

An Analysis of the Reliability of AC Optimal Power Flow Deep Learning Proxies

My H. Dinh, Ferdinando Fioretto
University of Virginia
{fqw2tz, fioretto}@virginia.edu

Mostafa Mohammadian, Kyri Baker
University of Colorado Boulder
{mostafa.mohammadian, kyri.baker}@colorado.edu

Abstract—Optimal Power Flow (OPF) is a challenging problem in power systems, and recent research has explored the use of Deep Neural Networks (DNNs) to approximate OPF solutions with reduced computational times. While these approaches show promising accuracy and efficiency, there is a lack of analysis of their robustness. This paper addresses this gap by investigating the factors that lead to both successful and suboptimal predictions in DNN-based OPF solvers. It identifies power system features and DNN characteristics that contribute to higher prediction errors and offers insights on mitigating these challenges when designing deep learning models for OPF.

I. INTRODUCTION AND BACKGROUND

The Optimal Power Flow (OPF) problem aims to find the generator dispatch with minimal cost to meet power system demands. However, the introduction of intermittent renewable energy sources requires frequent adjustment of generator set-points. To overcome the computational complexity of solving OPFs, system operators often use more computationally efficient approximations like the DC approximation. However, these approximations may lead to suboptimal infeasible solutions [1].

Recent research has explored the use of Deep Neural Networks (DNNs) to approximate AC-OPF [2]–[4]. Once trained, DNNs can quickly compute predictions in milliseconds. Although these learning models have shown high accuracy in approximating generator set-points for AC-OPF, little is known about why they achieve such accurate predictions and the robustness of their predictions. This paper aims to address this knowledge gap and contribute to the growing body of research on deep learning for OPF by providing insights into the inner workings of these black box models.

The paper investigates the relationship between the training data and their target outputs to understand why DNNs can accurately approximate OPF solutions with minimal errors. Fig. 1 (left) shows the change in generator outputs as the total demand varies for selected IEEE-118 generators. The blue curve indicates a linear dependence between generator outputs and loads, suggesting that a simple learning model can effectively capture this behavior. This is supported by low prediction errors reported in Fig. 1 (right), indicating accurate results for approximating OPF with DNNs when many generators exhibit such behavior.

However, certain generators are more challenging to predict, as shown by the volatile orange curve in the figure, leading to higher prediction errors and highlighting robustness concerns. The paper explains why standard learning models struggle with

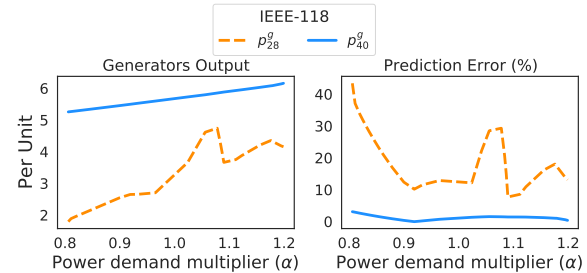


Fig. 1: Generator 28 and 40 outputs (left) and associated prediction errors (right) as a function of demand. Orange (blue) colors show high (low) variability in output and continuous (dashed) lines depict easy (hard) prediction tasks.

such behaviors due to training data instability affecting model approximation. Additionally, it examines factors impacting prediction accuracy for these generators, explores error-contributing OPF traits, and emphasizes modeling OPF constraints during training to capture prediction complexity.

Machine learning has gained traction in accelerating power system optimization procedures. Hasan et al. [5] provide a recent survey summarizing the developments in this area. Notably, Pan et al. [6] investigate DNN architectures for predicting DC-OPF solutions, while [7] utilizes a recurrent neural network (RNN) to predict solutions with intertemporal constraints. Furthermore, [8] employs a DNN architecture to learn the set of active constraints for DC-OPF.

Recent works aim to incorporate OPF constraints into deep learning-based models. One approach [3] combines deep learning and Lagrangian duality via OPF dual variables in the loss function to encourage feasible solution prediction. Other approaches directly enforce OPF constraints during the learning process. For instance, [2] uses a DNN to predict a partial OPF solution and solve for the remaining outputs using power flow equations. Additionally, [9] extends this approach by employing implicit layers for reasoning about hard constraints, with worst-case performance guarantees analyzed in [10].

While these approaches approximate high-quality solutions faster than traditional solvers, a deeper comprehension of their effectiveness and reliability is still lacking. This paper initiates the process of unraveling these insights. Importantly, this paper does **not** aim to introduce new ML techniques for learning AC-OPFs, but rather seeks to elucidate the conditions under which ML methods successfully approximate AC-OPFs.

II. PRELIMINARIES

Optimal Power Flow. A power network is viewed as a graph (N, E) where the set of nodes N describes n buses and the edges E describe e lines. Here E is a set of directed arcs and E^R denotes the reverse direction of the arcs in E . The power

This research is supported by National Science Foundation awards 2041835, 2143706, 2242930, 2242931.

Model 1: The AC Optimal Power Flow Problem (AC-OPF)

variables: $S_i^g, V_i \quad \forall i \in N, \quad S_{ij} \quad \forall (i, j) \in E \cup E^R$

minimize: $\sum_{i \in N} c_{2i}(\Re(S_i^g))^2 + c_{1i}\Re(S_i^g) + c_{0i} \quad (1)$

subject to: $v_i^l \leq |V_i| \leq v_i^u \quad \forall i \in N \quad (2)$

$-\theta_{ij}^{\Delta} \leq \angle(V_i V_j^*) \leq \theta_{ij}^{\Delta} \quad \forall (i, j) \in E \quad (3)$

$S_i^{gl} \leq S_i^g \leq S_i^{gu} \quad \forall i \in N \quad (4)$

$|S_{ij}| \leq s_{ij}^u \quad \forall (i, j) \in E \cup E^R \quad (5)$

$S_i^g - S_i^d = \sum_{(i,j) \in E \cup E^R} S_{ij} \quad \forall i \in N \quad (6)$

$S_{ij} = Y_{ij}^* |V_i|^2 - Y_{ij}^* V_i V_j^* \quad \forall (i, j) \in E \cup E^R \quad (7)$

generated and the load demand in bus $i \in N$ are denoted by S_i^g and S_i^d , respectively. S_{ij} describes the power flows associated with line ij . Finally, θ_i represents the phase angles at bus i .

The AC power flow equations are based on complex quantities for current I , voltage V , admittance Y , and power S . The quantities are linked by constraints expressing Kirchhoff's Current Law (KCL), i.e., $I_i^g - I_i^d = \sum_{(i,j) \in E \cup E^R} I_{ij}$, Ohm's Law, i.e., $I_{ij} = Y_{ij}(V_i - V_j)$, and the definition of AC power, i.e., $S_{ij} = V_i I_{ij}^*$. Combining these properties yields the AC Power Flow equations, i.e., (6) and (7) in Model 1. The objective function (1) captures dispatch cost. Constraints (2)-(3) capture complex voltage operational constraints, and (4)-(5) enforce the generator output and line flow limits. Finally, constraints (6) capture KCL and (7) capture Ohm's Law.

III. OPF LEARNING GOALS

The goal of this paper is to analyze the effectiveness of learning an OPF mapping $\mathcal{O} : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$: Given the loads $\{S_i^d\}_{i=1}^n$ (vectors of active and reactive power demand), predict the set-points $\{(\Re(S_i^g), |V_i|)\}_{i=1}^N$ of the generators, i.e., their active power (p^g) and the voltage magnitude ($|V|$) at their buses. Here, p^g and v are used as a shorthand for $\Re(S^g)$ and $|V|$.

The input of the learning task is a dataset $\mathcal{D} = \{(\mathbf{x}_\ell, \mathbf{y}_\ell)\}_{\ell=1}^N$, where $\mathbf{x}_\ell = S^d$ and $\mathbf{y}_\ell = (p^g, v)$ represent the ℓ^{th} observation of load demands and generator set-points which 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 following empirical risk minimization problem

$$\text{minimize: } \sum_{\ell=1}^N \mathcal{L}(\mathbf{y}_\ell, \hat{\mathcal{O}}(\mathbf{x}_\ell)) \quad (8a)$$

$$\text{subject to: } C(\mathbf{x}_\ell, \hat{\mathcal{O}}(\mathbf{x}_\ell)), \quad (8b)$$

where the loss function is specified by

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \|p^g - \hat{p}^g\|^2 + \|v - \hat{v}\|^2,$$

and $C(\mathbf{x}, \hat{\mathbf{y}})$ holds if there exists voltage angles and reactive power generated that produce a feasible solution to the OPF constraints with $\mathbf{x} = S^d$ and $\hat{\mathbf{y}} = (\hat{p}^g, \hat{v})$, where the *hat* notation is adopted to denote the predictions of the model.

A key difficulty of this learning task is the presence of the nonlinear feasibility OPF constraints. The approximation $\hat{\mathcal{O}}$ will typically focus on minimizing (8a) while ignoring the OPF constraints or using penalty-based methods [3]. Its predictions will thus not guarantee the satisfaction of the problem constraints.

As a result, the validation of the learning task uses a load flow computation Π_C that, given a prediction $\hat{\mathbf{y}} = \hat{\mathcal{O}}(\mathbf{x}_\ell)$, computes its projection onto the constraint set C , i.e., the closest feasible generator set-points $\Pi_C(\hat{\mathbf{y}}) = \arg\min_{\mathbf{y} \in C} \|\hat{\mathbf{y}} - \mathbf{y}\|^2$, with C being the OPF constraint set.

IV. DEEP LEARNING PROXIES FOR AC-OPF: ROADMAP

In this paper, the fundamental model assumes an OPF approximation $\hat{\mathcal{O}}$ provided by a feed-forward fully connected (FCC) neural network. This baseline employs 3 hidden layers, each containing $4n$ neurons and using ReLU activations. While this baseline minimizes (8a), it disregards AC-OPF constraints $C(\mathbf{x}_\ell, \hat{\mathbf{y}}_\ell)$. This foundational model, along with its variations detailed in Section I, often yields dependable and precise predictions. However, as later sections will explore, its robustness is not always guaranteed. The analysis categorizes generator complexity based on cost and solution trajectory smoothness, introducing a complexity score to identify more challenging predictions within each category.

The presence of nonlinear constraints has a significant impact on the learning process of the OPF problem. We examine how these constraints affect generator dispatch volatility under varying input loads and objective functions. Additionally, we investigate the characteristics of the learning model, analyzing parameter size, input scales, activation function, and architecture. The next sections shed light on the reasons for these behaviors. Prior to do so, we describe the training data generation setting.

Training Data This study examines learning models trained on NESTA library test cases [11], primarily focusing on IEEE 118, 162, and 300-bus networks for simplicity, but results are consistent across the benchmark set. The ground truth data are derived as follows: For each network, various benchmarks are created by adjusting nominal load $\mathbf{x} = S^d$ within $\pm 20\%$. Using a uniformly sampled *load multiplier* α from $[0.8, 1.2]$, a load vector $\mathbf{x}' = S^{d'}$ is formed by independently perturbing each load value S_i^d with Gaussian noise centered at α , while maintaining $\sum_i S_i^{d'} = \alpha \sum_i S_i^d$. Each dataset entry $(\mathbf{x}', \mathbf{y}' = \mathcal{O}(\mathbf{x}'))$ represents a feasible AC-OPF solution from solving the detailed Model 1. The data are normalized using the per unit (pu) system, and the experiments utilize an 80/20 train-test split, with results reported on the test set.

V. GENERATOR CHARACTERISTICS

A. Neural Network Input and Output Characteristics

This section relates learning challenges to changes in optimal dispatch across varying loading. As shown in Fig. 1, the trajectory of generator set-points under different input load parameters can often be approximated by piecewise linear functions. The objective of the mapping function $\hat{\mathcal{O}}$ is to accurately approximate these piecewise linear functions for each generator. It is intuitive that the more complex the function is to approximate, the more challenging the learning task becomes. To address this, we introduce the following concept:

Definition 1 (Complexity Score). *Given a piecewise linear function $f : \mathbb{R}^k \rightarrow \mathbb{R}$ with p pieces, each of width h_i for $i \in [p]$, the complexity score (CS) of f , CS_f is computed by:*

$$CS_f = p \mu_f \sigma_f^\lambda,$$

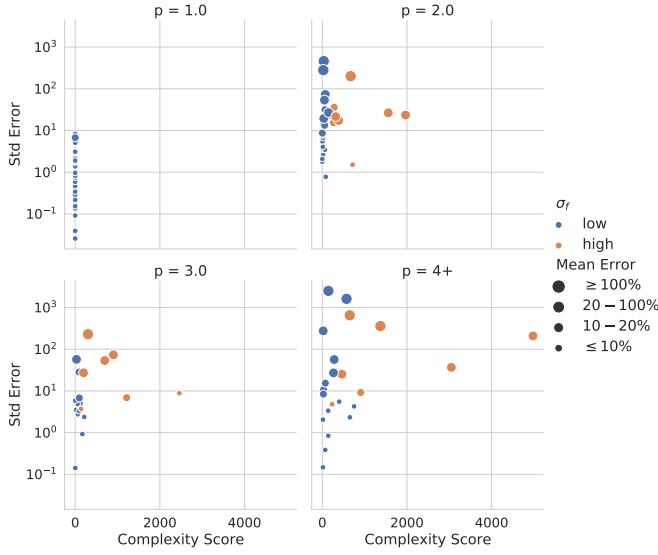


Fig. 2: Standard deviation of prediction error (in percentage) vs complexity score, of an FCC neural network without constraints.

where p is the number of pieces of f , μ_f is the average slope of f over p pieces, σ_f is the standard deviation of $\{L_i\}_{i=1}^p$ (slope of f on piece i), and λ is the weight multiplier.

Complexity score provides insights into the *variability* of a piecewise linear function, and its relevance to the robustness of ReLU-based neural networks will be explored later. Comparing the variability of two piecewise linear functions can be done by comparing their complexity scores using lexicographic ordering. Since generator dispatch can be approximated as piecewise linear, the complexity score of generator g represents the complexity score of the corresponding piecewise linear function for the optimal dispatch $O(S^d)$ of g under varying loads S^d .

Fig. 2 shows relationships between complexity score and the mean and standard deviation of prediction errors. The standard deviation (“Std Error”) represents the absolute values of the errors obtained when comparing the optimal dispatches p^g , associated with different input loads, to their predictions $\hat{p}^g$ obtained from an FCC learning model (described in Section IV). Standard deviation errors are reported as percentages and shown on the y-axis. The generators’ results for IEEE-89, -118, -162, -189, and -300 test cases are presented in ascending order based on their complexity score. σ_f is classified as “low” if it is under 5. As the figure shows, generators with a low number of piecewise linear segments generally have lower complexity score and mean/standard deviation of errors.

DNNs tend to struggle more in predicting the outputs of generators with higher complexity scores. There is generally a strong positive correlation between prediction errors and the complexity of the underlying piecewise linear function that describes the generators’ trajectories. Both the number of pieces p in these functions and the standard deviation components σ_f significantly influence the characterization of prediction errors. Importantly, the standard deviations of prediction errors reflect the notion of *robustness* in network predictions. It is noticeable that generators described by more variable piecewise linear functions also yield less robust predictions. These observations

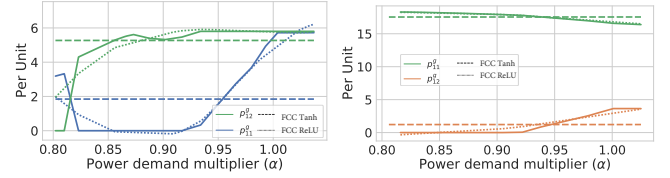


Fig. 3: Accuracy of ReLU FCC vs tanh FCC on selected generators IEEE-162 (left) and IEEE-300 (right).

shed light on why simple fully connected ReLU networks are able to approximate OPF solutions with relatively low error.

B. Which generator outputs are harder to predict?

Next, we investigate the relationship between cost coefficients, generator dispatch patterns, and the complexity score. The analysis utilizes k-means clustering to group generators based on their dispatch values and cost coefficients. Each generator cluster exhibits a distinct cost distribution and dispatch trajectory pattern. Clusters with small cost coefficients (≤ 25 th quantile) display nearly invariant solution trajectories, where generator dispatch remains relatively constant regardless of the load. Generators with cost coefficients in the 50-75th quantile exhibit more variable trajectories, while those with the highest cost coefficients (≥ 75 th quantile) display highly variable trajectories. Notably, these generators are typically inactive and only activated when there is a significant increase in load, resulting in spikes in their solution trajectories. These findings provide a quantitative and explainable factor that helps associate the robustness of DNN prediction errors.

VI. NEURAL NETWORK CHARACTERISTICS

The previous section revealed a correlation between prediction errors and generators with a high number of piecewise linear segments and larger standard deviations in segment magnitude. This section shifts focus to explore the characteristics of neural networks to elucidate the variability of OPF solution predictions across different network types. Specifically, we aim to understand how the selection of activation functions and model size can contribute to minimizing prediction errors.

a) **Activation function:** Fig. 3 compares two FCCs that differ only in their activation functions. The plots display the original generator trajectories (solid lines), approximations learned using ReLU activation (dotted lines), and those learned using tanh activation (dashed lines). The top and bottom plots correspond to selected generators from the IEEE-162 and IEEE-300 test cases, respectively. The results demonstrate that ReLU networks provide better approximations of the original generator trajectories, as they effectively capture piecewise linear functions, as observed in previous research [12]. Therefore, *ReLU activations are generally more suitable for approximating OPF solutions*, given that generator outputs can be described by piecewise linear functions.

b) **Model Capacity:** ReLU FCC models are suitable for predicting OPF solutions, but the required model capacity to accurately represent a target piecewise linear function depends on the number of pieces involved. The following section offers theoretical insights connecting the ability of an FCC model to learn effective approximations of generators’ trajectories across different Complexity Scores.

Theorem 1 (Model Capacity [13]). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a piecewise linear function with p pieces. If f is represented by a ReLU network with depth $k + 1$, then it must have size at least $\frac{1}{2}kp^{\frac{1}{k}} - 1$. Conversely, any piecewise linear function f that is represented by a ReLU network of depth $k + 1$ and size at most s , can have at most $\left(\frac{2s}{k}\right)^k$ pieces.*

The previous analysis establishes a lower bound on model complexity for representing piecewise linear functions, implying that larger models better capture complex relationships between loads and generator set points. Load values are found to relate to the total variation in generator outputs. A theorem [14] connects the approximation errors of piecewise linear functions to the total variation in their slopes.

Theorem 2. *Suppose a piecewise linear function $f_{p'}$, with p' pieces each of width h_k for $k \in [p']$, is used to approximate a piecewise linear f_p with p pieces, where $p' \leq p$. Then the approximation error*

$$\|f_p - f_{p'}\|_1 \leq \frac{1}{2} h_{\max}^2 \sum_{1 \leq k \leq p} |L_{k+1} - L_k|,$$

holds where L_k is the slope of f_p on piece k and $h_{\max}$ is the maximum width of all pieces.

The result indicates that higher variability in a generator's dispatch trajectory, as load increases, makes it more challenging to learn. Thus, for a fixed model size, an increase in the number of piecewise linear portions generally leads to larger errors.

The final observation is the fact that optimization problems typically satisfy a local Lipschitz condition, i.e., if the inputs of two instances are close, then they admit solutions that are close as well, i.e., for $\hat{\mathbf{y}}^{(i)} \in \mathcal{O}(\mathbf{x}^{(i)})$ and $\hat{\mathbf{y}}^{(j)} \in \mathcal{O}(\mathbf{x}^{(j)})$,

$$\|\hat{\mathbf{y}}^{(i)} - \hat{\mathbf{y}}^{(j)}\| \leq C \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|, \quad (9)$$

for some $C \geq 0$ and $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\| \leq \epsilon$, where ϵ is a small value. This observation suggests that, when this local Lipschitz condition holds, it may be possible to generate solution trajectories that are well-behaved and can be approximated effectively. Note that Lipschitz functions can be well approximated by neural networks, as the following result indicates.

Theorem 3 (Approximation [15]). *If $f : [0, 1]^n \rightarrow \mathbb{R}$ is L -Lipschitz continuous, then for every $\epsilon > 0$, there exists some single-layer neural network ρ of size N such that $\|f - \rho\|_{\infty} < \epsilon$, where $N = \left(n + \frac{3L}{\epsilon}\right)$.*

The result above illustrates that the model capacity required to approximate a given function depends to a non-negligible extent on the Lipschitz constant value of the underlying function. Combined with the observations reported in the previous section—showing that a large number of generators have a low complexity score—the results above further illustrate the ability of DNNs to approximate OPF solutions with small average errors. Larger models were expected to be better at learning complex solution trajectories (Theorem 1), but this was not consistently observed in our experiments. Fig. 4 demonstrates this surprising behavior, where increasing the model size did not significantly reduce prediction errors. Notice how prediction error improvements saturate quickly and even

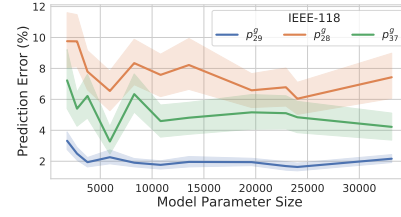


Fig. 4: Prediction error for three key IEEE-118 generators at increasing the FCC model complexity.

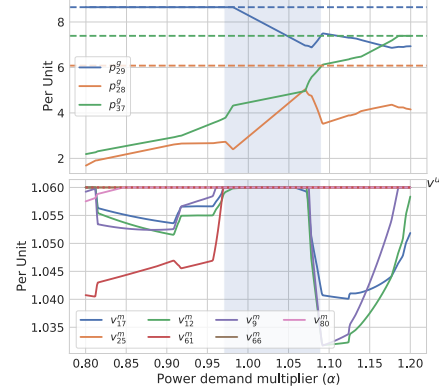


Fig. 5: Non-linear patterns of generators $\alpha \in [0.97, 1.05]$ (top) and associated voltage issues at various buses (bottom) for IEEE-118 case.

increasing the model size substantially does not produce notable error reductions. Theoretical bounds alone do not guarantee the training of effective approximators, as minimizing empirical risk (Equation (8a)) can also introduce errors. This suggests that additional factors, such as the presence of OPF constraints, influence the ability of DNN models to learn accurate OPF approximations, as discussed in the next section.

VII. CONSTRAINTS

Consider the relationship between generators with high complexity indices and corresponding high average prediction errors. Fig. 5 illustrates this using the IEEE-118 case (similar observations hold across all analyzed cases). The figure shows a region of *high variability* involving multiple generators. The top plot shows dispatch of three generators (solid lines) under varying loading¹ (denoted as α) and their upper limits (dashed lines) based on constraints (4) of Model 1. The shaded area highlights a region of high variability, aligning with high errors. The bottom plot presents voltage magnitude trajectories for a subset of buses, with the dashed line indicating the upper bounds imposed by constraints (2). Notably, while the generator dispatch remains within the feasible region, the bottom plot reveals multiple voltage issues. These buses yield voltages that trigger binding constraints (2) within the region of high generator volatility.

These errors are likely due to the DNN's hidden representation not accurately capturing the constraints that govern OPF solutions; e.g., the model is unaware of these constraints. Fig. 6 (first column) illustrates the significant variation in error (not in percentage but the actual value) for generators at buses with binding constraints. The error tends to have a higher mean

¹The load profiles consider a nominal load which is then approximately proportionately increased or decreased within a $\pm 20\%$ range.

TABLE I: Accuracy comparison of a fully connected DNN with and without constraints.

Test case	FCC Without Constraints				FCC With Constraints			
	Bound Vio	KCL Vio	LF Err. (%)	Opt. Gap (%)	Bound Vio	KCL Vio	LF Err. (%)	Opt. Gap (%)
IEEE-30	0.000	0.001	0.128	0.005	0.000	0.081	0.080	0.001
IEEE-118	0.007	0.087	23.59	2.41	0.002	0.052	2.901	0.131
IEEE-162	0.047	0.363	25.83	2.06	0.012	0.038	4.478	0.167
IEEE-300	0.000	0.015	0.205	17.34	0.023	0.0003	1.099	0.327

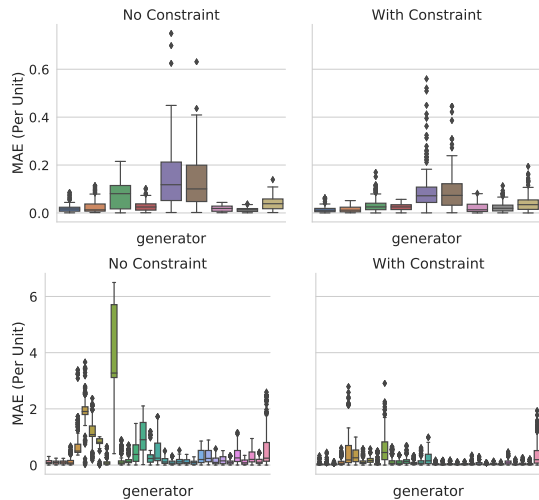


Fig. 6: Error distributions of generators associated with problematic buses (buses with voltage binding constraints) produced by FCC with and without constraints for IEEE-118 (top) IEEE-300 (bottom).

and standard deviation within the binding region. As observed previously [2], [3], [9], we confirm that actively incorporating constraints during training is effective for improving accuracy. The constraints were integrated using a similar model as [3], which uses a Lagrangian dual approach for constraint satisfaction. Note that the *constrained* and baseline models differ only in their loss function, not in the number of parameters.

Table I summarizes the results, displaying the *average prediction errors* $|\hat{\mathbf{y}} - \mathbf{y}|$ across the test set, the *average load flow (LF) errors* $|\Pi_C(\hat{\mathbf{y}}) - \mathbf{y}|$ by comparing feasible predictions $\Pi_C(\hat{\mathbf{y}})$ with optimal quantities $\mathbf{y}$, and the *average optimality gap*, represented as $\frac{|O(\Pi_C(\hat{\mathbf{p}}^s)) - O^* \mathbf{p}^s|}{O^* \mathbf{p}^s}$, where O is the OPF cost. The table also contrasts average absolute constraint violations (in p.u.) for set-point bounds (constraints (2)-(4)) and KCL (constraint (6)). As expected, the constrained model mitigates constraint violations compared to the model solely minimizing mean squared error. Fig. 6 further demonstrates diminished mean and standard deviation errors of generators linked to problematic buses by incorporating a Lagrange dual loss. Notably, constrained DNN variants exhibit greater robustness than unconstrained models. This finding contradicts empirical risk minimization expectations: Constraints introduced via Lagrangian-based penalties augment the loss function with regularization-like terms, which might theoretically reduce model variance. Consequently, integrating constraints in training diminishes prediction errors and error standard deviation, at the expense of heightened training complexity.

VIII. CONCLUSIONS

This study investigates the potential of deep neural networks (DNNs) for approximating OPF solutions. Despite promising

results, accuracy and robustness concerns persist. This paper addresses these gaps by analyzing the connection between generator output volatility and the model's approximation capability. It also explores how model structure influences prediction error, emphasizing the significance of operational and physical constraints for capturing prediction complexity. However, the interpretability of DNNs remains an issue. While these models exhibit impressive predictive capabilities, their limited ability to offer transparent insights into decision-making hampers their adoption in critical applications. The complexity of power systems further complicates capturing the full range of operational conditions, potentially affecting model robustness. Future research avenues could involve investigating alternative network architectures, activation functions, and power networks to broaden and generalize the findings. The authors intend to rigorously integrate domain-specific constraints, possibly through hybrid models that combine physics-based and data-driven approaches, potentially enhancing accuracy and interpretability.

REFERENCES

- [1] K. Baker, "Solutions of DC OPF Are Never AC Feasible," in *ACM International Conference on Future Energy Systems (e-Energy)*, 2021.
- [2] A. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *IEEE SmartGridComm*, Dec. 2020.
- [3] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting AC opf: Combining deep learning and lagrangian dual methods," in *AAAI*, 2020.
- [4] M. Mohammadian, K. Baker, and F. Fioretto, "Gradient-enhanced physics-informed neural networks for power systems operational support," *Electric Power Systems Research*, vol. 223, p. 109551, 2023.
- [5] 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.
- [6] X. Pan, T. Zhao, and M. Chen, "DeepOPF: Deep neural network for dc optimal power flow," in *SmartGridComm*, 2019, pp. 1–6.
- [7] M. Mohammadian, K. Baker, M. H. Dinh, and F. Fioretto, "Learning solutions for intertemporal power systems optimization with recurrent neural networks," in *17th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2022.
- [8] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," in *2019 IEEE Milan PowerTech*, June 2019.
- [9] P. L. Donti, D. Rolnick, and J. Z. Kolter, "Dc3: A learning method for optimization with hard constraints," in *ICLR*, 2020.
- [10] S. Chevalier and S. Chatzivasileiadis, "Global performance guarantees for neural network models of ac power flow," 2022. [Online]. Available: <https://arxiv.org/abs/2211.07125>
- [11] C. Coffrin, D. Gordon, and P. Scott, "NESTA, the NICTA energy system test case archive," *CoRR*, vol. abs/1411.0359, 2014.
- [12] C. Huang, "Relu networks are universal approximators via piecewise linear or constant functions," *Neural Computation*, vol. 32, no. 11, pp. 2249–2278, 2020.
- [13] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," *arXiv preprint arXiv:1611.01491*, 2016.
- [14] J. Kotary, F. Fioretto, and P. Van Hentenryck, "Learning hard optimization problems: A data generation perspective," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [15] K. F. E. Chong, "A closer look at the approximation capabilities of neural networks," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.