2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) | 978-1-6654-1502-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMARTGRIDCOMM51999.2021.9632308

Physics-Informed Neural Networks for Minimising Worst-Case Violations in DC Optimal Power Flow

Rahul Nellikkath, Spyros Chatzivasileiadis

Abstract—**Physics-informed neural networks exploit the existing models of the underlying physical systems to generate higher accuracy results with fewer data. Such approaches can help drastically reduce the computation time and generate a good estimate of computationally intensive processes in power systems, such as dynamic security assessment or optimal power flow. Combined with the extraction of worst-case guarantees for the neural network performance, such neural networks can be applied in safety-critical applications in power systems and build a high level of trust among power system operators. This paper takes the first step and applies, for the first time to our knowledge, Physics-Informed Neural Networks with Worst-Case Guarantees for the DC Optimal Power Flow problem. We look for guarantees related to (i) maximum constraint violations, (ii) maximum distance between predicted and optimal decision variables, and (iii) maximum sub-optimality in the entire input domain. In a range of PGLib-OPF networks, we demonstrate how physics-informed neural networks can be supplied with worst-case guarantees and how they can lead to reduced worst-case violations compared with conventional neural networks.**

Index Terms— **DC OPF, Physics-Informed Neural Network, Worst-Case Guarantees.**

I. Introduction

Optimal Power Flow (OPF) is a tool that is increasingly used by power system operators, electricity markets, and the rest of the power system industry both for planning and real-time operation. Countless instances of OPF need to be solved when it comes to assessing uncertain scenarios, identifying transmission investments, finding optimal control setpoints, or determining the electricity market clearing. However, the accurate representation of the AC Power Flow equations in the OPF problem renders the problem non-linear and non-convex [1], which usually results in significant challenges related to convergence and long computing times. As a substitute, the DC-OPF approximation is often used to reduce the complexity and improve speed [2]. However, even with DC-OPF, solving for multiple scenarios in a short period could be challenging. This has led to the development of various neural network (NN) architectures for predicting OPF problems. [3] [4]. However, these machine learning algorithms require a considerable amount of good quality datasets to train a neural network with reasonable accuracy, while we especially need OPF results that cover both normal and abnormal situations; this is often challenging to generate. As a result, researchers have investigated different NN training approaches to reduce the dependency on the training dataset size and improve accuracy. A convex relaxation technique to only focus on the regions closer to the security boundary was proposed in [5]. A method to incorporate adversarial examples into the training set to improve performance was introduced in [6]. Furthermore, an input-convex NN was introduced in [7] to improve its performance in unseen data points since the underlying DC-OPF problem is convex. A NN training framework to match both prediction and the sensitivity of the OPF problem was also proposed in [8] to improve data efficiency and convergence.

However, these proposed algorithms could be susceptible to outliers. Considering OPF is often used for safety-critical applications, outliers leading to massive system constraint violations such as line, voltage, or generator limits can have a catastrophic effect on system security. It is not easy to eliminate constraint violations when using NN to estimate the OPF result. To overcome this shortcoming, the generation limits can be enforced on the NN prediction, as Ref. [9] [10] proposed for the AC-OPF problem. However, this could lead to sub-optimality. So, a few methods have also suggested penalizing the constraint violations by adding them directly in the loss terms [11]. A few approaches have also used the Lagrangian dual scheme for DNN training, such as [12], and [13] for AC-OPF. Even then, these studies are trained to minimize the average prediction errors, and so far, none of the proposed machine learning algorithms have supplied any worst-case performance guarantees.

Here, we propose a physics-informed neural network (PINN) algorithm to predict the DC-OPF problem solutions, which will reduce the dependency on the training data set, and improve the worst-case guarantees while improving optimality. PINNs for power system applications is a NN training architecture that uses the underlying physical laws governing the power system model to improve prediction accuracy [14]. By doing that, the NN training is no longer dependent only on the size and quality of the training dataset. Instead, the training procedure can determine the NN optimal parameters based on the actual equations that the NN aims to emulate.

Our contributions in this paper are:

1) For the first time, we propose a PINN architecture which utilizes KKT conditions to predict the DC-OPF problem solutions. The work in this paper is seen as the first necessary step to identify opportunities and challenges towards the application of PINN on AC-OPF problems.
2) Extending our previous work, presented in [15], we develop approaches to determine the worst-case violations of PINN, and propose ways to reduce them.

This paper is structured as follows: Section II describes the optimal power flow problem, introduces the architecture of the physics-informed neural network, and discusses the MILP algorithm used to quantify the worst-case guarantees. Section

Rahul Nellikkath, and Spyros Chatzivasileiadis are with the Center for Electric Power and Energy, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark. e-mail: {rnelli, spchatz}@elektro.dtu.dk.

III presents simulation results demonstrating the performance of PINN. Section IV discusses the possible opportunities to improve the system performance, and concludes.

II. Methodology

A. DC - Optimal Power Flow

DC optimal power flow (DC-OPF) is a simplified linear approximation of the AC-OPF problem. A DC-OPF problem for generation cost minimization in an N_{bus} system with N_g number of generators and N_d number of loads can be represented as follows:

$$\min \mathbf{c^T} \, \mathbf{P_g} \tag{1}$$

$$\sum_{i=1}^{N_g} P_{g,i} - \sum_{i=1}^{N_d} P_{d,i} = 0 \tag{2}$$

$$\mathbf{P_g^{min}} \leq \mathbf{P_g} \leq \mathbf{P_g^{max}} \tag{3}$$

$$|\mathbf{PTDF}(\mathbf{P_g} - \mathbf{P_d})| \leq \mathbf{P_l^{max}} \tag{4}$$

where vector $\mathbf{c^T}$ is the linear cost term of each generator, vector $\mathbf{P_g}$ is the generator active power output and $\mathbf{P_d}$ is the active power demand. The minimum and maximum active power generation limit are denoted by $\mathbf{P_g^{min}}$ and $\mathbf{P_g^{max}}$ respectively, and $\mathbf{P_l^{max}}$ represents the line flow limit. $\mathbf{PTDF}$ is the power transfer distribution factors (for more details, see [16]).

The generation and line flow limits are guaranteed by (3) and (4), and (2) ensures load balance in the system. The corresponding Karush–Kuhn–Tucker (KKT) conditions for the DC-OPF can be formulated as follows:

$$\mathbf{c} + \lambda + \overline{\mu}_{\mathbf{g}} - \underline{\mu}_{\mathbf{g}} + \overline{\mu}_{\mathbf{l}} \mathbf{PTDF} - \underline{\mu}_{\mathbf{l}} \mathbf{PTDF} = 0 \tag{5}$$

$$\overline{\mu}_{g,i} [P_{g,i}^{max} - P_{g,i}] = 0 \tag{6}$$

$$\underline{\mu}_{g,i} [P_{g,i} - P_{g,i}^{min}] = 0 \tag{7}$$

$$\overline{\mu}_{l,j} \left[\mathbf{PTDF_j} (\mathbf{P_g} - \mathbf{P_d}) - P_{l,j}^{max} \right] = 0 \tag{8}$$

$$\underline{\mu}_{l,j} \left[-\mathbf{PTDF_j} (\mathbf{P_g} - \mathbf{P_d}) - P_{l,j}^{max} \right] = 0 \tag{9}$$

$$\overline{\mu}_g, \underline{\mu}_g, \overline{\mu}_l, \underline{\mu}_l \geq 0 \tag{10}$$

$$(2) - (4) \tag{11}$$

where λ, μ_g and μ_l denote the dual variables for (2) - (4) respectively. The stationarity condition is given in (5), and the complementary slackness conditions are described in (6) - (9). These KKT conditions are necessary and sufficient for optimality in the DC-OPF problem [17], given the DC-OPF problem is feasible.

B. Physics-Informed Neural Network

This section introduces the PINN architecture used for predicting the DC-OPF optimal $\mathbf{P_g}$ setpoints, given active power demand $\mathbf{P_d}$ as the input. A neural network is the group of interconnected nodes connecting the input and the output layers, as shown in Fig. 1. There are K number of hidden layers in the NN with N_k number of neurons in the hidden layer k. Each neuron in the NN has a nonlinear activation function linked with them, and the edges connecting the neurons have a weight $\mathbf{w}$ and a bias $\mathbf{b}$ associated with them.

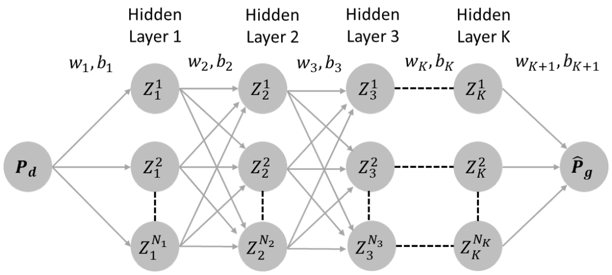

Fig. 1: Illustration of the neural network architecture to predict the optimal generation outputs $\mathbf{\hat{P}_g}$ using the active power demand $\mathbf{P_d}$ as input: There are K hidden layers in the NN with N_k neurons each. Where k = 1, ...,K.

The output of each layer in the neural network can be denoted as follows:

$$Z_{k+1} = \pi(w_{k+1} Z_k + b_{k+1}) \tag{12}$$

where Z_{k+1} is the output of layer $k+1$, w_{k+1} and b_{k+1} are the weights and biases connecting layer k and $k+1$. π is the nonlinear activation function. As in most recent literature, in this work we chose the ReLU as the non-linear activation function, as it is observed to accelerate the NN training [18]. The ReLU activation function will return the input if the input is positive and return zero if the input is negative. The ReLU activation function can be formulated as follows:

$$\hat{Z}_{k+1} = w_{k+1} Z_k + b_{k+1} \tag{13}$$

$$Z_{k+1} = max(\hat{Z}_{k+1}, 0) \tag{14}$$

When we use a NN to predict the optimal generator setpoints, these weights and biases are trained to predict the generation values of the optimal setpoint for the DC-OPF problem.

In a physics-informed neural network, the physical equations governing the problem will be incorporated into the NN loss function (see [19], and our previous work [14] for power systems applications) . In the case of a DC-OPF problem, the KKT conditions given in (5) - (11) act as a set of necessary and sufficient conditions that the optimal value shall satisfy. To incorporate the KKT conditions in the NN training (and by that also render it a PINN), we denote the discrepancies from zero in (5) - (11) with ϵ, as shown in (15)-(18), and minimize ϵ as part of the NN loss function, as shown in (19). The proposed PINN structure is given in Fig. 2. The dual variables required for calculating the discrepancy in the KKT conditions are predicted using a separate set of hidden layers.

The discrepancy in KKT conditions are calculated as follows:

$$\epsilon_{stat} = |\mathbf{c} + \hat{\lambda} + \hat{\overline{\mu}}_{\mathbf{g}} - \underline{\hat{\mu}}_{\mathbf{g}} + \hat{\overline{\mu}}_{\mathbf{l}} \mathbf{PTDF} - \underline{\hat{\mu}}_{\mathbf{l}} \mathbf{PTDF}| \tag{15}$$

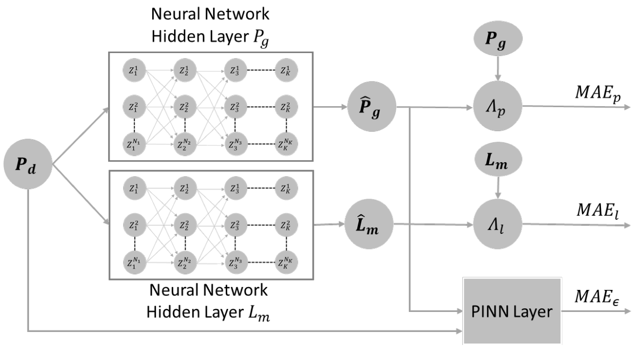

Fig. 2: Illustration of the physics-informed neural network architecture to predict the optimal generation outputs $\mathbf{\hat{P}_g}$ and dual variables $\mathbf{\hat{L}_m}$ using the active power demand $\mathbf{P_d}$ as input. Hidden layers used for predicting $\mathbf{\hat{P}_g}$ and $\mathbf{\hat{L}_m}$ are separate and not connected to each other. During training, the NN weights (W) and biases (b) are adjusted according to loss function (19), which minimizes the mean absolute errors MAE_p, MAE_l and MAE_ϵ

$$\begin{aligned}\epsilon_{comp} = &\sum_{i=1}^{N_g}\Big[|\hat{\bar{\mu}}_{g,i}\Big(P_{g,i}^{max}-\hat{P}_{g,i}\Big)|+|\hat{\underline{\mu}}_{g,i}\Big(\hat{P}_{g,i}-P_{g,i}^{min}\Big)|\Big]\\ &+\sum_{j=1}^{N_l}\Big[|\hat{\bar{\mu}}_{l,j}\Big(\mathbf{PTDF_j}(\mathbf{\hat{P}_g}-\mathbf{P_d})-\mathbf{P_{l,j}^{max}}\Big)|\Big]\\ &+\sum_{j=1}^{N_l}\Big[|\hat{\bar{\mu}}_{l,j}\Big(-\mathbf{PTDF_j}(\mathbf{\hat{P}_g}-\mathbf{P_d})-\mathbf{P_{l,j}^{max}}\Big)|\Big]\end{aligned} \tag{16}$$

$$\epsilon_{dual} = \pi(\hat{\bar{\mu}}_g)+\pi(\hat{\underline{\mu}}_g)+\pi(\hat{\bar{\mu}}_l)+\pi(\hat{\underline{\mu}}_l) \tag{17}$$

$$\begin{aligned}\epsilon_{prim} = &\sum_{i=1}^{N_g}\Big[\pi(P_{g,i}^{max}-\hat{P}_{g,i})+\pi\Big(\hat{P}_{g,i}-P_{g,i}^{min}\Big)\Big]\\ &+|\sum_{i=1}^{N_g}\hat{P}_{g,i}-\sum_{i=1}^{N_d}P_{d,i}|\\ &+\sum_{j=1}^{N_l}\pi\Big(\mathbf{PTDF_j}(\mathbf{\hat{P}_g}-\mathbf{P_d})-P_{l,j}^{max}\Big)\\ &+\sum_{j=1}^{N_l}\pi\Big(-\mathbf{PTDF_j}(\mathbf{\hat{P}_g}-\mathbf{P_d})-P_{l,j}^{max}\Big)\end{aligned} \tag{18}$$

where $\hat{\lambda}$, $\hat{\mu}_g$ and $\hat{\mu}_l$ are the dual variables predicted using the second set of hidden layers, represented by $\mathbf{L_m}$ in Fig. 2. The ϵ_{stat} is the absolute value of the stationarity condition, and ϵ_{com} is the sum of all the complementary slackness conditions given in (6)-(9). The ReLU activation function, represented by π, is used to measure the constraint violation, ϵ_{prim}, as shown in (18). If the NN prediction is the optimal value, then these error factors given in (15) - (18) will be zero.

By including the KKT conditions in the NN loss, we can have a set of collocation points in the training set. The collocation points are a set of random input values from the input domain. However, unlike the training data points, we have not pre-calculated and do not provide the optimal generation dispatch values $\mathbf{P_g}$ or dual variables $\mathbf{L_m}$ associated with them during training. Instead, the discrepancy in the KKT conditions caused by the predicted active power generation will be used to train the NN; i.e. based on (19) the NN during training seeks weights and biases to minimize this discrepancy, see MAE_ϵ. Furthermore, as we will see in Section III, the error factor corresponding to the primal conditions given in (18) can be used without the other conditions to penalize only the constraint violations; this will be similar to the method proposed in [13].

The shared parameters of the two neural networks are optimized by minimizing the loss function:

$$\begin{aligned}MAE = &\frac{1}{N_t}\sum_{i=1}^{N_t}\Lambda_P\underbrace{|\hat{P}_g-P_g|}_{MAE_p}+\Lambda_L\underbrace{|\hat{L}_m-L_m|}_{MAE_l}\\ &+\frac{\Lambda_\epsilon}{N_t+N_c}\sum_{i=1}^{N_t+N_c}\underbrace{\epsilon_{stat}+\epsilon_{comp}+\epsilon_{dual}+\epsilon_{prim}}_{MAE_\epsilon}\end{aligned} \tag{19}$$

where N_t is the number of training data points, and N_c is the number of collocation points. Furthermore, MAE_p, MAE_l, and MAE_ϵ are the mean absolute errors corresponding to active power dispatch, dual variables, and KKT condition violations and Λ_P, Λ_L, and Λ_ϵ are their corresponding weights. The PINN performance depends significantly on the weights Λ_P, Λ_L, and Λ_ϵ. So, they have to be selected appropriately to reduce either the average error or the maximum constraint violations.

For collocation points, since we do not provide the optimal generation dispatch values $\mathbf{P_g}$ or dual variables $\mathbf{L_m}$ associated with them, both MAE_p and MAE_l will be considered zero (in contrast with the points drawn from the training dataset), and MAE_ϵ will be used to calculate the training error.

C. Worst Case Guarantees

This section describes the worst-case guarantees used for evaluating the performance of the PINN. Worst-case guarantees provide an upper bound to constraint violations, sub-optimality, and distance between optimal generation value and the PINN predictions. To determine these worst-case guarantees, the trained NN will be reformulated into a MILP problem using the method proposed in [15]. After the training is finished, and the NN is ready to be deployed in practice, (15) -(18) are not used. So, we can ignore them during the MILP formulation.

The NN formulation given in (13) is linear, so we can directly use it in the MILP problem. However, the nonlinear ReLU activation (14) in the NN has to be reformulated into a mixed integer linear problem as follows:

$$Z_k^i \le Z_k^{\prime i} - Z_k^{min,i}(1-y_k^i) \qquad \forall k=1,...,K\ \forall i=1,...,N_k \tag{20}$$

$$Z_k^i \ge Z_k^{\prime i} \qquad \forall k=1,...,K\ \forall i=1,...,N_k \tag{21}$$

$$Z_k^i \leq Z_k^{max,i} y_k^i \qquad \forall k=1,...,K \; \forall i=1,...,N_k \tag{22}$$

$$Z_k^i \geq 0 \qquad \forall k=1,...,K \; \forall i=1,...,N_k \tag{23}$$

$$y_k \in \{0,1\}^{N_k} \qquad \forall k=1,...,K \tag{24}$$

where Z_k^i and $Z_k^{'i}$ are the outputs and inputs of the ReLU activation function, $Z^{min,i}$ and $Z^{max,i}$ are large value so that they won't be binding, and y_k^i is a binary variable. If $Z_k^{'i}$ is less than zero then y_k^i will be zero and (22) and (23) will be active and Z_k^i will be constrained to zero. Else, y_k^i will be equal to one and (20) and (21) will make sure Z_k^i is equal to $Z_k^{'i}$.

1) Worst-Case Guarantees for Constraint Violations: In this section, the MILP problem formulations used to determine the maximum constraint violations in generator active power outputs, denoted by v_g, and line flow violations, denoted by v_l, as a result of the physics-informed neural network predictions are discussed. The maximum constraint violations in generator active power outputs can be formulated as follows:

$$\max_{\hat{\mathbf{P}}_{\mathbf{g}},\mathbf{P_d},\mathbf{Z},\mathbf{Z'},\mathbf{y}} \quad v_g \tag{25}$$

$$v_g = \max(\hat{\mathbf{P}}_{\mathbf{g}} - \mathbf{P}_{\mathbf{g}}^{\mathbf{max}}, \mathbf{P}_{\mathbf{g}}^{\mathbf{min}} - \hat{\mathbf{P}}_{\mathbf{g}}, 0) \tag{26}$$

$$s.t. (13),(20)-(24) \tag{27}$$

Please note v_g is not the maximum constraint violation of a single generator, but rather it is the maximum constraint violation considering all the generators for the entire defined input domain. Similarly, v_l can be determined as follows:

$$\max_{\hat{\mathbf{P}}_{\mathbf{g}},\mathbf{P_d},\mathbf{Z},\mathbf{Z'},\mathbf{y}} \quad v_l \tag{28}$$

$$v_l = \max(|\mathbf{PTDF}(\hat{\mathbf{P}}_{\mathbf{g}} - \mathbf{P_d})| - P_l^{max}, 0) \tag{29}$$

$$s.t. (13),(20)-(24) \tag{30}$$

where v_l is the overall non-negative maximum line flow constraint violation in the entire input domain. When these MILP problems are solved to the zero MILP gap, we can ensure that the v_g and v_l values we obtain are the global optima. Thus, we can guarantee that there is no input $\mathbf{P_d}$ in the entire input domain, leading to constraint violations larger than the obtained values v_g and v_l.

2) Worst-Case Guarantees for Distance of Predicted to Optimal Decision Variables and for Sub-Optimality: This section establishes the MILP formulations used to determine the maximum distance between the physics-informed neural network prediction and the optimal value, denoted by v_{dist}, and the maximum sub-optimality, denoted by v_{opt}, in the entire input domain. The formulation used to determine v_{dist} is as follows:

$$v_{dist} = \max\left(\frac{|\hat{\mathbf{P}}_{\mathbf{g}} - \mathbf{P_g}|}{\mathbf{P}_{\mathbf{g}}^{\mathbf{max}} - \mathbf{P}_{\mathbf{g}}^{\mathbf{min}}}\right) \tag{31}$$

$$\max_{\hat{\mathbf{P}}_{\mathbf{g}},\mathbf{P_g},\mathbf{P_d},\mathbf{Z},\mathbf{Z'},\mathbf{y}} \quad v_{dist} \tag{32}$$

$$s.t. (13),(5)-(11),(20)-(24) \tag{33}$$

where $\mathbf{P_g}$ is the optimal generation active power output for a given $\mathbf{P_d}$ calculated in the lower-level optimization problem utilizing the KKT formulation given in (5) - (11), and $\hat{\mathbf{P}}_{\mathbf{g}}$ is the NN prediction. Similarly, the maximum sub-optimality of the PINN prediction can be formulated as follows:

Table I: TEST CASE CHARACTERISTICS

Test cases	N_{bus}	N_d	N_g	N_{line}	Max. loading MW
case39	39	21	10	46	6254
case118	118	99	19	186	4242
case162	162	113	12	284	7239

$$v_{opt} = \mathbf{c^T}(\hat{\mathbf{P}}_{\mathbf{g}} - \mathbf{P_g}) \tag{34}$$

$$\max_{\hat{\mathbf{P}}_{\mathbf{g}},\mathbf{P_g},\mathbf{P_d},\mathbf{Z},\mathbf{Z'},\mathbf{y}} \quad v_{opt} \tag{35}$$

$$s.t. (13),(5)-(11),(20)-(24) \tag{36}$$

By maximizing v_{opt} in the objective function, we can compute worst-case guarantees for the sub-optimality of the predicted solution. The complementary slackness conditions, given in (6) - (9), are non-linear so they have to be reformulated into linear equations using the Fortuny-Amat McCarl linearization [20] as follows:

$$\mathbf{P}_g^{min} - \mathbf{P}_g \geq -\mathbf{r}_g^{min}\mathbf{M}_g^{min} \tag{37}$$

$$\overline{\mu}_g \leq (1-\mathbf{r}_g^{min})\mathbf{M}_g^{min} \tag{38}$$

$$\mathbf{P}_g - \mathbf{P}_g^{max} \geq -\mathbf{r}_g^{max}\mathbf{M}_g^{max} \tag{39}$$

$$\underline{\mu}_g \leq (1-\mathbf{r}_g^{max})\mathbf{M}_g^{max} \tag{40}$$

$$\mathbf{P}_{line}^{min} - \mathbf{PTDF}(\mathbf{P}_g - \mathbf{P}_d) \geq -\mathbf{r}_{line}^{min}\mathbf{M}_{line}^{min} \tag{41}$$

$$\overline{\mu}_l \leq (1-\mathbf{r}_{line}^{min})\mathbf{M}_{line}^{min} \tag{42}$$

$$\mathbf{PTDF}(\mathbf{P}_g - \mathbf{P}_d) - \mathbf{P}_{line}^{max} \geq -\mathbf{r}_{line}^{max}\mathbf{M}_{line}^{max} \tag{43}$$

$$\underline{\mu}_l \leq (1-\mathbf{r}_{line}^{max})\mathbf{M}_{line}^{max} \tag{44}$$

where $\mathbf{r}$ is a binary variable and $\mathbf{M}$ is a sizeable non-binding constant for each condition. The constant M has to be chosen sufficiently large for it to be nonbinding. When the resulting MILP optimization problem is solved to zero MILP gap, we obtain the provable guarantee that there is no input $\mathbf{P_d}$ in the entire input domain that will result in a PINN output with distance or sub-optimality larger than the obtained values of v_{dist} and v_{opt}.

III. RESULTS & DISCUSSION

A. Simulation Setup

We evaluated the effectiveness of PINNs on three PGLib-OPF networks v19.05 [21]. The test case specifications are given in Table I. In each of these test cases, the input domain for each active power demand is assumed to be between 60% to 100% of its maximum loading. The maximum loading was defined according to [21], and the sum of maximum loading is given in Table I. We used Latin hypercube sampling [22] to randomly generate 100,000 samples from the input domain. Of the 100,000 samples generated, 20% was used as training and test dataset points (i.e we calculated and associated with them the DC-OPF results), 50% was used as collocation points, and the rest was used to determine average errors in an unseen test set. For the data points in training and test sets, the MATPOWER DC-OPF solver [23] was used to determine the optimal active power generation.

Then the KKT Conditions, given in (15) -(17), were utilized to determine the values of Lagrange multipliers.

The NN architecture consists of two sets of hidden layers, as shown in Fig 2. The NN used to predict the optimal active power generations has three hidden layers with 20 neurons each; the NN used to obtain the Lagrange multipliers has three hidden layers with 30 neurons each. We used TensorFlow [24] for NN training, we fixed the maximum number of training epochs to 5'000, and split the data set into two batches. The mean absolute error was used to determine the loss between NN predictions and the actual optimal solution during training.

The MILP problem used for worst-case guarantees was formulated in YALMIP [25] and solved using Gurobi. After solving the MILPs, we verified that the complementary slackness conditions are satisfied, and the constants are non-binding. A laptop with AMD Ryzen 7 pro CPU, 16 GB RAM, and Radeon GPU was used to carry out the computational experiments. The code to reproduce all simulation results is available online [26].

B. Physics-Informed Neural Network Average Performance over Test Set Samples

In the following, we evaluate the average performance of four different configurations of the PINN and compare them to a standard NN without the PINN layer, specified as NN in Table II. The first PINN configuration, represented by Pg Abs, includes the absolute value of the generation limit violation term given in (18), while all other KKT condition violations, given in (15) - (18), are ignored. We do this to understand how penalizing the generation constraint violation impacts the overall system performance. Similarly, the Pg Square and Pg Exp have the generation limit violation term given in (18) in square and exponential terms. Finally, in KKT, all the KKT constraint violations given in (15) - (18) are included in the PINN loss term.

The metrics used for comparing the average performance in Table II are:

1) Mean absolute error (MAE) in percentage.
2) Average generation active power constraint violation v_g in MW.
3) Average line flow limit violations v_l in MW.
4) Average distance of predicted value to optimal decision variables v_{dist} in percentage.
5) Average sub-optimality v_{opt} in percentage.

During the analysis, we observed that both the average and the worst-case performance of the PINN depends a lot on the hyper-parameter values, i.e., Λ_P, Λ_L, and Λ_ϵ weights. We experimented with different hyper-parameter values, and the ones which offered the lowest worst-case generation constraint violation are used to produce the results given in Table II. Because of this, the average performance compared to the standard NN has worsened in some cases, especially in the case of the 162 bus system compared to other sets of results we obtained. This could be because of the limited number of hyper-parameter values we explored. Even then, in most cases, the mean absolute error and the average constraint violations and sub-optimality have either improved or remain comparable. This indicates the satisfactory generalization capability of the PINN. It shall also be noted that due to additional number of equations and collocation points, the PINN was observed to take almost thrice as much time to train as opposed to the standard NN.

Table II: Performance Averaged Over Test Set Samples

TestCase			MAE (%)	v_g (MW)	v_{line} (MW)	v_{dist} (%)	v_{opt} (%)
Case 39	NN		0.23	2.58	0.00	0.35	0.15
	PINN	Pg Abs	0.05	0.44	0.01	0.13	0.02
		Pg Sqr	0.24	2.13	0.00	0.59	0.12
		Pg Exp	0.10	0.90	0.02	0.15	0.07
		KKT	0.12	0.92	0.04	0.39	0.01
Case 118	NN		0.68	0.66	2.75	8.42	0.65
	PINN	Pg Abs	0.89	1.57	5.87	9.49	0.74
		Pg Sqr	0.69	0.58	2.54	9.20	0.53
		Pg Exp	1.01	3.23	5.92	8.70	0.61
		KKT	1.24	3.01	4.60	4.60	8.28
Case 162	NN		3.48	8.65	11.54	23.70	0.45
	PINN	Pg Abs	3.35	10.31	11.03	23.54	0.62
		Pg Sqr	3.43	9.05	11.25	23.77	0.53
		Pg Exp	3.30	1.12	7.09	24.04	0.24
		KKT	3.11	5.34	7.54	22.31	0.49

Table III: Worst-Case Guarantees for Constraint Violations

Test Cases			v_g		v_{line}	
			MW	% wrt max load	MW	% wrt max load
Case 39	NN		365	6	146	2
	PINN	Pg Abs	265	4	63	1
		Pg Sqr	195	3	68	1
		Pg Exp	144	2	87	1
		KKT	133	2	54	1
Case 118	NN		572	13	252	6
	PINN	Pg Abs	246	6	131	3
		Pg Sqr	209	5	94	2
		Pg Exp	177	4	114	3
		KKT	209	5	95	2
Case 162	NN		1725	24	1224	17
	PINN	Pg Abs	861	12	1193	16
		Pg Sqr	792	11	1209	17
		Pg Exp	672	9	657	9
		KKT	696	10	475	7

C. Worst-Case Guarantees for Constraint Violations

Using the mixed-integer linear reformulation given in (25)-(30), we solve the MILPs to compute the corresponding worst-case guarantees. The results are shown in Table III. As hypothesized, when the absolute values of the generation limit violation were added to the NN loss, the worst-case generation constraint violation was reduced by at least 25%. Moreover, when we used higher-order terms to estimate the loss, the worst-case generation constraint violation values were reduced even further, and by at least 20% in all cases. This indicates that we can achieve a better worst-case guarantee by using higher-order terms to penalize constraint violations. When we added all the KKT conditions violations to the loss function (all in absolute terms), the worst-case generation and line flow constraint violation were reduced further in the 39-bus and 118-bus systems. In the 162-bus system, the results are comparable to that of the Pg Exp. This validates the hypothesis that we can achieve a better worst-case guarantee by incorporating KKT condition violations into the NN training.

Table IV: Worst-Case Guarantees For (i) Distance Of Predicted to Optimal Decision Variables and (ii) Suboptimality

Test Cases			v_{dist} %	v_{opt} %
Case 39	NN		78	10
	PINN	Pg Abs	71	10
		Pg Sqr	72	6
		Pg Exp	86	11
		KKT	69	9
Case 118	NN		542	13
	PINN	Pg Abs	465	9
		Pg Sqr	382	9
		Pg Exp	658	17
		KKT	306	7
Case 162	NN		788	58
	PINN	Pg Abs	488	65
		Pg Sqr	838	45
		Pg Exp	469	65
		KKT	482	69

D. Worst-Case Guarantees for (i) Distance of Predicted to Optimal Decision Variables and (ii) for Sub-Optimality

The MILP formulation given in (31) - (44) was used to find the worst-case guarantees for the distance of predicted to optimal decision variables and sub-optimality. The results given in Table IV are calculated for the same set of hyperparameter values used to obtain the results presented in Table III. In some cases, adding only the constraint violation terms to the loss function has a negative effect on the solution's optimality. Results still show, however, that adding all the KKT condition violations in the loss term obtains the best performance, as the maximum distance between the PINN prediction and the optimal value, as well as the worst-case suboptimality have improved in all three cases.

IV. Conclusion and Future Work

This paper presents two key contributions. First, to the best of our knowledge, this is the first paper to propose physics-informed neural networks for optimal power flow applications. We show that by combining the KKT conditions with the neural network, the physics-informed neural network achieves higher accuracy while utilizing substantially fewer data points. Second, we extend our previous work on worst-case guarantees to cover the physics-informed neural networks (PINNs), and we show that PINNs result in lower worst-case violations than conventional neural networks. Future work includes the extension of the proposed approaches to AC-OPF problems and a multilevel optimization algorithm to determine the key PINN hyper-parameters that improve average and worst-case performance.

References

[1] D. K. Molzahn, I. A. Hiskens *et al.*, "A survey of relaxations and approximations of the power flow equations," 2019.

[2] B. Stott, J. Jardim, and O. Alsac, "Dc power flow revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[3] D. Deka and S. Misra, "Learning for dc-opf: Classifying active sets using neural nets," in *2019 IEEE Milan PowerTech*, 2019, pp. 1–6.

[4] L. Duchesne, E. Karangelos, and L. Wehenkel, "Recent developments in machine learning for energy systems reliability management," *Proceedings of the IEEE*, vol. 108, no. 9, pp. 1656–1676, 2020.

[5] F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient database generation for data-driven security assessment of power systems," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 30–41, 2020.

[6] A. Venzke and S. Chatzivasileiadis, "Verification of neural network behaviour: Formal guarantees for power system applications," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 383–397, 2021.

[7] L. Zhang, Y. Chen, and B. Zhang, "A convex neural network solver for dcopf with generalization guarantees," *arXiv preprint arXiv:2009.09109*, 2020.

[8] M. K. Singh, V. Kekatos, and G. B. Giannakis, "Learning to solve the ac-opf using sensitivity-informed deep neural networks," *arXiv preprint arXiv:2103.14779*, 2021.

[9] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast ac optimal power flow," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–6.

[10] P. L. Donti, D. Rolnick, and J. Z. Kolter, "Dc3: A learning method for optimization with hard constraints," *arXiv preprint arXiv:2104.12225*, 2021.

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

[12] Y. Nandwani, A. Pathak, Mausam, and P. Singla, "A primal dual formulation for deep learning with constraints," in *NeurIPS*, 2019.

[13] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 630–637.

[14] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2020, pp. 1–5.

[15] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–7.

[16] S. Chatzivasileiadis, "Lecture notes on optimal power flow (opf)," *arXiv preprint arXiv:1811.00943*, 2018.

[17] S. S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[18] A. B. X. Glorot and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[19] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118307125

[20] J. Fortuny-Amat and B. McCarl, "A representation and economic interpretation of a two-level programming problem," *Journal of the operational Research Society*, vol. 32, no. 9, pp. 783–792, 1981.

[21] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang *et al.*, "The power grid library for benchmarking ac optimal power flow algorithms," *arXiv preprint arXiv:1908.02788*, 2019.

[22] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[23] R. D. Zimmerman, C. E. Murillo-Sánchez, 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, 2010.

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[25] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE, 2004, pp. 284–289.

[26] R. Nellikkath and S. Chatzivasileiadis, "Supplementary data and code: Physics-informed neural networks for minimising worst-case violations in dc optimal power flow," 2021. [Online]. Available: https://github.com/RahulNellikkath/Physics-Informed-Neural-Network-for-DC-OPF