

Towards Understanding the Unreasonable Effectiveness of Learning AC-OPF Solutions

My H. Dinh, Ferdinando Fioretto
Syracuse University
Syracuse, NY, USA
{mydinh, ffioretto}@syr.edu

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

Abstract—Optimal Power Flow (OPF) is a fundamental problem in power systems. It is computationally challenging and a recent line of research has proposed the use of Deep Neural Networks (DNNs) to find OPF approximations at vastly reduced runtimes, when compared to those obtained by classical optimization methods. While these works show encouraging results in terms of accuracy and runtime, little is known on why these models can predict OPF solutions accurately, as well as about their robustness. This paper provides a step forward to address this knowledge gap. The paper connects the volatility of the generators outputs to the ability of a learning model to approximate them, it sheds light on the characteristics affecting the DNN models to learn good predictors, and it proposes a new model that exploits the observations made by this paper to produce accurate and robust OPF predictions.

I. INTRODUCTION

The Optimal Power Flow (OPF) problem finds the generator dispatch of minimal cost that meets the demands in a power system. The problem is required to satisfy the AC power flow equations, which are non-convex and nonlinear, and is a core building block in many power system applications. While its resolution has benefited from decades of research in power systems and operational research, the introduction of intermittent renewable energy sources is forcing system operators to adjust the generators set-points with increasing frequency. 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, but, while more efficient computationally, their solutions may be sub-optimal and induce substantial economical losses.

Recently, an interesting line of research has focused on how to approximate AC-OPF using Deep Neural Networks (DNNs) [1]–[3]. Once a DNN is trained, predictions can be computed on the order of milliseconds. While the recent results show that these learning models can approximate the generator set-points of AC-OPF with high accuracy, little is known on why these models can predict OPF solutions accurately, as well as about their predictions robustness. This paper provides a step forward to address this knowledge gap and makes four main contributions.

It firstly asks: *Why are DNNs able to approximate OPF solutions with low errors?* To answer this question, the paper studies the relation between the training data and their target

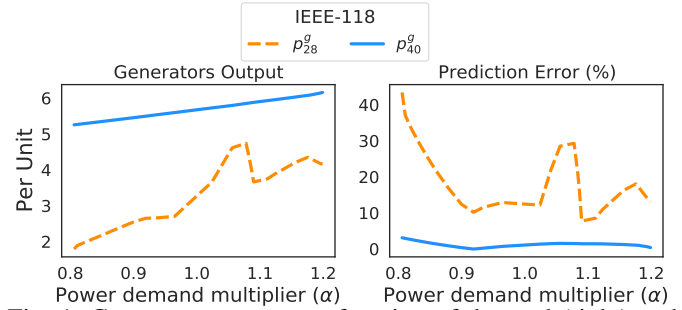


Fig. 1: Generator output as a function of demand (right) and associated predictions (left). Orange (blue) colors show high (low) volatile curves while continuous (dashed) lines depict easy (hard) prediction tasks.

outputs. Figure 1 (left) shows how generator outputs change as a function of the total demand for selected IEEE-118 generators. Notice that the blue curve suggests a linear dependence between the associated generator outputs and the loads, indicating that a simple learning model may effectively capture such behavior, as indeed confirmed in the corresponding low DNN prediction errors reported in Figure 1 (right). *The paper shows that when many generators exhibit this behavior, approximating OPF with DNNs produces accurate results, on average.*

There are, however, also generators whose outputs are inherently more difficult to predict. The orange curve in the figure depicts a much different scenario with a more volatile underlying function. The right plot shows the high prediction error attained, indicating robustness issues. *The paper sheds light on why these behaviors are not easily captured by standard learning models connecting the stability of the training data to the ability of a learning model to approximate it.*

Next, the paper asks: *What are the latent factors that affect the prediction accuracy of these generators?* To address this question, the paper studies which characteristics of the OPF may be responsible for these erroneous predictions, and indicates the need for modeling and predicting the behavior of the OPF engineering and physical constraints during training to capture the complexity of the predictions.

Finally, in light of the robustness issues observed in this study, the paper proposes a new framework that relies on a deep autoregressive Recurrent Neural Network to exploit the data generated by iterative nonlinear solvers during training.

The results show that this framework is not only able to improve the prediction robustness over existing DNN OPF predictors, but also it comes with a reduced memory footprint, thus, enabling it to predict very large instances, overcoming one of the limitations of existing DNN OPF predictors relying on fully-connected networks.

II. RELATED WORK

The use of machine learning to accelerate the resolution of power system optimization procedures has recently seen a growing number of results. A recent survey by Hasan et al. [4] summarizes the development in the area.

In particular, Pan et al. [5] explore DNN architectures for predicting DC-OPFs, a linear approximation of the full AC model. Deka et al. [6] and Ng et al. [7] use a DNN architecture to learn the set of active constraints. By exploiting the linearity of the DC-OPF problem, once the set of relevant active constraints is identified, an exhaustive search can be used to find a solution that satisfies the active constraints. A deep learning approach for AC-OPFs is also proposed by Yang et al. [8] to predict voltages and flows. This approach focuses on specific operational constraints while dismissing other physical and engineering constraints.

Other recent approaches have attempted to incorporate structure from OPF constraints into deep learning-based models. For instance, Fioretto et al. [3] propose a learning method which combines deep learning and Lagrangian duality, incorporating information about OPF dual variables into the learning loss function to promote the prediction of feasible solutions. Other approaches focus on enforcing OPF constraints directly within the learning process. For instance, Zamzam and Baker [2] use a DNN to predict a partial OPF solution, and then solve for the remaining outputs using power the flow equations. Donti et al. [9] extended this approach though the use of implicit layers which allows a DNN to reason about the hard constraints.

While these proposals have clearly shown that it is possible to approximate OPF solutions of high quality, and in vastly reduced computational times when compared to those required by traditional optimization solvers, a complete understanding of the reasons for the effectiveness of these learning models is missing. The rest of the paper provides a first step toward addressing this knowledge gap.

III. PRELIMINARIES

Optimal Power Flow. *Optimal Power Flow (OPF)* is the problem of determining the least-cost generator dispatch that meets the demands in a power network. 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 transmission lines. Here E is a set of directed arcs and E^R is used to denote the arcs in E but in reverse direction.

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.,

Model 1 The AC Optimal Power Flow Problem (AC-OPF)	
variables:	$S_i^g, V_i \forall i \in N, S_{ij} \forall (i, j) \in E \cup E^R$
minimize:	$\sum_{i \in N} c_{2i}(\Re(S_i^g))^2 + c_{1i}\Re(S_i^g) + c_{0i}$ (1)
subject to:	$v_i^l \leq V_i \leq v_i^u \forall i \in N$ (2)
	$-\theta_{ij}^A \leq \angle(V_i V_j^*) \leq \theta_{ij}^A \forall (i, j) \in E$ (3)
	$S_i^{g,l} \leq S_i^g \leq S_i^{g,u} \forall i \in N$ (4)
	$ S_{ij} \leq s_{ij}^u \forall (i, j) \in E \cup E^R$ (5)
	$S_i^g - S_i^d = \sum_{(i,j) \in E \cup E^R} S_{ij} \forall i \in N$ (6)
	$S_{ij} = Y_{ij}^* V_i ^2 - Y_{ij}^*V_i V_j^* \forall (i, j) \in E \cup E^R$ (7)

$S_{ij} = V_i I_{ij}^*$. Combining these three properties yields the AC Power Flow equations, i.e.,

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

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

These non-convex nonlinear equations are the core building blocks in many power system applications. Practical applications typically include various operational constraints on the flow of power, which are captured in the AC OPF formulation in Model 1. The objective function (1) captures the cost of the generator dispatch. Constraints (2) and (3) capture the voltage and phase angle difference operational constraints. Constraints (4) and (5) enforce the generator output and line flow limits. Finally, constraints (6) capture KCL and constraints (7) capture Ohm's Law. Notice that this is a non-convex nonlinear optimization problem and is NP-Hard [10]. Therefore, significant attention has been devoted to finding efficient approximation of Model 1.

Deep Learning Models. Supervised Deep Learning 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 [11]. Feed-forward neural networks are basic DNNs where the layers are fully connected and the function connecting the layer is given by $o = \sigma(Wx + b)$, where $x \in \mathbb{R}^n$ and is the input vector, $o \in \mathbb{R}^m$ the output vector, $W \in \mathbb{R}^{m \times n}$ a matrix of weights, and $b \in \mathbb{R}^m$ a bias vector. The function $\sigma(\cdot)$ is often non-linear (e.g., a rectified linear unit (ReLU)).

IV. 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 and the voltage magnitude at their buses. In the following p^g and v are used as a shorthand for $\Re(S^g)$ and $|V|$.

The input of the learning task is a dataset $\mathcal{D} = \{(x_\ell, y_\ell)\}_{\ell=1}^N$, where $x_\ell = S^d$ and $y_\ell = (p^g, v)$ represent the ℓ^{th} observation of load demands and generator set-points which satisfy $y_\ell = \mathcal{O}(x_\ell)$.

The output is a function $\hat{O}$ that ideally would be the result of the following constrained empirical minimization problem

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

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

where the loss function is specified by

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

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 $\mathbf{x} = \mathbf{S}^d$ and $\hat{\mathbf{y}} = (\hat{\mathbf{p}}^g, \hat{\mathbf{v}})$, where the *hat* notation is adopted to denote the predictions of the model.

One of the key difficulties of this learning task is the presence of the complex nonlinear feasibility constraints in the OPF. The approximation $\hat{O}$ 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 $\hat{\mathbf{y}} = \hat{O}(\mathbf{x}_\ell)$, computes its projection onto the constraint set C , i.e., the closest feasible generator set-points $\Pi_C(\hat{\mathbf{y}}) = \operatorname{argmin}_{\mathbf{y} \in C} \|\hat{\mathbf{y}} - \mathbf{y}\|^2$, with C being the OPF constraint set.

V. BASELINE LEARNING MODEL AND TRAINING DATA

The baseline model for this paper assumes that the OPF approximation $\hat{O}$ is given by a feed-forward fully connected (FCC) neural network, with 3 hidden layers, each of size $4n$ and equipped with ReLU activations. This baseline model minimizes (8a) but ignores the AC-OPF constraints $C(\mathbf{x}_\ell, \hat{\mathbf{y}}_\ell)$. This baseline, as well as its variants described in Section II, often produce reliable and accurate predictions, albeit, as the paper will discuss in the next sections, not always robust.

The next sections shed light on the reasons for these behaviors. Prior to do so, we describe the training data generation setting.

Training Data The paper analyzes the learning models behavior trained on test cases from the NESTA library [12]. For presentation simplicity, the analysis focuses primarily on the IEEE 118, 162 and 300-bus networks. However, the results are consistent across the entire benchmark set. The ground truth data are constructed as follows: For each network, different benchmarks are generated by altering the amount of nominal load $\mathbf{x} = \mathbf{S}^d$ within a range of $\pm 20\%$. For a given *load multiplier* α sampled uniformly in the interval $[0.8, 1.2]$, a load vector $\mathbf{x}' = \mathbf{S}^{d'}$ is generated by perturbing each load value S_i^d independently with additive Gaussian noise centered in α and such that $\sum_i S_i^{d'} = \alpha \sum_i S_i^d$. A network value that constitute a dataset entry $(\mathbf{x}', \mathbf{y}' = O(\mathbf{x}'))$ is a feasible OPF solution obtained by solving the AC-OPF problem detailed in Model 1. The data are normalized using the per unit (pu) system. The experiments use a 80/20 train-test split and report results on the test set.

VI. VOLATILITY ANALYSIS OF THE GENERATORS DISPATCH

The first aspect being investigated concerns *why deep learning models are able to approximate OPF solutions with low error*. To answer this question, this section first analyzes the change in magnitude of the optimal generators dispatch at varying of the input loads and then relates this analysis to the complexity of learning to approximate the generators dispatch. Finally, the section will show that, for many test cases analyzed, the generators outputs exhibit low *volatility*, enabling deep learning models to approximate them well.

The following discussion assumes that the data point set $\{\mathbf{x}_\ell\}_{\ell=1}^N$ is equipped with an ordering relation $\leq$ such that $\mathbf{x} \leq \mathbf{x}' \Rightarrow \|\mathbf{x}\|_p \leq \|\mathbf{x}'\|_p$ for some p -norm ($p \geq 1$). Since the training data is generated by increasing or decreasing the network demand at each bus the ordering relation naturally applies to this domain.

Observe that, as illustrated in the motivating Figure 1, the solution trajectory associated with the generator set-points on various input load parameters can often be naturally approximated by piecewise linear functions. The goal of the mapping function $\hat{O}$ is thus to approximate as best as possible these piecewise linear functions associated with each generator's output. Intuitively, the more volatile the function is to approximate, the harder the associated learning task will be. This aspect will be illustrated more formally in Section VII. To analyze this concept, the paper introduces the following notion.

Definition 1 (Complexity Index). *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 index (CI) of f is a pair $CI_f = (p, \omega)$, with p being the number of its pieces and*

$$\omega = \frac{1}{p} \sum_{i=1}^p h_i |L_i - L_{i-1}|,$$

where L_i is the slope of f on piece i . Value ω describes the weighted average change in the slopes of f .

The complexity index allows us to reason about the *volatility* of a piecewise linear function. It will become apparent later how this concept relates to the learning ability of ReLU neural networks. Notice that the two piecewise linear functions can be compared, in terms of their volatility, by their associated complexity indexes using a lexicographic ordering.

Since the generator dispatch trajectory can be approximated by a piecewise linear function, we refer to the complexity index of a generator g to denote the complexity index of the induced piecewise linear function of the optimal dispatch $O(\mathbf{S}^d)$ of g at varying of the loads $\mathbf{S}^d$ in the domain of interest.

Figure 2 illustrates the average prediction errors (in percentage) obtained when comparing the optimal dispatches $\mathbf{p}^g$, associated with different input load, to their predictions $\hat{\mathbf{p}}^g$ obtained by an FCC learning model as described in section V. The figure reports the errors of each generators for test cases IEEE-118, -162, and -300 ordered by their (normalized) CI values. Notice the strong correlation between the CI values and

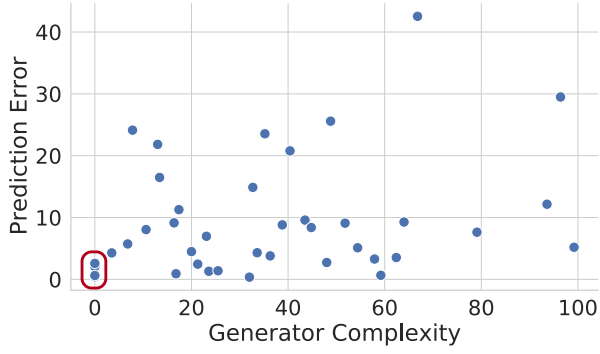


Fig. 2: Prediction errors, in percentage, of an FCC neural network. Generators are sorted by their CI values. The red box encloses generators with CI value $(1, \omega)$.

Test case	CI p-value (%)			Pred. Err. (%)	LF Err. (%)	Opt. Gap (%)
	1	≤ 2	≤ 3			
IEEE-30	100.0	100.0	100.0	0.12	0.128	0.005
IEEE-118	57.9	73.7	84.2	8.47	27.16	2.41
IEEE-162	25.0	41.7	66.7	5.76	25.09	2.06
IEEE-300	36.8	63.1	82.4	15.8	43.49	6.23

TABLE I: CI and average errors of FCC model.

the model errors: *More volatile generator dispatch trajectories correspond to generally less precise model predictions*. This observation connects the generators volatility with the hardness of the model to capture its output trajectory.

In particular, notice that generators with a CI index of $(1, \omega)$ (enclosed in a red box in the figure) can be approximated by linear functions, and, thus, represent an ideal case for the learning task. The underlying models can be described using only two parameters (representing slope and intercept) and are generally characterized by low prediction errors.

This aspect is further emphasized in Table I. The table reports the cumulative amount of generators whose trajectories are represented by a piecewise linear function with 1, at most 2, and at most 3 pieces, the *average prediction errors* $\|\hat{\mathbf{y}} - \mathbf{y}\|_1$ over the test set, the *average load flow (LF) errors* $\|\Pi_C(\hat{\mathbf{y}}) - \mathbf{y}\|_1$ which compare the closest feasible solution $\Pi_C(\hat{\mathbf{y}})$ of the predictions $\hat{\mathbf{y}}$ with the optimal quantities $\mathbf{y}$, and the *average optimality gap*, as $\frac{|O(\Pi_C(\hat{\mathbf{p}}^s)) - O^*(\mathbf{p}^s)|}{O^*(\mathbf{p}^s)}$, with O being the associated OPF cost. First, note that many of the generators trajectories in the test cases analyzed can indeed be approximated by linear functions (i.e., their complexity index is $(1, \omega)$) or have CI with a low p value (expressing the number of pieces of the associated piecewise linear function). Notice also that the predictions and load flow errors as well as the optimality gap correlates positively with the amount of generators with larger complexity indexes.

These observations shed light on why even simple fully connected ReLU networks, are able to approximate OPF solutions with relatively low average errors. The next section provides theoretical arguments to justify these observations.

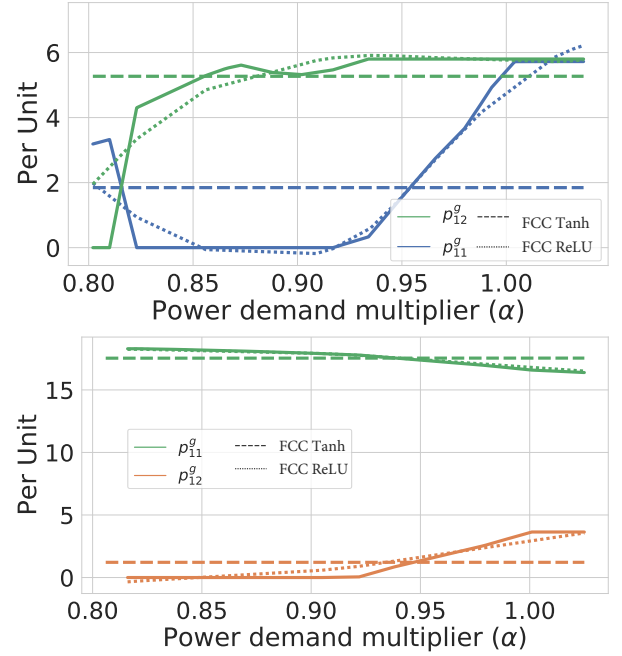


Fig. 3: Accuracy of ReLU FCC vs Tanh FCC on selected generators IEEE-162 (left) and IEEE-300 (right).

VII. CI AND PREDICTION ACCURACY: THEORETICAL INSIGHTS

As observed above, the trajectory of the generators outputs can be described by piecewise linear functions. Next, note that ReLU networks capture piecewise linear functions [13].

This observation justifies the choice of ReLU activation function for DNNs used to approximate OPF solutions. Figure 3 illustrates a comparison between two FCCs differing only in the type of activation functions they adopt. The plots show the original generators trajectories (solid lines), the approximations learned with a ReLU network (dotted lines) and those learned with a Tanh network (dashed lines). The top and bottom plots show results for selected generators from, respectively, the IEEE-162 and IEEE-300 test cases. Notice how the ReLU network predictions can represent piecewise linear functions that better approximate the original generator trajectories, when compared to those obtained from a Tanh network.

While these ReLU FCC models are compatible with the task of predicting the solutions of an OPF problem, the model capacity required to represent a target piecewise linear function exactly depends directly on the number of constituent pieces. Next, this section provides theoretical insights to link the ability of an FCC model to learn good approximations of generators trajectories of various CI complexities.

Theorem 1 (Model Capacity [14]). *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.*

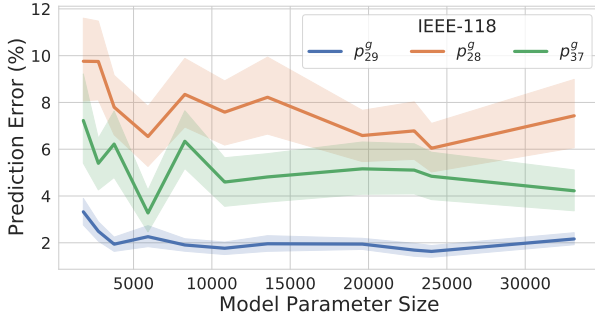


Fig. 4: Prediction error for three key IEEE-118 generators at increasing of the FCC model complexity.

The result above provides a lower bound on the model complexity to represent a given piecewise linear function. It implies that larger models may be able to better capture more complex relationships between inputs (loads) and output (generator set-points) values.

The second observation is from [15]. It relates the load values with the total variation of the generators outputs. The following theorem bounds the approximation error when using continuous piecewise linear functions: it connects the approximation errors of a piecewise linear function with the *total variation in its 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 above indicates that the more volatile the generators trajectory, the harder it will be to learn. Moreover, for a neural network of fixed size, the more volatile the generator trajectory, the larger the approximation error will be in general.

Combined with the observations reported in the previous section—showing that, for the test cases analyzed, a large number of generators have a low complexity index—the results above further illustrate the ability of DNNs to approximate OPF solutions with small average errors.

VIII. ROBUSTNESS ISSUES

The results in the previous section are bounds on the ability of neural networks to represent generic functions. In practice, however, these bounds rarely guarantee the training of good approximators, as the ability to minimize the empirical risk (see Equation (8a)) is often another significant source of error. This section demonstrates that there are also additional factors that may affect the ability of the DNN models to learn good OPF approximators, including the presence of the OPF constraints.

First notice that, in theory, it is to be expected that larger DNN models will be better suited to learning more complex

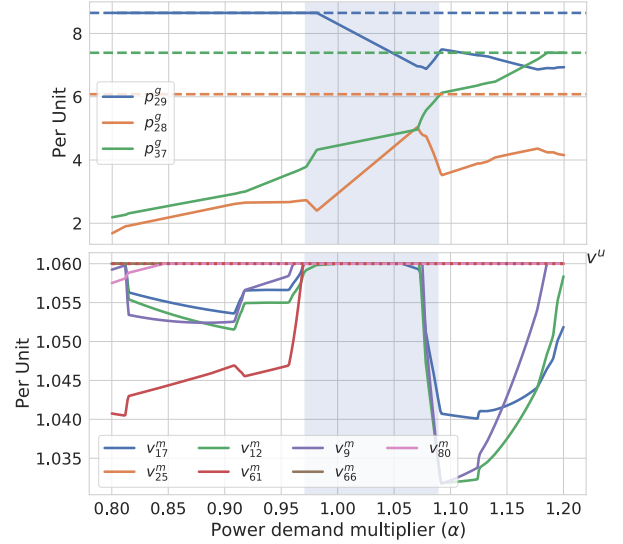


Fig. 5: Non-linear patterns of generators around load multiplier $\alpha \in [0.97, 1.05]$ (top) and associated voltage bounds issues at various buses.

solution trajectories (Theorem 1). However, this aspect was not observed in our experiments. Figure 4 illustrates this surprising behavior. It reports the prediction errors associated with the trajectories of three IEEE-118 generators at the varying of the model size. Notice how prediction errors improvements saturate quickly and that even increasing the model size substantially does not produce notable error reductions. The reminder of the section seeks to answer *why this behavior occurs*.

To answer this question the paper analyzes generators with large complexity indexes. Indeed, high prediction errors pertain commonly to the solution trajectories associated with these generators.

Figure 5 illustrates an example for the IEEE-118 test case, but these observations are consistent across the whole benchmark set analyzed. The figure highlights a region of *high volatility* involving several generators. The top plot shows the dispatch trajectories of three generators (continuous lines) at varying of the input load multipliers α and their associated upper bound limits (dashed lines) (see constraints (4) of Model 1). The shaded area highlights the region in which large volatilities are observed. This region also correspond to the portion associated with the higher dispatch error predictions. The bottom plot shows the trajectories of the voltage magnitude values for a selection of buses. The upper bounds (constraints (2)) are illustrated with a dashed line. Notice that, while the generators dispatch are within the feasible operating regions, the bottom plot highlights the presence of voltage issues on several buses. The reported buses all are associated with voltage magnitudes value which results in binding constraints (2) in the region of high volatility of the generators considered.

These prediction errors are thus likely to arise as the hidden representation of the DNN does not accurately learn the operational and physical constraints which regulate the

behavior of the OPF solutions. In other words, the model is unaware of these constraints.

Therefore, as investigated by several authors (including, [2], [3], [9]) this work found that actively exploiting the problem constraints during training to be an effective mechanism to enhance the model accuracy. The constraints were added using a model similar to [3] which encourages the satisfaction of the OPF constraints by the means of a Lagrangian dual approach. Notice that the *constrained* and baseline models differ solely in the loss function, and not in the number of their parameters.

Table II summarizes the results. It compares the average absolute constraint violations (in p.u.) for the set-points bounds (constraints (2) and (4)) and the KCL (constraint (6)), the load flow (LF) distance of the predictions $\hat{\mathbf{y}}$ to their optimal dispatches $\mathbf{y}$, and the optimality gaps, as defined in Section VI. Notice how the constrained model reduces the constraint violations, when compared to the baseline, as well as increases the associated prediction accuracy.

This aspect is also evident in Figure 6, which compares the prediction trajectories of the FCC model with (yellow curves) and without (red curves) constraints for two high-complexity IEEE-300 generators. Notice that the constrained model predictions follow more closely the original trajectories when compared to the simple model.

This aspect is surprising from an empirical risk minimization perspective: Including constraints using Lagrangian-based penalties adds additional terms to the loss function which can be interpreted as further regularizing terms, and thus, it may be expected they would reduce the model variance further.

However, Figure 6 also highlights some drawbacks of the constrained model. Despite its improved accuracy (and its ability to approximate precisely many *easy* generators) its predictions tend to discard the rapid changes in trajectories of the generators outputs (see bottom plot). From a data representation point of view, these cases (where the change in trajectory occurs) represent *outliers* and thus are hard to predict. This observation motivates the introduction of a novel model described next.

IX. A NOVEL RNN-BASED LEARNING FRAMEWORK

The issue observed above could be partially addressed by providing additional training data to the learning task with the goal of more suitably representing the inputs associated with the *outlier* set-points. Creating this data is, however, a very challenging task. It is unknown a-priori which set-point, within a trajectory, may be uncommon. Additionally, generating the input loads associated to a desired set-point would be an extremely challenging operation.

While generating additional targeted data is thus unfeasible, this section notices that iterative solvers, typically adopted to solve non-linear programs, generate a solution at each iteration of their execution. For example, IPOPT [16], a popular nonlinear solver adopted in this paper to generate OPF solutions, implements a primal-dual interior point line search filter method to find a local optimal solution to a given problem instance.

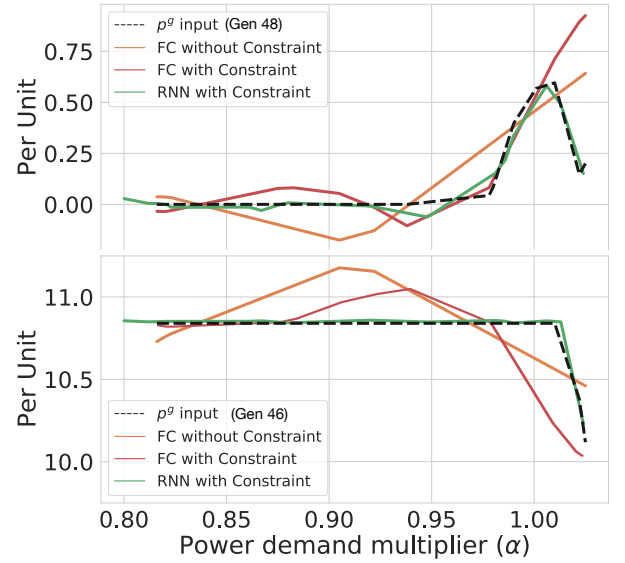


Fig. 6: IEEE-300. Optimal generators trajectory (red) for generator 36 (top) and 48 (bottom). Predictions: FCC without constraints (orange), FCC with constraints green), and RNN with constraint (blue).

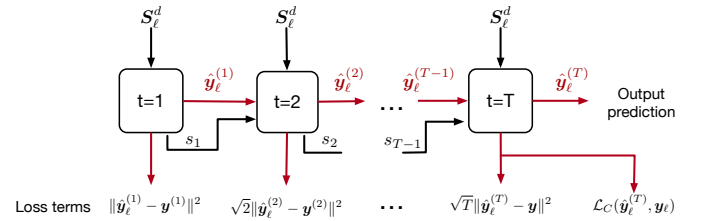


Fig. 7: RNN model Overview.

The underlying idea of the proposed model is thus to exploit these *solution trajectories* during training.

To do so, this section introduces a DNN model for OPF predictions which relies on deep autoregressive Recurrent Neural Networks (RNN). RNNs are a powerful tool to learn from sequential data and have been vastly adopted in domains including natural language processing and computer vision [17]–[19]. An autoregressive model is typically used in time-series modeling where the current time step value z_t depends linearly on some value $z_{t'}$ with $t' < t$. Similarly, autoregressive RNNs condition the prediction of the current time step on the predictions of the previous steps. They thus are a natural fit for the intended purpose.

The proposed model is illustrated in Figure 7. The model is composed by T sequential Long Short Memory Term (LSMT) units. For unit $t \in [T]$, the model takes as input the demands $\mathbf{x} = \mathbf{S}^d$ as well as the embedding $\mathbf{y}^{(t-1)}$ outputted by unit $t-1$, and the *state* s_{t-1} of unit $t-1$. The first unit $t=1$ is special and only considers the input demands $\mathbf{x}$. The model uses the following loss:

$$\sum_{t=1}^N \sum_{t=1}^T \sqrt{t} \mathcal{L}(\mathbf{y}_t^{(i)}, \hat{\mathbf{y}}_t^{(i)}) + \mathcal{L}_C(\mathbf{y}_t, \hat{\mathbf{y}}^{(T)}), \quad (9)$$

Test case	FCC Without Constraint				FCC With Constraint				RNN With Constraint			
	Bound Vio	KLC Vio	LF Err. (%)	Opt. Gap (%)	Bound Vio	KLC Vio	LF Err. (%)	Opt. Gap (%)	Bound Vio	KLC Vio	LF Err. (%)	Opt. Gap (%)
IEEE-30	0.000	0.001	0.128	0.005	0.000	0.081	0.080	0.001	0.0	0.13	0.384	0.270
IEEE-118	0.007	0.087	23.59	2.41	0.003	7.16	14.34	4.910	0.002	0.052	2.901	0.131
IEEE-162	0.047	0.363	25.83	2.06	0.016	8.35	19.17	2.191	0.012	0.038	4.478	0.167
IEEE-300	0.000	0.015	0.205	17.34	6.23	0.021	11.56	16.89	0.023	0.0003	1.099	0.327

TABLE II: Accuracy comparison: FCC with and without constraints and RNN models.

Test Case	FCC	RNN	Test Case	FCC	RNN
IEEE-118	11.4	0.007	IEEE-300	47.8	0.04
IEEE-162	14.8	0.005	PEGASE-1354	154	0.32
EDIN-189	3.4	0.013	RTE-2868	2907	1.64

TABLE III: RNN vs FCC: Model parameter size (Million).

where $\mathcal{L}_C$ is the Lagrangian loss involving the prediction from the last unit to encourage constraint satisfaction, equivalently to that adopted by the constrained variant of the FCC model. The $\sqrt{t}$ multiplicative factor is adopted to give larger weights to the latter units. The model returns $\hat{y}^{(T)}$ as its prediction, which is the output of the recurrent final unit.

The predictions of the proposed model are summarized in Table II (right). Notice how the model can reduce the load flow errors and optimality gaps by one order of magnitude when compared with the best FCC results. Notably, the RNN model predictions are much closer to satisfy the KLC than those produced by the constrained version of the FCC model. This is important as KLC are notoriously hard to satisfy for the predictions of DNN models [3]. The ability of this model to capture robustly rare changes in generators trajectory can be appreciated in Figure 6.

Finally, Table III reports a comparison of the number of parameters (proxy to memory footprint) required by the FCC and the proposed RNN models. Notice that the FCC grow very large with the size of the processed test case highlighting scalability issues, as also observed in [20], which reported the inability of these models to fit in memory for test cases larger than 2000 buses. In contrast, the proposed RNN model does not incur this drawback rendering it applicable to very large power systems.

X. CONCLUSIONS

This paper was motivated by the recent development around using deep neural networks (DNN) to approximate the solutions of Optimal Power Flow (OPF) problems. While these learning models show encouraging results, little is known on why they predict OPF solutions accurately, as well as about their predictions robustness. The paper provided a step forward to address this knowledge gap. It studied the connection between the volatility of the generators outputs with the ability of a learning model to approximate it, showing that many test cases are characterized by a large number of generators which are easy to predict. It also showed that operational and physical constraints are necessary to capture the complexity of the predictions. Finally, it proposed a new learning model based on recurrent neural networks, that was not only able improve

the prediction accuracy over existing supervised learning approaches, but also reduced the memory requirements.

ACKNOWLEDGEMENT

This research is partially supported by NSF grant 2007164 and NSF CAREER award 2041835.

REFERENCES

- [1] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," 2019, <https://arxiv.org/pdf/1902.05607>.
- [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 *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.
- [4] 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.
- [5] X. Pan, T. Zhao, and M. Chen, "DeepOPF: Deep neural network for dc optimal power flow," in *SmartGridComm*, 2019, pp. 1–6.
- [6] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," in *2019 IEEE Milan PowerTech*, June 2019.
- [7] Y. Ng, S. Misra, L. Roald, and S. Backhaus, "Statistical learning for DC optimal power flow," in *Power Systems Computation Conference*, 2018.
- [8] Y. Yang, Z. Yang, J. Yu, B. Zhang, Y. Zhang, and H. Yu, "Fast calculation of probabilistic power flow: A model-based deep learning approach," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2235–2244, 2020.
- [9] P. L. Donti, D. Rolnick, and J. Z. Kolter, "Dc3: A learning method for optimization with hard constraints," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.
- [10] A. Verma, "Power grid security analysis: An optimization approach," Ph.D. dissertation, Columbia University, 2009.
- [11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
- [12] C. Coffrin, D. Gordon, and P. Scott, "NESTA, the NICTA energy system test case archive," *CoRR*, vol. abs/1411.0359, 2014.
- [13] C. Huang, "Relu networks are universal approximators via piecewise linear or constant functions," *Neural Computation*, vol. 32, no. 11, pp. 2249–2278, 2020.
- [14] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," *arXiv preprint arXiv:1611.01491*, 2016.
- [15] J. Kotary, F. Fioretto, and P. V. Hentenryck, "Learning hard optimization problems: A data generation perspective," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
- [16] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [17] J. F. Kreider, D. E. Claridge, P. Curtiss, R. Dodier, J. S. Haberl, and M. Krarti, "Building Energy Use Prediction and System Identification Using Recurrent Neural Networks," *Journal of Solar Energy Engineering*, vol. 117, no. 3, pp. 161–166, 08 1995.
- [18] J. Connor, R. Martin, and L. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.
- [19] Y. Liu, C. Gong, L. Yang, and Y. Chen, "Dstp-rnn: A dual-stage attention-based rnn for long-term and multivariate time series prediction," *Expert Systems with Applications*, vol. 143, p. 113082, 2020.
- [20] M. Chatzos, F. Fioretto, T. W. K. Mak, and P. V. Hentenryck, "High-fidelity machine learning approximations of large-scale optimal power flow," *arXiv preprint arXiv:2006.16356*, 2020.