Learning-Accelerated ADMM for Stochastic Power System Scheduling with Numerous Scenarios

Rajaei, Ali; Arowolo, Olayiwola; Cremer, Jochen L.

DOI
10.1109/TSTE.2025.3562640

Publication date
2025

Published in
IEEE Transactions on Sustainable Energy

Learning-Accelerated ADMM for Stochastic Power System Scheduling With Numerous Scenarios

Ali Rajaei, *Graduate Student Member, IEEE*, Olayiwola Arowolo, *Graduate Student Member, IEEE*, and Jochen Lorenz Cremer, *Member, IEEE*

***Abstract*—The increasing share of uncertain renewable energy sources (RES) in power systems necessitates new efficient approaches for the two-stage stochastic multi-period AC optimal power flow (St-MP-OPF) optimization. The computational complexity of St-MP-OPF, particularly with AC constraints, grows exponentially with the number of uncertainty scenarios and the time horizon. This complexity poses significant challenges for large-scale transmission systems that require numerous scenarios to capture RES stochasticities. This paper introduces a scenario-based decomposition of theSt-MP-OPF based on the alternating direction method of multipliers (ADMM). Additionally, this paper proposes a machine learning-accelerated ADMM approach (ADMM-ML), facilitating rapid and parallel computations of numerous scenarios with extended time horizons. Within this approach, a recurrent neural network approximates the ADMM sub-problem optimization and predicts wait-and-see decisions for uncertainty scenarios, while a master optimization determines here-and-now decisions. Additionally, we develop a hybrid approach that uses ML predictions to warm-start the ADMM algorithm, combining the computational efficiency of ML with the feasibility and optimality guarantees of optimization methods. The numerical results on the 118-bus and 1354-bus system show that the proposed ADMM-ML approach solves the St-MP-OPF with 3-4 orders of magnitude speed-ups, while the hybrid approach provides a balance between speed-ups and optimality.**

***Index Terms*—Stochastic multi-period optimal power flow, ADMM, deep learning.**

NOMENCLATURE

Indices and Sets

$s \in \Omega^s$	Index for scenarios.
$t \in \mathcal{T}$	Index for time.
$g \in \Omega^g$	Index for generators.
$i \in \mathcal{V}$	Index for nodes.
k	Index for ADMM iterations.
$n \in \Omega^{\mathcal{N}}$	Index for ML data samples.

Variables

x_g	Active and reactive power generation (H&N decisions).
$z_{g,t,s}$	Active and reactive power generation (W&S decisions).
$\lambda^k_{g,s}$	Lagrangian multipliers.
r^k, s^k	Primal and dual residual of ADMM.
(χ^k_s, y^k_s)	Input-output pair of ML model.
A, B, C	Trainable linear operators for ML.

Parameters

$d_{i,t,s}$	Nodal active and reactive net load.
$\hat{x}, \hat{z}$	Fixed value of variables x and z.
ϵ_x	Exploration rate.

Received 23 August 2024; revised 17 December 2024 and 17 February 2025; accepted 20 March 2025. Date of publication 21 April 2025; date of current version 22 September 2025. This work was supported through Dutch Research Council, Veni Talent Program under Grant 19161. Paper no. TSTE-01059-2024. *(Corresponding author: Ali Rajaei.)*

The authors are with the Department of Electrical Sustainable Energy, Delft University of Technology, 2628 CD Delft, The Netherlands, and also with the Center for Energy, Austrian Institute of Technology, 1210 Wien, Austria (e-mail: a.rajaei@tudelft.nl; o.a.arowolo@tudelft.nl; j.l.cremer@tudelft.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSTE.2025.3562640.

Digital Object Identifier 10.1109/TSTE.2025.3562640

I. INTRODUCTION

POWER systems face significant operational challenges as uncertain renewable energy sources (RES) replace flexible dispatchable sources. The inherent stochastic nature of RES causes fluctuations in the net load (actual demand minus RES generation), posing a challenge in maintaining the supply-demand balance. Extensions of the optimal power flow (OPF) optimization, notably the stochastic multi-period AC OPF (St-MP-OPF) handle RES uncertainties [1]. However, solving the St-MP-OPF for large-scale systems with a high share of RES remains computationally challenging, as the computations grow exponentially with the number of uncertainty scenarios and the time horizon [2]. Therefore, this paper aims to accelerate solving the St-MP-OPF using a machine learning (ML)-based optimization proxy.

Two-stage stochastic programming (2SP) models uncertainty realizations by generating a set of scenarios and integrating them into the OPF formulation. When the number of scenarios is sufficiently large, 2SP provides an accurate solution that reflects the impact of the uncertainties [3], [4], [5]. For example, St-MP-OPF with DC and linearized AC equations are formulated in [6] and [7] to handle RES uncertainty. However, such approaches [6], [7] face scalability challenges with a large number of uncertainty scenarios. This challenge is particularly pronounced for large-scale transmission systems with a high RES share, demanding a multitude of scenarios to accurately capture their stochasticity [4], [5]. Decomposition methods, such as Benders decomposition (BD), improve the tractability of 2SP

and address uncertainties in various power system scheduling problems [8], [9], [10]. However, BD requires dual values of the sub-problems to generate cuts, and it can exhibit non-smooth convergence, introducing complexities for the use of ML-based proxies.

The alternating direction method of multipliers (ADMM) is a widely used distributed algorithm for convex optimizations [11]. [12] develops a scenario-decomposition based on ADMM for model predictive control problems as an example of generic 2SP; however, it faces scalability limitations with increasing time horizons. Several ADMM-based approaches for St-MP-OPF have been developed through geographical decomposition [13], [14], [15], [16], [17], [18]. These approaches decompose the original optimization into sub-areas [13], [14], AC and DC sub-grids [15], electric and heat systems [19], and multi-agent distribution systems [16], [17], to preserve privacy when multiple cooperative agents (e.g., microgrid or sub-grid operators) are responsible for their local resources [18]. However, these approaches [13], [14], [15], [16], [17], [18] are limited in their scalability to a large number of scenarios. Addressing this limitation, this paper proposes a scenario-based decomposition using ADMM, and uses ML to enhance scalability for handling a large number of scenarios with increased time horizons.

Motivated by the need for real-time OPF solutions, recent research has investigated supervised ML approaches to learn an OPF optimization proxy [20]. In works such as [21], [22], a deep neural network (DNN) predicts an estimate of the centralized single-period OPF solution. Other works focus on feasibility restoration with the predict-and-reconstruct approach [23], [24], [25], [26], predicting the active set of constraints [27], [28], self-supervised learning [29], [30], approaching the MP-OPF [31], [32], learning integer solutions [33], [34], and constraint-driven approach for N-k SC-OPF [35]. Several studies, such as [36], [37], [38], [39], have explored using ML within 2SP for computational improvements. In [36], a neural network approximates the second stage objective value but does not determine the second stage solution. [37] learns an iterative local search policy for the first stage solution of a combinatorial optimization, while [38] accelerates the L-shaped method substituting ML predictions for second stage solutions. [39] accelerate the BD using a cut classifier to identify effective cuts from sub-problems, reducing master problem complexity.

Learning-accelerated ADMM methods for multi-agent and regionally decentralized OPF problems are developed in [40], [41], [42], [43]. The approaches in [41], [42] learn convergence trajectories, and [43] learns converged dual and consensus values, which are used as high-quality warm-starts for ADMM, providing speed-ups compared to ADMM methods [13], [14], [15], [16], [17], [18]. These Learning-accelerated ADMM methods are advantageous for multi-agent settings, preserving regional privacy and convergence guarantees while ensuring feasibility through local controllers. However, our work focuses on scalability challenges of uncertainty scenarios in two-stage stochastic optimization, solved by a system operator, rather than regionally decentralized OPF problems. To handle numerous future uncertain scenarios derived from a probability distribution, we develop a statistical ML model to predict these future uncertain decisions.

This paper proposes an ML-accelerated ADMM approach for solving the two-stage St-MP-ACOPF problem. In this respect, we develop a scenario-based decomposition of St-MP-OPF via ADMM, where a master problem determines here-and-now (H&N) decisions, and numerous sub-problems determine future wait-and-see (W&S) decisions of the uncertainty scenarios. Additionally, we propose an ML-based approach that significantly accelerates the developed ADMM-based approach. A recurrent neural network (RNN) approximates scenario sub-problems, providing fast close-to-optimal solutions for W&S decisions; while a master optimization solves for H&N decisions and guides the scenario sub-problems towards a consensus. The RNN enables the parallelization of assessing numerous scenarios with GPU and the consideration of long time horizons. Moreover, the master optimization ensures physical feasibility for H&N decisions and consensus among scenarios by incorporating optimization constraints and objective penalization, while a differentiable repair layer ensures the feasibility of ML W&S decision predictions. Additionally, the ML predictions are used to warm-start the ADMM algorithm in the proposed hybrid approach. Therefore, the proposed approach bridges the gap between ML and optimization, harnessing the computational efficiency of ML models alongside the feasibility guarantees of optimization methods.

The main contributions of this paper are:

- The scenario-based decomposition of the St-MP-OPF based on the ADMM algorithm.
- The ADMM-ML approach that significantly speeds up the ADMM-based algorithm and facilitates scalability to numerous scenarios and long time horizons for large-scale transmission systems, while ensuring physical feasibility. The ML model is trained on samples with a few scenarios during the training phase but can be used for samples with numerous scenarios during the testing phase.
- The proposed hybrid approach to balance speed-ups and optimality of the solution. The hybrid approach uses ML predictions as a warm-start for the ADMM algorithm, enjoying the same convergence guarantees.
- The proposed ϵ-greedy approach to generate training data for the developed ADMM-ML. This approach balances exploration and exploitation to improve performance.

The paper is organized as follows. Section II introduces the proposed ADMM-based stochastic MP-SC-OPF formulation. Section III presents the proposed ADMM-ML and training data generation approach. Section IV presents the case studies, followed by conclusion in Section V.

II. Distributed Stochastic MP-OPF

A. Stochastic MP-OPF Formulation

The St-MP-OPF schedules the optimal dispatch of generators for a given time horizon to supply the demand, while considering the operational limits of the grid, the N-1 security criteria, and stochastic RES uncertainties. Consider a set of scenarios $s \in \Omega^s$ as shown in Fig. 1. Each scenario represents one realization

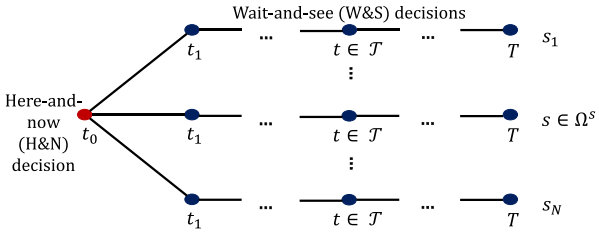

Fig. 1. Schematic of two-stage stochastic multi-period OPF on a scenario tree.

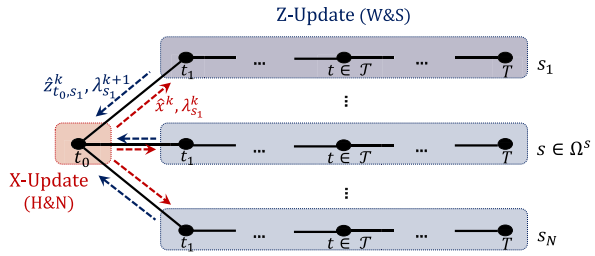

Fig. 2. ADMM update at iteration k of the proposed approach.

of the uncertain parameter (i.e. net load) for a time horizon of $t \in \mathcal{T}$. The two-stage stochastic optimization involves two sets of decisions: 1) the here-and-now (H&N) decision for the current time step t_0, denoted by x, 2) the wait-and-see (W&S) decisions for each stochastic scenario, denoted by z. This approach determines the H&N decision while considering all realizations of uncertain parameters through possible W&S decisions.

Consider a transmission network with a set of nodes $i \in \mathcal{V}$, and a scenario tree of uncertain nodal net load denoted by $d_{i,t,s}$. The St-MP-OPF optimization problem can be written:

$$\min_{z_{g,t,s}} \sum_{s\in\Omega^s}\sum_{t\in\mathcal{T}}\sum_{g\in\Omega^g} \pi_s f_g(z_{g,t,s}) \tag{1a}$$

$$\text{s.t.} \quad z_{g,t,s} \in \mathcal{Z}_g \tag{1b}$$

$$h(z_{t,s}, d_{i,t,s}) \le 0 \qquad i \in \mathcal{V}, t \in \mathcal{T}, s \in \Omega^s \tag{1c}$$

$$x_g = z_{g,t_0,s} : \lambda_{g,s} \qquad g \in \Omega^g s \in \Omega^s \tag{1d}$$

where $z_{g,t,s}$ is the dispatch of each generator $g \in \Omega^g$, over the time horizon and scenarios. Indices g, i are dropped in this paper to lighten the notation. Moreover, in the case of AC-PF, z shows the active and reactive dispatch. The objective function (1a) minimizes the expected operational cost of the generators, where f_g is a quadratic function. The probability of all scenarios is assumed equal, i.e., $\pi_s = 1/|\Omega^s|$. (1b) denotes generator constraints, including maximum/minimum generation, and ramping limits. h in (1c) is a convex function representing equality and inequality constraints of the grid. We use SOCP-relaxed AC-PF constraints [44]. Furthermore, we introduce an auxiliary variable for the H&N generator dispatches denoted by x_g. Respectively, the non-anticipativity constraint (1d) ensures that all scenarios have a *consensus* on the H&N decision for the current time step t_0. Moreover, $\lambda_{g,s}$ denotes the Lagrangian dual variable of (1d). The detailed formulation of (1) is provided in Appendix A.

B. Scenario-Based ADMM Approach

ADMM is a distributed algorithm to solve convex optimization problems [11]. This paper uses the consensus-based ADMM formulation to decompose the convex stochastic optimization (1) into scenario-based sub-problems. Note that the consensus constraint (1d) couples the variables of scenarios; which impedes the distributed decomposition. The Augmented Lagrangian of (1) can be written by relaxing (1d):

$$\mathcal{L}_\rho = \sum_{s\in\Omega^s}\left(\sum_{t\in\mathcal{T}} f(z_{t,s}) + \lambda_s\left(x - z_{t_0,s}\right) + \left(\frac{\rho}{2}\right)||x - z_{t_0,s}||_2^2\right)$$

$$\text{s.t. (1b)–(1c).} \tag{2}$$

where ρ is a positive constant parameter. The St-MP-OPF problem can now be decomposed into scenario sub-problems. The ADMM algorithm iteratively updates the variables in three steps, namely the X-update, Z-update, and dual-update:

$$x^{k+1} := \arg\min_x \mathcal{L}_\rho\left(x, z^k, \lambda^k\right) \tag{3a}$$

$$z_s^{k+1} := \arg\min_{z_s} \mathcal{L}_\rho\left(x^k, z_s, \lambda_s^k\right) \qquad s \in \Omega^s \tag{3b}$$

$$\lambda_s^{k+1} := \lambda_s^k + \rho(x - z_s) \qquad s \in \Omega^s \tag{3c}$$

Fig. 2 and Algorithm 1 summarize the ADMM-based approach for St-MP-OPF. The H&N decision x is derived by the X-update optimization, while the W&S decisions for each scenario $z_{t,s}$ are derived by independent local Z-update optimizations. In Line 2, the $\hat{x}^k, \hat{z}^k_{t_0,s}$ parameters can be initialized by historical data or solving a MP-OPF optimization for t_0; while $\lambda_s^{k_0}$ is initialized with zero values.

1) X-Update: Given $\hat{z}^k_{t_0,s}$ and the Lagrangian multipliers λ_s^k from previous iteration k, the X-update at iteration $k+1$ proceeds by solving (Algorithm 1-L4):

$$\min_x \sum_{s\in\Omega^s} \lambda_s^k\left(x - \hat{z}^k_{t_0,s}\right) + \left(\frac{\rho}{2}\right)||x - \hat{z}^k_{t_0,s}||_2^2 \tag{4a}$$

$$\text{s.t.} \quad h(x, d_{t_0}) \le 0 \tag{4b}$$

Algorithm 1: ADMM-Based St-MP-OPF.

1 Input: $d_{i,t,s} \quad i \in \mathcal{V}, t \in \mathcal{T}, s \in \Omega^s$
2 Initialize $\hat{x}^{k_0}, \hat{z}^{k_0}_{t_0,s}, \lambda^{k_0}_s$
3 **for** $k \in \{0, ..., k^{max}\}$ **do**
4 Solve X-update (4) for x
5 Update $\hat{x}^{k+1} \leftarrow x$
6 **for** $s \in \Omega^s$ **do**
7 Solve Z-update (5) for $z_{t,s}$
8 Update $\hat{z}^{k+1}_{t_0,s} \leftarrow z_{t_0,s}$
9 Update $\lambda_s^{k+1} \leftarrow \lambda_s^k + \rho(x - z_{t_0,s})$
10 **if** $r^k \le \epsilon_r$ *and* $s^k \le \epsilon_s$ **then**
11 Break for loop and report the results.

The subscripts t and s are removed from the operational variables, as the X-update only solves for the H&N dispatches at t_0. Moreover, as the scenario tree of Fig. 1 shows, the net loads of all scenarios are equal at the current time step t_0. (4b) satisfies the operational limits of the grid. The output of the X-update (i.e., x) is then passed to all scenario sub-problems as Algorithm 1-L5 and Fig. 2 shows.

2) Z-Update: The Z-update optimization determines the generator dispatches over the given time horizon for each scenario $s \in \Omega^s$ in a distributed manner. In Algorithm 1-L7, considering $\hat{x}^k$ from X-update and the Lagrangian multipliers λ_s^k at previous iteration k, the Z-update at iteration $k+1$ is:

$$\min_{z_{t,s}} \left(\sum_{t \in \mathcal{T}} f(z_{t,s}) + \lambda_s^k \left(\hat{x}^k - z_{t_0,s} \right) + \left(\frac{\rho}{2} \right) ||\hat{x}^k - z_{t_0,s}||_2^2 \right) \quad \forall s \in \Omega^s \tag{5}$$

$$\text{s.t. (1c).}$$

The Z-update optimization (5) solves for each scenario $s \in \Omega^s$ independently. Subsequently, the Lagrangian multipliers are updated:

$$\lambda_s^{k+1} := \lambda_s^k + \rho \left(x - z_{t_0,s} \right) \quad \forall s \in \Omega^s \tag{6}$$

The new Lagrangian values and the Z-update outputs are passed to the next ADMM iterations as Fig. 2 shows.

The ADMM algorithm stops when it reaches the maximum iteration k^{max}, or if the primal and dual residual, r^k, s^k, become less than specified thresholds of ϵ_r, ϵ_s (Line 10):

$$r^k := \frac{1}{|\Omega^s|} \sum_{s \in \Omega^s} ||x^k - z_s^k||_2^2 \tag{7a}$$

$$s^k := ||\rho(x^k - x^{k-1})||_2^2 \tag{7b}$$

By using convex SOCP-relaxed AC equations, the ADMM algorithm is guaranteed to converge ($r^k, s^k \to 0$) as $k \to \infty$ [11], [44].

III. Learning-Accelerated ADMM

A. Method Overview

The previous section presented an ADMM-based formulation of the St-MP-OPF. Scenario sub-problems can be solved in parallel to significantly speed up the ADMM algorithm and improve scalability to a large number of scenarios. However, the parallel implementation of ADMM with off-the-shelf solvers like Gurobi requires many CPU cores [45]. Moreover, the computations grows exponentially with the time horizon. These factors pose challenges for near-real-time solutions of St-MP-OPF for large-scale transmission systems that require numerous scenarios. Therefore, this section presents the ADMM-ML approach.

Fig. 3 depicts the proposed ML workflow. First, training samples are generated based on historical data. Second, the proposed approach in Section III-D generates a training dataset with output labels. Third, the RNN is trained with supervised imitation learning to approximate the sub-problem optimization

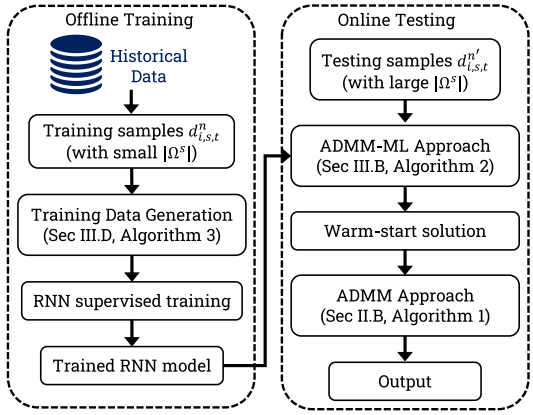

Fig. 3. Workflow of the proposed Hybrid ADMM-ML approach.

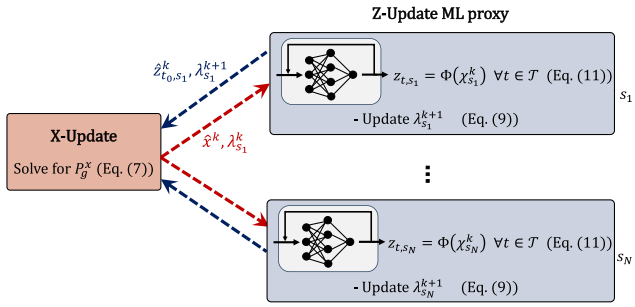

Fig. 4. The ADMM-ML approach for stochastic scheduling at iteration k.

(5). During online testing, the trained RNN provides a high quality solution in Section III-B. This solution is then used to warm-start the ADMM algorithm in the proposed hybrid approach in Section III-C.

Fig. 4 presents the schematic of the proposed ADMM-ML approach during testing. Similar to Fig. 2, the H&N dispatch of the generators is obtained by solving the X-update optimization (4), and then passed to the Z-update sub-problems. However, here the Z-update optimizations (5) are replaced with an ML optimization proxy. Given the X-update output, the Lagrangian multipliers and the net load over the time horizon $\mathcal{T}$, the trained RNN predicts the W&S dispatch of the generators over $\mathcal{T}$ for each scenario $s \in \Omega^s$. The dual-update step and the required communications are the same as the ADMM Algorithm 1.

The following properties of the ADMM-ML approach provide computational speed-ups and ensure physical feasibility:

- As RNNs use simple matrix multiplications and the same RNN is used for all sub-problems, several z-update sub-problems can be run together in mini-batches by an RNN forward pass.
- The recurrent structure of RNNs provides scalability to increasing time horizons.
- As the uncertainty scenarios are drawn from the same probability distribution, the same RNN is used for all sub-problems. Notably, the RNN can be trained on problem samples where each sample has a few scenarios (small $|\Omega^s|$), and later be used for problems where each sample has many scenarios (large $|\Omega^s|$). Note that *scenarios*

refer to the uncertainty scenarios $s \in \Omega_n^s$ for each ML sample n.

- The X-update optimization ensures the feasibility of H&N decisions, while a differentiable feasibility restoration layer satisfies power balance and ramping limits of W&S decisions.
- The proposed hybrid approach leverages close-to-optimal ML predictions to provide a high-quality warm starting point for ADMM, enjoying the same convergence guarantees while significantly reducing computations.

B. ADMM-ML Approach

1) Recurrent Neural Networks: We propose RNNs to learn an optimization proxy, imitating the Z-update sub-problem optimization (5). RNN is a widely used approach to capture temporal relationships in time sequence data such as hourly scheduling [32]. For each time step $t \in \mathcal{T}$, the hidden state $h_t \in \mathbb{R}^{F_h}$ and the output $y_t \in \mathbb{R}^{F_o}$ are calculated based on the current input $\chi_t \in \mathbb{R}^{F_i}$ and the previous hidden state h_{t-1}:

$$h_t = \sigma\left(A\chi_t + Bh_{t-1}\right) \tag{8a}$$

$$y_t = \sigma(Ch_t) \tag{8b}$$

where $A \in \mathbb{R}^{F_h \times F_i}$, $B \in \mathbb{R}^{F_h \times F_h}$ and $C \in \mathbb{R}^{F_o \times F_h}$ are linear operators, and σ is a pointwise non-linear function. Multiple RNN blocks can be stacked together to capture more complex relations.

In the case of the Z-update optimization (5) at ADMM iteration k, the nodal net load $d_{t,s}$ for a given scenario s, the output of X-update x, and the Lagrangian multipliers λ_s^k are the input features $\chi_s^k = \{d_{t,s}, x^k, \lambda_s^k\}_{t\in\mathcal{T}}$. Furthermore, the output of the Z-update optimization, i.e., $y_s^k = \{z_{t,s}^k\}_{t\in\mathcal{T}}$ for the given scenario s is the output of the last layer. Therefore, the Z-update optimization proxy Φ can be expressed as:

$$y_s^k = \Phi\left(\chi_s^k\right) \tag{9}$$

While we use RNNs in this paper, the proposed ADMM-ML approach is general and can be extended to use other neural architectures.

2) ADMM-ML Approach: Algorithm 2 details the proposed ADMM-ML approach for the stochastic MP-SC-OPF. In comparison to the ADMM-based Algorithm 1, here a forward-pass of Φ in (9) predicts the Z-update output (Line 7). Additionally, in Line 8, a feasibility restoration layer ensures the prediction feasibility, which is discussed in the next section. Note that the forward pass of the RNN is done for a mini-batch of the scenarios. In this respect, the ADMM-ML approach can parallelize the Z-update sub-problems, achieving significant speed-ups for numerous scenarios. The rest of Algorithm 2 is the same as Algorithm 1.

3) Feasibility Restoration: Supply-demand imbalances and violations of generator ramping constraints can compromise system security, making the feasibility of ML predictions essential for the secure operation of power systems. To address this, we develop a *feasibility restoration layer* (FRL) to *repair* ML dispatch W&S predictions, ensuring the satisfaction of generator and power balance constraints. The implicit differentiable layers [46] embed the feasibility restoration within the neural architecture and enable end-to-end training. Considering an initial dispatch prediction $\{z_{t,s}\}_{t\in\mathcal{T}}$ from the RNN block as input, the proposed FRL determines a feasible prediction $z'_{t,s}$ by:

$$z'_{t,s} = \underset{z'_{t,s}}{\text{argmin}} \quad ||z'_{t,s} - z_{t,s}||_2 \tag{10a}$$

$$\text{s.t.} \quad z'_{t,s} \in \mathcal{Z}_g \tag{10b}$$

$$h'(z'_{t,s}, d_{t,s}) = 0 \tag{10c}$$

where (10a) finds the feasible point $z'_{t,s}$ closest to $z_{t,s}$. (10b) ensures that $z'_{t,s}$ satisfies the ramping limits of the generator, and (10c) satisfies the supply-demand balance. Therefore, during the forward pass, (10) solves a simple LP to project the RNN prediction into the feasible space, and during the backward pass, gradients propagate through the FRL to update the RNN weights. Moreover, the minimum and maximum generation limits are satisfied using a Sigmoid function as the activation function of the RNN output layer, which predicts a value between [0,1] and then scales to $[z^{\min}, z^{\max}]$.

Algorithm 2: ADMM-ML Approach.

1 Input: $d_{i,t,s} \quad i \in \mathcal{V}, t \in \mathcal{T}, s \in \Omega^s$
2 Initialize $\hat{x}^{k_0}, \hat{z}_{t_0,s}^{k_0}, \lambda_s^{k_0}$
3 **for** $k \in \{0, ..., k^{max}\}$ **do**
4 Solve X-update (4) for x
5 Update $\hat{x}^{k+1} \leftarrow x$
6 **for** mini-batch of $\chi_s^k \quad \forall s \in \Omega^s$ **do**
7 Predict $z_{t,s}$ with the RNN model Φ (9)
8 Restore feasibility by $z'_{t,s} = \text{FRL}(z_{t,s})$ in (10)
9 Update $\hat{z}_{t_0,s}^{k+1} \leftarrow z'_{t_0,s}$
10 Update $\lambda_s^{k+1} \leftarrow \lambda_s^k + \rho(x - z_{t_0,s})$
11 **if** $r^k \le \epsilon_r$ *and* $s^k \le \epsilon_s$ **then**
12 Break for loop and report the results.

C. Hybrid Approach (ADMM-ML-H)

Although the FRL ensures the satisfaction of the power balance and generator constraints, other PF constraints, such as line thermal limits and voltage limits for W&S decisions in (1c), still need to be addressed. Additionally, while the ADMM approach is guaranteed to converge to optimality, the ADMM-ML approach lacks such assurances. We propose a *hybrid* approach of ML-based Algorithm 2 and optimization-based Algorithm 1, denoted by *ADMM-ML-H*. Initially, the ADMM-ML approach is used for a fixed number of iterations to provide fast and high-quality predictions $z_{t,s}$ and Lagrangian values λ_s. These values are then used to warm-start the ADMM approach, which runs for one or a few iterations. Therefore, the proposed ADMM-ML-H approach benefits from prediction speed-ups, all constraints feasibility, and convergence guarantees.

Algorithm 3: Training Data Generation.

1 **for** data sample $n \in \Omega^{\mathcal{N}}$ **do**
2 Input: $d^n_{i,t,s} \quad i \in \mathcal{V}, t \in \mathcal{T}, s \in \Omega^s$
3 Initialize $\hat{x}, \hat{z}_{t_0,s}$
4 Set $\lambda^{k_0}_s$ with normal random values $\mathcal{N}(0, \sigma^2)$
5 Initialize exploration rate $\epsilon_x = 1.0$
6 **for** $k \in \{0, ..., k^{max}\}$ **do**
7 Solve X-update (4) for x
8 Update $\hat{x}^{k+1} \leftarrow x$
9 **for** $s \in \Omega^s$ **do**
10 Solve Z-update (5) for $z_{t,s}$
11 Update $\hat{z}^{k+1}_{t_0,s} \leftarrow z_{t_0,s}$
12 Store $\chi^k_s = \{d^n_{i,t,s}, \hat{x}^k, \lambda^k_s\}_{t\in\mathcal{T}}$ as features
13 Store $y^k_s = \{\hat{z}^k_{t_0,s}\}_{t\in\mathcal{T}}$ as labels
14 Generate random uniform values $u_s = \mathcal{U}[0, 1]$
15 **for** $\forall s \in \Omega^s$ **do**
16 **if** $u_s \leq \epsilon_x$ **then**
17 Exploration:
18 Set λ^{k+1}_s with random values $\mathcal{N}(0, \sigma^2)$
19 **if** $u_s > \epsilon_x$ **then**
20 Exploitation:
21 Update $\lambda^{k+1}_s \leftarrow \lambda^k_s + \rho(x - z_{t_0,s})$
22 Update $\epsilon_x \leftarrow \max(\epsilon_x - \Delta\epsilon_x, \epsilon^{\min}_x)$

D. ϵ-Greedy Training Data Generation

The previous subsections presented the ADMM-ML approach during the test phase. This section presents the training procedure of the ML model.

Inspired by the ϵ-greedy algorithm [47], we propose a novel approach to generate training data for the ADMM-ML approach. The ϵ-greedy algorithm is commonly used in reinforcement learning (RL). With probability ϵ the RL agent explores a random action, and with probability $1 - \epsilon$, it exploits the currently known best action. Trading off between exploration and exploitation balances the learning and the performance in the test phase [47]. Previously developed ML proxies use either a full-exploration [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32] or a full-exploitation algorithm [40], [41], [43]. A full-exploration algorithm (i.e., random sampling of the input parameters) encounters diverse samples during training but may struggle with similarity during testing. Moreover, a full-exploitation algorithm (i.e., sampling from ADMM iterations) focuses on exploiting similar samples during training, potentially leading to difficulties in adapting to novel situations during testing.

We propose Algorithm 3 that explores the space around the optimal λ_s values with ϵ-greedy exploration. $\epsilon_x \in [0, 1]$ is the exploration rate. The Lagrangian multipliers for iteration k_0 are set by generating normal random values $\mathcal{N}(0, \sigma^2)$, where σ^2 is a pre-defined hyper-parameter (Line 4). In Line 7-11, the X-update and Z-update optimizations are solved similarly to Algorithm 1. Respectively, $\chi^k_s = \{P^d_{t,s}, \hat{x}^k, \lambda^k_s\}_{t\in\mathcal{T}}$ are stored as input features, and $y^k_s = \{\hat{z}^k_{t_0,s}\}_{t\in\mathcal{T}}$ values are stored as output labels for the Z-update ML proxy (Line 12-13). Then, random uniform values u_s are generated and compared with ϵ_x to decide between exploration and exploitation. For exploration (Line 16-18), new random values are generated for Lagrangian multipliers of the next ADMM iteration $k + 1$. Whereas for exploitation (Line 19-21), Lagrangian multipliers are updated based on (6). The Algorithm 3 starts with full-exploration ($\epsilon_x = 1$) in the first iteration, but gradually decays over ADMM iterations with $\Delta\epsilon_x$ until it reaches a minimum exploration rate $\epsilon^{\min}_x$ (Line 22). The training data generation algorithm is run for all data samples of $n \in \Omega^{\mathcal{N}}$ for a specified number of ADMM iterations $k^{\max}$.

Given a training dataset of input-output values $\{(\chi^k_s, y^k_s)\}$ for the Z-update optimizations, the RNN is trained with supervised learning using backpropagation on a loss function over the training dataset $\Omega^{\mathcal{N}}$. The trained RNN is used during the testing phase according to Algorithm 2. Notably, the RNN is trained on samples with small Ω^s, and tested on samples with large Ω^s.

IV. Case Studies

A. Settings and Test Networks

The case studies are performed on the IEEE 118-bus, and the PEGASE 1354-bus transmission networks [48] for two St-MP-OPF formulations: 1-St-MP-SC-DCOPF considering DC PF, 2-St-MP-ACOPF considering AC PF equations with SOCP relaxation [44]. The critical contingency set Ω^c for the SC-DCOPF is obtained offline by performing a contingency screening assuming 20% additional demand than the nominal demand. The hourly load demand and wind profile of the Netherlands in 2015 are obtained from publicly-available data in [49], [50] to generate $|\Omega^{\mathcal{N}}| = 8^{\sim}K$ net load samples. 10 wind farms of size 50MW (100MW) are added to the IEEE 118-bus (PEGASE 1354-bus) networks [51]. Stochastic hourly scenarios for a time horizon $\mathcal{T} = 24$ hours are generated using ARIMA models [3] with StatsModels package [52]. Moreover, the hourly profiles are scaled to the nominal values of the benchmark networks in addition to a $\pm 10\%$ random uniform noise. See Appendix B for details on the scenario generation algorithm. The samples are then randomly split between training, validation and testing with 70%-10%-20% ratio. For each training sample n, $|\Omega^s_n| = 10$ uncertainty scenarios are generated and used in Algorithm 3, to generate the training dataset. However, for each testing sample n', $|\Omega^s_{n'}|$ =100-1000 scenarios are generated and solved based on Algorithm 2, investigating the scalability of the proposed ADMM-ML approach. Note the difference between *sample* and *scenario*. *Sample* $n \in \Omega^{\mathcal{N}}$ refers to each training or testing sample n, while *scenarios* refer to the stochastic uncertainty scenarios $s \in \Omega^s_n$ for each sample n. Therefore, testing samples with $|\Omega^s| = 100$ does NOT mean 100 test samples, rather it means each test sample considers 100 uncertainty scenarios.

The AC-OPF and the DC-SCOPF problems are modelled in Pyomo [53] and solved with Gurobi [45]. The ADMM stopping criteria $\epsilon_r = \epsilon_s = 10^{-4}$ and ADMM step $\rho = 0.01$ are assumed. Moreover, the BD model stops when the optimality gap reaches 0.01%. $S_{base} = 100$MW is assumed for per-unit calculations.

The RNN model is implemented with PyTorch package [54], with 3 hidden layers, $f_h = 64$, ReLU activation functions, and Sigmoid function for the last layer. The mean squared error (MSE) is used as the loss function during training. All models are trained with the Adam optimizer with 16 mini-batches, 50 epochs, 10^{-3} learning rate. The hyper-parameters are selected through local tuning on the validation dataset, based on MSE loss on MP-OPF prediction task. The mini-batch size of 100 (1000) is considered during the testing phase of the 118-bus (1354-bus) system. The numerical experiments are run on the DelftBlue supercomputer with 32GB RAM and NVIDIA V100S GPUs with Python 3.9 [55].

The following models are compared to investigate the efficacy of the proposed approach for the St-MP-OPF:

- *Central optimization (CO):* this baseline solves the St-MP-OPF in a single optimization. This is the ground truth that the accuracy of other models is compared with.
- *Benders decomposition (BD):* this baseline solves the St-MP-OPF using a scenario-based multi-cut Benders decomposition [8].
- *ADMM:* the proposed ADMM-based Algorithm 1.
- *ADMM-ML:* the proposed Algorithm 2, integrating ADMM and the RNN model with feasibility restoration layer.
- *ADMM-ML-H:* the proposed hybrid approach, where the output of the ADMM-ML model warm-starts the ADMM model that runs for one step.

When reporting the computational time of distributed optimization approaches (BD and ADMM), scenario sub-problems are assumed to be solved in parallel. Therefore, the computational time is $t = \sum_k (t^x_k + \max_{s\in\Omega^s}(t^z_{s,k}))$, where $t^x_k, t^z_{s,k}$ are the computational times of the X-update and Z-update subproblem in iteration k, respectively. We assume 50 CPU cores to ensure a fair comparison between ADMM and ADMM-ML, as the latter relies on GPU-accelerated computations, while ADMM achieves speed-ups through parallelization of sub-problems. Although we implement ADMM sequentially due to hardware limitations, we report results equivalent to parallel computation with 50 cores, reflecting a realistic setup.

B. Scalability to Scenarios With Distributed Optimization

Fig. 5 investigates the scalability issues of CO and distributed optimization models (i.e. BD and ADMM) to the number of scenarios. Fig. 5 presents the computational time of the models when solving St-MP-SC-DCOPF with 1 to 1000 scenarios in the IEEE 118-bus system. The computational time of the CO exhibits exponential growth, reaching a point where the solution for CO with $|\Omega^s| = 1000$ scenarios is intractable. The CO for $|\Omega^s| = 1000$ scenarios was run on a system with 128GB RAM and encountered an out-of-memory error. In contrast, the BD and ADMM demonstrate better scalability to the number of scenarios, as they apply a scenario-based decomposition and solve scenario sub-problems in parallel. However, the increase in computation time for 1000 scenarios is due to the assumption of 50 CPU cores, which leads to 20 batches of sub-problems. while the BD and ADMM enable solving for numerous scenarios, they still require many CPU cores for parallel computations. This limitation motivates using ML approaches to facilitate parallel computations on GPU and provide better scalability to the number of scenarios.

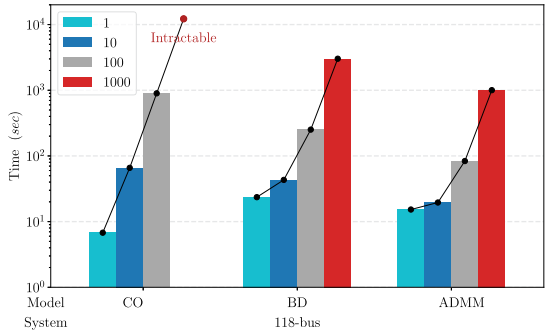

Fig. 5. Computational time of different models to solve St-MP-SC-DCOPF with 1 to 1000 scenarios on the 118-bus system. The CO solution for $|\Omega^s| = 1000$ is intractable, and thus interpolated.

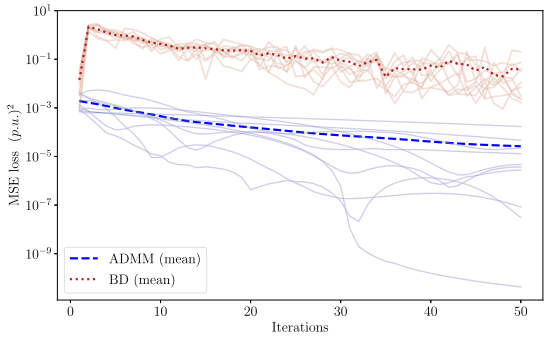

Fig. 6. Convergence of ADMM and BD models in solving the St-MP-SC-DCOPF with $|\Omega^s| = 100$ on the 118-bus system.

TABLE I
CONVERGENCE PERFORMANCE ON THE 118-BUS SYSTEM (IN PU)

	MSE loss mean (std)	convergence rate mean (std)
BD	0.149 (0.132)	1.3 (0.4)
ADMM	0.003 (0.004)	1.7 (1.1)

C. ADMM Convergence Analysis

Fig. 6 and Table I provide a comparison of the convergence performance between the ADMM and BD in solving St-MP-SC-DCOPF samples with $|\Omega^s| = 100$ on the IEEE 118-bus system. Ten random data samples are executed for 50 iterations. Fig. 6 shows the MSE loss of the solution obtained by the ADMM and BD over iterations, with respect to the ground truth of CO. The convergence rate is defined as $cr = (\log(e_{i_{\max}}) - \log(e_{i_0}))/\log(i_{\max})$, where e_i is the MSE loss at the i-th iteration. The BD solutions change more abruptly between iterations due to the added cuts, which can introduce complexities to ML-based proxies. In contrast, ADMM shows a faster and smoother convergence due to objective penalization, making it more suitable for ML-based proxies.

TABLE II
COMPUTATIONAL TIME, OPTIMALITY GAP, AND ACCURACY RESULTS FOR ST-MP-SC-DCOPF ON THE 118-BUS SYSTEM TEST SAMPLES WITH $|\Omega^s| = 500$

	Time (sec)	Gap (%)		MAE (pu)	
		Mean	Max	Mean	Max
CO	3942.4	-	-	-	-
ADMM	277.2	0.00	0.01	0.02	0.11
ADMM-ML	0.9	0.59	0.85	0.11	0.19
ADMM-ML-H	28.5	0.02	0.08	0.03	0.09

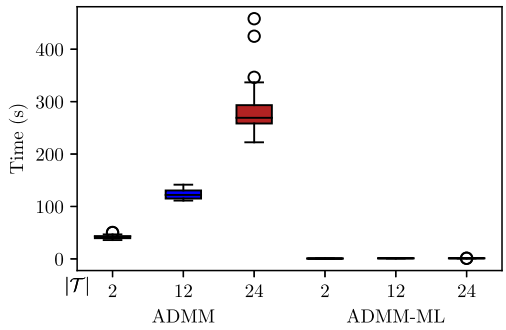

Fig. 7. Computational time of ADMM and ADMM-ML for different time horizons $\mathcal{T}$.

D. ADMM-ML Accuracy and Computational Time

Table II presents the computational time, optimality gap, and accuracy of different models on testing samples of St-MP-SC-DCOPF with $|\Omega^s| = 500$, $|\mathcal{T}| = 24$ for the 118-bus system. The proposed ADMM model reduces computational time by one order of magnitude compared to the CO model; however, it remains insufficient for real-time operations, particularly in large-scale systems. Moreover, the ADMM model computes the Z-updates in 10 batches of 50 scenarios, leading to scalability issues with numerous scenarios. The proposed ADMM-ML model significantly reduces the computational time by 2.5 orders of magnitude, with a minor 0.6% optimality gap and 0.1 pu MAE error. Additionally, the proposed ADMM-ML-H offers a balance between time and accuracy. By using the ADMM-ML predictions to warm-start the ADMM, the ADMM-ML-H achieves an optimality gap of 0.02% and MAE error of 0.03%, with an order of magnitude speed-up compared to ADMM. Therefore, we recommend the ADMM-ML model when fast solutions are essential, and the ADMM-ML-H model when the optimality and feasibility of the solution is preferred.

Fig. 7 investigates the scalability of ADMM and ADMM-ML with respect to the time horizon $|\mathcal{T}|$ on testing samples of St-MP-SC-DCOPF with $|\Omega^s| = 500$ for the 118-bus system. The computational time of ADMM exhibits exponential growth as $|\mathcal{T}|$ increases. In contrast, the computational time of ADMM-ML remains relatively constant, as the RNN efficiently determines the Z-update solutions over $\mathcal{T}$.

E. ML for MP-OPF Prediction

Table III compares SOCP-relaxed AC-OPF with several neural architectures for MP-OPF prediction over a 24-hour period on the IEEE 118-bus system. The following models are compared: feed-forward neural network (FNN), graph convolutional network (GCN), recurrent neural network (RNN), long short-term memory (LSTM), and the proposed RNN-FR with the feasibility restoration layer. While the RNN and LSTM are comparable with the FNN in terms of accuracy and optimality gap, the RNN and LSTM exhibit improved feasibility for ramp constraints, as the ignorance of inter-temporal dependencies by the FNN and GCN models results in more constraint violations. The GCN demonstrates poor performance with a 2.4% optimality gap, primarily due to parameter sharing among generator nodes. The proposed RNN-FR eliminates constraint violations and reduces the optimality gap to 0.3%. In terms of inference time, all ML models achieve significant speed-ups of 2 to 5 orders of magnitude compared to the SOCP optimization for the 118-bus system. Regarding training time, the RNN and LSTM require less training time than FNN and GCN due to their recurrent structure. However, the FRL in RNN-FR increases the training time, as the FRL optimization, though a simple LP, is solved on CPU cores at each forward pass.

TABLE III
PERFORMANCE COMPARISON OF NEURAL ARCHITECTURES FOR MP-OPF OVER A 24-HOUR HORIZON ON THE 118-BUS SYSTEM

	Gap (%)	MAE (pu)	Violation (%)		Time (sec)	
			Balance	Ramp	Testing	Training
Gurobi	-	-	-	-	4.0E+0	-
FNN	0.9	.017	1.4	0.6	2.6E-4	3.4E+2
GCN	2.4	.045	2.3	5.3	1.8E-3	1.1E+3
RNN	0.4	.012	0.8	0.2	2.3E-5	1.9E+1
LSTM	0.6	.010	2.7	0.1	3.0E-5	3.5E+1
RNN-FR	0.3	.011	0	0	2.4E-2	2.4E+3

It is noteworthy that the proposed ADMM-ML approach is general. While we focus on the day-ahead operational scheduling of power systems, if one were to apply this approach to other problems or datasets, the selection of the neural architecture should be based on the specific characteristics of the problem or datasets.

F. ML Scalability to Large Systems With Numerous Scenarios

Fig. 8 presents the primal residual r^k of the ADMM-ML model on testing samples of St-MP-SC-DCOPF with $|\Omega^s| = 1000$ for the 1354-bus system. Notably, solving the St-MP-SC-DCOPF on a 1354-bus system with 1000 uncertainty scenarios over a 24-hour horizon is highly time-consuming for the baseline CO, BD, and the proposed ADMM model. The ADMM-ML model efficiently provides a solution in under 5 minutes, while the CO is intractable and the ADMM model would require approximately 5 hours. The ADMM-ML model achieves the stopping criteria of $\epsilon_r = 10^{-4}$ after 20 iterations, a value sufficiently low for a consensus, ensuring no generator ramping constraint will be violated. Moreover, the ML model here is trained on $|\Omega^{\mathcal{N}}| = 6\tilde{}K$ samples with $|\Omega^s| = 10$, whereas the testing samples consider $|\Omega^s| = 1000$. This demonstrates the efficacy of the proposed ADMM-ML approach, which can be

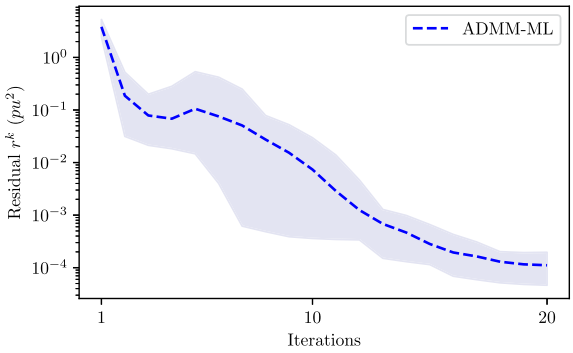

Fig. 8. The residual r^k during testing of the proposed ADMM-ML model over ADMM iterations on the 1354-bus system with $|\Omega^s| = 1000$ and $|\mathcal{T}| = 24$.

TABLE IV
COMPUTATIONAL TIME, OPTIMALITY GAP, AND ACCURACY RESULTS FOR ST-MP-ACOPF ON THE 118-BUS SYSTEM TEST SAMPLES WITH $|\Omega^s| = 100$ UNCERTAINTY SCENARIOS

		Time (sec)	Gap (%)		MAE (pu)	
			Mean	Max	Mean	Max
AC	CO	27716.2	-	-	-	-
	ADMM	457.5	0.0	0.0	0.01	0.1
	ADMM-ML	7.7	1.6	3.6	0.15	0.4
	ADMM-ML-H	53.5	0.7	1.6	0.08	0.2
DC	CO	211.0	17.2	58.2	0.4	0.8
	ADMM	16.0	17.2	58.2	0.4	0.8
	ADMM-ML	0.8	18.0	65.0	0.4	0.8
	ADMM-ML-H	2.6	17.4	60.2	0.4	0.8

trained on samples with a few scenarios, and later be used for samples with many scenarios. This property of the proposed model significantly reduces computations during offline date generation, especially for large-scale system with a high RES share.

G. Application to AC-OPF

Table IV presents the computational time, optimality gap, and accuracy of different models on testing samples of St-MP-ACOPF with $|\Omega^s| = 100$ for the 118-bus system. The accuracy of each model is determined by comparing the active generation against the solution of CO with AC-PF equations. Models relying on DC approximation significantly reduce computational time by orders of magnitude; however, the DC approximation results in high optimality gaps of 17% and MAE errors of 0.4 pu. The proposed AC-ADMM model significantly reduces the optimality gap and error to negligible values, with x60 speed-ups compared to CO. The proposed AC-ADMM-ML model improves computational time by $\times 10^{3.6}$ to only 7 seconds, with a minor optimality gap of 1.6% and an MAE error of 0.1 pu. Furthermore, the proposed AC-ADMM-ML-H improves the optimality gap to 0.7% and the MAE error to 0.1 pu, with $\times 500$ speed-up compared to the CO, offering a balance between time and accuracy. These results underscore the computational efficiency of the ADMM-ML model, especially for large-scale systems with numerous uncertainty scenarios, where its advantages become even more pronounced.

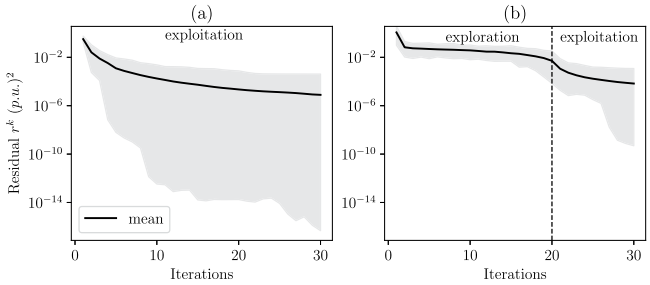

Fig. 9. The primal residual r^k during training data generation: (a) without exploration, (b) with the proposed exploration algorithm.

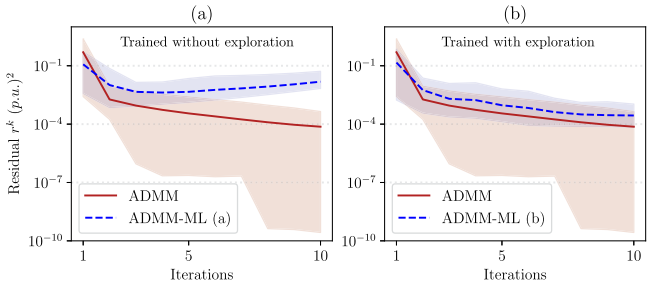

Fig. 10. The primal residual r^k during testing for the ADMM and two ADMM-ML models: (a) trained on the dataset without exploration, (b) trained on the dataset with the proposed exploration algorithm.

H. Training Data Generation

Figs. 9 and 10 investigate the efficacy of the proposed training data generation Algorithm 3 on the IEEE 118-bus system. To this end, two distinct training datasets, labelled (a) and (b) are generated by solving $\Omega^{\mathcal{N}}$ samples of the St-MP-SC-DCOPF with $|\Omega^s| = 10$ scenarios for 30 ADMM iterations. Dataset (a) uses a data generation algorithm without exploration (i.e., $\epsilon_x = 0$), whereas dataset (b) employs the proposed data generation algorithm with an exploration-exploitation trade-off. For dataset (b), the exploration rate is initially set to $\epsilon_x = 1$, and gradually decayed by $\Delta\epsilon_x = 0.05$ over 20 iterations, reaching $\epsilon_x^{\min} = 0$ when only exploitation is performed.

Fig. 9 presents the primal residual r^k over ADMM iterations during training procedure of both datasets. Dataset (a) achieves significantly low values of r^k, while dataset (b) initially explores the input space for the first 20 iterations, resulting in higher values of r^k. Subsequently, full exploration starts after the 20th iteration, leading to lower r^k values.

Furthermore, two RNNs are trained for the ADMM-ML approach: model (a) trained on dataset (a) without exploration, and model (b) trained on dataset (b) with the proposed exploration algorithm. It is noteworthy that, although the RNNs are trained on $|\Omega^{\mathcal{N}}| = 6\tilde{}K$ samples with $|\Omega^s| = 10$ scenarios, they are tested on samples with $|\Omega^s| = 100$ scenarios. Fig. 10 presents the primal residual r^k during testing for the ADMM model and the two ADMM-ML models (a) and (b). The ADMM-ML

model (a) diverges after a few iterations due to insufficient exploration during the training (a). Inaccurate predictions of model (a) during testing lead to λ values that the RNN has not encountered during training, triggering a positive feedback loop with further inaccurate predictions. In contrast, the ADMM-ML model (b) effectively adapts to these testing samples, achieving consensus and mitigating the impact of deviations in its predictions. Furthermore, the mean values of r^k for the ADMM-ML (b) closely follow the mean values of the ADMM. This showcases the adaptability and effectiveness of the proposed exploration-exploitation trade-off in improving the robustness of the ADMM-ML during the testing phase.

I. Discussion

The presented case studies demonstrate promising results for the proposed ML-accelerated ADMM-based model approaching the St-MP-OPF with convex AC constraints. The scenario-based decomposition using ADMM enables scalability to a large number of uncertainty scenarios and shows comparable convergence to the BD baseline. The proposed ADMM-ML approach significantly accelerates computations by $\times 10^{3.6}$ with a minor 1.6% optimality gap, while the proposed hybrid approach (ADMM-ML-H) reduces the optimality gap to 0.7%, offering a balance between speed and optimality. The ADMM-ML speed-up is particularly pronounced for the large-scale 1354-bus system with 1000 uncertainty scenarios, where the ADMM-ML approach provides a solution in just a few minutes. Additionally, as the stochastic scenarios are drawn the same probability distribution, the RNN can be trained on training samples that each have a few uncertainty scenarios (small $|\Omega^s|$), and later be used for testing samples that each have numerous uncertainty scenarios (large $|\Omega^s|$). Moreover, the proposed ϵ-greedy exploration enhances learning performance. Thus, the proposed ADMM-ML approach enables near-real-time solving of the St-MP-OPF with AC constraints on large-scale systems requiring many stochastic scenarios to fully reflect the impact of RES uncertainties. This is a fundamental step forward towards the real-time stochastic scheduling of industry-scale power systems, which is essential for meeting market clearing requirements.

In comparison to other decomposition approaches, the smoother convergence of ADMM without requiring dual values makes it suitable for integration with ML proxies; however, the ADMM-ML approach can be extended to other decomposition techniques with faster convergence rates. In comparison to other ADMM [13], [14], [15], [16], [17], [18] and ML-accelerated ADMM approaches [40], [41], [42], [43], which focus on regionally decentralized OPF problems, our work is tailored to addresses the scalability challenges of uncertainty scenarios in two-stage stochastic operational scheduling. Similarly to non-intrusive ML approaches in [41], [42], [43] where ML warm-starts ADMM, our hybrid ADMM-ML-H approach provides convergence to optimality guarantees and explainability for operators. Additionally, in our ADMM-ML approach, the H&N decisions (applied to the system) are derived through optimization to ensure system operator guarantees, while the uncertain future W&S decisions are approximated by ML. In practice, the proposed approach can integrate with a rolling horizon predictive control (RHPC) framework, where H&N decisions are applied step-by-step as the horizon advances; while also addressing practical aspects such as data integration (combining diverse data sources and preparing scenario-based data), re-training with newly available data, and aligning with market requirements.

However, some limitations of the approach can be noted. The proposed ADMM-ML approach uses RNNs to learn the mapping between net load and generator output for a fixed network topology, whereas network topology is frequently changing in power systems [56]. Topology-aware convolutional [57], [58] and graph neural networks with re-training [59] and transfer learning [60] can be used to optimize the ML model for samples with topological changes. Additionally, this work focuses on day-ahead operational scheduling, but the proposed ADMM-ML approach is general and can be adapted to other problems or datasets, such as those with long-term dependencies, by selecting neural architectures suited to their specific characteristics. Furthermore, although the training data generation is done offline and each training sample has a small $|\Omega^s|$, generating the training dataset still imposes significant computations. Self-supervised approaches may address this limitation. Moreover, future work can explore strategies to reduce ML model training time by improving the efficiency of optimization layers within the training loop.

While ADMM is guaranteed to converge to optimality, the ADMM-ML approach does not offer the same assurance. This challenge may arise during testing due to distribution shifts of the λ values. This challenge is also highlighted as the proposed ϵ-greedy data generation algorithm outperforms a data generation algorithm without exploration. Relying solely on ML predictions also poses security risks [61]. To address these challenges, we propose the hybrid approach, which uses ML predictions to warm-start ADMM, offering the same convergence and feasibility guarantees. Nevertheless, the ADMM-ML approach can be further developed in the future to improve robustness to distribution shifts. For instance, i) clipping mechanisms, such as in proximal policy optimization (PPO) [62], can limit the Lagrangian update and prevent distribution shifts during testing phase, and ii) distribution shift bounds used for imitation learning [63], can learn a policy guaranteed to perform well under its induced distribution of states.

V. Conclusion

As power systems become more complex with an increasing share of renewables, this paper proposes an ML-accelerated ADMM approach to provide fast and high-quality solutions for stochastic multi-period AC OPF (St-MP-OPF) optimization with numerous scenarios. We develop a scenario-based decomposition using ADMM and propose the ADMM-ML approach, where an RNN approximates the ADMM sub-problem optimizations. This approach enables rapid and parallel computations of numerous scenarios by using an RNN to predict W&S decisions, while a master optimization determines the H&N decisions. Additionally, we propose a hybrid approach

that uses ML predictions to warm-start the ADMM, maintaining feasibility and optimality guarantees. The proposed ADMM-ML approach significantly accelerates computations by $\times 10^{3.6}$ with a minor 1.6% optimality gap for the 118-bus system with AC constraints, while the proposed hybrid approach improves the optimality gap to 0.7%, offering a balance between speed and optimality. Case studies on the challenging 1354-bus system with $|\Omega^s| = 1000$ uncertainty scenarios further demonstrate the scalability of the ADMM-ML approach to large-scale systems with numerous scenarios. This work, for the first time, provides a near-real-time solution for the St-MP-OPF on industry-scale power systems with numerous uncertainty scenarios and AC constraints, paving the way towards sustainable power systems with a high share of renewable sources. Future works should explore self-supervised ML approaches, addressing input distribution shifts in ML-based optimization proxies, topology adaptivity, extending to other scheduling problems, other PF formulations, and other distributed optimization methods.

Appendix A
Stochastic AC Multi-Period Optimal Power Flow Formulation

The detailed formulation of the St-MP-ACOPF problem in (1) is presented here. Equations (1a), (1b), (1d) can be written:

$$\min \sum_{s\in\Omega^s}\sum_{t\in\mathcal{T}}\sum_{g\in\Omega^g} \pi_s c_g p_{g,t,s} \tag{11a}$$

$$\underline{p_g} \le p_{g,t,s} \le \overline{p_g} \qquad g\in\Omega^g, t\in\mathcal{T}, s\in\Omega^s \tag{11b}$$

$$\underline{q_g} \le q_{g,t,s} \le \overline{q_g} \qquad g\in\Omega^g, t\in\mathcal{T}, s\in\Omega^s \tag{11c}$$

$$\underline{r_g} \le p_{g,t,s} - p_{g,t-1,s} \le \overline{r_g} \qquad g\in\Omega^g, t\in\mathcal{T}, s\in\Omega^s \tag{11d}$$

$$p_g = p_{g,t_0,s} : \lambda_{g,s} \qquad t=t_0, g\in\Omega^g, s\in\Omega^s \tag{11e}$$

where c_g is the cost coefficient of generators, and p_g, q_g show the active and reactive generation. (11b), (11c) limit active and reactive generation output, and (11d) limits the upward and downward ramping. (11e) is the non-anticipativity constraint in (1d).

The SOCP-relaxed AC-PF constraints (1c) are as follows, which apply to $i\in\mathcal{V}, t\in\mathcal{T}, s\in\Omega^s$:

$$\sum_{g\in\Omega_i^g} p_{g,t,s} - d^p_{i,t,s} = \sum_{ij\in\mathcal{E}_i} (f^p_{ij,t,s} - r_{ij} l_{ij,t,s}) \tag{12a}$$

$$\sum_{g\in\Omega_i^g} q_{g,t,s} - d^q_{i,t,s} = \sum_{ij\in\mathcal{E}_i} (f^q_{ij,t,s} - x_{ij} l_{ij,t,s}) \tag{12b}$$

$$(f^p_{ij,t,s})^2 + (f^q_{ij,t,s})^2 \le v_{i,t,s} l_{ij,t,s} \qquad ij\in\mathcal{E}_i \tag{12c}$$

$$v_{i,t,s} - v_{j,t,s} = 2\left(r_{ij} f^p_{ij,t,s} + x_{ij} f^q_{ij,t,s}\right) - \left(r_{ij}^2 + x_{ij}^2\right) l_{ij,t,s} \qquad ij\in\mathcal{E}_i \tag{12d}$$

$$-\overline{S_{ij}}^2 \le \left(f^p_{ij,t,s}\right)^2 + \left(f^q_{ij,t,s}\right)^2 \le \overline{S_{ij}}^2 \qquad ij\in\mathcal{E}_i \tag{12e}$$

$$-\overline{v_i} \le v_{i,t,s} \le \overline{v_i} \tag{12f}$$

Algorithm 4: Scenario Generation.

```
Input: load consumption historical data h_{t'}   t' ∈ T
Set values for T, Ω^s
Divide h_{t'} monthly → h^m_{t'}   t' ∈ T^m, m ∈ Ω^m
for month m ∈ Ω^m do
    Fit ARIMA model f_θ to h^m_{t'}
    for t' ∈ T^m do
        initial_state = f_θ.state[t']
        d_{t',t,sc} = f_θ.simulate(|T|, |Ω^s|, initial_state,
          anchor = t')
        Filter the scenarios d_{t',t,sc}
Project d_{t',t,s} to the grid → d_{t',i,t,s}
Add random noise to d_{t',i,t,s}
```

where d^p, d^q are nodal active and reactive net loads, v is the nodal squared voltage magnitude, f^p, f^q, l are active, reactive power flows and squared current of lines $ij\in\mathcal{E}$. Equations (12a), (12b) present the nodal active and reactive power balance. The SOCP-relaxed AC-PF equations for line flows are presented in (12c)–(12d), where r, x are the line resistance and reactance. (12e) shows the line thermal limit, and (12f) limits voltage violations. The DC OPF equations can be derived by discarding the reactive power, i.e., $q_g = d^q = f^q = 0$, and assuming $r_{ij} = 0, v_i = 1.0$. The DC PF equation for line flows can be written as $f^p_{ij} = (\theta_i - \theta_j)$, where θ is the nodal voltage angle.

Appendix B
Scenario Generation

In this paper, stochastic hourly scenarios are generated using ARIMA models [3] with the StatsModels package [52]. We use the hourly load demand and wind profile data of the Netherlands from 2015, obtained from publicly available sources [49], [50]. Algorithm 4 presents the scenario generation procedure.

The algorithm begins with the historical net load consumption data $h_{t'}$, where $t'\in T$ represents the time steps corresponding to training samples (i.e., $|T| = 8760$). The time horizon $\mathcal{T} = 24$ hours, and the number of scenarios Ω^s are set. The historical data $h_{t'}$ is divided into monthly segments $h^m_{t'}$, to remove seasonal trends. For each month m, an ARIMA model f_θ is fitted to the monthly data. For each sample t', the *simulate* function of the StatsModels package is then used to generate the required number of scenarios starting from t' to $|\mathcal{T}|$ steps in the future. The scenarios are filtered to remove any outliers. The anchor point then moves to the next sample $t' += 1$. Finally, the generated scenarios for total net load are scaled to the test grid to match the nominal loads, followed by adding random noise to simulate a more realistic grid dataset.

References

[1] L. A. Roald, D. Pozo, A. Papavasiliou, D. K. Molzahn, J. Kazempour, and A. Conejo, "Power systems optimization under uncertainty: A review of methods and applications," *Electr. Power Syst. Res.*, vol. 214, 2023, Art. no. 108725.

[2] M. Usman and F. Capitanescu, "Three solution approaches to stochastic multi-period AC optimal power flow in active distribution systems," *IEEE Trans. Sustain. Energy*, vol. 14, no. 1, pp. 178–192, Jan. 2023.

[3] A. J. Conejo et al. *Decision Making Under Uncertainty in Electricity Markets*. Berlin, Germany: Springer, 2010.

[4] I. Mezghani, S. Misra, and D. Deka, "Stochastic AC optimal power flow: A data-driven approach," in *Proc. 21nd Power Syst. Computation Conf.*, 2020, Art. no. 106567.

[5] H. Li et al., "A review of scenario analysis methods in planning and operation of modern power systems: Methodologies, applications, and challenges," *Electric Power Syst. Res.*, 2022, Art. no. 107722.

[6] H. Sharifzadeh, N. Amjady, and H. Zareipour, "Multi-period stochastic security-constrained OPF considering the uncertainty sources of wind power, load demand and equipment unavailability," *Electric Power Syst. Res.*, vol. 146, pp. 33–42, 2017.

[7] M. I. Alizadeh and F. Capitanescu, "A tractable linearization-based approximated solution methodology to stochastic multi-period AC security-constrained optimal power flow," *IEEE Trans. Power Syst.*, vol. 38, no. 6, pp. 5896–5908, Nov. 2023.

[8] A. Nasri, S. J. Kazempour, A. J. Conejo, and M. Ghandhari, "Network-constrained AC unit commitment under uncertainty: A benders' decomposition approach," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 412–422, Jan. 2016.

[9] J. Soares, B. Canizes, M. A. F. Ghazvini, Z. Vale, and G. K. Venayagamoorthy, "Two-stage stochastic model using benders' decomposition for large-scale energy resource management in smart grids," *IEEE Trans. Ind. Appl.*, vol. 53, no. 6, pp. 5905–5914, Nov./Dec. 2017.

[10] F. García-Muñoz, S. Dávila, and F. Quezada, "A benders decomposition approach for solving a two-stage local energy market problem under uncertainty," *Appl. Energy*, vol. 329, 2023, Art. no. 120226.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[12] J. Kang, A. U. Raghunathan, and S. Di Cairano, "Decomposition via ADMM for scenario-based model predictive control," in *Proc. 2015 Amer. Control Conf.*, 2015, pp. 1246–1251.

[13] C. Zhang and L. Yang, "Distributed AC security-constrained unit commitment for multi-area interconnected power systems," *Electric Power Syst. Res.*, vol. 211, 2022, Art. no. 108197.

[14] V. Rostampour, O. Ter Haar, and T. Keviczky, "Distributed stochastic reserve scheduling in AC power systems with uncertain generation," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1005–1020, Mar. 2019.

[15] N. Meyer-Huebner, M. Suriyah, and T. Leibfried, "Distributed optimal power flow in hybrid AC–DC grids," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 2937–2946, Jul. 2019.

[16] M. Bazrafshan and N. Gatsis, "Decentralized stochastic optimal power flow in radial networks with distributed generation," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 787–801, Mar. 2017.

[17] A. Rajaei, S. Fattaheian-Dehkordi, M. Fotuhi-Firuzabad, M. Moeini-Aghtaie, and M. Lehtonen, "Developing a distributed robust energy management framework for active distribution systems," *IEEE Trans. Sustain. Energy*, vol. 12, no. 4, pp. 1891–1902, Oct. 2021.

[18] V. Dvorkin, P. Van Hentenryck, J. Kazempour, and P. Pinson, "Differentially private distributed optimal power flow," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 2092–2097.

[19] Z. Li et al., "Stochastic accelerated alternating direction method of multipliers for hedging communication noise in combined heat and power dispatch," *CSEE J. Power Energy Syst.*, vol. 9, no. 2, pp. 696–706, 2022.

[20] F. Hasan, A. Kargarian, and A. Mohammadi, "A survey on applications of machine learning for optimal power flow," in *Proc. 2020 IEEE Texas Power Energy Conf.*, 2020, pp. 1–6.

[21] K. Baker, "Learning warm-start points for AC optimal power flow," in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process.*, 2019, pp. 1–6.

[22] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *Proc. 2020 IEEE Int. Conf. Commun. Control Comput. Technol. Smart Grids*, 2020, pp. 1–6.

[23] 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, May 2021.

[24] P. L. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A learning method for optimization with hard constraints," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=V1ZHVxJ6dSS

[25] 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, Jul. 2021.

[26] J. Han et al., "FRMNet: A feasibility restoration mapping deep neural network for AC optimal power flow," *IEEE Trans. Power Syst.*, vol. 39, no. 5, pp. 6566–6577, Sep. 2024.

[27] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," in *Proc. 2019 IEEE Milan PowerTech*, 2019, pp. 1–6.

[28] F. Hasan, A. Kargarian, and J. Mohammadi, "Hybrid learning aided inactive constraints filtering algorithm to enhance AC OPF solution time," *IEEE Trans. Ind. Appl.*, vol. 57, no. 2, pp. 1325–1334, Mar./Apr. 2021.

[29] S. Park and P. Van Hentenryck, "Self-supervised primal-dual learning for constrained optimization," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 4052–4060.

[30] W. Chen, M. Tanneau, and P. V. Hentenryck, "End-to-end feasible optimization proxies for large-scale economic dispatch," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 4723–4734, Mar. 2024.

[31] W. Chen, S. Park, M. Tanneau, and P. Van Hentenryck, "Learning optimization proxies for large-scale security-constrained economic dispatch," in *Proc. 22nd Power Syst. Computation Conf.*, 2022, Art. no. 108566.

[32] M. Mohammadian, K. Baker, M. H. Dinh, and F. Fioretto, "Learning solutions for intertemporal power systems optimization with recurrent neural networks," in *Proc. 17th Int. Conf. Probabilistic Methods Appl. Power Syst.*, 2022, pp. 1–6.

[33] M. Li, W. Wei, Y. Chen, M.-F. Ge, and J. P. Catalao, "Learning the optimal strategy of power system operation with varying renewable generations," *IEEE Trans. Sustain. Energy*, vol. 12, no. 4, pp. 2293–2305, Oct. 2021.

[34] T. Wu, Y.-J. A. Zhang, and S. Wang, "Deep learning to optimize: Security-constrained unit commitment with uncertain wind power generation and bESSs," *IEEE Trans. Sustain. Energy*, vol. 13, no. 1, pp. 231–240, Jan. 2022.

[35] B. N. Giraud, A. Rajaei, and J. L. Cremer, "Constraint-driven deep learning for N-K security constrained optimal power flow," *Electric Power Syst. Res.*, vol. 235, 2024, Art. no. 110692.

[36] R. M. Patel, J. Dumouchelle, E. Khalil, and M. Bodur, "Neur2SP: Neural two-stage stochastic programming," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 23992–24005.

[37] V. Nair, D. Dvijotham, I. Dunning, and O. Vinyals, "Learning fast optimizers for contextual stochastic integer programs," in *Proc. Conf. Uncertainty Artif. Intell.*, 2018, pp. 591–600.

[38] E. Larsen, E. Frejinger, B. Gendron, and A. Lodi, "Fast continuous and integer L-shaped heuristics through supervised learning," *Informs J. Comput.*, vol. 36, no. 1, pp. 203–223, 2024.

[39] H. Jia and S. Shen, "Benders cut classification via support vector machines for solving two-stage stochastic programs," *Informs J. Optim.*, vol. 3, no. 3, pp. 278–297, 2021.

[40] M. Li, S. Kolouri, and J. Mohammadi, "Learning to optimize distributed optimization: ADMM-based DC-OPF case study," in *Proc. 2023 IEEE Power Energy Soc. Gen. Meeting*, 2023, pp. 1–5.

[41] D. Biagioni, P. Graf, X. Zhang, A. S. Zamzam, K. Baker, and J. King, "Learning-accelerated ADMM for distributed DC optimal power flow," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 1–6, 2020.

[42] B. Li and Q. Xu, "A machine learning-assisted distributed optimization method for inverter-based volt-VAR control in active distribution networks," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 2668–2681, Mar. 2024.

[43] T. W. K. Mak, M. Chatzos, M. Tanneau, and P. V. Hentenryck, "Learning regionally decentralized AC optimal power flows with ADMM," *IEEE Trans. Smart Grid*, vol. 14, no. 6, pp. 4863–4876, Nov. 2023.

[44] S. H. Low, "Convex relaxation of optimal power flow—part I: Formulations and equivalence," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 15–27, Mar. 2014.

[45] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2023. [Online]. Available: https://www.gurobi.com

[46] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9562–9574.

[47] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[48] S. Babaeinejadsarookolaee et al., "The power grid library for benchmarking ac optimal power flow algorithms," 2019, *arXiv:1908.02788*.

[49] ENTSO-E, "Power statistics," (n.d.) [Online]. Available: https://www.entsoe.eu/data/power-stats/

[50] I. Gonzalez-Aparicio et al., "Wind hourly generation time series at country, nuts 1, nuts 2 level and bidding zones," 2016. [Online]. Available: http://data.europa.eu/89h/jrc-emhires-wind-generation-time-series
[51] M. Sun, J. Cremer, and G. Strbac, "A novel data-driven scenario generation framework for transmission expansion planning with high renewable energy penetration," *Appl. Energy*, vol. 228, pp. 546–555, 2018.
[52] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," *SciPy*, vol. 7, no. 1, pp. 92–96, 2010.
[53] M. L. Bynum et al., *Pyomo-Optimization Modeling in Python*. Berlin, Germany: Springer, 2021.
[54] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[55] Delft high performance computing centre (DHPC), "DelftBlue supercomputer (Phase 2)," 2024. [Online]. Available: https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2
[56] F. Bellizio, J. L. Cremer, and G. Strbac, "Machine-learned security assessment for changing system topologies," *Int. J. Elect. Power Energy Syst.*, vol. 134, 2022, Art. no. 107380.
[57] T. Falconer and L. Mones, "Leveraging power grid topology in machine learning assisted optimal power flow," *IEEE Trans. Power Syst.*, vol. 38, no. 3, pp. 2234–2246, May 2023.
[58] 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.*, vol. 38, no. 1, pp. 853–860, Jan. 2023.
[59] S. Liu, C. Wu, and H. Zhu, "Topology-aware graph neural networks for learning feasible and adaptive AC-OPF solutions," *IEEE Trans. Power Syst.*, vol. 38, no. 6, pp. 5660–5670, Nov. 2023.
[60] M. Yang et al., "Topology-transferable physics-guided graph neural network for real-time optimal power flow," *IEEE Trans. Ind. Informat.*, vol. 20, no. 9, pp. 10857–10872, Sep. 2024.
[61] J. L. Cremer and G. Strbac, "A machine-learning based probabilistic perspective on dynamic security assessment," *Int. J. Elect. Power Energy Syst.*, vol. 128, 2021, Art. no. 106571.
[62] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[63] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 627–635.

Ali Rajaei (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees from the Sharif University of Technology, Tehran, Iran, in 2017 and 2019, respectively. He is currently working toward the Ph.D. degree with the Delft AI Energy Lab, Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, The Netherlands. His research interests include machine learning and optimization in power systems operation.

Olayiwola Arowolo (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the University of Ibadan, Ibadan, Nigeria, and the M.Sc. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, Pennsylvania, PA, USA. He is currently working toward the Ph.D. degree with the Delft AI Energy Lab, The Netherlands. His current research is focused on using machine learning to predict power system dynamics. His research interests include graph representation learning, generalization in machine learning and data-driven dynamic simulations.

Jochen Lorenz Cremer (Member, IEEE) received the Ph.D. degree from Imperial College London, U.K., in 2020. He was an Associate Professor with the Delft AI Energy Lab, Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, The Netherlands. His research interests include machine learning and mathematical programming applied to the operation and planning of power systems.