# Constrained-Informed Optimal Power Flow

### Preliminary Notes and Results

Ferdinando Fioretto
Syracuse Universty
ffiorett@syr.edu

## 1 Introduction

The *Optimal Power Flow* (OPF) problem determines the generator dispatch of minimal cost that meets the demands while satisfying the physical and engineering constraints of the power system [6]. The OPF (aka AC-OPF) is a non-convex non-linear optimization problem and the building bock of many applications, including security-constrained OPFs [18], optimal transmission switching [12], capacitor placement [3], expansion planning [20], and security-constrained unit commitment [26].

Typically, generation schedules are updated in intervals of 5 minutes [24], possibly using a solution to the OPF solved in the previous step as a starting point. In recent years, 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 in order to serve the power demand while ensuring stable network operations. However, the resolution frequency to solve OPFs is limited by their computational complexity. To address this issue, system operators typically solve OPF approximations such as the linear DC model (DC-OPF). While these approximations are more efficient computationally, their solution may be sub-optimal and induce substantial economical losses, or they may fail to satisfy the physical and engineering constraints.

Similar issues also arise in expansion planning and other configuration problems, where plans are evaluated by solving a massive number of multi-year Monte-Carlo simulations at 15-minute intervals [21; 10]. Additionally, the stochasticity introduced by renewable energy sources further increases the number of scenarios to consider. Therefore, modern approaches recur to the linear DC-OPF approximation and focus only on the scenarios considered most pertinent [21] at the expense of the fidelity of the simulations.

To address these challenges, this paper studies how to approximate OPFs using a Deep Neural Network (DNN) approach. The main goal of the OPF is to find generator setpoints, i.e., the amount of real power and the voltage magnitude for each generator. Approximating the OPF using DNNs can thus 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 machine learning-based approaches, as shown in [19; 9]. To address these difficulties, this paper presents a DNN approach to the OPF (OPF-DNN) that borrows ideas from Lagrangian duality and models the learning task as the Lagrangian dual of the empirical risk minimization problem under the OPF constraints.

In addition to the difficulty of handling hard constraints, a second challenge is the need to generate large and high-fidelity datasets, which consist of solutions to the OPF problems. To address this second challenge, this paper proposes to train the proposed model in an unsupervised manner, exploiting the optimization problem objective.

OPF-DNN is evaluated on realistic medium-sized power system benchmarks: The computational results show significant improvements in accuracy and efficiency compared to the ubiquitous DC model. In particular, OPF-DNN provides accuracy improvements of up to two orders of magnitude and efficiency speedups of several orders of magnitude. *These results may open new avenues for power system analyses and operations under significant penetration of renewable energy.*

## 2 Preliminaries

The paper uses the following notations: *Variables* are denoted by calligraph lowercase symbols, *constants* by dotted symbols, and *vectors* by bold symbols. The hat notation $\hat{x}$ describes the prediction of a value $x$ and $\|\cdot\|$ denotes the $L2$-norm. The power flow equations are expressed in terms of complex *powers* of the form

**Model 1** AC Optimal Power Flow (AC-OPF)

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

**subject to:**

$$\dot{v}_i^{\min} \leq v_i \leq \dot{v}_i^{\max} \qquad \forall i \in \mathcal{N} \quad (2\text{a})$$

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

$$\dot{p}_i^{g\,\min} \leq p_i^g \leq \dot{p}_i^{g\,\max} \qquad \forall i \in \mathcal{N} \quad (\bar{3}a)$$

$$\dot{q}_i^{g\,\min} \leq q_i^g \leq \dot{q}_i^{g\,\max} \qquad \forall i \in \mathcal{N} \quad (3\text{b})$$

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

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

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

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

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

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

**Model 2** The Load Flow Model

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

$$\textbf{subject to:} \;\; (2\text{a}) - (6\text{b})$$

$S = (p + jq)$, where $p$ and $q$ denote active and reactive powers, *admittance* 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 $\theta$.

## 2.1 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 $\mathcal{E}$ represent 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 magnitude 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 quadratic function. Constraints (2a) and $(\bar{2}b)$ restrict the voltage magnitudes and the phase angle differences within their bounds. Constraints $(\bar{3}a)$ and (3b) enforce the generator active and reactive output limits. Constraints $(\bar{4})$ enforce the line flow limits. Constraints $(\bar{5}a)$ and (5b) capture *Ohm's Law*. Finally, Constraint $(\bar{6}a)$ and (6b) capture *Kirchhoff's Current Law* enforcing flow conservation.

**The DC Relaxation** The DC model is a ubiquitous linear approximation to the OPF [27]. It ignores reactive power and assumes that the voltage magnitudes are at their nominal values (1.0 in per unit notation). The model uses only the barred constraints in Model 1. Constrains $(\bar{4})$ considers only the active flows and hence can be trivially linearized and Constraints $(\bar{5}a)$ becomes $p_{ij}^f = -\dot{b}_{ij}(\theta_i - \theta_j)$. The quadratic objective is also replaced by a piecewise linear function. Being an approximation, a DC solution $\hat{\boldsymbol{p}}^g$ may not satisfy the AC model constraints. As result, prior to being deployed, one typically solves a *load flow optimization*, described in Model 2. It is a least squares minimization problem that finds the closest AC-feasible solution to the approximated one.

### 2.2 Deep Learning Models

Supervised Deep Learning (SDL) can be viewed as the task of approximating a complex non-linear mapping from labeled data. Deep Neural Networks (DNNs) are deep learning architectures composed of a sequence of layers, each typically taking as inputs the results of the previous layer [16]. Feed-forward neural networks are basic DNNs where the layers are fully connected and the function connecting the layer is given by

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

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

## 3 OPF Learning Goals

The goal of this paper is to learn the OPF mapping $\mathcal{O} : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$: Given the loads $\left(\boldsymbol{p}^d, \boldsymbol{q}^d\right)$, predict the setpoints $(\boldsymbol{p}^g, \boldsymbol{v})$ of the generators, i.e., their active power and the voltage magnitude at their buses. The input of the learning task is a dataset $\mathcal{D} = \{(\boldsymbol{x}_\ell)\}_{\ell=1}^N$, where $\boldsymbol{x}_\ell = (\boldsymbol{p}^d, \boldsymbol{q}^d)$ represent the $\ell^{th}$ observation of load demands and generator setpoints. The output is a function $\hat{\mathcal{O}}$ that ideally would be the result of the following optimization problem

$$\textbf{minimize:} \sum_{\ell=1}^{N} \text{cost}(\hat{\mathcal{O}}(\boldsymbol{x}_\ell)) \qquad \textbf{subject to:} \ \mathcal{C}(\boldsymbol{x}_\ell, \hat{\mathcal{O}}(\boldsymbol{x}_\ell))$$

where $\text{cost}(\hat{\mathcal{O}}(\boldsymbol{x}_\ell))$ captures the AC OPF objective cost component applied to the predictions $\hat{\mathcal{O}}(\boldsymbol{x}_\ell)$ and $\mathcal{C}(\boldsymbol{x}, \boldsymbol{y})$ holds if there exist voltage angles $\boldsymbol{\theta}$ and reactive power generated $\boldsymbol{q}^g$ that produce a feasible solution to the OPF constraints with $\boldsymbol{x} = (\boldsymbol{p}^d, \boldsymbol{q}^d)$ and $\boldsymbol{y} = (\boldsymbol{p}^g, \boldsymbol{v})$.

One of the key difficulties of this learning task is the presence of the complex nonlinear feasibility constraints in the OPF. The approximation $\hat{\mathcal{O}}$ will typically not satisfy the problem constraints. As a result, like in the case of the DC model discussed earlier, the validation of the learning task uses a load flow computation that, given a prediction $\hat{\boldsymbol{y}} = \hat{\mathcal{O}}(\boldsymbol{x}_\ell)$, computes the closest feasible generator setpoints.

## 4 Baseline Deep Learning Model

The baseline model for this paper assumes that function $\hat{\mathcal{O}}$ is given by a feed-forward neural network. While this baseline model is often accurate for many regression problems, the experimental results show that it has low fidelity for complex AC-OPF tasks. More precisely, a load flow computation on the predictions of this baseline model to restore feasibility produces generator setpoints with substantial errors. The rest of the paper shows how to improve the accuracy of the model by exploiting the problem structure.

## 5 Capturing the OPF Constraints

To capture the OPF constraints, this paper uses a Lagrangian relaxation approach based on constraint violations [13] used in generalized augmented Lagrangian relaxation [14]. The Lagrangian relaxation of an optimization problem

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

is given by

$$\textbf{minimize:} \ f(\boldsymbol{x}) + \lambda_h h(\boldsymbol{x}) + \lambda_g g(\boldsymbol{x})$$

where $\lambda_h$ and $\lambda_g \geq 0$ are the Lagrangian multipliers. In contrast, the violation-based Lagrangian relaxation is

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

with $\lambda_h, \lambda_g \geq 0$. In other words, the traditional Lagrangian relaxation exploits the satisfiability degrees of constraints, while 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, while the violation degree is always non-negative and represents how much the constraint is violated. More formally, the satisfiability degree of a constraint $c : \mathbb{R}^n \to \mathit{Bool}$ is a function $\sigma_c : \mathbb{R}^n \to \mathbb{R}$ such that $\sigma_c(\boldsymbol{x}) \leq 0 \equiv c(\boldsymbol{x})$. The violation degree of a constraint $c : \mathbb{R}^n \to \mathit{Bool}$ is a function $\nu_c : \mathbb{R}^n \to \mathbb{R}^+$ such that $\sigma_c(\boldsymbol{x}) \equiv 0 \equiv c(\boldsymbol{x})$. For instance, for a linear constraints $c(\boldsymbol{x})$ of type $A\boldsymbol{x} \geq b$, the *satisfiability degree* is defined as

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

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

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

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 extended to predict the reactive power dispatched $\boldsymbol{q}^g$ and the voltage angles $\boldsymbol{\theta}$ of the power network. Given the predicted values $\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^g$, and $\hat{\boldsymbol{q}}^g$, the satisfiability degree of the OPF constraints can be expressed as: This section extends the homonym section of the main paper by reporting the complete set of satisfiability and violation degrees of the OPF constraints.

Given the predicted values $\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^g$, and $\hat{\boldsymbol{q}}^g$, the satisfiability degree of the OPF constraints are expressed as follows:

$$\begin{aligned}
\sigma^L_{2a}(\hat{v}_i) &= (\dot{v}_i^{\min} - \hat{v}_i) && \forall i \in \mathcal{N}\\
\sigma^R_{2a}(\hat{v}_i) &= (\hat{v}_i - \dot{v}_i^{\max}) && \forall i \in \mathcal{N}\\
\sigma^L_{2b}(\hat{\theta}_{ij}) &= ((\hat{\theta}_j - \hat{\theta}_i) - \dot{\theta}^{\Delta}_{ij}) && \forall (ij) \in \mathcal{E}\\
\sigma^R_{2b}(\hat{\theta}_{ij}) &= ((\hat{\theta}_i - \hat{\theta}_j) - \dot{\theta}^{\Delta}_{ij}) && \forall (ij) \in \mathcal{E}\\
\sigma^L_{3a}(\hat{p}_i{}^g) &= \dot{p}_i^{g\,\min} - \hat{p}^g_i && \forall i \in \mathcal{N}\\
\sigma^R_{3a}(\hat{p}_i{}^g) &= \hat{p}^g_i - \dot{p}_i^{\max} && \forall i \in \mathcal{N}\\
\sigma^L_{3b}(\hat{q}_i{}^g) &= \dot{q}_i^{g\,\min} - \hat{q}^g_i && \forall i \in \mathcal{N}\\
\sigma^R_{3b}(\hat{q}_i{}^g) &= \hat{q}^g_i - \dot{q}_i^{\max} && \forall i \in \mathcal{N}\\
\sigma_4(\tilde{p}^f_{ij}, \tilde{q}^f_{ij}) &= (\tilde{p}^f_{ij})^2 + (\tilde{q}^f_{ij})^2 - \dot{S}^{\max}_{ij} && \forall (ij) \in \mathcal{E}\\
\sigma_{5a}(\tilde{p}^f_{ij}, p^f_{ij}) &= \tilde{p}^f_{ij} - p^f_{ij} && \forall (ij) \in \mathcal{E}\\
\sigma_{5b}(\tilde{q}^f_{ij}, q^f_{ij}) &= \tilde{q}^f_{ij} - q^f_{ij} && \forall (ij) \in \mathcal{E}\\
\sigma_{6a}(\hat{p}^g_i, \dot{p}^d_i, \tilde{\boldsymbol{p}}^f) &= \sum_{(ij)\in\mathcal{E}} \tilde{p}^f_{ij} - (\hat{p}^g_i - \dot{p}^d_i) && \forall i \in \mathcal{N}\\
\sigma_{6b}(\hat{q}^g_i, \dot{q}^d_i, \tilde{\boldsymbol{q}}^f) &= \sum_{(ij)\in\mathcal{E}} \tilde{q}^f_{ij} - (\hat{q}^g_i - \dot{q}^d_i) && \forall i \in \mathcal{N}
\end{aligned}$$

where $\sigma^L_{2a}$ and $\sigma^R_{2a}$ correspond to Constraints (2a) and capture the distance of the predictions $\hat{v}_i$ from exceeding the voltage bounds. The functions $\sigma^L_{2b}$ and $\sigma^R_{2b}$ correspond to Constraints (2b) and express how much the difference between two voltage angles exceeds the bound. Similarly, $\sigma^L_{3a}$, $\sigma^R_{3a}$, and $\sigma^L_{3b}$, $\sigma^R_{3b}$, relate to Constraints (3a) and (3b), respectively, and describe the distance of the predicted generator active and reactive dispatch from their bounds. Function $\sigma_4$ corresponds to Constraints (4) and captures the distance of the power flow on line $(ij)$ from its bound. Therein, $\tilde{p}^f_{ij}$ and $\tilde{q}^f_{ij}$ are, respectively, the active and reactive power flow for line $(ij) \in \mathcal{E}$. Notice that $\tilde{p}^f_{ij}$ and $\tilde{q}^f_{ij}$ are not predicted directly, as an output of the DNN. Instead, they are computed using the predicted quantities $\hat{v}_i, \hat{v}_j, \hat{\theta}_i$, and $\hat{\theta}_j$ according to Constraints (5a) and (5b). The quantities $p^f_{ij}$ and $q^f_{ij}$ correspond to the ground truths values. Functions $\sigma_{5a}$ and $\sigma_{5b}$ measure the deviation of the predicted flow (based on the other predicted quantities) to the ground truth values according to the Ohm's Law (Constraints (5a) and (5b)). Finally, the functions $\sigma_{6a}$ and $\sigma_{6b}$ relate to the Kirchhoff Current Law (Constraints (6a) and (6b)) and express the violation of flow conservation at a bus.

The violation degrees associated to the satisfiability degree above are defined as follows:

$$
\begin{aligned}
\nu_{2a}(\hat{\boldsymbol{v}}) &= \frac{1}{n} \sum_{i \in \mathcal{N}} \left( \nu_c^{\geq}(\sigma^L_{2a}(\hat{v}_i)) + \nu_c^{\geq}(\sigma^R_{2a}(\hat{v}_i)) \right) \\
\nu_{2b}(\hat{\boldsymbol{\theta}}) &= \frac{1}{e} \sum_{(ij) \in \mathcal{E}} \left( \nu_c^{\geq}(\sigma^L_{2b}(\hat{\theta}_{ij})) + \nu_c^{\geq}(\sigma^R_{2b}(\hat{\theta}_{ij})) \right) \\
\nu_{3a}(\hat{\boldsymbol{p}}) &= \frac{1}{n} \sum_{i \in \mathcal{N}} \left( \nu_c^{\geq}(\sigma^L_{3a}(\hat{p}_i)) + \nu_c^{\geq}(\sigma^R_{3a}(\hat{p}_i)) \right) \\
\nu_{3b}(\hat{\boldsymbol{q}}) &= \frac{1}{n} \sum_{i \in \mathcal{N}} \left( \nu_c^{\geq}(\sigma^L_{3b}(\hat{q}_i)) + \nu_c^{\geq}(\sigma^R_{3b}(\hat{q}_i)) \right) \\
\nu_4(\tilde{\boldsymbol{p}}^f, \tilde{\boldsymbol{q}}^f) &= \frac{1}{e} \sum_{(ij) \in \mathcal{E}} \nu_c^{\geq}(\sigma_4(\tilde{p}^f_{ij}, \tilde{q}^f_{ij})) \\
\nu_{5a}(\tilde{\boldsymbol{p}}^f, \boldsymbol{p}^f) &= \frac{1}{e} \sum_{(ij) \in \mathcal{E}} \nu_c^{=}(\sigma_{5a}(\tilde{p}^f_{ij}, p^f_{ij})) \\
\nu_{5b}(\tilde{\boldsymbol{q}}^f, \boldsymbol{q}^f) &= \frac{1}{e} \sum_{(ij) \in \mathcal{E}} \nu_c^{=}(\sigma_{5a}(\tilde{q}^f_{ij}, q^f_{ij})) \\
\nu_{6a}(\hat{\boldsymbol{p}}^g, \dot{\boldsymbol{p}}^d, \boldsymbol{p}^f) &= \frac{1}{e} \sum_{(ij) \in \mathcal{E}} \nu_c^{=}(\sigma_{6a}(\hat{p}^g_i, \dot{p}^d_i, \tilde{\boldsymbol{p}}^f)) \\
\nu_{6b}(\hat{\boldsymbol{q}}^g, \dot{\boldsymbol{q}}^d, \boldsymbol{q}^f) &= \frac{1}{e} \sum_{(ij) \in \mathcal{E}} \nu_c^{=}(\sigma_{6b}(\hat{q}^g_i, \dot{q}^d_i, \tilde{\boldsymbol{q}}^f)).
\end{aligned}
$$

where $n$ and $e$ denote the number of buses and transmission lines, respectively. These functions capture the average deviation by which the prediction violates the associated constraint. The violations degrees define penalties that will be used to enrich the DNN loss function to encourage their satisfaction. Prior describing the DNN objective, we introduce a further extension that exploits yet another aspect of the structure of the OPF.

## 6 The Network Architecture

An illustration of the network architecture is provided in Figure 1. The input layers on the left process the tensor of loads $(\boldsymbol{p}^d, \boldsymbol{q}^d)$. The network has four basic units, each following a decoder-encoder structure and composed by a number of layers as outlined in the table above. Each subnetwork predicts a target variable: voltage magnitudes $\hat{\boldsymbol{v}}$, phase angles $\hat{\boldsymbol{\theta}}$, active power generations $\hat{\boldsymbol{p}}^g$, and reactive power generations $\hat{\boldsymbol{q}}^g$. Each sub-network takes as input the last hidden layer of its input subnetwork, that processes the load tensors. Each of the four prediction outputs is used to compute the penalties associated to the quantity bounds ($\nu_{2a}$ for the voltage magnitudes, $\nu_{2b}$ for voltage angles, $\nu_{3a}$ for the active generator power, and $\nu_{3b}$ for the reactive generator power). The predictions for the voltage magnitude $\hat{v}$ and angle $\hat{\boldsymbol{\theta}}$ are used to compute the load flows $(\tilde{\boldsymbol{p}}^f, \tilde{\boldsymbol{q}}^f)$, as illustrated on the bottom of the Figure and produce penalties $\nu_4, \nu_{5a}$, and $\nu_{5b}$. Finally, the resulting flows $(\tilde{\boldsymbol{p}}^f, \tilde{\boldsymbol{q}}^f)$ and the predictions for active $\hat{\boldsymbol{p}}^g$ and reactive $\hat{\boldsymbol{q}}^g$ generator power are used to compute the penalties associated to the Kirchhoff's Current Law ($\nu_{6a}$ and $\nu_{6b}$).

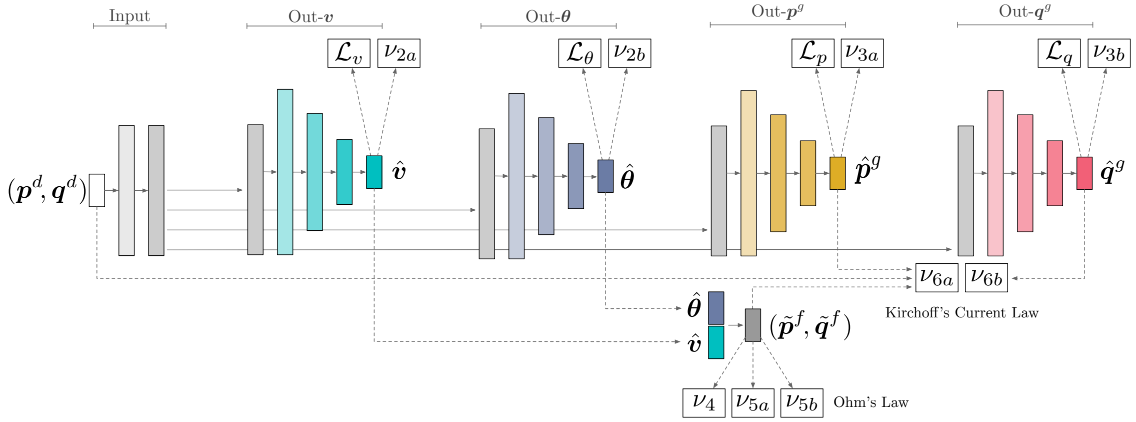

**Fig. 1:** A representation of the AC-OPF DNN model $\mathcal{M}_C$. Each layer is fully connected with ReLU activation. White boxes correspond to input tensors, dark, colored, boxes correspond to output layers. Loss components and violation degrees are shown in the rectangles with black borders.

# 7 Lagrangian Duality

Let $\hat{\mathcal{O}}[\boldsymbol{w}]$ be the resulting OPF-DNN with weights $\boldsymbol{w}$ 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 $\boldsymbol{w}$ that minimize the loss function for a given set of Lagrangian multipliers, i.e., it computes

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

It remains to determine appropriate Lagrangian multipliers. This paper proposes the use of Lagrangian duality to obtain the optimal Lagrangian multipliers when training the OPF-DNN, i.e., it solves

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

The Lagrangian dual is solved through a subgradient method that computes a sequence of multipliers $\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^k, \ldots$ by solving a sequence of trainings $LR(\boldsymbol{\lambda}^0), \ldots, LR(\boldsymbol{\lambda}^{k-1}), \ldots$ and adjusting the multipliers using the violations, i.e.,

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

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

In the implementation, step (L1) is approximated using a Stochastic Gradient Descent (SGD) method. Importantly, this step does not recomputes the training from scratch but uses a hot start for the weights $\boldsymbol{w}$.

The overall training scheme is presented in Algorithm 1. It takes as input the training dataset $(\mathcal{X}, \mathcal{Y})$, the optimizer step size $\alpha > 0$ and the Lagrangian step size $\rho > 0$. The Lagrangian multipliers are initialized in line 1. The training is performed for a fixed number of epochs, and each epoch optimizes the weights using a minibatch of size $b$. After predicting the voltage and generation power quantities (line 4), the objective and constraint losses are computed (lines 5 and 6). The latter uses the Lagrangian multipliers $\boldsymbol{\lambda}^k$ associated with current epoch $k$. The model weights are updated in line 7. Finally, after each epoch, the Lagrangian multipliers are updated following step (L2) described above (lines 8 and 9).

# 8 Experiments

This section evaluates the predictive accuracy of OPF-DNN and compares it to the AC model and its linear DC approximation. It also analyzes various design decisions in detail.

**Algorithm 1:** *Learning Step*

```
input: $(\mathcal{X}, \mathcal{Y})$ : Training data
       $\alpha, \rho$ : Optimizer and Lagrangian step sizes, reps.
$\boldsymbol{\lambda}^0 \leftarrow 0 \;\; \forall c \in \mathcal{C}$
for epoch $k = 0, 1, \ldots$ do
    foreach $(\boldsymbol{x}, \boldsymbol{y}) \leftarrow minibatch(\mathcal{X}, \mathcal{Y})$ of size $b$ do
        $\hat{\boldsymbol{y}} \leftarrow \hat{\mathcal{O}}[\boldsymbol{w}](\boldsymbol{x})$
        $\mathcal{L}_o(\hat{\boldsymbol{y}}, \boldsymbol{y}) \leftarrow \frac{1}{b} \sum_{\ell \in [b]} \mathcal{L}_v(\boldsymbol{y}_\ell, \hat{\boldsymbol{y}}_\ell) + \mathcal{L}_\theta(\boldsymbol{y}_\ell, \hat{\boldsymbol{y}}_\ell) +$
                $\mathcal{L}_p(\boldsymbol{y}_\ell, \hat{\boldsymbol{y}}_\ell) + \mathcal{L}_q(\boldsymbol{y}_\ell, \hat{\boldsymbol{y}}_\ell)$
        $\mathcal{L}_c(\boldsymbol{x}, \hat{\boldsymbol{y}}) \leftarrow \frac{1}{b} \sum_{\ell \in [b]} \sum_{c \in \mathcal{C}} \lambda_c^k \nu_c(\boldsymbol{x}_\ell, \hat{\boldsymbol{y}}_\ell)$
        $\omega \leftarrow \omega - \alpha \nabla_\omega (\mathcal{L}_o(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \mathcal{L}_c(\boldsymbol{x}, \hat{\boldsymbol{y}}))$
    foreach $c \in \mathcal{C}$ do
        $\lambda_c^{k+1} \leftarrow \lambda^k + \rho \nu_c(\boldsymbol{x}, \hat{\boldsymbol{y}})$
```

| **Test Case** | $\lvert\mathcal{N}\rvert$ | $\lvert\mathcal{E}\rvert$ | $l$ | $g$ | $\lvert(\mathcal{X}, \mathcal{Y})\rvert$ |
|---|---|---|---|---|---|
| **14_ieee** | 14 | 40 | 11 | 2 | 395806 |
| **30_ieee** | 30 | 82 | 21 | 2 | 273506 |
| **39_epri** | 39 | 92 | 21 | 10 | 287390 |
| **89_pegase** | 89 | 420 | 35 | 12 | 338132 |
| **118_ieee** | 118 | 372 | 99 | 19 | 395806 |
| **162_ieee_dtc** | 162 | 568 | 113 | 12 | 237812 |

**Table 1:** The Power Networks Adopted as Benchmarks.

**Data sets and Settings** The experiments examine the proposed models on a variety of mid-sized power networks from the NESTA library [8]. The ground truth data are constructed as follows: For each network, different benchmarks are generated by altering the amount of nominal load $\boldsymbol{x} = (\boldsymbol{p}^d, \boldsymbol{q}^d)$ within a range of $\pm 20\%$. The loads are thus sampled from the distributions $\boldsymbol{x}' = (\boldsymbol{p}^{d\prime}, \boldsymbol{q}^{d\prime}) \sim \text{Uniform}(0.8\boldsymbol{x}, 1.2\boldsymbol{x})$. Notice that the resulting benchmarks have load demands that vary by a factor of up to 20% of their nominal values: Many of them become congested and significantly harder computationally than their original counterparts. A network value that constitutes a dataset entry $(\boldsymbol{x}', \boldsymbol{y}')$ is a feasible OPF solution obtained by solving the AC-OPF problem detailed in Model 1.

The experiments use a 80/20 train-test split and report results on the test set.

The experiments examine the following OPF-DNN models:

- $\mathcal{M}_{\text{B}}$ refers to the baseline model: It minimizes a loss which includes only an MSE component of the predictions against the real quantities labels ($p^g, q^g, v$, and $\theta$).

  $\mathcal{M}_{\text{C}}$ exploits the problem constraints and minimizes the loss of $\mathcal{M}_{\text{B}}$ augumented with the Lagrangian terms as expressed in Section (5).

  $\mathcal{M}_I$ exploits the problem constraints using a Lagrangian loss function but minimizes the problem OPF cost directly, without the need for data labels.

The models were implemented using the Julia package PowerModels.jl [7] with the nonlinear solver IPOPT [25] for solving the nonlinear AC model and its the DC approximation. The DDN models were implemented using PyTorch [22] with Python 3.0.

## 8.1 Prediction Errors

This section first analyzes the prediction error of the DNN models. Table 2 reports the average L1 distance between the predicted generator active $\hat{\boldsymbol{p}}^g$ and reactive $\hat{\boldsymbol{q}}^g$ power, voltage magnitude $\hat{\boldsymbol{v}}$ and angles $\hat{\boldsymbol{\theta}}$ and the

| Test case | Model | $\hat{p}^g$ | $\hat{q}^g$ | $\hat{v}$ | $\hat{\theta}$ | $\tilde{p}^f$ | Test case | Model | $\hat{p}^g$ | $\hat{q}^g$ | $\hat{v}$ | $\hat{\theta}$ | $\tilde{p}^f$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **14_ieee** | $\mathcal{M}_\mathrm{B}$ | 5.7820 | 11.004 | 0.7310 | 1.4050 | 1.9070 | **89_pegase** | $\mathcal{M}_\mathrm{B}$ | 0.2516 | 0.2250 | 90.689 | 37.176 | 3133.4 |
| | $\mathcal{M}_I$ | 0.2711 | 0.7443 | 0.1451 | 0.1861 | 0.1530 | | $\mathcal{M}I.$ | 0.1032 | 0.0990 | 9.1190 | 0.9111 | 7.5400 |
| | $\mathcal{M}_\mathrm{C}$ | 0.2756 | 0.6980 | 0.1180 | 0.1480 | 0.1050 | | $\mathcal{M}_\mathrm{C}$ | 0.1074 | 0.0860 | 9.4168 | 0.8240 | 6.4130 |
| **30_ieee** | $\mathcal{M}_\mathrm{B}$ | 3.3465 | 2.0270 | 14.699 | 4.3400 | 27.213 | **118_ieee** | $\mathcal{M}_\mathrm{B}$ | 0.2150 | 2.9910 | 7.1520 | 4.2600 | 38.863 |
| | $\mathcal{M}_I$ | 0.3211 | 0.1344 | 0.3155 | 0.0611 | 0.2115 | | $\mathcal{M}I.$ | 0.0511 | 0.6170 | 0.1210 | 1.3301 | 0.6143 |
| | $\mathcal{M}_\mathrm{C}$ | 0.3052 | 0.1104 | 0.3130 | 0.0580 | 0.2030 | | $\mathcal{M}_\mathrm{C}$ | 0.0380 | 0.6900 | 0.1170 | 1.2750 | 0.6640 |
| **39_epri** | $\mathcal{M}_\mathrm{B}$ | 0.2299 | 1.2600 | 98.726 | 58.135 | 202.67 | **162_ieee** | $\mathcal{M}_\mathrm{B}$ | 0.2310 | 1.2070 | 9.1810 | 5.4800 | 82.076 |
| | $\mathcal{M}_I$ | 0.0511 | 0.1342 | 2.2414 | 0.2991 | 2.0110 | | $\mathcal{M}I$ | 0.0855 | 0.3695 | 0.1718 | 0.3950 | 0.7410 |
| | $\mathcal{M}_\mathrm{C}$ | 0.0559 | 0.1080 | 2.2350 | 0.2880 | 1.8360 | | $\mathcal{M}_\mathrm{C}$ | 0.0750 | 0.3760 | 0.1760 | 0.3720 | 0.7520 |

**Table 2:** Prediction Errors (%).

original quantities. It also reports the errors of the predicted flows $\tilde{\boldsymbol{p}}^f$ (which use the generator power and voltage predictions) and are important to assess the fidelity of the predictions. The distances are reported in percentage: $\frac{\|\hat{\boldsymbol{x}}-\boldsymbol{x}\|_1}{\|\boldsymbol{x}\|_1} \times 100$, for quantity $\boldsymbol{x}$, and best results are highlighted in bold.

Model $\mathcal{M}_\mathrm{C}$ and $\mathcal{M}_I$, which exploits the problem constraints, predicts much better voltage quantities and power flows than $\mathcal{M}_\mathrm{B}$. In addition, the usage of the model without labels $\mathcal{M}_I$, does not appear to degrade performance, despite the significant advantage of not utilizing labeled data.

| Test case | | DC | $\mathcal{M}_\mathrm{B}$ | $\mathcal{M}_I$ | $\mathcal{M}_C$ | DC | LF$_S$ | $\mathcal{M}_\mathrm{B}$ | $\mathcal{M}_I$ | $\mathcal{M}_C$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **14_ieee** | $\boldsymbol{p}^g$ | 2.4020 | 5.7048 | 0.2134 | 0.2052 | 0.1359 | 4.3518 | 1.1061 | 0.2689 | 0.2571 |
| | $\boldsymbol{v}$ | 1.8352 | 0.9174 | 0.0937 | 0.0944 | 3.0365 | 0.3450 | 0.3075 | 0.2118 | 0.2170 |
| **30_ieee** | $\boldsymbol{p}^g$ | 2.6972 | 2.0793 | 0.1615 | 0.1320 | 0.1907 | 13.504 | 2.1353 | 0.2901 | 0.2853 |
| | $\boldsymbol{v}$ | 1.2929 | 83.138 | 0.0954 | 0.0728 | 3.4931 | 0.4829 | 6.2996 | 0.4238 | 0.4168 |
| **39_epri** | $\boldsymbol{p}^g$ | 0.0731 | 0.2067 | 0.0556 | 0.0488 | 0.2163 | 2.1260 | 0.1350 | 0.0178 | 0.0155 |
| | $\boldsymbol{v}$ | 1.1086 | 95.944 | 0.3433 | 0.3131 | 1.7251 | 0.8573 | 6.8089 | 1.9818 | 2.3476 |
| **89_pegase** | $\boldsymbol{p}^g$ | 0.1641 | 0.1440 | 0.0460 | 0.0443 | 3.0662 | 7.3622 | 0.1834 | 0.0989 | 0.1035 |
| | $\boldsymbol{v}$ | 3.8584 | 86.795 | 6.6610 | 6.4734 | 1.1776 | 0.6913 | 4.1315 | 3.9880 | 3.9410 |
| **118_ieee** | $\boldsymbol{p}^g$ | 0.2011 | 0.1071 | 0.0044 | 0.0056 | 0.5865 | 3.8034 | 0.1353 | 0.0378 | 0.0391 |
| | $\boldsymbol{v}$ | 1.9971 | 3.4391 | 0.0959 | 0.0920 | 2.2780 | 0.9772 | 4.5972 | 0.1686 | 0.1768 |
| **162_ieee_dtc** | $\boldsymbol{p}^g$ | 0.6727 | 0.1054 | 0.0611 | 0.0571 | 0.6917 | 17.873 | 0.1648 | 0.0981 | 0.0981 |
| | $\boldsymbol{v}$ | 3.7718 | 3.6372 | 0.3106 | 0.3202 | 0.5820 | 1.4595 | 0.4378 | 0.2998 | 0.2906 |

**Table 3:** Average distances (%) for the active power (top rows) and voltage magnitude (bottom rows) of the Load Flow solutions w.r.t. the corresponding predictions (left table) and w.r.t. the AC-OPF solutions (right table).

## 8.2 Load Flow Analysis

The next results focus on evaluating the practicality of the proposed approach by simulating the prediction results in an operational environment. The idea is to measure how much the predictions need to be adjusted in order to satisfy the operational and physical constraints. The experiments perform a load flow (Model 2) on the predicted $\hat{\boldsymbol{p}}^g$ and $\hat{\boldsymbol{v}}$ values. In addition to comparing the DNN model variants, the results also report the deviations of the linear DC model from an AC-feasible solution. The DC model is widely used in power system industry. The results also reports the performance of a baseline load flow model LF$_S$ that finds a feasible solution using the hot-start state $\boldsymbol{s}_0$ as reference point in its objective function. These results highlight the value of learning in OPF-DNN: The reference point alone is not sufficient to find high quality solutions.

The results are tabulated in Table 3. The left table reports the L1 distances, in percentage, of the predictions $\hat{\boldsymbol{p}}^g$ and $\hat{\boldsymbol{v}}$ to the solutions $\boldsymbol{p}^g$ and $\boldsymbol{v}$ of the load flows. Trends similar to the previous section are observed. Note that the very high accuracy of the proposed method may render the use of a load flow optimization, to restore feasibility, unnecessary. *These results are significant: They suggest that OPF-DNN has the potential to replace the DC model as an AC-OPF approximation and deliver generator setpoints with greater fidelity.*

# 9 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, Anwar et al. **(year?)** 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. **(year?)** studied a DDN framework to predict the electricity costs of solar power plants coupled with a fuel backup system and energy storage. [5] 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*: [15] proposed a convolutional neural network (CNN) model for real-time power system fault classification to detect faulted power system voltage signals. [2] proposed a convolutional neural network to identify safe vs. unsafe operating points to reduce the risks of a blackout. [11] use a ResNet architecture to predict the effect of interventions that reconnect disconnected transmission lines in a power network.

In terms of OPF prediction, the literature is much sparser. The most relevant work uses a DNN architecture to learn the set of active constraints (e.g., those that, if removed, would improve the value of the objective function) at optimality in the linear DC model [19; 9]. Once the set of relevant active constraints are identified, exploiting the fact that the DC OPF is a linear program, one can run an exhaustive search to find a solution that satisfies the active constraints. While this strategy is efficient when the number of active constraints is small, its computational efficiently decreases drastically when its number increases due to the combinatoric nature of the problem. Additionally, this strategy applies only to the linear DC approximation.

This work departs from these proposals and predicts the optimal setpoints for the network generators and bus voltages in the AC-OPF setting. Crucially, the presented model actively exploits the OPF constraints during training, producing reliable results that significantly outperform classical model approximations (e.g., DC-OPF). This work also provides a compelling alternative to real-time OPF tracking [23; 17]: OPF-DNN always converges instantly with very high accuracy and can be applied to a wider class of applications.

# References

[1] Muhammad Bashar Anwar, Mohamed Shawky El Moursi, and Weidong Xiao. Novel power smoothing and generation scheduling strategies for a hybrid wind and marine current turbine system. *IEEE Transactions on Power Systems*, 32(2):1315–1326, 2016.

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

[3] M. E. Baran and F. F. Wu. Optimal capacitor placement on radial distribution systems. *IEEE Transactions on Power Delivery*, 4(1):725–734, Jan 1989.

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

[5] P. Chatziagorakis, C. Ziogou, C. Elmasides, G. Ch. 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*, 27(5):1093–1118, Jul 2016.

[6] B. H. Chowdhury and S. Rahman. A review of recent advances in economic dispatch. *IEEE Transactions on Power Systems*, 5(4):1248–1259, Nov 1990.

[7] Carleton Coffrin, Russell Bent, Kaarthik Sundar, Yeesian Ng, and Miles Lubin. Powermodels.jl: An open-source framework for exploring power flow formulations. In *PSCC*, June 2018.

[8] Carleton Coffrin, Dan Gordon, and Paul Scott. NESTA, the NICTA energy system test case archive. *CoRR*, abs/1411.0359, 2014.

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

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

[11] Benjamin Donnot, Balthazar Donon, Isabelle Guyon, Zhengying Liu, Antoine Marot, Patrick Panciatici, and Marc Schoenauer. LEAP nets for power grid perturbations. In *European Symposium on Artificial Neural Networks*, April 2019.

[12] E. B. Fisher, R. P. O'Neill, and M. C. Ferris. Optimal transmission switching. *IEEE Transactions on Power Systems*, 23(3):1346–1355, Aug 2008.

[13] Daniel Fontaine, Michel Laurent, and Pascal Van Hentenryck. Constraint-based lagrangian relaxation. In *Principles and Practice of Constraint Programming*, pages 324–339, 2014.

[14] Magnus R Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

[15] Turker Ince, Serkan Kiranyaz, Levent Eren, Murat Askar, and Moncef Gabbouj. Real-time motor fault detection by 1-D convolutional neural networks. *IEEE Transactions on Industrial Electronics*, 63(11):7067–7075, 2016.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[17] J. Liu, J. Marecek, A. Simonetta, and M. Takač. A coordinate-descent algorithm for tracking solutions in time-varying optimal power flows. In *Power Systems Computation Conference*, June 2018.

[18] A Monticelli, MVF Pereira, and S Granville. Security-constrained optimal power flow with post-contingency corrective rescheduling. *IEEE Transactions on Power Systems*, 2(1):175–180, 1987.

[19] Yeesian Ng, Sidhant Misra, Line Roald, and Scott Backhaus. Statistical learning for DC optimal power flow. In *Power Systems Computation Conference*, 2018.

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

[21] 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.

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[23] Y. Tang, K. Dvijotham, and S. Low. Real-time optimal power flow. *IEEE Transactions on Smart Grid*, 8(6):2963–2973, Nov 2017.

[24] J. Tong and H. Ni. Look-ahead multi-time frame generator control and dispatch method in PJM real time operations. In *IEEE Power and Energy Society General Meeting*, July 2011.

[25] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[26] Jianhui Wang, Mohammad Shahidehpour, and Zuyi Li. Security-constrained unit commitment with volatile wind power generation. *IEEE Transactions on Power Systems*, 23(3):1319–1327, 2008.

[27] Allen J. Wood and Bruce F. Wollenberg. *Power Generation, Operation, and Control*. Wiley-Interscience, 1996.