High-Fidelity Machine Learning Approximations of Large-Scale Optimal Power Flow

Minas Chatzos, Ferdinando Fioretto, Terrence W.K. Mak, *Member, IEEE,*
and Pascal Van Hentenryck, *Member, IEEE*

***Abstract*—The AC Optimal Power Flow (AC-OPF) is a key building block in many power system applications. It determines generator setpoints at minimal cost that meet the power demands while satisfying the underlying physical and operational constraints. It is non-convex and NP-hard, and computationally challenging for large-scale power systems. Motivated by the increased stochasticity in generation schedules and increasing penetration of renewable sources, this paper explores a deep learning approach to deliver highly efficient and accurate approximations to the AC-OPF. In particular, the paper proposes an integration of deep neural networks and Lagrangian duality to capture the physical and operational constraints. The resulting model, called OPF-DNN, is evaluated on real case studies from the French transmission system, with up to 3,400 buses and 4,500 lines. Computational results show that OPF-DNN produces, in milliseconds, highly accurate AC-OPF approximations whose costs are within 0.01% of optimality and which capture the problem constraints with high fidelity.**

***Index Terms*—Deep Learning; Optimal Power Flow; Lagrangian Dual**

I. Introduction

The *AC Optimal Power Flow* (AC-OPF) determines the most economical generation dispatch that meets the demands while satisfying physical and engineering constraints [1]. It is a non-convex and NP-hard [2] problem, and the building bock of many applications, including security-constrained OPFs (Monticelli et al. [3]), optimal transmission switching [4], capacitor placement [5], expansion planning (Verma et al. [6]), and security-constrained unit commitment [7].

This paper explores a Machine-Learning (ML) approach to the AC-OPF. It is motivated by the recognition that the integration of renewable energy in sub-transmission and distribution systems has introduced significant stochasticity in front and behind the meter, making load profiles much harder to predict and introducing significant variations in load and generation. This uncertainty forces system operators to adjust the generators' setpoints with increasing frequency. However, the resolution frequency to solve OPFs is limited by the AC-OPF runtimes. As a result, system operators typically solve OPF approximations such as the linearized DC model (DC-OPF). While these approximations are more efficient computationally, their solutions may be sub-optimal and induce substantial financial losses [8] or they may fail to satisfy the physical and engineering constraints. An ML approach to the AC-OPF is also promising for expansion planning and other design problems, where plans are typically evaluated by solving a massive number of multi-year Monte-Carlo simulations at 15-minute intervals [9], [10]. For computational reasons, modern approaches recur to the linear DC-OPF approximation and focus only on the scenarios considered most pertinent [9] at the expense of the fidelity of the simulations. Adopting ML to approximate AC-OPF solutions may fundamentally increase the accuracy of these simulations by handling the non-convexities and a larger collection of scenarios.

The ML approach explored in this paper approximates the AC-OPF using a Deep Neural Network (DNN) model. This approximation can be seen as an empirical risk minimization problem. However, the resulting setpoints must also satisfy the physical and engineering constraints that regulate power flows, and these constraints introduce significant difficulties for ML-based approaches, as shown in [11], [12]. To address this challenge, this paper presents a DNN approach (OPF-DNN) to the AC-OPF that borrows ideas from Lagrangian duality. OPF-DNN models the learning task as the Lagrangian dual of the empirical risk minimization problem under the AC-OPF constraints.

The paper makes the following contributions. At the methodological level, it shows how to model the physical and engineering constraints of the AC-OPF during training by combining Lagrangian duality and a DNN architecture. This integration is a key element to produce AC-OPF approximations of high fidelity, that satisfy more than 99% of the constraints and have small violations for the remaining ones. At the conceptual level, this paper shows, for the first time, that DNN architectures can provide high-fidelity approximations for OPF problems with thousands of buses and transmission lines, significantly expanding prior results. The proposed framework is evaluated on real case studies based on the French transmission system and is shown to produce highly accurate AC-OPF approximations in a few milliseconds. The paper is a revised and extended version of our conference paper [13] with significant improvements in the algorithm design, training procedures, and the application of OPF-DNN to real-life case studies that are one order of magnitude larger than those reported in the prior study.

II. Related Work

Within the energy research landscape, DNN architectures have mainly been adopted to predict exogenous factors affecting renewable resources, such as solar or wind. For instance,

Chatzos, Mak and Van Hentenryck are affiliated with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. Fioretto is affiliated with the Electrical Engineering And Computer Science Department, Syracuse University, Syracuse, NY 13244, USA. E-mail: minas@gatech.edu, ffiorett@syr.edu, wmak@gatech.edu, pvh@isye.gatech.edu.

Anwar et al. [14] uses a DNN-based system to predict wind speed and adopt the predictions to schedule generation units ahead of the trading period, and Boukelia et al. [15] studied a DDN framework to predict the electricity costs of solar power plants coupled with a fuel backup system and energy storage. Chatziagorakis [16] studied the control of hybrid renewable energy systems, using recurrent neural networks to forecast weather conditions. Another power system area in which DNNs have been adopted is that of *security assessment*: Ince [17] proposed a convolutional neural network (CNN) model for real-time power system fault classification to detect faulted power system voltage signals. Arteaga [18] proposed a convolutional neural network to identify safe vs. unsafe operating points to reduce the risks of a blackout. Donnot [19] use a ResNet architecture to predict the effect of interventions that reconnect disconnected transmission lines.

The literature on predicting the AC-OPF is much sparser. A survey by Hasan et al. [20] summarize recent developments in the area. Pan et al. [21] explore DNN architectures for predicting DC-OPFs. Deka et al. [22] and Ng et al. [11] use a DNN architecture to learn the set of active constraints. By exploiting the linearity of the DC-OPF problem, once the set of relevant active constraints is identified, an exhaustive search can be used to find a solution that satisfies the active constraints. While this strategy is efficient when the number of active constraints is small, it becomes computationally challenging when the number of active constraints increases. Additionally, this strategy only applies to the DC-OPF. The deep learning approach in [23] predicts voltages and flows for the AC-OPF. However, this approach focuses on specific operational constraints but ignores other physical and engineering constraints, and is only validated on test cases with up to 600 buses. Zamzam and Baker [24] propose another approach, which does not consider problem constraints and is validated only on small test cases. The OPF-DNN approach proposed in this paper is validated on large and realistic transmission grids from the French transmission system, and attains higher prediction quality compared to existing approaches. The approach predicts all the variables and models all physical and engineering constraints through Lagrangian duality, and provides validation results for case studies up to 3400 buses and 4500 lines.

III. Preliminaries

Variables are in lowercase, *constants* are denoted by dotted symbols and *vectors* by bold symbols. The hat notation $\hat{x}$ describes the prediction of a value x and $\| \cdot \|$ is used to denote the L_1-norm. The power flow equations are expressed in terms of complex *power* of the form $S = (p + jq)$, where p and q denote active and reactive power, *admittances* of the form $Y = (g + jb)$, where g and b denote the conductance and susceptance, and *voltages* of the form $V = (v \angle \theta)$, with magnitude v and phase angle θ.

a) Optimal Power Flow: The *Optimal Power Flow (OPF)* determines the least-cost generator dispatch that meets the load (demand) in a power network. A power network is viewed as a graph $(\mathcal{N}, \mathcal{E})$ where the nodes $\mathcal{N}$ represent the set of n *buses* and the edges $\mathcal{E}$ the set of e *transmission lines*. The OPF constraints include physical and engineering constraints, which are captured in the AC-OPF formulation of Model 1. The model uses $\boldsymbol{p}^g$, and $\boldsymbol{p}^d$ to denote, respectively, the vectors of active power generation and load associated with each bus and $\boldsymbol{p}^f$ to describe the vector of active power flows associated with each transmission line. Similar notations are used to denote the vectors of reactive power $\boldsymbol{q}$. Finally, the model uses $\boldsymbol{v}$ and $\boldsymbol{\theta}$ to describe the vectors of voltage magnitudes and angles associated with each bus. The OPF takes as inputs the loads $(\dot{\boldsymbol{p}}^d, \dot{\boldsymbol{q}}^d)$ and the admittance matrix $\dot{\boldsymbol{Y}}$, with entries $\dot{g}_{ij}$ and $\dot{b}_{ij}$ for each line $(ij) \in \mathcal{E}$; It returns the active power vector $\boldsymbol{p}$ of the generators, as well the voltage magnitude $\boldsymbol{v}$ at the generator buses. The objective function (1) captures the cost of the generator dispatch, and is typically expressed as a linear or quadratic function. Constraints (2a) and (2*b*) restrict the voltage magnitude and the phase angle differences within their bounds. Constraints (3*a*) and (3b) enforce the generator active and reactive output limits. Constraints (4) enforce the line flow limits. Constraints (5*a*) and (5b) capture *Ohm's Law*. Finally, constraint (6*a*) and (6b) capture *Kirchhoff's Current Law* enforcing flow conservation.

Model 1 AC Optimal Power Flow (AC-OPF)

$$\mathcal{O}(\dot{\boldsymbol{p}}^d, \dot{\boldsymbol{q}}^d) = \text{argmin}_{\boldsymbol{p}^g, \boldsymbol{v}} \;\; \textstyle\sum_{i \in \mathcal{N}} \text{cost}(p_i^g) \tag{1}$$

subject to:

$$\dot{v}_i^{\min} \leqslant v_i \leqslant \dot{v}_i^{\max} \qquad \forall i \in \mathcal{N} \tag{2a}$$

$$-\dot{\theta}_{ij}^{\Delta} \leqslant \theta_i - \theta_j \leqslant \dot{\theta}_{ij}^{\Delta} \qquad \forall (ij) \in \mathcal{E} \tag{2b}$$

$$\dot{p}_i^{g\,\min} \leqslant p_i^g \leqslant \dot{p}_i^{g\,\max} \qquad \forall i \in \mathcal{N} \tag{3a}$$

$$\dot{q}_i^{g\,\min} \leqslant q_i^g \leqslant \dot{q}_i^{g\,\max} \qquad \forall i \in \mathcal{N} \tag{3b}$$

$$(p_{ij}^f)^2 + (q_{ij}^f)^2 \leqslant \dot{S}_{ij}^{f\,\max} \qquad \forall (ij) \in \mathcal{E} \tag{4}$$

$$p_{ij}^f = \dot{g}_{ij} v_i^2 - v_i v_j (\dot{b}_{ij} \sin(\theta_i - \theta_j) + \dot{g}_{ij} \cos(\theta_i - \theta_j)) \qquad \forall (ij) \in \mathcal{E} \tag{5a}$$

$$q_{ij}^f = -\dot{b}_{ij} v_i^2 - v_i v_j (\dot{g}_{ij} \sin(\theta_i - \theta_j) - \dot{b}_{ij} \cos(\theta_i - \theta_j)) \qquad \forall (ij) \in \mathcal{E} \tag{5b}$$

$$p_i^g - \dot{p}_i^d = \textstyle\sum_{(ij) \in \mathcal{E}} p_{ij}^f \qquad \forall i \in \mathcal{N} \tag{6a}$$

$$q_i^g - \dot{q}_i^d = \textstyle\sum_{(ij) \in \mathcal{E}} q_{ij}^f \qquad \forall i \in \mathcal{N} \tag{6b}$$

output: $(\boldsymbol{p}^g, \boldsymbol{v})$ – The system operational parameters

Model 2 The Load Flow Model

$$\textbf{minimize:} \;\; \|\boldsymbol{p}^g - \hat{\boldsymbol{p}}^g\|^2 + \|\boldsymbol{v} - \hat{\boldsymbol{v}}\|^2 \tag{7}$$

subject to: (2a) – (6b)

b) Deep Learning Models: Supervised learning approximates a complex non-linear mapping from labeled data. Deep neural networks (DNNs) are ML models composed of a sequence of layers, each typically taking as inputs the results of the previous layer [25]. Feed-forward neural networks are basic DNNs where the layers are fully connected and the function connecting the layers is given by

$$\boldsymbol{o} = \pi(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}),$$

where $\mathbf{x} \in \mathbb{R}^n$ is the input vector, $\boldsymbol{o} \in \mathbb{R}^m$ the output vector, $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ a weight matrix, and $\boldsymbol{b} \in \mathbb{R}^m$ a bias vector. The function $\pi(\cdot)$ is non-linear (e.g., a rectified linear unit (ReLU)).

The AC-OPF is an ideal candidate for supervised learning for a number of reasons:

1) **Data availability:** A large amount of historical training data is available in industry.
2) **Input pattern:** The problem inputs are reasonably consistent, lie in specific intervals, and present spatial and temporal correlations between individual components. These properties make it possible to learn AC-OPFs without the need of a dataset that is exponential w.r.t. the input size.
3) **Speed:** Once trained, a DNN can approximate the AC-OPF extremely fast, as only matrix operations are needed and GPU technology can be used. This opens a new realm of applications for AC-OPF that were hitherto limited because of computational challenges.
4) **Difficult instances:** In congested instances, the AC-OPF or its approximations (e.g., the DC model) may need significantly more time to produce a solution. This is not the case for a DNN-based model. Moreover, it has been shown experimentally that AC-OPF approximations result to large optimality gaps in real transmission systems [8]. These gaps do not occur using a DNN approach, as shown in Section IX.

IV. OPF Learning Goals

Given loads $\left(\boldsymbol{p}^d, \boldsymbol{q}^d\right)$, the goal is to predict the control setpoints $(\boldsymbol{p}^g, \boldsymbol{v})$ of the generators. The resulting predictor, called OPF-DNN, learns an OPF mapping $\mathcal{O} : \mathbb{R}^{2n} \to \mathbb{R}^{2m}$, where n is the number of loads and m is the number of generators. The input of the learning task is a dataset $\mathcal{D} = (\mathcal{X}, \mathcal{Y}) = \{(\mathbf{x}_\ell, \mathbf{y}_\ell)\}_{\ell=1}^N$, where $\mathbf{x}_\ell = (\boldsymbol{p}^d, \boldsymbol{q}^d)$ and $\mathbf{y}_\ell = (\boldsymbol{p}^g, \boldsymbol{v})$ represents the ℓ^{th} observation of load demands and generator setpoints that satisfy $\mathbf{y}_\ell = \mathcal{O}(\mathbf{x}_\ell)$. The output is a function $\hat{\mathcal{O}}$ that ideally would be the result of the optimization problem

$$\begin{aligned} \textbf{minimize:} \quad & \sum_{\ell=1}^{N} \mathcal{L}_o(\mathbf{y}_\ell, \hat{\mathcal{O}}(\mathbf{x}_\ell)) \\ \textbf{subject to:} \quad & \mathcal{C}(\mathbf{x}_\ell, \hat{\mathcal{O}}(\mathbf{x}_\ell)) \end{aligned}$$

where the loss function is specified by

$$\mathcal{L}_o(\mathbf{y}, \hat{\mathbf{y}}) = \underbrace{\|\boldsymbol{p}^g - \hat{\boldsymbol{p}}^g\|}_{\mathcal{L}_p(\mathbf{y}, \hat{\mathbf{y}})} + \underbrace{\|\boldsymbol{v} - \hat{\boldsymbol{v}}\|}_{\mathcal{L}_v(\mathbf{y}, \hat{\mathbf{y}})} \tag{8}$$

and $\mathcal{C}(\mathbf{x}, \mathbf{y})$ holds if there exist voltage angles $\boldsymbol{\theta}$ and reactive power $\boldsymbol{q}^g$ that produce a feasible solution to the OPF constraints with $\mathbf{x} = (\boldsymbol{p}^d, \boldsymbol{q}^d)$ and $\mathbf{y} = (\boldsymbol{p}^g, \boldsymbol{v})$.

A baseline DNN model can obtained by ignoring the problem constraints and minimizing the loss function. It will produce an approximation $\hat{\mathcal{O}}$ will typically not satisfy the OPF constraints. A key challenge of the learning task is thus to design a DNN that incorporate these constraints.

V. Modeling the OPF Constraints

To capture OPF constraints, this paper uses a Lagrangian relaxation approach based on constraint violations [26] used in generalized augmented Lagrangian relaxation [27]. The Lagrangian relaxation of an optimization problem

$$\begin{aligned} \textbf{minimize:} \quad & f(\mathbf{x}) \\ \textbf{subject to:} \quad & h(\mathbf{x}) = 0 \\ & g(\mathbf{x}) \leqslant 0 \end{aligned}$$

is given by **minimize**$f(\mathbf{x}) + \lambda_h h(\mathbf{x}) + \lambda_g g(\mathbf{x})$ where λ_h and $\lambda_g \geqslant 0$ are the Lagrangian multipliers. In contrast, the violation-based Lagrangian relaxation is

$$\textbf{minimize:} \quad f(\mathbf{x}) + \lambda_h |h(\mathbf{x})| + \lambda_g \max(0, g(\mathbf{x}))$$

with $\lambda_h, \lambda_g \geqslant 0$. In other words, the traditional Lagrangian relaxation exploits the satisfiability degrees of constraints, whereas the violation-based Lagrangian relaxation is expressed in terms of violation degrees. The satisfiability degree of a constraint measures how well the constraint is satisfied, with negative values representing the slack and positive values representing violations. The violation degree is always non-negative and represents how much the constraint is violated. More formally, the satisfiability degree of a constraint $c : \mathbb{R}^n \to$ Bool is a function $\sigma_c : \mathbb{R}^n \to \mathbb{R}$ such that $c(\mathbf{x})$ holds whenever $\sigma_c(\mathbf{x}) \leqslant 0$. The violation degree of a constraint c is a function $\nu_c : \mathbb{R}^n \to \mathbb{R}^+$ such that $c(\mathbf{x})$ holds whenever $\nu_c(\mathbf{x}) \equiv 0$. For instance, for a linear constraints $c(\mathbf{x})$ of type $A\mathbf{x} \geqslant b$, the *satisfiability degree* is defined as

$$\sigma_c(\mathbf{x}) \equiv \boldsymbol{b} - A\mathbf{x}$$

and the *violation degrees* for inequality and equality constraints are specified by

$$\nu_c^{\geqslant}(\mathbf{x}) = \max\left(0, \sigma_c(\mathbf{x})\right) \qquad \nu_c^{=}(\mathbf{x}) = \left|\sigma_c(\mathbf{x})\right|.$$

Although the resulting term is not differentiable (but admits subgradients), computational experiments indicated that violation degrees are more appropriate for predicting OPFs than satisfiability degrees. Observe also that an augmented Lagrangian method uses both the satisfiability and violation degrees in its objective.

To define the violation degrees of the AC-OPF constraints, the baseline model needs to be extended to predict the reactive power dispatched $\boldsymbol{q}^g$ and the voltage angles $\boldsymbol{\theta}$. Given the predicted values $\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^g$, and $\hat{\boldsymbol{q}}^g$, the constraints can be captured naturally in terms of satisfiability and violation degrees. For instance, the satisfiability degree of a constraint in (4) can be expressed as

$$\sigma_4(\tilde{p}^f_{ij}, \tilde{q}^f_{ij}) = (\tilde{p}^f_{ij})^2 + (\tilde{q}^f_{ij})^2 - \dot{S}^{\max}_{ij}$$

for all $(ij) \in \mathcal{E}$ and the violation degree becomes

$$\nu_4(\tilde{\boldsymbol{p}}^f, \tilde{\boldsymbol{q}}^f) = \frac{1}{e} \sum_{(ij)\in\mathcal{E}} \nu_c^{\geqslant}\left(\sigma_4(\tilde{p}^f_{ij}, \tilde{q}^f_{ij})\right).$$

VI. Objective

The loss function used to train OPF-DNN can now be derived systematically. First, the loss is augmented to consider the predictions of voltage phase angles and the reactive power of generators, since these are required to compute the violation degrees associated with the OPF constraints. The resulting loss function $\mathcal{L}_o(\mathbf{y}, \hat{\mathbf{y}})$ is:

$$\underbrace{\|\boldsymbol{v} - \hat{\boldsymbol{v}}\|}_{\mathcal{L}_v(\mathbf{y}, \hat{\mathbf{y}})} + \underbrace{\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|}_{\mathcal{L}_\theta(\mathbf{y}, \hat{\mathbf{y}})} + \underbrace{\|\boldsymbol{p}^g - \hat{\boldsymbol{p}}^g\|}_{\mathcal{L}_p(\mathbf{y}, \hat{\mathbf{y}})} + \underbrace{\|\boldsymbol{q}^g - \hat{\boldsymbol{q}}^g\|}_{\mathcal{L}_q(\mathbf{y}, \hat{\mathbf{y}})}. \tag{9}$$

It minimizes the mean L_1-error between the optimal voltage and generator settings $\boldsymbol{y}$ and their predictions $\hat{\boldsymbol{y}}$.

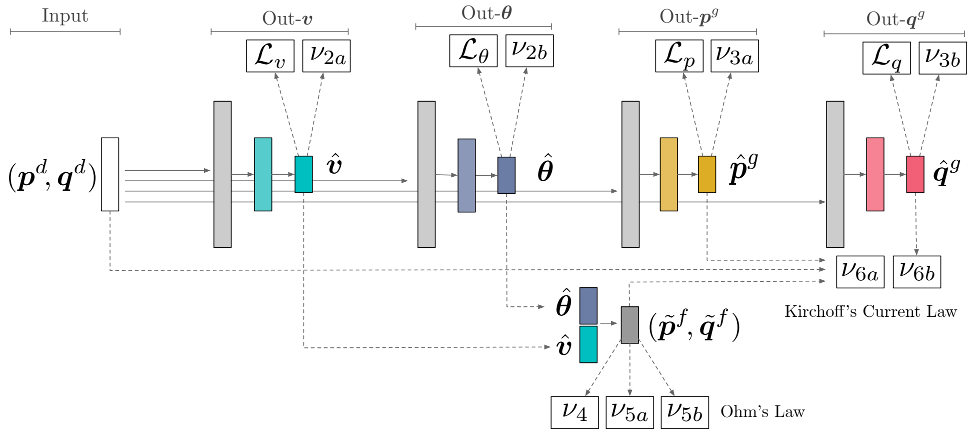

Fig. 1: The OPF-DNN Model: Each layer is fully connected with ReLU activation. White boxes correspond to input tensors, dark-colored boxes correspond to output layers. Loss components and violation degrees are shown as white rectangles.

Moreover, the objective function includes the Lagrangian relaxation that uses the violation degrees of the physical and engineering constraints. Given the set $\mathcal{C}$ of OPF constraints, the associated loss is captured by the expression

$$\mathcal{L}_c(\mathbf{x}, \hat{\mathbf{y}}) = \sum_{c \in \mathcal{C}} \lambda_c \nu_c(\mathbf{x}, \hat{\mathbf{y}}),$$

where $\nu_c(\mathbf{x}, \hat{\mathbf{y}})$ is the violation degree of constraint c for input $\mathbf{x}$ and prediction $\hat{\mathbf{y}}$. The model loss function sums these two terms, i.e.,

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \hat{\mathbf{y}}) = \mathcal{L}_o(\mathbf{y}, \hat{\mathbf{y}}) + \mathcal{L}_c(\mathbf{x}, \hat{\mathbf{y}}).$$

VII. The Network Architecture

The OPF-DNN architecture is outlined in Figure 1. Its inputs are load vectors $(\boldsymbol{p}^d, \boldsymbol{q}^d)$. The network has four basic units composed of fully connected layers with ReLU activations. Each subnetwork predicts a target variable: voltage magnitude $\hat{\boldsymbol{v}}$, phase angle $\hat{\boldsymbol{\theta}}$, active power generation $\hat{\boldsymbol{p}}^g$, and reactive power generation $\hat{\boldsymbol{q}}^g$. The predictions for the voltage magnitude $\hat{\boldsymbol{v}}$ and angle $\hat{\boldsymbol{\theta}}$ are used to compute the load flows $(\tilde{\boldsymbol{p}}^f, \tilde{\boldsymbol{q}}^f)$, as illustrated in the bottom of Figure 1. Predicting the values $(\tilde{\boldsymbol{p}}^f, \tilde{\boldsymbol{q}}^f)$ indirectly is beneficial for two reasons. First, a direct prediction would require additional components in the neural network topology, greatly increasing its complexity, its training time, and memory utilization. These are of critical importance for scaling to large-scale power systems. Second, experimental results indicated that indirect flow predictions may increase the overall accuracy of the predictions.

VIII. Lagrangian Duality

Let $\hat{\mathcal{O}}[\mathbf{w}]$ be the resulting OPF-DNN with weights $\mathbf{w}$ and let $\mathcal{L}[\boldsymbol{\lambda}]$ be the loss function parametrized by the Lagrangian multipliers $\boldsymbol{\lambda} = \{\lambda_c\}_{c \in \mathcal{C}}$. The training aims at finding the weights $\mathbf{w}$ that minimize the loss function for a given set of Lagrangian multipliers, i.e., it computes

$$\mathrm{LR}(\boldsymbol{\lambda}) = \min_{\mathbf{w}} \mathcal{L}[\boldsymbol{\lambda}](\mathbf{x}, \mathbf{y}, \hat{\mathcal{O}}[\mathbf{w}](\mathbf{x})).$$

It remains to determine appropriate Lagrangian multipliers, which can be achieved by solving the Lagrangian dual:

$$\mathrm{LD} = \max_{\boldsymbol{\lambda}} \mathrm{LR}(\boldsymbol{\lambda}).$$

The OPF-DNN training solves the Lagrangian dual through a subgradient method: it computes a sequence of multipliers $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^k, \ldots$ by solving a sequence of trainings $\mathrm{LR}(\boldsymbol{\lambda}^0), \ldots, \mathrm{LR}(\boldsymbol{\lambda}^{k-1}), \ldots$ and adjusting the multipliers using constraint violations, i.e.,

$$\boldsymbol{w}^{k+1} = \operatorname*{argmin}_{\boldsymbol{w}} \mathcal{L}[\boldsymbol{\lambda}^k](\mathbf{x}, \mathbf{y}, \hat{\mathcal{O}}[\mathbf{w}^k](\mathbf{x})) \qquad \text{(L1)}$$

$$\boldsymbol{\lambda}^{k+1} = \Big(\lambda_c^k + \rho\, \nu_c(\mathbf{x}, \hat{\mathcal{O}}[\mathbf{w}^{k+1}](\mathbf{x})) \mid c \in \mathcal{C}\Big). \qquad \text{(L2)}$$

Algorithm 1: *The Training of OPF-DNN.*

```
input: $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ : Training data
       $\alpha, \rho, u_\lambda$: Parameters
1  $\lambda_c^1 \leftarrow \lambda_c^{\text{init}}, \quad \forall c \in \mathcal{C}$
2  for epoch $k = 1, 2, \ldots$ do
3      $\hat{\mathbf{y}} \leftarrow \hat{\mathcal{O}}[\mathbf{w}](\mathbf{x})$
4      $\mathcal{L}_o(\hat{\mathbf{y}}, \mathbf{y}) \leftarrow \frac{1}{b} \sum_{\ell \in [b]} \mathcal{L}_v(\mathbf{y}_\ell, \hat{\mathbf{y}}_\ell) + \mathcal{L}_\theta(\mathbf{y}_\ell, \hat{\mathbf{y}}_\ell) +$
           $\mathcal{L}_p(\mathbf{y}_\ell, \hat{\mathbf{y}}_\ell) + \mathcal{L}_q(\mathbf{y}_\ell, \hat{\mathbf{y}}_\ell)$
5      $\mathcal{L}_c(\mathbf{x}, \hat{\mathbf{y}}) \leftarrow \frac{1}{b} \sum_{\ell \in [b]} \sum_{c \in \mathcal{C}} \lambda_c^k \nu_c(\mathbf{x}_\ell, \hat{\mathbf{y}}_\ell)$
6      $\omega \leftarrow \omega - \alpha \nabla_\omega (\mathcal{L}_o(\hat{\mathbf{y}}, \mathbf{y}) + \mathcal{L}_c(\mathbf{x}, \hat{\mathbf{y}}))$
7      $\boldsymbol{\lambda}^{k+1} \leftarrow \boldsymbol{\lambda}^k$
8      if $k$ mod $u_\lambda = 0$ then
9          for $c \in \mathcal{C}$ do
10             $\lambda_c^{k+1} \leftarrow \lambda_c^k + \rho \nu_c(\mathbf{x}, \hat{\mathbf{y}})$
```

The overall training scheme is presented in Algorithm 1. It takes as input the training dataset $\mathcal{D}$, the optimizer step size $\alpha > 0$, the Lagrangian step size $\rho > 0$, and the parameter u_λ. The Lagrangian multipliers are initialized in line 1. The training is performed for a fixed number of epochs. The Lagrangian multipliers are only updated after every u_λ epochs which allows an accurate approximation of (L1) to be computed before the updating step. After predicting the voltage and generation power quantities (line 3), the objective

Test Case	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	l	g
89_pegase	89	210	35	12
118_ieee	118	186	99	54
300_ieee	300	411	201	69
MSR	403	550	538	115
France_EHV	1,737	2,350	1,731	290
France_Lyon	3,411	4,499	3,273	771

TABLE I: The power system networks.

and constraint losses are computed (lines 4 and 5). The constraint losses use the Lagrangian multipliers $\boldsymbol{\lambda}^k$. The model weights are updated in line 6. Finally, every u_λ epochs, the Lagrangian multipliers are updated (line 10).

IX. Experimental Results

A. Dataset Generation and ML Models

OPF-DNN is evaluated on real-world network instances from RTE, the French transmission operator, as well as from the NESTA library[28] for context. Table I shows the network names and some of their characteristics. The test cases `MSR`, `France EHV`, and `France_Lyon` correspond to configurations of sub-networks of the French Transmission system. `MSR` (Marseille Sous Realtor) is the South-East region of France, which is challenging from an operational standpoint due to voltage issues. `France_EHV` describes the EHV level of the French system, and `France_Lyon` is the EHV level of the French system with a more detailed network for the Lyon region. This last network has 3,411 buses, 4,499 lines or transformers, 3,273 loads, and 771 generators at bus-branch representation for OPF studies.

The datasets used for training and testing OPF-DNN are generated as follows. Initially, for each load (p^d, q^d), lower and upper bounds are randomly generated from two uniform distributions U(0.8ℓ_0, 0.9ℓ_0) and U(1.1ℓ_0, 1.2ℓ_0), where $\ell_0 = ({p_0}^d, {q_0}^d)$ is the nominal load. Then, starting from their lower bounds, the loads are uniformly increased towards their corresponding upper bounds. Due to the different bounds for each load, some loads have larger variations than others. In addition, for each such snapshot, uncorrelated noise is added to each load separately. In some cases, the network becomes too congested and the AC-OPF (Model 1) may become infeasible, in which case the snapshot is discarded. Therefore, each training data point consists of a load snapshot and its AC-OPF solution. The characteristics of the resulting datasets represent realistic load profiles: the uniform load increase and the fixed active/reactive power ratio represent the typical behavior for aggregated demand in a given geographical region. Randomly perturbing each individual load in an uncorrelated fashion would produce unrealistic load profiles. The datasets were generated by solving AC-OPFs with the Julia package PowerModels.jl [29] and the nonlinear solver IPOPT [30] using Intel 2.5 GHz-i7 CPUs and 16GBs of RAM.

B. Machine-Learning Models

The experimental results consider three versions of OPF-DNN (Algorithm 1) specified by three sets of values for the hyperparameters $\boldsymbol{\lambda}_{\text{init}}$ and ρ:

- $\mathcal{M}_B$ ($\boldsymbol{\lambda}^{\text{init}} = 0, \rho = 0$): the base model that ignores the OPF constraints.
- $\mathcal{M}_C$ ($\boldsymbol{\lambda}^{\text{init}} = 1, \rho = 0$): the base model with constraint violations in the objective but no updates of Lagrangian multipliers.
- $\mathcal{M}_C^D$ ($\boldsymbol{\lambda}^{\text{init}} = 0, \rho$): the proposed OPF-DNN.

The ML models were implemented in Python 3.0 using PyTorch [31]. The training was performed using NVidia Tesla V100 GPUs with 16 GB of memory, and the Adam optimizer with learning rate $lr = 10^{-4}$ and a maximum of 50000 epochs. The Lagrangian step size ρ is set to a value in the interval $[10^{-4}, 10^{-2}]$ and the multipliers are updated every $1000 - 5000$ epochs. These hyperparameters were tuned for each benchmark. The evaluation study uses a 80/20 train-test split and reports results only on the test set. For all test cases, the dataset size N was set to 2000.

C. Model Comparison

Figure 2 displays, for each learning algorithm, the convergence of the training procedure for voltage magnitudes, active power generations, and active flows on the topologies of the French transmission system. It reports the maximum average error for buses, generators, and branches over all training instances. The figure highlights that $\mathcal{M}_C$ fails to converge to a satisfying point for the three predictions. This behavior is due to the non-convex part of the loss function stemming from constraints (5a) and (5b). The error for each such constraints is extremely large (10^3 in p.u., on average), leading to exploding gradients and an inefficient training procedure. In contrast, $\mathcal{M}_B$ and $\mathcal{M}_C^D$ exhibit similar convergence profiles for voltage and active power values. However, $\mathcal{M}_C^D$ enjoys stronger convergence behaviors for branch flows, which are indirectly predicted through Ohm's law. The more accurate prediction of branch flows is crucial to satisfy the AC-OPF constraints.

D. Prediction Errors

The predictive accuracy of OPF-DNN is assessed by comparing the directly predicted values $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^g, \hat{\boldsymbol{q}}^g)$, as well as the indirectly predicted value $\hat{\boldsymbol{p}}^f$, to their respective ground truth values $(\boldsymbol{v}, \boldsymbol{\theta}, \boldsymbol{p}^g, \boldsymbol{q}^g, \boldsymbol{p}^f)$. Figure 3 depicts the absolute errors for voltage magnitude $\boldsymbol{v}$, active generation $\boldsymbol{p}$, and active flow $\boldsymbol{p^f}$ for the French benchmarks. To communicate the results more clearly, the buses, generators, and branches are *sorted in ascending orders* of their mean ground-truth values. The figure reports the mean L_1-error over the test cases as well as intervals that capture 95% of the test cases errors. For the voltage magnitudes, the mean error is under 0.1% of the nominal value. For the active power, the mean error is extremely low and, in almost all cases, well below 1% of the nominal value. The intervals indicate that there are some cases with larger errors. These cases are instances with large loads, which induce significant changes in solution values. Adding more instances in these operating regions would improve results. Finally, OPF-DNN yields extremely low errors (well under 1%) for the indirectly predicted active flows for lines carrying large flows (typically 90% of the lines). For lines with small flows, although the absolute error is still very low,

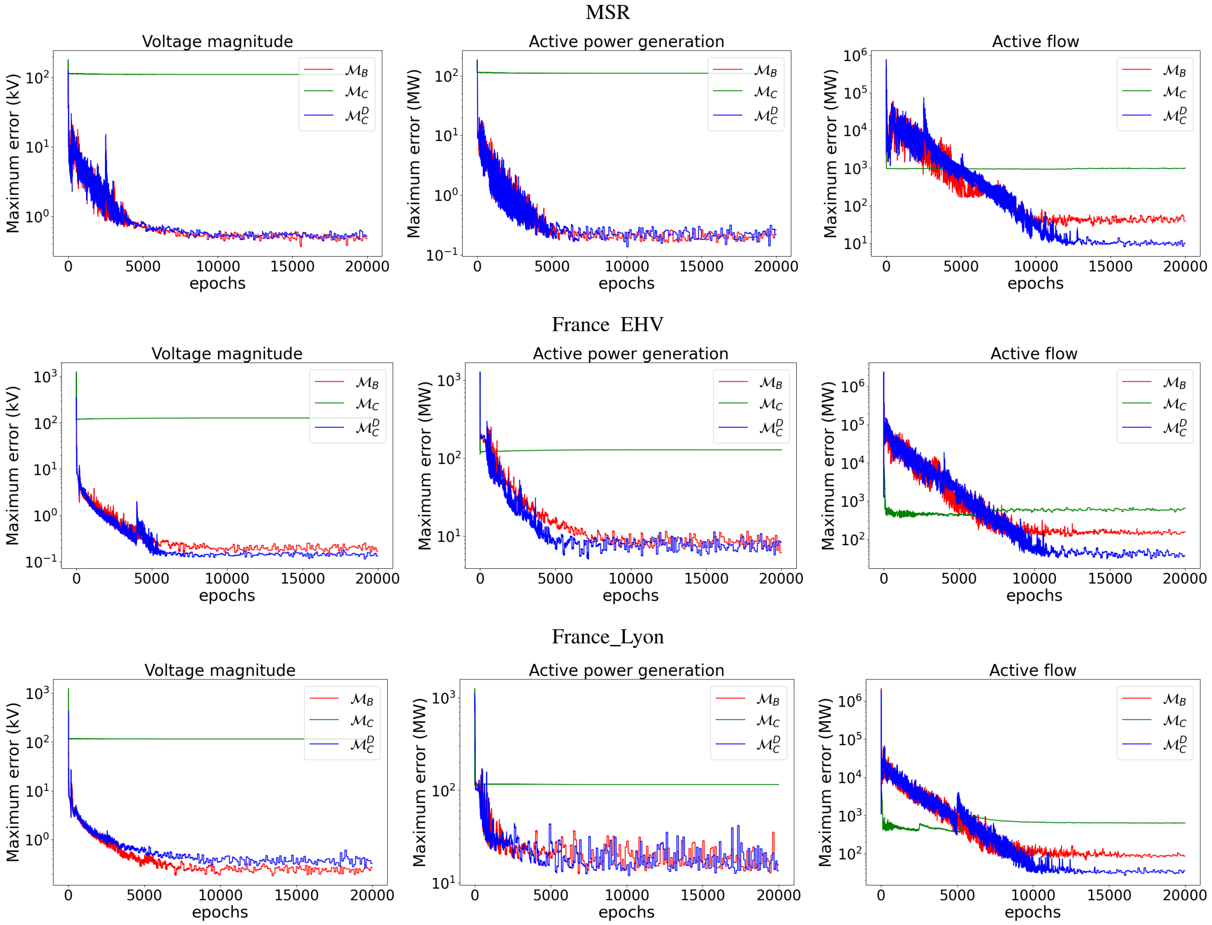

Fig. 2: Convergence analysis for $\mathcal{M}_B, \mathcal{M}_C, \mathcal{M}_C^D$ on the French Transmission System topologies.

it is larger in percentage, as the training minimizes the error in absolute scale (but not in relative scale). Branch errors can be strongly correlated with their impedance values, as impedance values directly affect the power flows (indirect) predictions. Table II reports comprehensive results on the absolute errors. For each quantity $\mathbf{x}$, it reports the value $\frac{1}{mN}||\mathbf{x} - \hat{\mathbf{x}}||_1$, where m is the number of buses, generators, or branches. The results are consistent across all the benchmarks and demonstrate the high accuracy of the OPF-DNN predictions.

Test Case	$\hat{\boldsymbol{v}}$	$\hat{\boldsymbol{\theta}}$	$\hat{\boldsymbol{p}}^g$	$\hat{\boldsymbol{q}}^g$	$\hat{\boldsymbol{p}}^f$	$\hat{\boldsymbol{q}}^f$
89_pegase	0.025	0.005	0.585	0.425	0.152	0.353
118_ieee	0.009	0.004	0.013	0.016	0.035	0.059
300_ieee	0.006	0.020	0.183	0.068	0.152	0.101
MSR	0.002	0.001	0.007	0.005	0.039	0.066
France_EHV	0.012	0.008	0.032	0.104	0.146	0.099
France_Lyon	0.025	0.005	0.098	0.093	0.087	0.153

TABLE II: Mean prediction errors in kV, degrees, MW, MVA for $\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, (\hat{\boldsymbol{p}}^g, \hat{\boldsymbol{p}}^f), (\hat{\boldsymbol{q}}^g, \hat{\boldsymbol{q}}^f)$, respectively.

Test Case	(2a)	(3a)	(3b)	(4)
89_pegase	99.9	99.8	98.8	99.9
118_ieee	99.9	99.7	99.8	100.0
300_ieee	98.7	98.3	98.9	99.7
MSR	99.8	99.6	99.9	100.0
France_EHV	99.7	99.2	99.7	100.0
France_Lyon	99.8	99.8	99.9	100.0

TABLE III: Percentage of bound constraints satisfied.

E. Feasibility Errors

It is important to study how the predictions $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^g, \hat{\boldsymbol{q}}^g, \hat{\boldsymbol{p}}^f)$ sastify the OPF constraints. There are no violations due to constraints (5a) and (5b) since these equations (Ohm's law) are used to compute the predicted flow values $(\hat{\boldsymbol{p}}^f, \hat{\boldsymbol{q}}^f)$. Table III reports results on the satisfaction of constraints (2a), (3a), (3b), and (4) in percentage. Table IV complements these results and reports the mean violations for the violated constraints in Table III. Power balance constraints (6a) and (6b) are also included. The mean violations are small, indicating the strong accuracy of the OPF-DNN predictions. Moreover, the violation pattern over the buses for constraints (6a) displays the same behavior as that observed for the errors on the branch flows: the errors tend to be very small for buses adjacent to branches carrying large flows. Figure 4 visualizes this information for a representative case study: it reports the mean absolute violations of constraints (6a) for the MSR test case. Observe how small the constraint violations are in absolute terms.

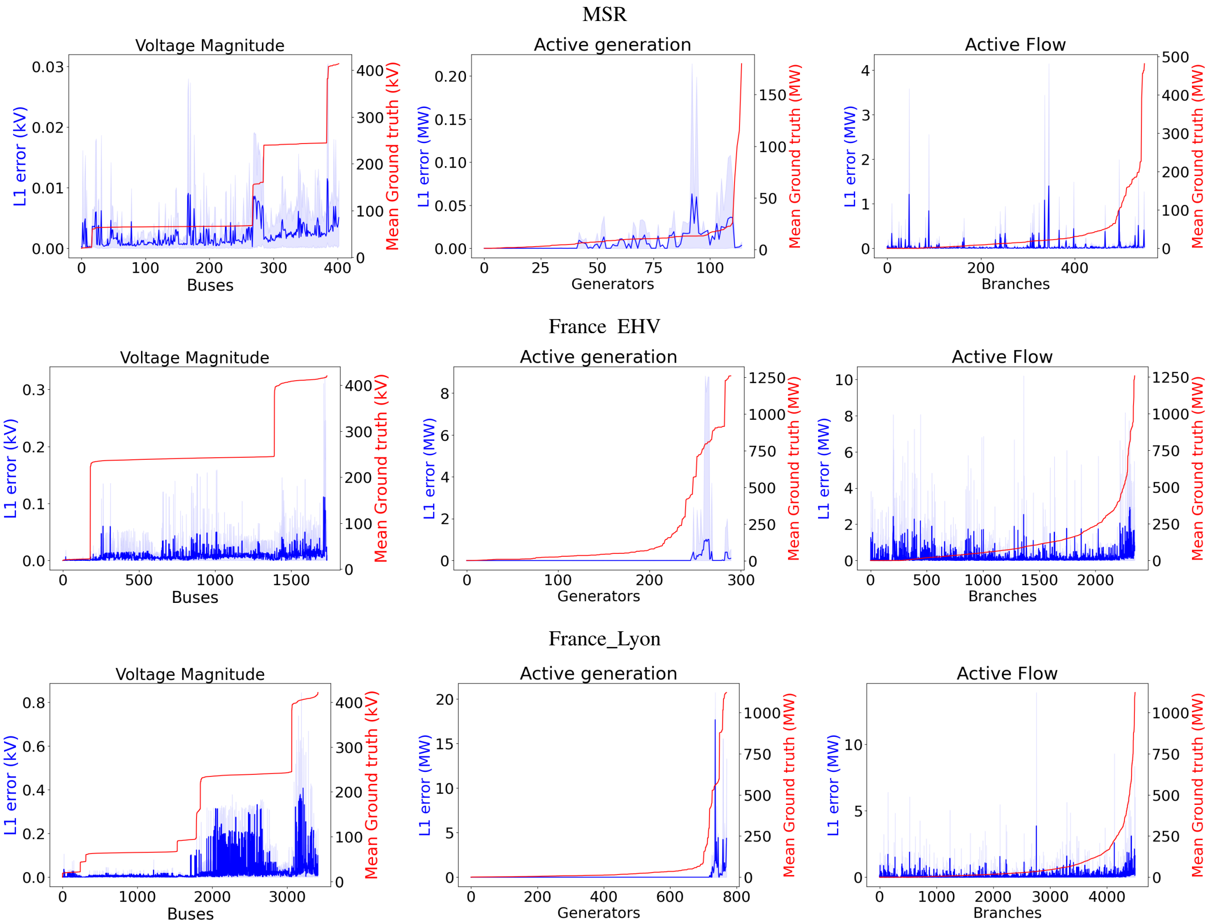

Fig. 3: Prediction Errors over Buses, Generators, and Branches. The Ground Truth and Error Graphs are in Different Scales.

Test Case	(2a)	(3a)	(3b)	(4)	(6a)	(6b)
89_pegase	<0.01	4.52	0.45	1.39	0.25	1.01
118_ieee	<0.01	1.83	0.18	-	0.04	0.10
300_ieee	<0.01	0.43	0.20	0.21	0.16	0.17
MSR	<0.01	0.62	0.12	-	0.09	0.15
France_EHV	0.01	2.94	0.79	-	0.24	0.14
France_Lyon	<0.01	1.30	0.50	-	0.16	0.16

TABLE IV: Mean violation for violated AC-OPF constraints. The violation is expressed in kV, MW and MVA for constraints 2a, (3a, 6a) and (3b, 4, 6b), respectively.

F. Objective Values

Table V compares the objective values from the AC-OPF optimizations with those obtained by the OPF-DNN predictions. In addition, the table also compares the costs obtained from the load flow model (2) seeded with the OPF-DNN predictions. In particular, Table V reports the average relative objective error $\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}}|1-\frac{\hat{c}_i}{c_i}| \times 100$, where c_i is the AC-OPF objective value and $\hat{c}_i$ is either the objective value produced by OPF-DNN or by the load flow solution. The results demonstrate that the OPF-DNN produces objective values closely approximate the AC-OPF costs. The closest feasible solutions obtained by solving load flow are also extremely accurate.

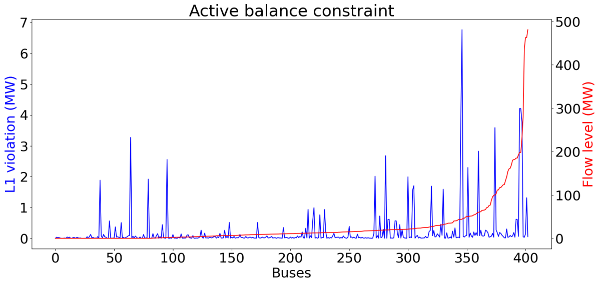

Fig. 4: Mean absolute violations of the active power balance constraint (6a) for the MSR test case across all buses. The ground truth and the violations are in different scales.

Test Case	OPF-DNN	Load Flow
89_pegase	$25 \cdot 10^{-3}$	$0.13 \cdot 10^{-3}$
118_ieee	$16 \cdot 10^{-3}$	$0.18 \cdot 10^{-3}$
300_ieee	$10 \cdot 10^{-3}$	$0.42 \cdot 10^{-3}$
MSR	$6 \cdot 10^{-3}$	$0.21 \cdot 10^{-3}$
France_EHV	$5 \cdot 10^{-3}$	$0.16 \cdot 10^{-3}$
France_Lyon	$25 \cdot 10^{-3}$	$2.75 \cdot 10^{-3}$

TABLE V: Differences (in %) between the OPF, the OPF-DNN objective values and the associated load flow objective.

	Train time	Predict time	Train Mem.	AC-OPF
Test Case	(min)	(sec)	(GB)	(sec)
89_pegase	48	0.0013	1.1	0.2
118_ieee	51	0.0013	1.1	0.2
300_ieee	54	0.0014	1.2	1.9
MSR	59	0.0016	1.5	2.2
France_EHV	142	0.0016	4.6	4.2
France_Lyon	444	0.0020	13.9	47.9

TABLE VI: Training and inference computational costs.

G. Time and Memory

Finally, table VI reports the average computation times and the GPU memory utilization for the OPF-DNN training across all benchmarks. In addition, it reports the time needed to produce a prediction for a given instance. To give the proper context, the table also reports the average time to solve the AC-OPFs. For the case studies, the training times are reasonable and the prediction times are up to 4 orders of magnitude faster than the AC-OPF runtimes. *OPF-DNN, thus, provides an appealing tradeoff between accuracy and efficiency.*

X. Conclusions

The paper proposed OPF-DNN, a deep learning approach to produce high-fidelity approximations of large-scale optimal power flows in milliseconds. The approach combines deep learning and Lagrangian duality to model the physical and engineering constraints. Computational experiments on real-life case studies based on the French transmission system (up to 3,400 buses and 4,500 lines) demonstrate the potential of the approach. These results open a new avenue in approximating the AC-OPF, a key building block in many power system applications, including expansion planning and security assessment studies. Current work aims at scaling the approach to even larger systems with over 10^4 buses, which raise challenges in GPU memory and training time.

References

[1] B. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1248–1259, Nov 1990.

[2] K. Lehmann, A. Grastien, and P. Van Hentenryck, "Ac-feasibility on tree networks is np-hard," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 798–801, 2016.

[3] A. Monticelli, M. Pereira, and S. Granville, "Security-constrained optimal power flow with post-contingency corrective rescheduling," *IEEE Transactions on Power Systems*, vol. 2, no. 1, pp. 175–180, 1987.

[4] E. B. Fisher, R. P. O'Neill, and M. C. Ferris, "Optimal transmission switching," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1346–1355, Aug 2008.

[5] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 725–734, Jan 1989.

[6] Niharika, S. Verma, and V. Mukherjee, "Transmission expansion planning: A review," in *International Conference on Energy Efficient Technologies for Sustainability*, April 2016, pp. 350–355.

[7] J. Wang, M. Shahidehpour, and Z. Li, "Security-constrained unit commitment with volatile wind power generation," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1319–1327, 2008.

[8] T. W. K. Mak, L. Shi, and P. V. Hentenryck, "Phase transitions for optimality gaps in optimal power flows a study on the french transmission network," 2018.

[9] C. Pache, J. Maeght, B. Seguinot, A. Zani, S. Lumbreras, A. Ramos, S. Agapoff, L. Warland, L. Rouco, and P. Panciatici, "Enhanced pan-european transmission planning methodology," in *IEEE Power Energy Society General Meeting*, July 2015.

[10] Deutche-Energue-Agentur, "The e-highway2050 project," http://www.e-highway2050.eu, 2019, accessed: 2019-11-19.

[11] Y. Ng, S. Misra, L. Roald, and S. Backhaus, "Statistical learning for DC optimal power flow," in *Power Systems Computation Conference*, 2018.

[12] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," in *2019 IEEE Milan PowerTech*, June 2019.

[13] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 630–637.

[14] M. B. Anwar, M. S. El Moursi, and W. Xiao, "Novel power smoothing and generation scheduling strategies for a hybrid wind and marine current turbine system," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1315–1326, 2016.

[15] T. Boukelia, O. Arslan, and M. Mecibah, "Potential assessment of a parabolic trough solar thermal power plant considering hourly analysis: ANN-based approach," *Renewable Energy*, vol. 105, pp. 324 – 333, 2017.

[16] P. Chatziagorakis, C. Ziogou, C. Elmasides, G. C. Sirakoulis, I. Karafyllidis, I. Andreadis, N. Georgoulas, D. Giaouris, A. I. Papadopoulos, D. Ipsakis, S. Papadopoulou, P. Seferlis, F. Stergiopoulos, and S. Voutetakis, "Enhancement of hybrid renewable energy systems control with neural networks applied to weather forecasting: the case of Olvio," *Neural Computing and Applications*, vol. 27, no. 5, pp. 1093–1118, Jul 2016.

[17] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[18] J. H. Arteaga, F. Hancharou, F. Thams, and S. Chatzivasileiadis, "Deep learning for power system security assessment," in *2019 IEEE Milan PowerTech*, June 2019.

[19] B. Donnot, B. Donon, I. Guyon, Z. Liu, A. Marot, P. Panciatici, and M. Schoenauer, "LEAP nets for power grid perturbations," in *European Symposium on Artificial Neural Networks*, April 2019.

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

[21] X. Pan, T. Zhao, and M. Chen, "Deepopf: Deep neural network for dc optimal power flow," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019, pp. 1–6.

[22] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," in *2019 IEEE Milan PowerTech*, June 2019.

[23] Y. Yang, Z. Yang, J. Yu, B. Zhang, Y. Zhang, and H. Yu, "Fast calculation of probabilistic power flow: A model-based deep learning approach," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2235–2244, 2020.

[24] A. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," *CoRR*, vol. abs/1910.01213, 2019. [Online]. Available: http://arxiv.org/abs/1910.01213

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[26] D. Fontaine, M. Laurent, and P. Van Hentenryck, "Constraint-based lagrangian relaxation," in *Principles and Practice of Constraint Programming*, 2014, pp. 324–339.

[27] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[28] C. Coffrin, D. Gordon, and P. Scott, "NESTA, the NICTA energy system test case archive," *CoRR*, vol. abs/1411.0359, 2014. [Online]. Available: http://arxiv.org/abs/1411.0359

[29] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "Powermodels.jl: An open-source framework for exploring power flow formulations," in *PSCC*, June 2018.

[30] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.