feasibility of the proposed method.

Figure 9 illustrates total electricity purchasing costs given by different methods. Although **B1** and **B2** may achieve low energy costs in some scenarios, their feasibility is poor. In S1-S3, the SOCP relaxation **B3** is exact and its solutions are globally optimal [41]. In the rest of scenarios with high DRG penetration, **B3** becomes inexact, though it derives the lowest cost. Benchmark **Baseline** is directly solved by IPOPT, so global optimality can not be guaranteed. In scenarios S1–S3, costs of the proposed method are only slightly higher than optimal ones (i.e., the solutions of **B3**), while they are lower compared to **Baseline**. For instance, the cost of the proposed method is only 0.26% larger than that of **B3** and 0.76% lower than of **Baseline**. In S4–S6, costs of the proposed method and **Baseline** are almost the same in S4-S6. These results demonstrate the proposed method's optimality is comparable to or even better compared to model-based approaches like **B3**.

Figure 10 depicts solving time required by different methods. In **B1**, the neural network is trained using an extremely imbalanced historical dataset, leading it to consistently predict "feasible" outcomes. Consequently, a simple constant constraint $\boldsymbol{y}_t = (1, 0)$ would yield the same results as constraints

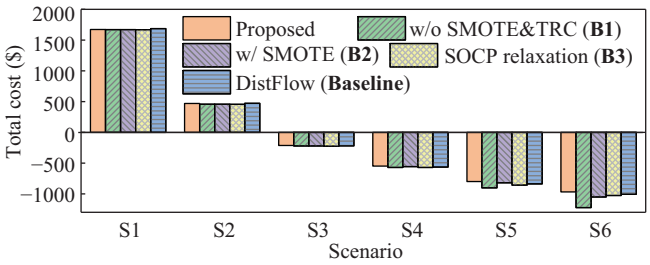


Fig. 9. Overall costs in the six scenarios.

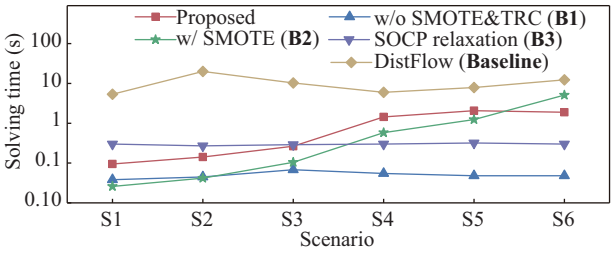


Fig. 10. Solving times of different methods in the six scenarios.

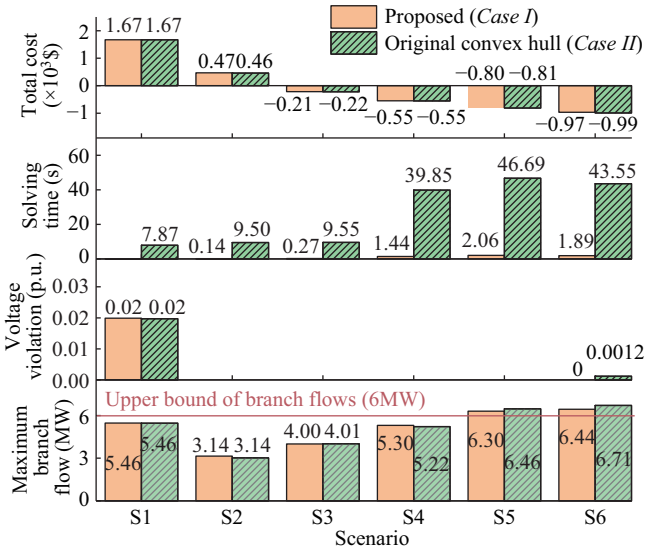


Fig. 11. Total costs, solving times, voltage violations, and maximum branch flows in *Case I* (i.e., proposed method) and *Case II* (i.e., the method based on the convex hull of all majority class instances).

(13)–(14), and (17)–(19). Note entries of $\boldsymbol{y}_t$ represent probabilities of a decision belonging to "feasible" and "infeasible" categories, respectively. As a result, **P2** degrades into a simple linear problem, resulting in consistently low computational complexity. Benchmark **B2** employs SMOTE to balance the historical dataset, and the trained neural network can differentiate between infeasible and feasible samples, allowing constraints (13)–(14), and (17)–(19) to take effect, resulting in a mixed-integer linear problem. Thus, its solving time is higher than of **B1**. **B3** is a SOCP relaxation. Its inherent convexity ensures high computational efficiency. Benchmark **Baseline** is nonconvex and solved by IPOPT. Thus, its computational complexity is much higher compared to the other methods. The proposed method not only employs SMOTE but also introduces trust-region constraints, so its computational burden is higher than of **B2**. Nevertheless, the proposed method can always find its optimal solution in 2.06 seconds, much lower compared to **Baseline**. Considering this paper focuses on a 24-hour optimal power dispatch problem, this computational efficiency is acceptable for practical utilization.

3) Effectiveness of the Approximated Trust Region

As explained in Section III-C, the trust region in the proposed method is constructed by the convex hull of support vectors, which is an approximation of the convex hull of all feasible samples. In order to evaluate effectiveness of the proposed trust region, we compare the following two cases:

1) *Case I (Proposed)*: Use convex hull of support vectors as trust region.
2) *Case II*: Use convex hull of all feasible samples as trust region.

Energy costs, solving time, voltage violations, and maximum branch flows for the two cases are summarized in Fig. 11. According to Proposition 1, trust region in *Case I* is a subset of *Case II*. As a result, overall cost in *Case I* is slightly higher. However, cost difference between the two cases is consistently negligible. Meanwhile, *Case I* exhibits better feasibility performance, as evidenced by its significantly smaller branch flow violations. Moreover, solving time of *Case I* is much lower than *Case II*. *Case I* uses only 107 support vectors to form its trust region, whereas *Case II* relies on 9,900 feasible samples. Thus, *Case I* involves much fewer additional variables and constraints and can achieve much better computational efficiency compared to *Case II*.

4) Robustness Under Variant Loads/generation

Monte-Carlo simulations are performed to assess robustness of the proposed method under diverse load/generation profiles. A total of 100 distinct load/generation profiles were considered as individual scenarios. Fig. 12 illustrates base active and reactive power demands, along with available DRG, across these 100 scenarios.

Figure 13(a) and (b) showcases constraint violations obtained by different methods. Benchmark **B3** is a SOCP relaxation and may become inexact when reverse power flows occur. Thus, it exhibits undesirable feasibility: its voltage and branch flow violations are very significant. Benchmark **Baseline** directly solves non-convex DistFlow and does not introduce any approximation or relaxation. Thus, its solution can always satisfy power flow constraints. Both benchmarks **B1** and **B2** use constraint learning method to replicate power flow constraints. However, **B1** directly employ extremely imbalanced historical dataset as training set, so the trained neural network is unable to identify infeasible decisions. Furthermore, **B1** lacks trust-region constraints to limit the solution, resulting in very poor feasibility. **B2** utilizes SMOTE to improve the neural network's ability to identify infeasible instances. However, without trust-region constraints, its solution may lie in an area the trained neural network has not learned, leading to significant constraint violations. The proposed method enhances prediction performance of its neural network with SMOTE and incorporates a trust region to constrain the solution. As a result, it achieves better

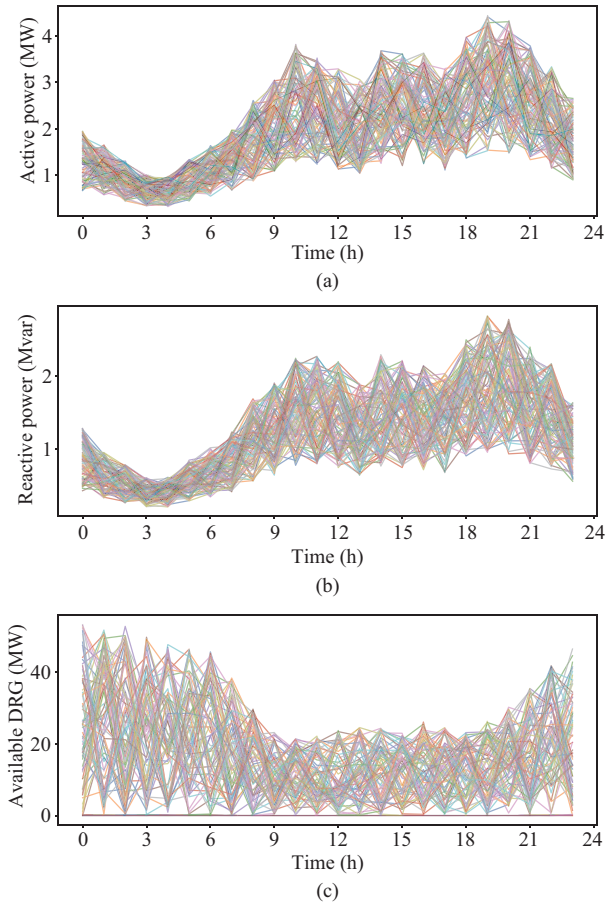


Fig. 12. Parameters used in Monte-Carlo simulations in the 33-bus test case. (a) Base active power demands. (b) Base reactive power demands. (c) Available DRG.

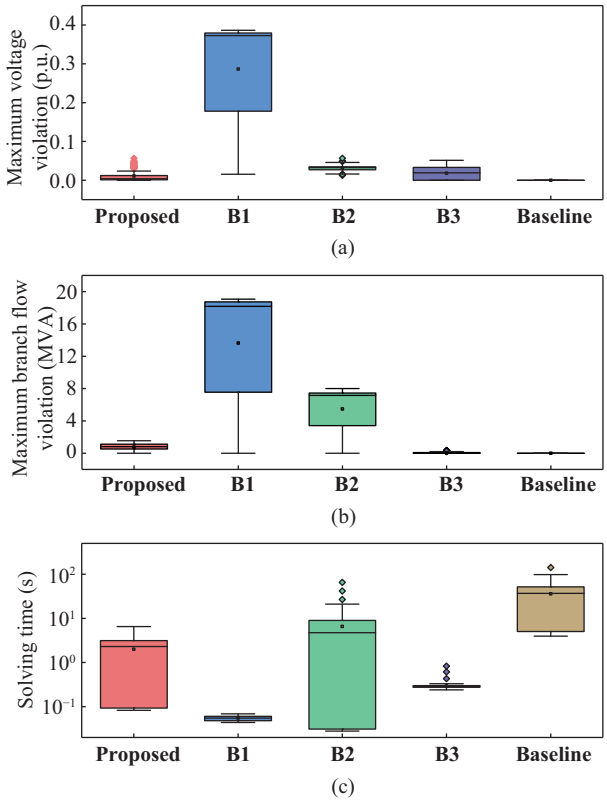


Fig. 13. Results of the Monte-Carlo simulations based on the 33-bus test system. (a) Voltage violations. (b) Branch flow violations. (c) Solving times obtained by the proposed method and benchmarks.

feasibility compared to **B2**. Although its feasibility is worse than of **Baseline**, it is fully data-driven and does not require exact topology and line impedance information. Moreover, it only uses an extremely imbalanced dataset for training. Thus, these results confirm effectiveness of the proposed method in learning power flow constraints.

Figure 13(c) shows solving times required by different methods. Benchmark **Baseline** is non-convex and solved by the nonlinear solver IPOPT. IPOPT is based on primal and dual interior point algorithm. It needs hundreds of seconds for solving, showing a low computational efficiency. Benchmark **B3** is a convex relaxation, which ensures its high computational efficiency. Computational efficiency of **B1** is very high because the trained neural network consistently predicts "feasible" outcomes and does not take any effect. **B2** introduces SMOTE to make the trained neural network has the ability to differentiate between infeasible and feasible samples, resulting in a mixed-integer linear program. Thus, its computational complexity is higher compared to **B1**. The proposed method further involves trust-region constraints. On one hand, these constraints increase computational complexity of the problem. On the other hand, they also reduce search space of the B&B. Thus, its solving time is almost the same as of **B2**. These results further validate the excellent computational efficiency of the proposed method.

D. Results of the 33-bus Test Case (Time step = 15 min)

In many distribution networks, time step for power dispatch is 15 min. Thus, we further implement another case study based on a 33-bus test system with a one-day optimization horizon and 15-minute time intervals. Fig. 14 illustrates base power demands, indoor heat loads, outdoor temperatures, and available DRG used in this new case. Prices and building thermal parameters are the same as those in **Section** IV-C. It's worth noting we have included only benchmarks **B3** and **Baseline** for comparison, as **B1** and **B2** showed poor performance in Section IV-C.

Figure 15 illustrates voltage violations and maximum branch flows obtained by different methods. Benchmark **Baseline** is based on original DistFlow without any relaxation, so its solution is always feasible and does not cause any constraint violation. Benchmark **B3** is SOCP relaxation of DistFlow. This relaxation may become inexact if reverse power flow occurs [41]. Thus, both its voltage and branch flow violations are significant S4–S6 because high DRG penetration causes reverse power flows. The proposed method uses a trained neural network to replicate power flow constraints without any additional relaxation. Its constraint violations are much smaller compared to **B3**, which indicates its great feasibility performance.

Figure 16 demonstrates total electricity purchasing costs of different methods in the new case study. Scenarios S1 and S2 exhibit relatively low DRG penetration, resulting in no reverse power flow. Consequently, SOCP relaxation **B3** is exact and its

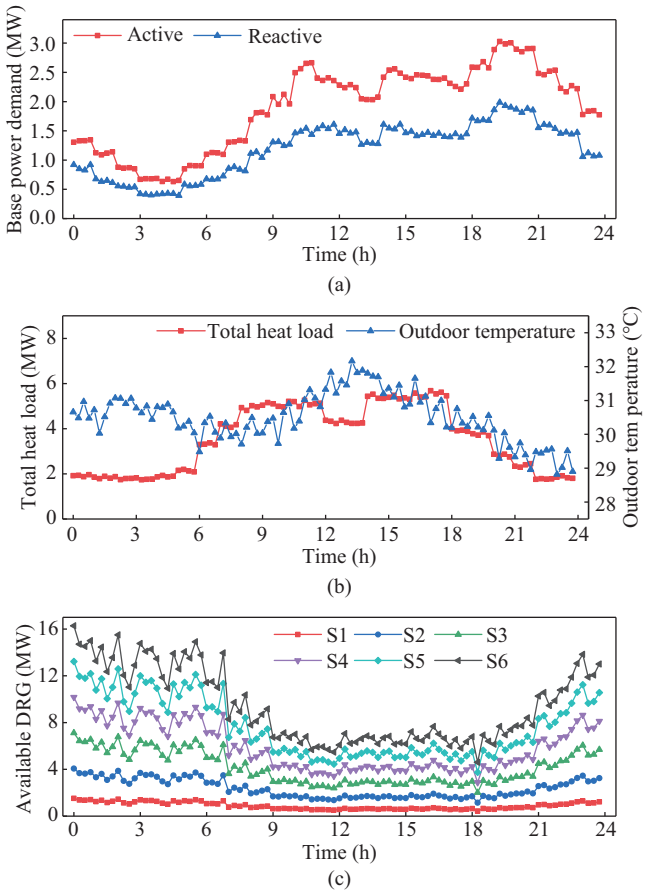


Fig. 14. Parameters used in the new 33-bus test case with 15 min as its time step. (a) Base power demands. (b) Indoor heat loads and outdoor temperature. (c) Six scenarios with variant penetration levels of DRG.

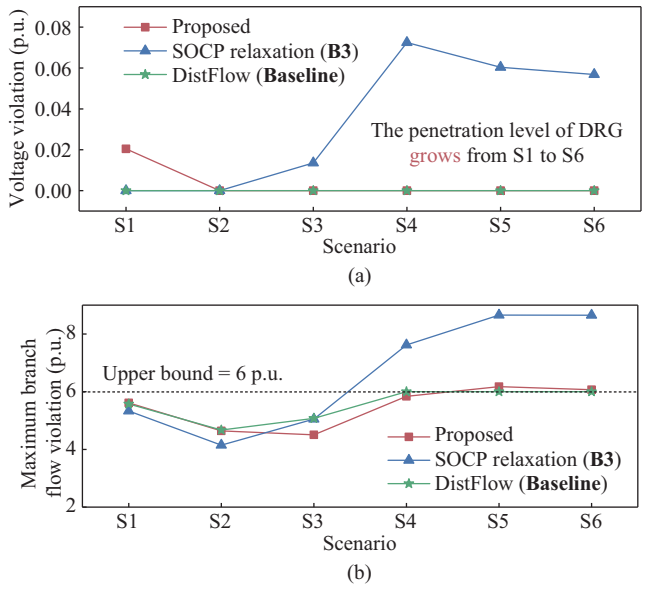


Fig. 15. Results of (a) voltage violations and (b) maximum branch flows obtained by different methods in the new 33-bus test case with 15 min as its time step.

solution can be regarded as ideal ones [41]. In these scenarios, costs of the proposed method is almost the same as of **B3**, while it is lower compared to **Baseline**. In the other scenarios

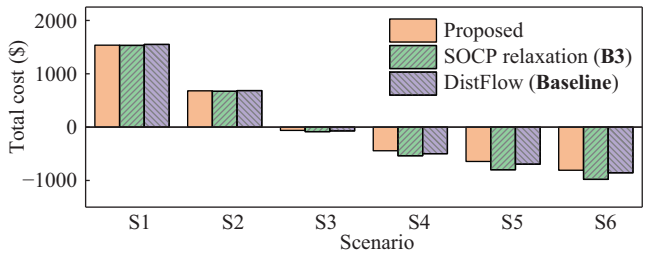


Fig. 16. Total electricity purchasing costs of different methods in the new 33-bus test case with 15 min as its time step.

with high DRG penetration, reverse power flow occurs. Thus, **B3** becomes inexact and can not guarantee feasibility, though it gives lowest costs. Cost of the proposed method is only slightly higher than of **Baseline** in these scenarios. These simulation results confirm optimality of the proposed method is comparable to those of model-based ones.

Figure 17 shows solving times of different methods. The convex nature of **B3** ensures efficient computations. **Baseline**, on the other hand, is nonconvex and solved by IPOPT, leading to significantly higher computational complexity compared to the other methods. As total number of time intervals increases from 24 to 96, the proposed method introduces more binary variables, which raises its computational burdens. Nevertheless, its solving time remains considerably lower than of **Baseline**. These results validate desirable computational efficiency of the proposed method.

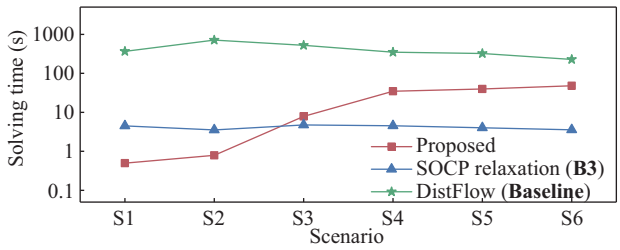


Fig. 17. Solving times of different methods in the new 33-bus test case with 15 min as its time step.

E. Results of the 123-bus Test Case

We have implemented another case study based on the IEEE 123-bus test system. To better validate performance of the proposed method, Monte-Carlo simulations are performed. A total of 100 distinct load/generation profiles are considered as individual scenarios. Fig. 18 illustrates base active and reactive power demands, along with available DRG, across these 100 scenarios.

Figure 19(a) and (b) illustrates constraint violations obtained by different methods. Results are very similar to those of the 33-bus test case: Benchmark **B3** exhibits undesirable feasibility because of its inexactness, while benchmark can always satisfy power flow constraints without any violation. Benchmark **B1** causes very large violations since the trained neural network can not identity infeasible samples. Benchmark **B2** shows much better feasibility compared to **B1** due to employment of SMOTE. The proposed method further introduces trust-region constraints to restrict its solutions with the area

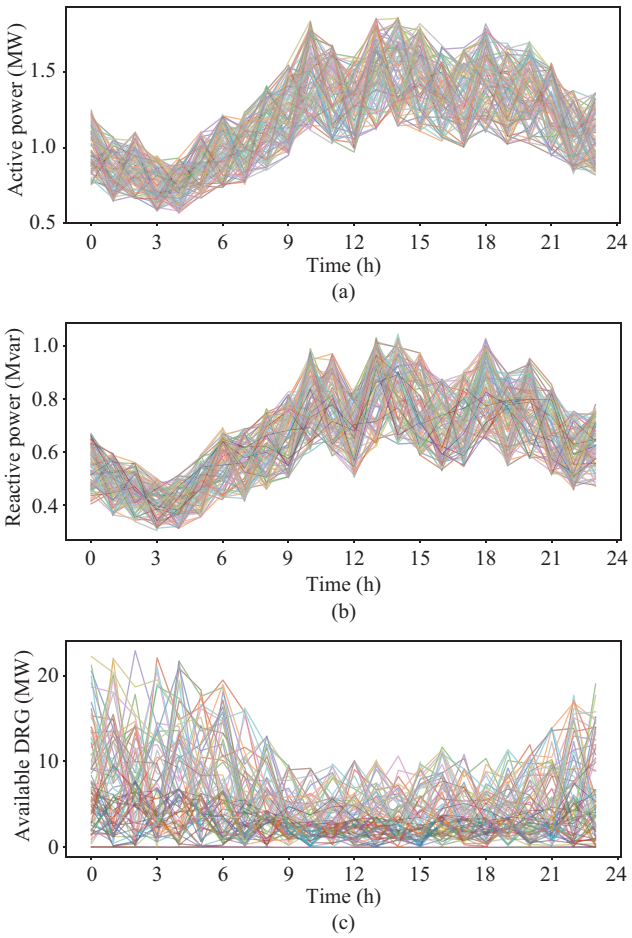


Fig. 18. Parameters used in Monte-Carlo simulations in the 123-bus test case. (a) Base active power demands. (b) Base reactive power demands. (c) Available DRG.

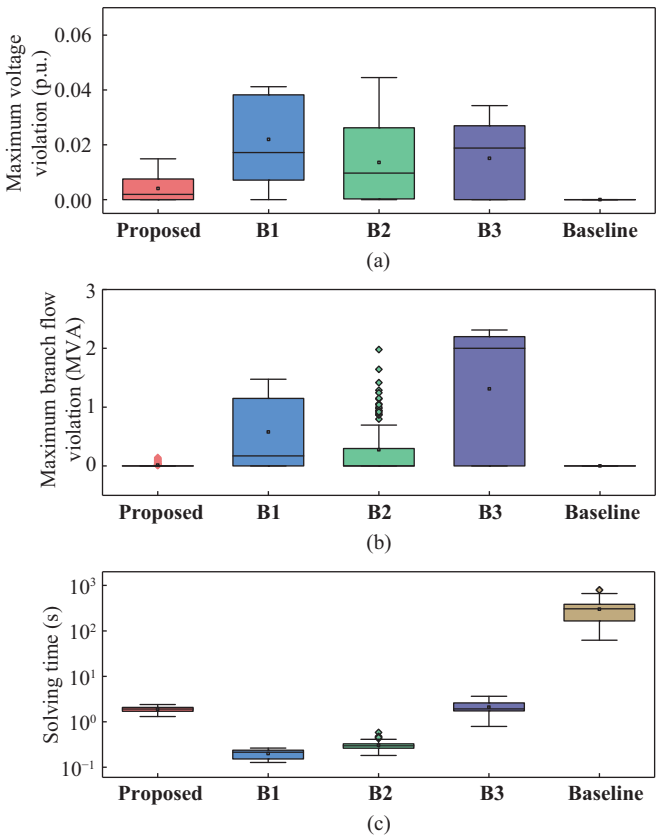


Fig. 19. Results of the Monte-Carlo simulations based on the 123-bus test system. (a) Voltage violations. (b) Branch flow violations. (c) Solving times obtained by the proposed method and benchmarks.

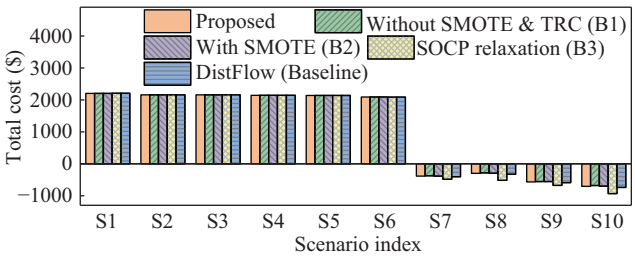


Fig. 20. Total costs obtained by different methods in the 123-bus test case. The 10 scenarios, i.e., S1-S10, are randomly selected scenarios from the 100 scenarios in the Monte-Carlo simulation. Their scenario indices are sorted by the penetration levels of renewable generation.

close to training samples, so it achieves better feasibility than of **B2**. Fig. 19(c) shows solving times required by different methods. Similarly, **Baseline** shows poor computational efficiency. **B3** exhibits excellent efficiency due to its convexity. **B1** can find its solution in a very short time because its neural network-based surrogate does not take effect, while **B2** needs more time. The proposed method involves trust-region constraints, which increase computational complexity. Nevertheless, the proposed method can always find its optimal solution in 4 s, which indicates its desirable computational performance.

Figure 20 demonstrates total costs of different methods in ten randomly selected scenarios in the Monte-Carlo simulation. The total cost of benchmark **B3** is always lowest because it is a relaxation. However, as mentioned early, its feasibility may be poor. The costs obtained by the other methods are almost the same in all scenarios. These results confirm the optimality of the proposed method is comparable to model-based ones.

V. Conclusion

Due to unavailability of network topology and line impedance information in many distribution networks, commonly used physical model-based methods may be unable to properly operate ADNs. Constraint learning is an alternative way that can bypass above information. However, operational dataset of ADNs often exhibits a significant imbalance, with normal operations accounting for most of the time. This poses a challenge for neural network training in constraint learning. To address this issue, we propose an improved constraint learning-based dispatch method. First, constraint learning is employed to train a neural network as surrogate of OPF model. Second, SMOTE is introduced to generate infeasible samples and alleviate dataset imbalance. By adding these samples into the training set, we can greatly improve neural network's ability for identifying infeasible samples. Third, a trust region is constructed based on convex hull of all feasible samples to constrain the solution. Then, we can ensure the solution in the

area the trained neural network has well learned, which guarantees reliability of the solution. To mitigate computational complexity, we further approximate this trust region based on the OC-SVC algorithm. This approximation is only formed by very a few additional constraints. Thus, its computational complexity is much lower compared to the original one. Our numerical experiments validate effectiveness of our proposed method in achieving desirable optimality and feasibility without network topology and line impedance, even when the historical dataset is highly imbalanced.

As outlined in Section IV, our proposed method may still yield solutions that marginally violate power flow constraints. In the future, we plan to focus on developing a calibration method to further enhance feasibility. Additionally, our model relies on sufficient historical data for effective training. This means it might perform suboptimally when applied to a newly installed distribution network. To address this issue, we intend to leverage state-of-the-art transfer learning techniques in conjunction with our method to extract valuable insights from a well-established model.

REFERENCES

[1] P. J. Heptonstall and R. J. K. Gross, "A systematic review of the costs and impacts of integrating variable renewables into power grids," *Nature Energy*, vol. 6, no. 1, pp. 72–83, Nov. 2021.

[2] X. D. Yang, C. B. Xu, H. B. He, W. Yao, J. Y. Wen, and Y. B. Zhang, "Flexibility provisions in active distribution networks with uncertainties," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 1, pp. 553–567, Jan. 2021.

[3] Y. F. Guo, Q. W. Wu, H. L. Gao, X. Y. Chen, J. Østergaard, and H. H. Xin, "MPC-based coordinated voltage regulation for distribution networks with distributed generation and energy storage system," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 4, pp. 1731–1739, Oct. 2019.

[4] K. Oikonomou, M. Parvania, and R. Khatami, "Deliverable energy flexibility scheduling for active distribution networks," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 655–664, Jan. 2020.

[5] G. Chen, H. C. Zhang, H. X. Hui, and Y. H. Song, "Fast wasserstein-distance- based distributionally robust chance-constrained power dispatch for multi-zone HVAC systems," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4016–4028, Sep. 2021.

[6] W. Q. Sun, W. Liu, J. Zhang, and K. P. Tian, "Bi-level optimal operation model of mobile energy storage system in coupled transportation-power networks," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 6, pp. 1725–1737, Nov. 2022.

[7] Y. Y. Chai, L. Guo, C. S. Wang, Y. X. Liu, and Z. Z. Zhao, "Hierarchical distributed voltage optimization method for HV and MV distribution networks," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 968–980, Mar. 2020.

[8] Z. G. Li, W. J. Huang, J. H. Zheng, and Q. H. Wu, "Dispatchable region for active distribution networks using approximate second-order cone relaxation," *CSEE Journal of Power and Energy Systems*, vol. 9, no. 6, pp. 1999–2007, Nov. 2023.

[9] J. W. Zhang, Y. Wang, Y. Weng, and N. Zhang, "Topology identification and line parameter estimation for non-PMU distribution network: a numerical method," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4440–4453, Sep. 2020.

[10] L. Ma, L. F. Wang, and Z. X. Liu, "Topology identification of distribution networks using a split-EM based data-driven approach," *IEEE Transactions on Power Systems*, vol. 37, no. 3, pp. 2019–2031, May 2022.

[11] V. L. Srinivas and J. Z. Wu, "Topology and parameter identification of distribution network using smart meter and μpmu measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, Jan. 2022.

[12] M. Khodayar, G. Y. Liu, J. H. Wang, and M. E. Khodayar, "Deep learning in power systems research: a review," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 209–220, Mar. 2021.

[13] M. S. Gao, J. Yu, Z. F. Yang, and J. B. Zhao, "A physics-guided graph convolution neural network for optimal power flow," *IEEE Transactions on Power Systems*, 2023, doi: 10.1109/TPWRS.2023.3238377.

[14] R. Nellikkath and S. Chatzivasileiadis, "Physics-informed neural networks for AC optimal power flow," *Electric Power Systems Research*, vol. 212, pp. 108412, Nov. 2022.

[15] F. Fioretto, T. W. K. Mak, and P. Van Hentenryck, "Predicting AC optimal power flows: combining deep learning and lagrangian dual methods," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2020, pp. 630–637.

[16] S. Park and P. Van Hentenryck, "Self-supervised primal-dual learning for constrained optimization," in *Proceedings of the 37th AAAI Conference on Artificial Intelligence Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, Thirteenth Symposium on Educational Advances in Artificial Intelligence*, 2023, pp. 4052–4060.

[17] L. Zhang and B. S. Zhang, "Learning to solve the AC optimal power flow via a lagrangian approach," in *Proceedings of 2022 North American Power Symposium (NAPS)*, 2022, pp. 1–6.

[18] M. Zhou, M. H. Chen, and S. H. Low, "DeepOPF-FT: one deep neural network for multiple AC-OPF problems with flexible topology," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 964–967, Jan. 2023.

[19] S. Bahrami, Y. C. Chen, and V. W. S. Wong, "Deep reinforcement learning for demand response in distribution networks," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1496–1506, Mar. 2021.

[20] H. P. Li and H. B. He, "Learning to operate distribution networks with safe deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 1860–1872, May 2022.

[21] J. H. Wang, W. K. Xu, Y. J. Gu, W. B. Song, and T. C. Green, "Multi-agent reinforcement learning for active voltage control on power distribution networks," in *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021, pp. 3271–3284.

[22] Y. Xiang, Y. Lu, and J. Y. Liu, "Deep reinforcement learning based topology-aware voltage regulation of distribution networks with distributed energy storage," *Applied Energy*, vol. 332, pp. 120510, Feb. 2023.

[23] D. Cao, J. B. Zhao, J. X. Hu, Y. S. Pei, Q. Huang, Z. Chen, and W. H. Hu, "Physics- informed graphical representation-enabled deep reinforcement learning for robust distribution system voltage control," *IEEE Transactions on Smart Grid*, vol. 15, no. 1, pp. 233–246, Jan. 2024.

[24] A. Venzke and S. Chatzivasileiadis, "Verification of neural network behaviour: Formal guarantees for power system applications," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 383–397, Jan. 2021.

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

[26] G. Chen, H. C. Zhang, H. X. Hui, N. Y. Dai, and Y. H. Song, "Scheduling thermostatically controlled loads to provide regulation capacity based on a learning-based optimal power flow model," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 4, pp. 2459–2470, Oct. 2021.

[27] A. Kody, S. Chevalier, S. Chatzivasileiadis, and D. Molzahn, "Modeling the AC power flow equations with optimally compact neural networks: application to unit commitment," *Electric Power Systems Research*, vol. 213, pp. 108282, Dec. 2022.

[28] G. Chen, H. C. Zhang, and Y. H. Song, "Efficient constraint learning for data- driven active distribution network operation," *IEEE Transactions on Power Systems*, 2023, doi: 10.1109/TPWRS.2023.3251724.

[29] G. Chen, H. C. Zhang, H. X. Hui, and Y. H. Song, "Deep-quantile-regression- based surrogate model for joint chance-constrained optimal power flow with renewable generation," *IEEE Transactions on Sustainable Energy*, vol. 14, no. 1, pp. 657–672, Jan. 2023.

[30] A. Alcántara and C. Ruiz, "On data-driven chance constraint learning for mixed-integer optimization problems," *Applied Mathematical Modelling*, vol. 121, pp. 445–462, Sep. 2023.

[31] L. W. Sang, Y. L. Xu, and H. B. Sun, "Encoding carbon emission flow in energy management: a compact constraint learning approach," *IEEE Transactions on Sustainable Energy*, vol. 15, no. 1, pp. 1–13, Jan. 2024.

[32] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, "Branch-and-bound algorithms: a survey of recent advances in searching, branching, and pruning," *Discrete Optimization*, vol. 19, pp. 79–102, Feb. 2016.

[33] F. X. Li, "Successful applications and future challenges of machine learning for power systems: A summary of recent activities by the IEEE

WG on machine learning for power systems [what's popular]," *IEEE Electrification Magazine*, vol. 10, no. 4, pp. 90–96, Dec. 2022.
[34] D. Yang, P. Balaprakash, and S. Leyffer, "Modeling design and control problems involving neural network surrogates," *Computational Optimization and Applications*, vol. 83, no. 3, pp. 759–800, Nov. 2022.
[35] X. Chen, E. Dall'Anese, C. H. Zhao, and N. Li, "Aggregate power flexibility in unbalanced distribution systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 258–269, Jan. 2020.
[36] M. Baran and F. F. Wu, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 735–743, Jan. 1989.
[37] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of Artificial Intelligence Research*, vol. 61, no. 1, pp. 863–905, Jan. 2018.
[38] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, Apr. 2011.
[39] M. Bicego and M. A. T. Figueiredo, "Soft clustering using weighted one- class support vector machines," *Pattern Recognition*, vol. 42, no. 1, pp. 27–32, Jan. 2009.
[40] 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 Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018.
[41] S. H. Low, "Convex relaxation of optimal power flow—part II: exactness," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 177–189, Jun. 2014.



Yonghua Song received B.E. and Ph.D. degrees from the Chengdu University of Science and Technology, Chengdu, China, and China Electric Power Research Institute, Beijing, China, in 1984 and 1989, respectively, all in Electrical Engineering. He was awarded D.Sc. degree by Brunel University in 2002, Honorary D.Eng. degree by University of Bath in 2014 and Honorary D.Sc. degree by University of Edinburgh in 2019. He is currently a Chair Professor of University of Macau and Director of the State Key Laboratory in the Internet of Things for Smart City. He is also the Vice President of Chinese Electrotechnical Society, an International Member of Academia Europaea, a Fellow of the Royal Academy of Engineering (UK) and an IEEE Fellow. He has long been engaged in renewable electrical power systems and smart energy research. He won a second prize of the State Scientific and Technological Progress Award and the prize for Scientific and Technological Progress from the Ho Leung Ho Lee Foundation.



Ge Chen received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 2015 and the M.S. degree from Xi'an Jiaotong University in 2018, both in Thermodynamic Engineering. He received the Ph.D. degree in Electrical and Computer Engineering from University of Macau, Macao, China, in 2023. He is currently a Postdoctoral Research Associate with Purdue University. His research interests include Internet of Things for smart energy, optimal operation and data-driven optimization under uncertainty.



Hongcai Zhang received B.S. and Ph.D. degrees in Electrical Engineering from Tsinghua University, Beijing, China, in 2013 and 2018, respectively. He is currently an Assistant Professor with the State Key Laboratory of Internet of Things for Smart City and Department of Electrical and Computer Engineering, University of Macau, Macau, China. From 2018 to 2019, he was a postdoctoral scholar with the Energy, Controls, and Applications Lab at University of California, Berkeley, where he also worked as a visiting student researcher in 2016. His current research interests include Internet of Things for smart energy, optimal operation and optimization of power and transportation systems, and grid integration of distributed energy resources.