Learning Optimal Solutions for Extremely Fast AC Optimal Power Flow

Preprint

Ahmed S. Zamzam[1] and Kyri Baker[2]

1 National Renewable Energy Laboratory
2 University of Colorado Boulder

Presented at the IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (IEEE SmartGridComm)
November 11–13, 2020

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5D00-77736
November 2020

Learning Optimal Solutions for Extremely Fast AC Optimal Power Flow

Preprint

Ahmed S. Zamzam[1] and Kyri Baker[2]

1 National Renewable Energy Laboratory
2 University of Colorado Boulder

Suggested Citation
Zamzam, Ahmed, and Kyri Baker. 2020. *Learning Optimal Solutions for Extremely Fast AC Optimal Power Flow: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-5D00-77736. https://www.nrel.gov/docs/fy21osti/77736.pdf.

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5D00-77736
November 2020

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via www.OSTI.gov.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Learning Optimal Solutions for Extremely Fast AC Optimal Power Flow

Ahmed S. Zamzam* and Kyri Baker†
* National Renewable Energy Laboratory, Golden, CO, USA
† University of Colorado Boulder, Boulder, CO, USA

***Abstract*—In this paper, we develop a machine learning approach to optimize the real-time operation of electric grids. In particular, we learn feasible solutions to the AC optimal power flow (OPF) problem with negligible optimality gaps. The AC OPF problem aims to identify optimal operating conditions of electric grids that minimize power losses and/or generation costs. Because of the computational challenges with solving this nonconvex problem, many efforts have focused on linearizing or approximating the problem to solve the AC OPF problem in faster timescales; however, many of these approximations can be fairly poor representations of the actual system state and still require solving an optimization problem, which can be time-consuming for large networks. In this work, we learn a mapping between the system loading and optimal generation values, enabling us to find near-optimal and feasible AC OPF solutions. This allows us to bypass solving the traditionally nonconvex AC OPF problem, which results in a significant decrease in computational burden for grid operators.**

I. Introduction

AC optimal power flow (OPF) problems are solved by grid operators to achieve the most economic generation dispatch to meet electric power demand while adhering to the physical constraints of the network. With the increasing integration of fluctuating renewable energy sources, system operators are required to perform more frequent adjustments of the generator set points. Currently, real-time adjustments are realized using automatic generation control, which could lead to suboptimal operational points because of the use of restrictive affine control policies. Various linearizations or approximations of the AC OPF problem have been developed throughout the years to address the computational burden of solving the AC OPF in real time, but in many cases they cannot guarantee feasibility of the approximate solution, and they still require solving an optimization problem, which can be prohibitive as the network size increases. With faster system dynamics introduced by intermittent generation sources such as wind and solar, it is imperative that grid operations a) accurately represent power flows and b) can obtain optimal set points for controllable grid devices on timescales that can adapt to fluctuations in renewable energy [1].

Increasing renewable energy integration requires an increase in not only installed generation reserve but also in intelligent, computationally efficient ways of operating generation resources and associated components throughout the grid. Heavily increasing computing power is currently one of the only solutions that would allow for calculating AC OPF solutions in real time; and in most cases, this would not be not sustainable because it would require solving large-scale, nonconvex problems thousands of times daily, leading to increases in greenhouse gas emissions. Alternatively, could we periodically train a neural network (to reflect new installations or topology changes in the grid) and use it to efficiently produce solutions?

The need to operate the grid in a fast, clean, efficient manner is a pressing environmental and economic concern, and it has been analyzed for many years. Although this is not an exhaustive list, three general techniques have been used to pursue OPF solutions on fast timescales [2]: a) leveraging approximations of the AC power flow equations (e.g., [3]–[7]); b) distributed techniques (e.g., [8]–[10]); and c) so-called "real-time" or "online" optimization techniques (e.g., [11]–[13]), which borrow techniques from control theory to constantly pursue optimal solutions. Combinations of these three distributed online approaches that use linearizations of power flow equations are also used [12], [14].

Because of the importance of this issue, researchers have recently started using artificial intelligence techniques to assist in grid operations [15]–[20]. The data-driven nature of machine learning approaches is pertinent because large quantities of measurements are being generated in modern systems but are not yet being fully used [21]. In the area of power systems, machine learning approaches have been proposed to recover the system state [22], enhance solving optimal power flow problems (i.e., learning a good starting point for AC OPF [23] and learning the active set of the DC OPF problem [24], [25]), for directly bypassing the use of solvers to learn solutions in an online fashion (i.e., leveraging machine learning to obtain the solution of mixed-integer quadratic programs [26], and for learning a feasible and optimal DC OPF solution [27]).

In [25], neural networks are used to learn a mapping from uncertainty realizations to the active set of a DC OPF problem

This work was authored in part by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

This report is available at no cost from the National Renewable Energy Laboratory at www.nrel.gov/publications.

as an intermediate step toward learning the optimal solution. Once the active set is determined, the optimal solution to the original problem can be recovered by solving a linear system of equations. This work illustrated the benefits of using data to exploit the structure of the OPF problem and solve the DC OPF problem on timescales appropriate for corrective control; however, showing the benefit of this particular framework for AC OPF problems—which are very difficult to solve in real time—has not yet been demonstrated. In addition, if the correct active set is not identified during the classification, the resulting solution of the DC OPF will be incorrect, and additional resources must be used to identify the correct active set.

The work in [27] demonstrated that it is possible to learn the solution of the DC OPF problem directly and preserve feasibility. Encouraged by the extremely fast solution times for DC OPF with feasible solutions and negligible optimality losses when using a neural network in [27], we explored the potential and limitations of learning a mapping from the system loading to optimal generation set points and voltages in AC OPF. Our approach transcends traditional approaches to solve the AC OPF problem because it eliminates the the need to a) directly tackle the AC OPF optimization problem; b) use approximations, linearizations, or convexifications of the power flow equations; and c) use distributed techniques or computationally intensive platforms.

Recently, the authors of [18] proposed a machine learning model that targets approximating the AC OPF solutions directly. The AC OPF problem is treated as a regression task where the generator set points are estimated from the grid demand profile. Fixing the active power injection at all generators, including the slack bus, drastically affects the feasibility of the solution; hence, the reported feasibility percentage is almost 50% for a maximum of 10% change in the base load profile in a small network. The approach in [18] does not offer feasibility adjustments, and the design of the test scenarios with only 10% deviation without considering correlation makes the representativeness of the study questionable.

In the power systems realm, a large quantity of data is generated by grid operators by repeatedly solving AC OPF problems throughout the day [21]. This paper aims to use these data to train a neural network (offline) that can produce optimal solutions to the AC OPF problem (online). To achieve this, we learn a mapping from the network loads to the optimal generator set points, where a novel parameterization of the AC OPF solutions is used to enforce generation limits. Strictly feasible solutions are used to train the learning model, and the output of the model is constrained to adhere to active power generation and voltage magnitude constraints. To circumvent solutions that violate the system requirements, the solution feasibility is pursued by solving a set of power flow equations. Fig. 1 shows the architecture of the proposed approach in the operation phases. By using this framework, we can determine generation set points on timescales that are appropriate for balancing fast fluctuations in renewable generation and load. To summarize, the main technical contributions of this work

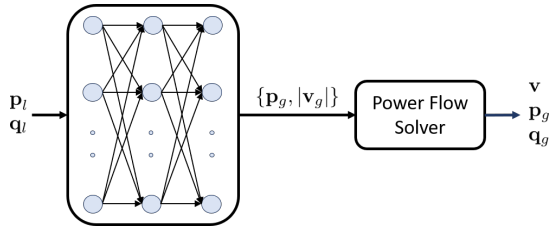

Fig. 1. The flow of the proposed learning approach.

are: i) the two-stage learning-based AC OPF solver that uses efficient modeling of the AC OPF solutions and feasibility recovering stage, ii) using strictly feasible training samples to steer the learned model toward feasible solutions, and iii) the implicit constraint of the set points using the saturated output layer.

II. Problem Formulation

In this section, we describe the AC OPF problem and our data-driven approach to appropriately reformulate the problem. First, consider a power network with N buses collected in set $\mathcal{N}$, set $\mathcal{G}$ collects the set of nodes with generators, and set $\mathcal{L}$ collects the set of load buses. The nodal power injection at bus $n \in \mathcal{N}$ is denoted by $p_n + j\ q_n$. Also, let $p_{g,n} + j\ q_{g,n}$, and $p_{l,n} + j\ q_{l,n}$ denote the power generated at bus $n \in \mathcal{G}$ and the load demand and bus $n \in \mathcal{L}$, respectively. Note that the sets $\mathcal{G}$ and $\mathcal{L}$ are both subsets of $\mathcal{N}$ but do not need to be disjointed. The voltage at bus n is denoted by v_n, and define $\mathbf{v} := [v_n]_{n\in\mathcal{N}}$, $\mathbf{p} := [p_n]_{n\in\mathcal{N}}$, and $\mathbf{q} := [q_n]_{n\in\mathcal{N}}$, which collect the voltage phasors, active power injection, and reactive power injection at all buses, respectively. Therefore, the AC OPF problem can be formulated as:

$$\underset{\mathbf{v},\{p_{g,n},q_{g,n}\}_{n\in\mathcal{G}}}{\text{minimize}} \quad c(\mathbf{p}_g) \ = \ \sum_{n\in\mathcal{G}} c_n(p_{g,n}) \tag{1a}$$

$$\text{subject to} \quad \underline{p}_{g,n} \le p_{g,n} \le \overline{p}_{g,n} \tag{1b}$$

$$\underline{q}_{g,n} \le q_{g,n} \le \overline{q}_{g,n} \tag{1c}$$

$$\underline{|v|} \le |\mathbf{v}| \le \overline{|v|} \tag{1d}$$

$$\mathbf{h}(\mathbf{v}, \mathbf{p}, \mathbf{q}) = \mathbf{0} \tag{1e}$$

where $\underline{p}_{g,n}$ ($\underline{q}_{g,n}$) and $\overline{p}_{g,n}$ ($\overline{q}_{g,n}$) denote the minimum and maximum active (reactive) power injection from the generator installed at bus $n \in \mathcal{G}$, and $\underline{|v|}$ and $\overline{|v|}$ represent the lower and upper limits on the voltage magnitudes. The vector collecting the active power injection from all generators is denoted by $\mathbf{p}_g$, and the cost of generating power from the generator installed at bus n is denoted by $c_n(p_{g,n})$. The *nonlinear* power flow equations are collected in the equality constraint (1e).

Given the active and reactive power demand at the load buses, the active power generated by the generators, and the voltage magnitudes at the generator buses, the vector $\mathbf{v}$ can be recovered by solving the power flow equations [28]. Using this property, we treat the AC OPF problem as an operator with inputs representing the load profile, and the generators' active power injections and voltage magnitudes are the output of this operator. In other words, we treat the OPF as an operator [29]

This report is available at no cost from the National Renewable Energy Laboratory at www.nrel.gov/publications.

with inputs comprising $p_{l,n}$ and $q_{l,n}$ for all $n \in \mathcal{L}$ and with output representing $|v_n|$ for buses $n \in \mathcal{G}$ and $p_{g,n}$ for nodes $n \in \mathcal{G}/\{n_0\}$, where n_0 is the index of the reference bus. Therefore, the goal of the proposed data-driven approach is to learn the underlying OPF mapping between the load demand at all load buses and the generators' active power and voltage magnitude set points.

Any active power generation $p_{g,n}$ that satisfies (1b) can be written as:

$$p_{g,n} = \underline{p}_{g,n} + \alpha_n \Big(\overline{p}_{g,n} - \underline{p}_{g,n} \Big),$$

where $0 \leq \alpha_n \leq 1$. Similarly, the voltage magnitude at bus $n \in \mathcal{G}$ can be parameterized by $\beta_n \in [0,1]$, which is given by:

$$|v_n| = \underline{|v|} + \beta_n \Big(\overline{|v|} - \underline{|v|} \Big).$$

Define the sets $\mathcal{A} := \big\{ [\alpha_n]_{n \in \mathcal{G}/\{n_0\}} \ | 0 \leq \alpha_n \leq 1 \big\}$ and $\mathcal{B} := \big\{ [\beta_n]_{n \in \mathcal{G}} \ | 0 \leq \beta_n \leq 1 \big\}$. Therefore, the AC OPF operator can be written as: $\mathbf{\Omega} : \mathcal{P}_l \times \mathcal{Q}_l \rightarrow \mathcal{A} \times \mathcal{B}$,, where $\mathcal{P}_l$ and $\mathcal{Q}_l$ represent the sets of active and reactive load demand at all buses $n \in \mathcal{L}$, respectively. Paremterizing the solution of the AC OPF problem using the parameters α_n and β_n will prove useful when approximating the mapping Ω.

III. Learning Feasible AC OPF Solutions

Neural networks are efficient approximators of high-dimensional, nonlinear mappings. With the availability of massive amounts of training data (here, solutions to the AC OPF problem), neural networks can uncover natural relationships in the data to perform classification or regression extremely fast. This is why they have been used previously to predict OPF solutions [18], [27] and the active set of OPF problems [18], [25]. In this section, we discuss generating the training data for the neural networks, training the neural networks, and obtaining the overall feasible solution to the AC OPF problem.

A. Obtaining mappings between loads and optimal solutions

To approximate the mapping $\mathbf{\Omega}$, we use a deep neural network model that leverages sigmoid activation functions. For finite-width deep neural networks, smooth functions can be approximated up to a specified accuracy that depends on the smoothness parameters of the function as well as the neural network size [22], [30]. That is, if a neural network, $g(\cdot;\theta)$, is optimally trained with samples of $f(\cdot)$ on a closed set, then the distance between the output of the neural network, $\hat{\mathbf{y}} = g(\cdot;\theta)$, and ground-truth mapping, $\mathbf{y} = f(\cdot)$, can be bounded as:

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_\infty \leq \epsilon,$$

where ϵ is function of the smoothness parameters of $f(\cdot)$ and the size of the neural network $g(\cdot;\theta)$. Because the solution of the AC OPF is often on the boundary of the feasibility set [31], we aim to ensure that the neural network outputs are in the interior of the AC OPF feasibility set.

To avoid cases where the voltage magnitude is exactly at the maximum or the lower voltage limits, we generate the training by solving the following restricted AC OPF problem (R-AC OPF).

$$\underset{\mathbf{v}, \{p_{g,n}, q_{g,n}\}_{n \in \mathcal{G}}}{\text{minimize}} \quad \sum_{n \in \mathcal{G}} c_n(p_{g,n}) \tag{2a}$$

$$\text{subject to} \quad \text{(1b), (1c), (1e)}$$

$$\underline{|v|} + \lambda \ \leq \ |\mathbf{v}| \ \leq \ \overline{|v|} - \lambda \tag{2b}$$

Therefore, the generated solutions are guaranteed to be in the interior of the feasibility region with respect to the voltage magnitudes. Note that this data-driven approach approximates a mapping that deviates from the original AC OPF mapping; however, the training samples generated by solving (2) are strictly in the interior of the voltage magnitudes feasibility set. Therefore, any bounded deviations in the learned mapping are expected to remain within the voltage limits. The parameter λ can thus be considered an algorithmic tuning parameter that addresses the optimality and feasibility trade-off. Notice that large values of λ could render the original problem infeasible, and hence the parameter λ must be tuned to obtain strictly feasible solutions.

In addition, to ensure that the generation limits adhere to the prescribed AC OPF constraints, we use sigmoid functions at the output layer of the neural network. This ensures that the output is bounded between 0 and 1 for the values of α_n and β_n. Designing the output layer of the neural network to be a sigmoid function acts as an *implicit* constraint on the learned AC OPF mapping, which illustrates the reasoning for reparameterizing the AC OPF solutions in terms of α_n and β_n. Although enforcing general constraints on the output of the neural network makes the training process challenging, incorporating a sigmoid function at the output layer does not increase the complexity of the training process.

B. Generating training samples

In a physical grid, historical AC OPF runs would be used as the training set for the neural network, yielding an abundance of data representing a wide variety of system states; however, for simulation and testing, we must generate this set of training data. Let $\mathbf{p}_l \in \mathcal{R}^{|\mathcal{L}|}$ denote the vector collecting the load demand at all load buses. We generate the instances of load demand using the truncated Gaussian distribution:

$$\mathbf{p}_l \sim \mathcal{TN} \left(\mathbf{p}_{l,0}, \ \mathbf{\Sigma}_l, \ (1-\mu)\mathbf{p}_{l,0}, \ (1+\mu)\mathbf{p}_{l,0} \right),$$

where $\mathbf{p}_{l,0}$ denotes a baseload profile; $\mathbf{\Sigma}_l$ denotes the covariance matrix; and μ denotes the maximum allowable deviation from the baseload profile, which we set to 0.7 in our simulations. The covariance matrix accounts for the relationships between loading patterns at different locations throughout the network. Here, we generated load profile realizations using a Gibbs sampler algorithm [32]. To consider deviations in the reactive power demand, we randomly chose the power factor of each load at each instance from a uniform distribution between 0.8 and 1. Rarely, the generated load profile can

This report is available at no cost from the National Renewable Energy Laboratory at www.nrel.gov/publications.

result in an infeasible solution for the AC OPF problem, hence making it infeasible for the R-AC OPF problem as well. In such cases, we discard this load profile from the training set because it does not represent a physically realizable state of the power system.

C. Learning model

A deep neural network model (i.e., a neural network with more than two layers) is used to approximate the mapping between the load profile and the optimal voltage magnitude and active power injections of the generators. The size of the input of the neural network is $2|\mathcal{L}|$, which represents the active and reactive load demand at the load buses; and the number of outputs is $2|\mathcal{G}|-1$, which represents the voltage magnitude at the generator buses and the active power injection at all generators except the one at the slack/reference bus. Let $(\mathbf{x}_i, \mathbf{y}_i)$ denote the i-th pair of data within the training set. The model is thus trained using the classical empirical loss minimization formulation, which is given by:

$$\min_{\theta} \frac{1}{I} \sum_{i}^{I} \|\mathbf{y}_i - g(\mathbf{x}_i;\theta)\|_2^2,$$

where I represents the total number of the training samples; nd $g(\cdot;\theta)$ represents the neural network mapping, where θ collects the trainable parameters.

D. Recovering feasible solutions

As mentioned in Section II, we pursue feasibility by both restricting the training set to include strictly feasible solutions and to adhere to generation and voltage limits by constraining the output of the neural network. Although this ensures feasibility of the learned values, the overall power flow equations and other generation constraints must be satisfied as well. Thus, the power flow equation is solved to assess the general feasibility of the solution with respect to constraints in (1). Let $p^o_{g,n}$ and $|v^o_n|$ denote the output of the neural network representing the active power injection and voltage magnitude of bus $n \in \mathcal{G}$. Then, solving the power flow problem can be formulated as follows, where the output of the neural network representing the voltage magnitude and active power injection at generator n are denoted by $|v^o_n|$ and $p^o_{g,n}$, respectively.

$$\begin{aligned}
\text{find} \quad & \mathbf{v}, \mathbf{p}_g, \mathbf{q}_g && \text{(3a)}\\
\text{subject to} \quad & |v_n| = |v^o_n| \quad \forall n \in \mathcal{G} && \text{(3b)}\\
& p_{g,n} = p^o_{g,n} \quad \forall n \in \mathcal{G}/n_0 && \text{(3c)}\\
& \mathbf{h}(\mathbf{v}, \mathbf{p}_g, \mathbf{q}_g) = \mathbf{0} && \text{(3d)}
\end{aligned}$$

In rare cases, the solution of the power flow equations result in reactive power injections at the generators that exceed the limits in (1c). Let the solution of the power flow equations in (3) representing the reactive power injection at bus $n \in \mathcal{G}$ be denoted by $q^o_{g,n}$. Also, denote the set of buses where the reactive power limits are violated by $\mathcal{G}_Q$. Then, we define $q^r_{g,n}$ at the buses $\mathcal{G}_Q$ as:

$$q^r_{g,n} = \min(\overline{q}_{g,n},\ \max(q^o_{g,n},\ \underline{q}_{g,n})). \tag{4}$$

Consequently, a modified power flow problem must be solved to obtain the feasible operating point of the network, which is formulated as:

$$\begin{aligned}
\text{find} \quad & \mathbf{v}, \mathbf{p}_g, \mathbf{q}_g && \text{(5a)}\\
\text{subject to} \quad & |v_n| = |v^o_n| \quad \forall n \in \mathcal{G}/\mathcal{G}_Q && \text{(5b)}\\
& p_{g,n} = p^o_{g,n} \quad \forall n \in \mathcal{G}/n_0 && \text{(5c)}\\
& q_{g,n} = q^r_{g,n} \quad \forall n \in \mathcal{G}_Q && \text{(5d)}\\
& \mathbf{h}(\mathbf{v}, \mathbf{p}_g, \mathbf{q}_g) = \mathbf{0} && \text{(5e)}
\end{aligned}$$

The proposed approach is summarized in Algorithm 1. Although a small amount of time is needed to solve the power flow equations and check the feasibility of all the remaining variables, the overall solution procedure from end-to-end is *extremely fast* because we are dealing with a simpler problem, which is solving the power equations instead of solving the more complex nonconvex AC OPF problem. In addition, note that the objective function value in the original problem (1) is not altered by adjusting reactive power and voltage to preserve feasibility because the cost function is the function of only the active power generated $p^o_{g,n}$.

Algorithm 1 Proposed learning-based approach for AC OPF

Input: $\{p_{l,n}, q_{l,n}\}_{n\in\mathcal{L}}$
Output: $\{\mathbf{v}^r, \mathbf{p}^r_g, \mathbf{q}^r_g\}$
begin
  $\{|v^o_n|\}_{n\in\mathcal{G}}, \{p^o_{g,n}\}_{n\in\mathcal{G}/n_0} \longleftarrow \mathbf{g}(\{p_{l,n}, q_{l,n}\}_{n\in\mathcal{L}}; \boldsymbol{\theta})$
  $\{\mathbf{v}^o, \mathbf{p}^o_g, \mathbf{q}^o_q\} \longleftarrow$ solution of (3)
  if (1c) *is satisfied* **then**
    $\{\mathbf{v}^r, \mathbf{p}^r_g, \mathbf{q}^r_g\} \longleftarrow \{\mathbf{v}^o, \mathbf{p}^o_g, \mathbf{q}^o_q\}$
  else
    for $n \in \mathcal{G}_Q$ **do**
      $q^r_{g,n} \leftarrow \min\{\overline{q}_{g,n},\ \max\{q^o_{g,n},\ \underline{q}_{g,n}\}\}$
    end
    $\{\mathbf{v}^r, \mathbf{p}^r_g, \mathbf{q}^r_g\} \longleftarrow$ solution of (5)
  end
end

IV. Simulation Results

We now present the considered networks and analyze the optimality and feasibility obtained using the proposed approach. MATPOWER [33] was used to generate $100,000$ training samples for each considered network and to solve the power flow equations for the obtained generator set points.

A. Test networks and setup

Table I shows the number of nodes, generators, lines, and the base operating costs of the IEEE 118-, 57-, and 39-bus systems used in this paper.

For the neural network network model, we used a neural network with 3 hidden layers, with sigmoid activation function for all layers. The learning model was implemented using the Python-based TensorFlow software library [34] and trained using the Adam optimizer [35].

This report is available at no cost from the National Renewable Energy Laboratory at www.nrel.gov/publications.

TABLE I
DETAILS OF THE CONSIDERED TEST NETWORKS.

Test Case	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{G}\rvert$	$\lvert\mathcal{L}\rvert$	Base Cost ($/hr)
IEEE 118-bus	118	54	99	129.66×10^3
IEEE 57-bus	57	7	42	41.74×10^3
IEEE 39-bus	39	10	21	41.86×10^3

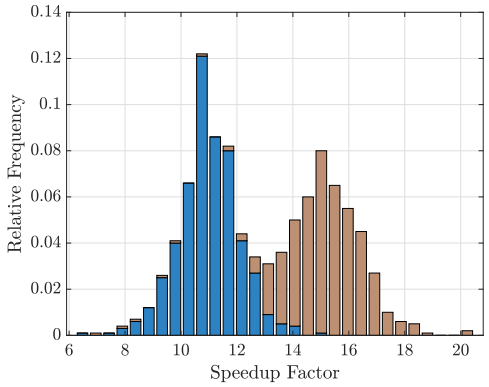

Fig. 2. Relative histogram of speedup factor for solving the AC OPF for the IEEE 118-bus test case ($\lambda = 5 \times 10^{-3}$). Blue bars represent cases where (5) is solved at least once to recover a feasible solution, and brown bars represent cases where the output of the neural network is already feasible.

B. Time comparisons

Denote the time to obtain an AC OPF solution using our method for the t-th test instance by $\tau_{o,t}$, which includes the time needed to evaluate the neural network, the time needed to solve the power flow equations, and the time consumed by solving any additional power flow equations to ensure feasibility. For comparison, let $\tau_{\text{OPF},t}$ denote the time consumed by the 'PDIPM' (primal-dual interior point method) solver to solve the original AC OPF. In the considered experiments, this solver was on average the fastest and most successful solver. To measure the computational improvements, define a speedup factor (SF) as $\text{SF} = \frac{1}{T}\sum_{t=1}^{T} \tau_{\text{OPF},t}/\tau_{o,t}$, which we can use to measure the relative speedup of the learning-based framework versus solving the AC OPF directly.

Fig. 2 shows the relative histogram of the speedup factor for the IEEE 118-bus network with $\lambda = 0.005$. The speedup factor ranges from 6x to 22x, which shows the potential of the proposed approach to identify feasible AC OPF solutions extremely quickly, even in the slower cases. The two humps in the histogram result from the fact that the second stage of the algorithm is needed for only a portion of the test cases.

C. Feasibility and optimality

After passing the network demands through the learning model, we obtain the generator set points. The set points are guaranteed to satisfy constraints (1b) and (2b), given the design of the output layer of the neural network. Further, to ensure feasibility of the *overall* AC OPF solution, the infeasibility of the generated solution with respect to the reactive power limit constraints (1c) needs to be ensured by solving the power flow equations, as outlined in problem (5). We evaluate the infeasibility of this constraint for the solutions obtained directly using our learning model by defining a metric δ_q, which is given by:

$$\delta_q = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{|\mathcal{G}|} \|\boldsymbol{\xi}_{q,t}\|_2,$$

where $\boldsymbol{\xi}_{q,t}$ collects all the reactive power violations across all buses. The n-th element of the vector $\boldsymbol{\xi}_{q,t}$ is given by:

$$\xi_{q,t,n} = \max\{\underline{q}_{g,n} - q^o_{g,n}, 0\} + \max\{q^o_{g,n} - \overline{q}_{g,n}, 0\},$$

where $q^o_{g,n}$ denotes the recovered reactive power injection at bus n by solving the power flow equations at the set point given by the output of the neural network.

Let $\mathbf{p}^\star_g$ denote the optimal generation set points obtained by solving the AC OPF problem (1), and denote the generators' active power injection set points obtained by our data-driven approach by $\mathbf{p}^o_g$. The average optimality of the solutions obtained through the proposed approach is evaluated as follows:

$$\text{Optimality} = \frac{1}{T}\sum_{t=1}^{T} \frac{c(\mathbf{p}^o_{g,t}) - c(\mathbf{p}^*_{g,t})}{c(\mathbf{p}^*_{g,t})},$$

where $c(\mathbf{p}^o_{g,t})$ and $c(\mathbf{p}^\star_{g,t})$ denote the total objective function value resulting from the solution obtained using our approach and the optimal cost of the original AC OPF, respectively.

Denote by $\mathbf{p}^o_t$, $\mathbf{q}^o_t$, and $\mathbf{v}^o_t$ the solution obtained using our approach for a particular test scenario, t. The feasibility of the solution is measured by the satisfaction of the power flow equations. The maximum infeasibility is thus evaluated as:

$$\text{Feasibility} = \max_t \|\mathbf{h}(\mathbf{v}^\mathbf{r}_\mathbf{t}, \mathbf{p}^\mathbf{r}_\mathbf{t}, \mathbf{q}^\mathbf{r}_\mathbf{t})\|_\mathbf{2}.$$

Table II illustrates the results from evaluating $1,000$ AC OPF test scenarios for each network. As shown in all the network simulations, the maximum infeasibility observed in the all the test scenarios is very low (and well within most solver tolerances for feasibility). The average computational speedup by using the data-driven approach versus solving the AC OPF directly is also significant, ranging from a speedup factor of nearly 8x to more than 15x. These tables also show the average infeasibility, δ_q, of the reactive power limits (1c) encountered, in MVAR, after solving the power flow equations with the learned generator active power set points and voltages. Note, however, that the feasibility of these constraints is ensured using the procedure in Algorithm 1, and the time required to do this is considered in the results.

Further restricting the feasible set of voltages by increasing λ from 0.005 to 0.01 generally has a positive impact on feasibility while mildly impacting optimality. Regardless of the λ chosen here, however, the impact on the optimality of the learned solution is minimal for all considered networks, indicating that using the learning-assisted approach of recovering AC OPF solutions can preserve optimality and feasibility while providing significant computational speedups.

This report is available at no cost from the National Renewable Energy Laboratory at www.nrel.gov/publications.

TABLE II
PERFORMANCE RESULTS FOR EACH TEST CASE.

Case	λ	δ_q	Optimality	SF	Feasibility
118-bus	0.005	0.30	$2.97{\cdot}10^{-4}$	11.83	$1.41{\cdot}10^{-8}$
	0.01	0.29	$5.55{\cdot}10^{-4}$	11.74	$3.58{\cdot}10^{-9}$
57-bus	0.005	1.58	$4.56{\cdot}10^{-3}$	9.49	$3.36{\cdot}10^{-8}$
	0.01	1.37	$7.05{\cdot}10^{-3}$	9.09	$1.33{\cdot}10^{-8}$
39-bus	0.005	1.32	$3.26{\cdot}10^{-3}$	15.38	$2.51{\cdot}10^{-8}$
	0.01	1.07	$6.25{\cdot}10^{-3}$	12.86	$1.11{\cdot}10^{-8}$

V. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a framework for solving AC OPF problems that uses machine learning to learn a mapping between system loading and optimal generation values. We provided a method to pursue feasibility of the solution in the original AC OPF problem by training the model on strictly feasible solutions and then solving the power flow equations to recover the remainder of the complex network voltages and the reactive power generation values. From start to finish, the framework can recover a feasible solution to the AC OPF problem extremely quickly with negligible sacrifices in optimality. Thus, overall the framework is faster and can provide a more accurate solution than other methods that are used to quickly solve AC OPF, such as approximating power flow equations.

REFERENCES

[1] D. Rolnick *et al.*, "Tackling climate change with machine learning," *arXiv preprint arXiv:1906.05433v2*, 2019.

[2] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov 2017.

[3] J. Lavaei and S. H. Low, "Convexification of optimal power flow problem," in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2010, pp. 223–232.

[4] S. Guggilam, E. Dall'Anese, Y. Chen, S. Dhople, and G. B. Giannakis, "Scalable optimization methods for distribution networks with high PV integration," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, July 2016.

[5] S. Bolognani and F. Dörfler, "Fast power system analysis via implicit linearization of the power flow manifold," in *2015 53rd Annual Allerton Conf. on Communication, Control, and Computing*, Oct. 2015.

[6] A. Bernstein and E. Dall'Anese, "Linear power-flow models in multiphase distribution networks," in *The 7th IEEE Intl. Conf. on Innovative Smart Grid Technologies*, Sep. 2017.

[7] A. Bernstein, C. Wang, E. Dall'Anese, J.-Y. Le Boudec, and C. Zhao, "Load flow in multiphase distribution networks: Existence, uniqueness, non-singularity and linear models," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 5832–5843, 2018.

[8] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Trans. on Power Systems*, vol. 29, no. 5, pp. 2370–2380, Sep. 2014.

[9] W. Zheng, W. Wu, B. Zhang, H. Sun, and Y. Liu, "A fully distributed reactive power optimization and control method for active distribution networks," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1021–1033, March 2016.

[10] K. Baker, J. Guo, G. Hug, and X. Li, "Distributed MPC for efficient coordination of storage and renewable energy sources across control areas," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 992–1001, Mar. 2016.

[11] Y. Tang, K. Dvijotham, and S. Low, "Real-time optimal power flow," *IEEE Trans. on Smart Grid*, vol. 8, no. 6, pp. 2963–2973, Nov 2017.

[12] E. Dall'Anese and A. Simonetto, "Optimal power flow pursuit," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 942–952, March 2018.

[13] A. Hauswirth, A. Zanardi, S. Bolognani, F. Dörfler, and G. Hug, "Online optimization in closed loop on the power flow manifold," in *2017 IEEE Manchester PowerTech*, June 2017.

[14] A. Bernstein and E. Dall'Anese, "Real-time feedback-based optimization of distribution grids: A unified approach," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 1197–1209, 2019.

[15] K. Baker and A. Bernstein, "Joint chance constraints in AC optimal power flow: Improving bounds through learning," *IEEE Transactions on Smart Grid*, pp. 6376 – 6385, Mar 2019.

[16] D. Gunduz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. Murthy, and M. van der Schaar, "Machine learning in the air," *arXiv preprint arXiv:1904.12385*, 2019.

[17] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2018.

[18] N. Guha, Z. Wang, and A. Majumdar, "Machine learning for AC optimal power flow," in *Climate Change Workshop at The Thirty-sixth International Conference on Machine Learning (ICML)*, June 2019.

[19] A. S. Zamzam and N. D. Sidiropoulos, "Physics-aware neural networks for distribution system state estimation," *IEEE Transactions on Power Systems (to appear)*, 2019, [Online] Available: https://arxiv.org/abs/1903.09669.

[20] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Real-time voltage control using deep reinforcement learning," 2019, [Online] Available: https://arxiv.org/abs/1904.09374.

[21] H. Akhavan-Hejazi and H. Mohsenian-Rad, "Power systems big data analytics: An assessment of paradigm shift barriers and prospects," *Energy Reports*, vol. 4, pp. 91 – 100, 2018.

[22] A. S. Zamzam, X. Fu, and N. D. Sidiropoulos, "Data-driven learning-based optimization for distribution system state estimation," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4796 – 4805, Apr. 2019.

[23] K. Baker, "Learning warm-start points for AC optimal power flow," in *IEEE Machine Learning for Signal Proc. Conf.*, Pittsburgh, PA 2019.

[24] S. Misra, L. Roald, and Y. Ng, "Learning for constrained optimization: Identifying optimal active constraint sets," 2019, [Online] Available at: https://arxiv.org/pdf/1802.09639.

[25] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," 2019, [Online] Available at: https://arxiv.org/pdf/1902.05607.

[26] D. Bertsimas and B. Stellato, "Online mixed-integer optimization in milliseconds," July 2019, [Online] Available at: https://arxiv.org/pdf/1907.02206.

[27] X. Pan, T. Zhao, and M. Chen, "DeepOPF: Deep neural network for dc optimal power flow," in *IEEE Intl. Conf. on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019.

[28] A. Araposthatis, S. Sastry, and P. Varaiya, "Analysis of power-flow equation," *International Journal of Electrical Power & Energy Systems*, vol. 3, no. 3, pp. 115–126, July 1981.

[29] F. Zhou, J. Anderson, and S. H. Low, "The optimal power flow operator: Theory and computation," *arXiv preprint arXiv:1907.02219*, 2019.

[30] P. Grohs, D. Perekrestenko, D. Elbrächter, and H. Bölcskei, "Deep neural network approximation theory," *arXiv preprint arXiv:1901.02220*, 2019.

[31] D. Molzahn, S. Baghsorkhi, and I. Hiskens, "Semidefinite relaxations of equivalent optimal power flow problems: An illustrative example," in *IEEE Int. Symposium on Circuits and Sys.*, May 2015, pp. 1887–1890.

[32] J. H. Kotecha and P. M. Djuric, "Gibbs sampling approach for generation of truncated multivariate gaussian random variables," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, vol. 3, 1999, pp. 1757–1760.

[33] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb 2011.

[34] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May*, 2015.

This report is available at no cost from the National Renewable Energy Laboratory at www.nrel.gov/publications.