Electric Power Systems Research 212 (2022) 108412

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Physics-Informed Neural Networks for AC Optimal Power Flow☆

Rahul Nellikkath *, Spyros Chatzivasileiadis

Center for Electric Power and Energy, Technical University of Denmark (DTU), Kgs. Lyngby, Denmark

ARTICLE INFO

Keywords:
AC-OPF
Physics-informed neural network
Worst-case guarantees

ABSTRACT

This paper introduces, for the first time to our knowledge, physics-informed neural networks to accurately estimate the AC-Optimal Power Flow (AC-OPF) result and delivers rigorous guarantees about their performance. Power system operators, along with several other actors, are increasingly using Optimal Power Flow (OPF) algorithms for a wide number of applications, including planning and real-time operations. However, in its original form, the AC OPF problem is often challenging to solve as it is non-linear and non-convex. Besides the large number of approximations and relaxations, recent efforts have also been focusing on Machine Learning approaches, especially neural networks. So far, however, these approaches have only partially considered the wide number of physical models available during training. And, more importantly, they have offered no guarantees about potential constraint violations of their output. Our approach (i) introduces a framework to incorporate AC power flow equations inside neural network training and (ii) integrates methods that rigorously determine and reduce the worst-case constraint violations across the entire input domain, while maintaining the optimality of the prediction. We demonstrate how physics-informed neural networks achieve higher accuracy and lower constraint violations than standard neural networks, and show how we can further reduce the worst-case violations for all neural networks.

1. Introduction

Power system operators, electricity market operators, and several other actors increasingly use Optimal Power Flow (OPF) for the planning and real-time operation of power systems. Countless instances of OPF need to be solved when it comes to evaluating uncertain scenarios, finding optimal control setpoints, bidding strategies, or determining the electricity market clearing. However, the exact formulation of the AC-Power Flow equations in the OPF problem is non-linear and non-convex [1], which could result in significant difficulties for convergence and long computational times. A large body of literature exists on deriving approximations of the AC-OPF problem, with the most popular ones being the linearized DC-OPF, other linear or convex approximations, and convex relaxations, that will ensure fast computation speed and convergence guarantees [2].

Most recently, there is a revived interest in machine learning methods to accurately estimate the AC-OPF solution [3,4], which have demonstrated a computation speedup of 100–1000 times compared with the conventional methods. This means that in the time it would take to assess one scenario by solving one AC-OPF instance, we can now assess up to 1'000 scenarios simultaneously. However, these machine learning algorithms experience two significant challenges. First, the availability and quality of training datasets: to train a neural network with considerable accuracy, we need OPF results for a huge set of operating points that will cover both normal and abnormal situations. Such datasets often do not exist, or it is often challenging to generate. Convex relaxation techniques were proposed in [5,6] to efficiently generating such large datasets, concentrating closer to the security boundary. Along the lines of improving the performance of such Machine Learning algorithms, a method was proposed in [7] to identify and include adversarial examples in the training data set during training.

The second major issue limiting the Neural Network (NN) widespread adaptation is that, so far, none of the proposed machine learning algorithms have supplied any worst-case performance guarantee. With OPF often used for safety–critical applications, the neural network estimates must not violate any OPF constraints, e.g., line, voltage, or generator limits. To mitigate these limitations, the NN predictions can be post-processed to satisfy generation constraints as proposed in [8,9] for AC-OPF. However, this could negatively impact the optimality of the solution. A few methods have also offered to include the constraint violations in the NN loss function [10,8]. However, none of these proposed algorithms provide any worst-case performance guarantees for the AC-OPF problem. The only works that have derived

☆ This work is supported by the FLEXGRID project, funded by the European Commission Horizon 2020 program, Grant Agreement No. 863876, and by the ERC Starting Grant VeriPhIED, Grant Agreement No. 949899.

* Corresponding author.
E-mail addresses: rnelli@elektro.dtu.dk (R. Nellikkath), spchatz@elektro.dtu.dk (S. Chatzivasileiadis).

https://doi.org/10.1016/j.epsr.2022.108412
Received 3 October 2021; Received in revised form 16 April 2022; Accepted 2 July 2022
Available online 28 July 2022

worst-case performance guarantees have so far focused only on the DC-OPF problem [11,12].

This paper attempts to address both of these challenges by using a physics-informed neural network for AC-OPF applications. First, we introduce the physical equations in the form of the AC-OPF Karush–Kuhn–Tucker (KKT) conditions inside the neural network training. By doing that, the neural network can reduce its dependency on the size and quality of the training dataset, and instead, it can determine its optimal parameters based on the actual equations that it aims to emulate [13].

Second, we introduce methods that extract worst-case guarantees for generation and line flow constraint violations of the neural network AC-OPF predictions, extending our previous work presented in [11,12]. Through that, we (i) determine the worst violations that can result from any neural network output across the whole input domain, and (ii) propose methods to reduce them.

This paper is structured as follows: Section 2 describes the AC-OPF problem and its KKT formulation, introduces the proposed physics-informed neural network training architecture, and the optimization algorithm used to determine the worst-case guarantees. Section 3 provides the simulation setup used and delivers the results demonstrating the performance of physics-informed neural networks. Section 4 discusses the possible opportunities to improve the system performance and concludes.

2. Methodology

2.1. AC - optimal power flow

The AC Optimal Power Flow (AC-OPF) problem for generation cost minimization in a system with N_b number of buses, N_g number of generators and N_d number of loads is a Quadratically Constrained Quadratic Programing (QCQP) problem. The objective function for the cost minimization can be written as follows:

$$\min_{\mathbf{P}_g,\mathbf{Q}_g} \mathbf{c}_p^T\mathbf{P}_g + \mathbf{c}_q^T\mathbf{Q}_g \tag{1}$$

where vector $\mathbf{c}_p^T$ and $\mathbf{c}_q^T$ refers to the linear cost terms for the active and reactive power generations $\mathbf{P}_g$ and $\mathbf{Q}_g$, respectively. The optimal generation values depend on the active and reactive power demand, denoted by $\mathbf{P}_d$ and $\mathbf{Q}_d$, and the network constraints. For a given demand, the active and reactive power injection at each node $n \in N_b$ can be represented as follows:

$$p_n = p_n^g - p_n^d \qquad \forall n \in N_b \tag{2}$$

$$q_n = q_n^g - q_n^d \qquad \forall n \in N_b \tag{3}$$

where p_n and q_n denote the active and reactive power injection at bus n and p_n^g, q_n^g, p_n^d and q_n^d specifies the active and reactive power generation and demand, respectively at bus n. The power flow equations in the network can be expressed as follows:

$$p_n = \sum_{k=1}^{N_b} v_n^r(v_k^r G_{nk} - v_k^i B_{nk}) + v_n^i(v_k^i G_{nk} + v_k^r B_{nk}) \tag{4}$$

$$q_n = \sum_{k=1}^{N_b} v_n^i(v_k^r G_{nk} - v_k^i B_{nk}) - v_n^r(v_k^i G_{nk} + v_k^r B_{nk}) \tag{5}$$

where real and imaginary part of the voltage at bus n is given by v_n^r and v_n^i and, the conductance and susceptance of the line nk is denoted by G_{nk} and B_{nk}. If the real and reactive parts of voltage are combined into a vector of size $2N_b \times 1$ as $\mathbf{v} = [(\mathbf{v}^r)^T, (\mathbf{v}^i)^T]^T$. Then, the power flow equation in (4) and (5) can be simplified as follows [14]:

$$\mathbf{v}^T\mathbf{M}_p^n\mathbf{v} = p_n \qquad \forall n \in N_b \tag{6}$$

$$\mathbf{v}^T\mathbf{M}_q^n\mathbf{v} = q_n \qquad \forall n \in N_b \tag{7}$$

The optimal generation set point should also satisfy the active and reactive power generation limits.

$$\underline{p}_n^g \le \mathbf{v}^T\mathbf{M}_p^n\mathbf{v} + p_n^d \le \overline{p}_n^g \qquad \forall n \in N_g \tag{8}$$

$$\underline{q}_n^g \le \mathbf{v}^T\mathbf{M}_q^n\mathbf{v} + q_n^d \le \overline{q}_n^g \qquad \forall n \in N_g \tag{9}$$

Similarly, the voltage and line current flow constraints for the power system can be represented in a matrix form as follows:

$$\underline{\mathbf{v}}^n \le \mathbf{v}^T\mathbf{M}_v^n\mathbf{v} \le \overline{\mathbf{v}}^n \qquad \forall n \in N_b \tag{10}$$

$$\ell_{mn} = \mathbf{v}^T\mathbf{M}_{i_{mn}}\mathbf{v} \le \overline{\ell}_{mn} \qquad \forall mn \in N_l \tag{11}$$

where $\mathbf{M}_v^n := e_n e_n^T + e_{N_b+n}e_{N_b+n}^T$ and e_n is a $2N_b \times 1$ unit vector with zeros at all the locations except n. The square of magnitude of upper and lower voltage limit is denoted by $\overline{\mathbf{v}}^n$ and $\underline{\mathbf{v}}^n$ respectively, and the square of magnitude of line current flow in line mn is represented by ℓ_{mn} and matrix $\mathbf{M}_i^{mn} = |y_{mn}|^2(e_m - e_n)(e_m - e_n)^T + |y_{mn}|^2(e_{N_b+m} - e_{N_b+n})(e_{N_b+m} - e_{N_b+n})^T$ where y_{mn} is the admittance of branch nm. Assuming the slack bus N_{sb} acts as an angle reference for the voltage, we will have:

$$v_{N_{sb}}^i = \mathbf{v}^T\mathbf{e}_{N_b+N_{sb}}\mathbf{e}_{N_b+N_{sb}}^T\mathbf{v} = 0 \tag{12}$$

The constraints (2)–(3), (6)–(12) and the objective function for the AC-OPF problem (1) can be simplified as follows:

$$\min_{\mathbf{v},\mathbf{G}} \mathbf{c}^T\mathbf{G} \tag{13a}$$

$$s.t\ \mathbf{v}^T\mathbf{F}_l\mathbf{v} = a_l^T\mathbf{G} + b_l^T\mathbf{D}, \qquad l = 1:F\ ;\lambda_l \tag{13b}$$

$$\mathbf{v}^T\mathbf{M}_m\mathbf{v} \le d_m^T\mathbf{D} + f_m, \qquad m = 1:M\ ;\mu_m \tag{13c}$$

where $\mathbf{G} = [\mathbf{P}_g^T, \mathbf{Q}_g^T]^T$, $\mathbf{D} = [\mathbf{P}_d^T, \mathbf{Q}_d^T]^T$, and $\mathbf{c}^T$ is the combined linear cost terms for the active and reactive power generations. Then the equality constraints (6)–(7) and (12) can be represented by the $\mathbf{F} = 2N_b + 1$ constraints in (13b). Similarly, the inequality constraints (8)–(11) can be represented by the $\mathbf{M} = 4N_g + 2N_b + N_l$ constraints in (13c). The corresponding Lagrange multipliers are denoted by λ_l and μ_m.

The Lagrangian function $\mathcal{L}$ for the AC-OPF can be formulated as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{x},\lambda,\mu,D) = c^T\mathbf{G} &+ \sum_{l=1}^{L}\lambda_l(\mathbf{v}^T\mathbf{F}_l\mathbf{v} - a_l^T\mathbf{G} - b_l^T\mathbf{D}) \\ &+ \sum_{m=1}^{M}\mu_m(\mathbf{v}^T\mathbf{M}_m\mathbf{v} - d_m^T\mathbf{D} - f_m)\end{aligned} \tag{14}$$

where $\mathbf{x} = \{\mathbf{G}, \mathbf{v}\}$. So, the KKT conditions can be formulated as follows:

$$c = \sum_{l=1}^{L}\lambda_l a_l \tag{15}$$

$$\left(\sum_{l=1}^{L}\lambda_l\mathbf{F}_l + \sum_{m=1}^{M}\mu_m\mathbf{M}_m\right) = 0 \tag{16}$$

$$\mu_m(\mathbf{v}^T\mathbf{M}_m\mathbf{v} - d_m^T\mathbf{D} - f_m) = 0 \qquad m = 1:M \tag{17}$$

$$\mu_m \ge 0 \qquad m = 1:M \tag{18}$$

(13b)–(13c) (19)

where the stationarity condition is given in (15) and (16), the complementary slackness condition is given by (17) and dual feasibility is given by (18). For an AC-OPF problem these KKT conditions act as a necessary condition for optimality.

2.2. Physics informed neural network

This section introduces the physics-informed neural network architecture used for predicting the AC-OPF active and reactive power generation setpoints $\mathbf{G}$, given active and reactive power demand $\mathbf{D}$ as the input. A conventional neural network (NN) is a group of interconnected nodes that are trained to correlate the input and the output layers, as shown in Fig. 1. Nodes connecting the input and output layers

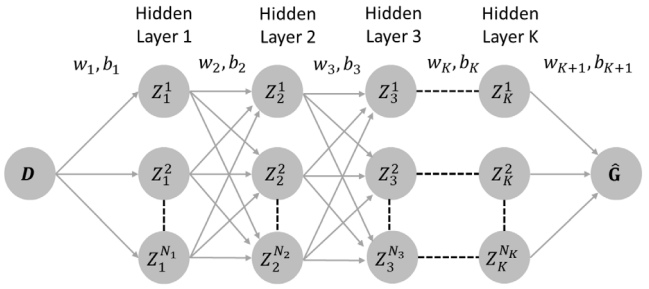

Fig. 1. Illustration of the neural network architecture to predict the optimal generation active and reactive power outputs $\hat{\mathbf{G}}$ using the active and reactive power demand $\mathbf{D}$ as input: There are K hidden layers in the neural network with N_k neurons each. Where k = 1, …, K. During training, the weights $\mathbf{w}$ and biases $\mathbf{b}$ are modified to minimize the mean absolute errors, as formulated in (23), in the training dataset.

will be divided into K number of hidden layers with N_k number of neurons in hidden layer k and the edges connecting the neurons will have a weight $\mathbf{w}$ and a bias $\mathbf{b}$ associated with them. Also, each neuron in the neural network will have a nonlinear activation function linked with them.

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

$$\mathbf{Z}_k = \sigma(\mathbf{w}_k \mathbf{Z}_{k-1} + \mathbf{b}_k) \tag{20}$$

where $\mathbf{Z}_k$ is the output of layer k, $\mathbf{w}_k$ and $\mathbf{b}_k$ are the weights and biases connecting layer $k-1$ and k. σ is the nonlinear activation function. We chose the ReLU as the nonlinear activation function in this work, as it is observed to accelerate the neural network training [14]. The ReLU activation function will return the input if the input is positive and return zero if the input is negative or zero. The output of each layer after the ReLU activation function can be formulated as follows:

$$\hat{\mathbf{Z}}_k = \mathbf{w}_k \mathbf{Z}_{k-1} + \mathbf{b}_k \tag{21}$$

$$\mathbf{Z}_k = \max(\hat{\mathbf{Z}}_k, 0) \tag{22}$$

During the NN training, the backpropagation algorithm will modify the weights and biases to minimize the average prediction error, as formulated in (23), in the training dataset.

$$\mathcal{L}_0 = \frac{1}{N_t} \sum_{i=1}^{N_t} |\mathbf{G}_i - \hat{\mathbf{G}}_i| \tag{23}$$

where N_t is the number of training data points.

In case of a physics-informed neural network, on top of comparing the NN predictions to the AC-OPF setpoints of the training database, the validity of the physical equations governing the problem will also be accessed during NN training (see [15,13], and our previous work [16] for DC-OPF applications). Since the optimal value should satisfy the KKT conditions given in (15)–(19), the disparities in the KKT condition, denoted by ϵ, as shown in (24a)–(24d) are added to the NN training loss function (25) and minimized during training. The proposed physics-informed neural network structure is given in Fig. 2.

The voltage values and the dual variables required for calculating the discrepancy in the KKT conditions are predicted using a separate set of hidden layers since we observed an increase in accuracy while maintaining a smaller neural network size.

The discrepancy in KKT conditions are calculated as follows:

$$\epsilon_{stat,i} = |c - \sum \hat{\lambda}_{l,i} a_l| + |\left(\sum \hat{\lambda}_{l,i} \mathbf{F}_l + \sum \hat{\mu}_{m,i} \mathbf{M}_m \right)| \tag{24a}$$

$$\epsilon_{comp,i} = \sum |\hat{\mu}_{m,i} (\hat{\mathbf{v}}_i^T \mathbf{M}_m \hat{\mathbf{v}}_i - d_m^T \mathbf{D} - f_m)| \tag{24b}$$

$$\epsilon_{dual,i} = \sigma(\hat{\mu}_{m,i}) \tag{24c}$$

$$\epsilon_{prim,i} = \sum |\hat{\mathbf{v}}_i^T \mathbf{F}_l \hat{\mathbf{v}}_i - a_l^T \mathbf{G} - b_l^T \mathbf{D}| \tag{24d}$$

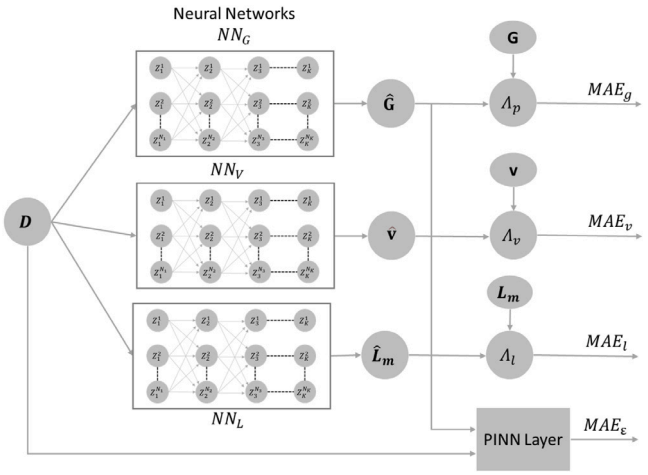

Fig. 2. Illustration of the physics-informed neural network architecture to predict the optimal active and reactive power generation outputs $\hat{\mathbf{G}}$, voltage setpoints $\hat{\mathbf{v}}$ and dual variables $\hat{\mathbf{L}}_\mathbf{m}$ utilizing the active and reactive power demand $\mathbf{D}$ as input. The neural networks that predict $\hat{\mathbf{G}}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{L}}_\mathbf{m}$ are independent of each other. During training, the weights $\mathbf{w}$ and biases $\mathbf{b}$ are modified to minimize the mean absolute errors MAE_g, MAE_v, MAE_l and MAE_ϵ as formulated in (25).

$$+ \sum \sigma(\mathbf{v}^T \mathbf{M}_m \mathbf{v} - d_m^T \mathbf{D} - f_m)$$

where $\hat{\mathbf{v}}$ is the voltage values predicted by the hidden layers NN_V and $\hat{\lambda}_l$ and $\hat{\mu}_m$ are the dual variables predicted by hidden layers NN_L in Fig. 2. The absolute value of the stationarity condition is measured in ϵ_{stat}, and the error in complementary slackness conditions (17) is given by ϵ_{com}. The ReLU activation function, represented by σ in (24c), is used to measure the dual feasibility violation. If the neural network prediction is the optimal value, then the errors calculated in (24) will be zero. Considering we also have the KKT conditions to measure the accuracy of the NN prediction, we can have collocation points in the training set. Like the NN training data points, the collocation points are a set of random input values from the input domain. However, unlike the training data points, we do not spend resources to determine the optimal generation dispatch values, the voltage setpoints, or dual variables before training. Instead, the error factors given in (24a)–(24d) will be used to measure the prediction accuracy and train the NN.

The following loss function is used to modify the shared parameters of the three neural networks:

$$\begin{aligned} MAE = \frac{1}{N_t} \sum_{i=1}^{N_t} & \Lambda_P \underbrace{|\hat{\mathbf{G}} - \mathbf{G}|}_{MAE_g} + \Lambda_V \underbrace{|\hat{\mathbf{v}} - \mathbf{v}|}_{MAE_v} + \Lambda_L \underbrace{|\hat{\mathbf{L}}_m - \mathbf{L}_m|}_{MAE_l} \\ & + \frac{\Lambda_\epsilon}{N_t + N_c} \sum_{i=1}^{N_t + N_c} \underbrace{\epsilon_{stat,i} + \epsilon_{comp,i} + \epsilon_{dual,i} + \epsilon_{prim,i}}_{MAE_\epsilon} \end{aligned} \tag{25}$$

where N_t is the number of training data points, and N_c is the number of collocation points. The mean absolute error of active and reactive power generation prediction to the actual value is denoted by MAE_g. MAE_v and MAE_l indicates the mean absolute error of voltage and dual value prediction, and MAE_ϵ is the mean absolute value of KKT condition violations given in (24a)–(24d). Λ_P, Λ_V, Λ_L, and Λ_ϵ are the corresponding weights of each loss functions. The physics-informed neural network performance depends significantly on these weights. So, they have to be selected appropriately to reduce either the average error or the maximum constraint violations.

Since the three NNs are independent, the NN will be trained to minimize the corresponding MAE along with MAE_ϵ. For the collocation points, as discussed earlier, we do not compute the OPF output a priori. As we have not computed the optimal generation dispatch values $\mathbf{G}$, voltage $\mathbf{v}$, and dual variables $\mathbf{L}_m$ for the collocation points, MAE_g,

MAE_v, and MAE_l will be considered zero for these points, and only MAE_ϵ will be considered in the NN training.

2.3. Evaluating the worst case performance of the physics-informed neural network

The average performance of NN prediction on an unseen test data set is typically used to evaluate and compare different NN architectures. However, in our case, other than the average performance on the test dataset, we will also be using the worst-case generation and line flow constraint violations in the entire input domain to evaluate and improve the performance of the proposed NN training architecture. This section describes the optimization problem used for extracting the worst-case guarantees. The NNs used for predicting the voltages and dual variables are independent of the NN used for predicting the optimal generation set-points. Since they are not required after the training and when the system is ready to be deployed, we can ignore them and instead focus only on the hidden layers used for predicting the generation values.

2.3.1. Worst-case guarantees for generation constraint violations

This section discusses the Mixed Integer Linear Programming (MILP) problem formulation used to determine the maximum active and reactive power generation constraint violations, denoted by v_g. The maximum constraint violations in the input domain can be formulated as follows:

$$\max_{\hat{\mathbf{G}},\mathbf{D},\mathbf{Z},\mathbf{Z}',\mathbf{y}} \; v_g \tag{26a}$$

$$v_g = \max(\hat{\mathbf{G}} - \overline{G}, \underline{\mathbf{G}} - \hat{G}, 0) \tag{26b}$$

$$s.t.\ \ (21),\ (22) \tag{26c}$$

where $\overline{\mathbf{G}}$ and $\underline{\mathbf{G}}$ are the maximum and minimum active and reactive power generation capacity. Since the ReLU activation function, used in (22), is nonlinear, the mixed-integer reformulation of ReLU activation function proposed in [11] is used as follows:

$$\mathbf{Z}_k = \max(\hat{\mathbf{Z}}_k, 0) \Rightarrow \begin{cases} \mathbf{Z}_k \le \hat{\mathbf{Z}}_k - \mathbf{Z}_k^{\min}(1-\mathbf{y}_k) & \text{(a)} \\ \mathbf{Z}_k \ge \hat{\mathbf{Z}}_k & \text{(b)} \\ \mathbf{Z}_k \le \mathbf{Z}_k^{\max}\mathbf{y}_k & \text{(c)} \\ \mathbf{Z}_k \ge \mathbf{0} & \text{(d)} \\ \mathbf{y}_k \in \{0,1\}^{N_k}. & \text{(e)} \end{cases} \tag{27}$$

where $\hat{\mathbf{Z}}_k$ and $\mathbf{Z}_k$ are the inputs and outputs of the ReLU activation function. $\mathbf{Z}_k^{max}$ and $\mathbf{Z}_k^{min}$ are the maximum and minimum values possible for the respective $\hat{\mathbf{Z}}_k$. They should be large enough so that the constraints (27)a and (27)c will not be binding and small enough so that the constraints are not unbounded. $\mathbf{y}_k$ denotes the binary variables. If $\hat{\mathbf{Z}}_k$ is less than zero, then (27)d will be active. Z_k^i and y_k^i will be zero to satisfy the other constraints. Similarly, if $\hat{\mathbf{Z}}_k$ is greater than zero, then (27)b and (27)a will be active, and y_k^i will be one to satisfy the other constraints.

2.3.2. Worst-case guarantees for line flow constraint violations

Similar to the generation constraints, the line flow limits play an equally significant role in ensuring the proper functioning of the network. However, unlike the generation set-points, we do not directly get the line flow values as an output from the NN prediction. In our effort to formulate the most computationally efficient approach to obtain the line current flow from the generation set points predicted by the NN, we observed that simplified convex relaxations such as semidefinite programming for AC Power Flow (AC-PF) were not binding for worst-case guarantees. Therefore, we had to formulate a non-convex power flow problem to obtain the line current flow from the generation set points as follows:

$$\hat{p}_n^g - p_n^d = \hat{\mathbf{v}}_{pf}^T \mathbf{M}_p^n \hat{\mathbf{v}}_{pf} \qquad \forall n \in N_b \tag{28}$$

Table 1
Test case characteristics.

Test case	N_b	N_d	N_g	N_l	Max loading	
					MW	MVA
case 14	14	11	5	20	259	276
case 39	39	21	10	46	6254	6626
case 118	118	99	19	186	4242	4537
case 162	162	113	12	284	7239	12005

$$\hat{q}_n^g - q_n^d = \hat{\mathbf{v}}_{pf}^T \mathbf{M}_q^n \hat{\mathbf{v}}_{pf} \qquad \forall n \in N_b \tag{29}$$

$$\hat{\ell}_{mn} = \hat{\mathbf{v}}_{pf}^T \mathbf{M}_{i_{mn}} \hat{\mathbf{v}}_{pf} \qquad \forall mn \in N_l \tag{30}$$

$$\hat{v}_{N_{sb}}^i = 0 \tag{31}$$

where $\hat{\mathbf{v}}_{pf}$ and $\hat{\ell}_{mn}$ are the voltage and square of line flow obtained from the power flow equations. The AC-PF equations in (28)–(31) are non-convex and quadratic. So, we have to use a Mixed Integer Quadratic Constrained Quadratic Programming (MIQCQP) solver to obtain the worst-case guarantees. The MIQCQP problem can be formulated as follows:

$$\max_{\hat{\mathbf{G}},\mathbf{D},\mathbf{Z},\mathbf{Z}',\mathbf{y}} \; v_l \tag{32a}$$

$$v_l = \max(\hat{\ell}_{mn} - \overline{\ell}_{mn}^2, 0) \tag{32b}$$

$$s.t.\ \ (28)\text{–}(31),\ (21),\ (27)\text{a}\text{–}(27)\text{e} \tag{32c}$$

When the MILP and MIQCQP problems are solved to zero MILP gap, we can ensure that the v_g and v_l values we obtained are the global optima. Thus, we can guarantee that there is no input $\{\mathbf{P}_d, \mathbf{Q}_d\} \in \mathbf{D}$ in the entire input domain, leading to constraint violations larger than the obtained values v_g and v_l.

3. Results & discussion

3.1. Simulation setup

The accuracy and scalability of the proposed physics-informed neural network training architecture are analyzed against a standard NN on four different test systems. The test system characteristics are given in Table 1. The network details for case 39, case 118, and case 162 are from the PGLib-OPF network library v19.05 [17], and the network details for case 14 is from [18]. In all of the test cases, the active and reactive power demand at each node is assumed to be independent of each other and they are specified to be between 60 to 100% of their respective maximum loading as follows:

$$0.6\,\underline{\mathbf{D}} \le \mathbf{D} \le \overline{\mathbf{D}} \tag{33}$$

where $\overline{\mathbf{D}}$ denotes the maximum active and reactive power demand. The sum of the maximum loading over all nodes for each system is given in Table 1.

Ten thousand sets of random active and reactive power input values were generated using latin hypercube sampling [19]. From these, 50% were allotted to the collocation data set, for which we do not have to calculate and provide the OPF setpoints. 20% of the rest was considered training data points and the remaining 30% as the unseen test data set. AC-OPF in MATPOWER was used to determine the optimal active and reactive power generation values and voltage setpoints for the input data points in the training and test data sets, and afterwards the KKT conditions given in (15)–(19) were used to obtain the dual variables.

The properties of both standard and physics-informed neural networks used for the analysis are given in the Table 2. We used TensorFlow [20] with Python for neural network training, and we fixed the number of training epochs to 1000 and split the data set into 200 batches while training.

A High-Performance Computing (HPC) server with an Intel Xeon E5-2650v4 processor and 256 GB RAM was used to train the neural

Table 2
Neural network properties.

Test case	N_k in hidden layer			Training time	
	G	V	L_m	NN (min)	PINN (xNN)
Case 14	5	10	20	3	2
Case 39	20	30	50	8	2.5
Case 118				88	3.2
Case 162				105	2.6

networks. The training time for the standard NN, denoted by NN, and physics-informed neural network, indicated by PINN, as a factor of time taken by the NN is given in Table 2. In our experiments the proposed physics-informed neural network took almost thrice as much time to train compared to the standard NN. On the other hand, physics-informed neural networks result in significant computational savings when we count in the generation of the training database, as PINNs require considerably less training samples for which both input and output needs to be computed (i.e. we do not need to run an OPF for any collocation point).

The MILP problem used for solving worst-case guarantees for generation violation and the MIQCQP problem used for solving the worst-case guarantees for line flow violations is formulated in YALMIP [21] and solved using Gurobi. The code to reproduce all the simulation results is available online [22].

3.2. Average performance over test data set

This section examines the performance of the physics-informed neural networks versus the standard NNs, considering the most common performance metric for regression NN: the mean absolute error over an unseen test data set. As shown in (34)–(37) and explained below, along with MAE we also define metrics that measure the average constraint violation, sub-optimality, and distance to optimal setpoint:

$$MAE_T = \frac{1}{N_T}\sum_{i=1}^{N_T} |\hat{\mathbf{G}} - \mathbf{G}| \tag{34}$$

$$v_g^{avg} = \frac{1}{N_T}\sum_{i=1}^{N_T} \max\left(\frac{\hat{\mathbf{G}} - \overline{\mathbf{G}}}{\overline{\mathbf{G}}}, \frac{\underline{\mathbf{G}} - \hat{\mathbf{G}}}{\overline{\mathbf{G}}}, 0\right) \tag{35}$$

$$v_{opt}^{avg} = \frac{1}{N_T}\sum_{i=1}^{N_T} \mathbf{c}^{\mathbf{T}}(\hat{\mathbf{G}} - \mathbf{G}) \tag{36}$$

$$v_{dist}^{avg} = \frac{1}{N_T}\sum_{i=1}^{N_T} \left(\frac{|\hat{\mathbf{G}} - \mathbf{G}|}{\overline{\mathbf{G}} - \underline{\mathbf{G}}}\right) \tag{37}$$

where MAE_T is the mean absolute error with respect to the total generation in test data set, N_T is the test data size, v_g^{avg} is the average active and reactive power generation constraint violation, v_{opt} is the average sub-optimality and v_{dist} is the average distance of predicted value to optimal decision variables v_{dist} in percentage.

During training, both the standard and the physics-informed neural network were optimized to minimize the MAE. However, it was observed that the average performance of the physics-informed neural network in the test data set depends on the hyperparameters used in (25), while training. So different hyperparameters values were tested, and the combination which produced the least MAE is used to generate the results given in Table 3.

In Table 3, NN denotes the standard NN, and $PINN_{avg}$ indicates the results from the physics-informed neural network.

Incorporating the KKT conditions in the neural network training resulted in a 20% to 30% reduction in the MAE on the test dataset. Similarly, a considerable improvement was observed in the average generation constraint violation, suboptimality, and distance from optimality as well. This indicates that we can achieve higher prediction accuracy using a physics-informed neural network with the same amount of data.

Table 3
Improving average performance.

Test case		MAE_T (%)	v_g^{avg} (%)	v_{opt}^{avg} (%)	v_{dist}^{avg} (%)
Case 14	NN	0.59	0.02	0.08	5.63
	$PINN_{avg}$	0.42	0.00	0.05	4.43
Case 39	NN	0.89	0.93	0.21	10.78
	$PINN_{avg}$	0.70	0.88	0.08	10.38
Case 118	NN	1.07	1.76	0.31	11.46
	$PINN_{avg}$	0.84	0.76	0.26	10.71
Case 162	NN	2.22	1.42	0.27	37.81
	$PINN_{avg}$	2.00	1.84	0.26	36.72

Table 4
Worst-case performance.

Test case		v_g (MW)	v_g % wrt max loading	v_l (MVA)	v_l % wrt max loading
Case 14	NN	0.09	0.03	38	14.67
	$PINN_{avg}$	0.08	0.03	35	13.51
	$PINN$	0.01	0.01	34	13.13
Case 39	NN	246	3.93		
	$PINN_{avg}$	184	2.94		
	$PINN$	129	2.06		
Case 118	NN	266	6.27		
	$PINN_{avg}$	187	4.41		
	$PINN$	141	3.32		
Case 162	NN	2771	38.27		
	$PINN_{avg}$	1029	14.21		
	$PINN$	899	12.42		

3.3. Worst case performance

While examining the worst-case guarantees for generator violations, we observed no direct correlation between the average performance of the physics-informed neural network in a test data set and the maximum constraint violations. Since worst-case guarantees play a crucial role in the trustworthiness of a NN, similar to the previous case, different hyperparameters were tested to see which one would produce the lowest worst-case generation constraint violations. Then those worst-case violations values, denoted by $PINN$, are compared against (i) the standard NN and (ii) the physics-informed neural network that delivered the least MAE, denoted by $PINN_{avg}$, in Table 4 to access the importance of hyperparameters in the worst-case performance.

From Table 4, we can infer that the physics-informed neural network which produced the lowest MAE also offered a significant reduction in worst-case generation constraint violation compared to the standard NN in all the test cases. This establishes the fact that the physics-informed neural networks offer both better generalization capabilities and lower worst-case guarantees with the same training data compared to a standard NNs. However, an additional 15% to 30% reduction in worst-case generation constraint violation is achieved when we tune the hyperparameters of the physics-informed neural networks with the objective to improve specifically the worst-case guarantees instead of the objective to minimize the mean absolute error. This indicates we can further tighten the worst-case guarantees by optimizing the hyperparameters to minimize the worst-case constraint violations.

In 'Case 14', along with the generation constraint violations, we were also able to compute the worst-case line flow constraint violations. As Table 4 shows, physics-informed neural networks also resulted in lower line flow constraint violations. For Case 39, Case 118, and Case 162, we were unable to compute the worst-case line flow constraint violation since the MIQCQP problem could not be solved to zero optimality gap within 5 hr. This highlights the computational challenges associated with the extraction of the worst-case guarantees for the AC-OPF, and in general non-linear, problems. The present work focuses on

Table 5
Average performance.

Test case		MAE_T (%)	v_g^{avg} (%)	v_{opt}^{avg} (%)	v_{dist}^{avg} (%)
Case 14	$PINN$	0.60	0.07	0.06	5.75
Case 39	$PINN$	2.17	1.19	0.40	12.73
Case 118	$PINN$	1.04	1.80	0.18	10.31
Case 162	$PINN$	2.27	0.97	0.25	39.13

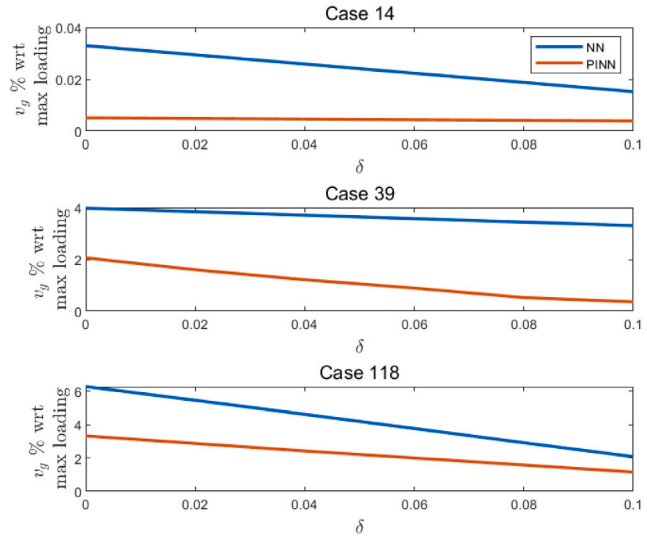

Fig. 3. Input domain reduction.

introducing the first formulations that can extract such guarantees for a non-linear optimization problem. Future work shall focus on addressing the computational issues in order to arrive at scalable algorithms for any system size.

The average performance of $PINN$, (i.e mean absolute error, average generator constraint violations, average suboptimality, and average distance to the optimal setpoint) is given in Table 5. If we compare Table 5 with Table 3, we observe that in most cases the average performance worsens when we tune the hyperparameters with the objective to reduce the worst-case generation constraint violations instead of the objective to obtain the least mean absolute error. This possibly indicates that there is a trade-off in the performance of the NN between trying to achieve the best average performance and achieving the least worst case violations. However, further analysis is required to ascertain the relationship between good average performance and least worst case violations.

3.4. Input domain reduction

Ref. [11] has shown that for linear optimization programs, the worst-case constraint violation of the NN can be reduced by training the NN on a broader input domain than the one on which it will be deployed. We explore how this approach performs for non-linear programs, both for standard NNs and PINNs. For both neural network types, the input domain for the test set was symmetrically reduced by δ as follows:

$$(0.6 + \delta)\overline{\mathbf{D}} \leq \mathbf{D} \leq (1 - \delta)\overline{\mathbf{D}} \tag{38}$$

The resulting worst-case generation constraint violation with respect to the maximum active power loading is given in Fig. 3.

We observe that similar to linear programs, for the AC-OPF as well input domain reduction of test sets improves the worst-case guarantees considerably, especially for the standard NNs. We notice that the reduction of the violations is significantly higher for standard NNs compared with the Physics-Informed Neural Networks (PINNs), although across the whole input domain reduction the PINNs exhibit always lower worst-case violations. This implies that the physics-informed neural network will only require a slightly larger input domain of the training data set to maintain the worst-case constraint violations below a certain threshold, while standard NNs would require significantly larger input domains to achieve a similarly low level of worst-case constraint violation, if at all possible.

4. Conclusion

This paper offers two key contributions. First, we introduced a framework to incorporate the physical equations of the AC-OPF inside the neural network training. Additionally, we show that we can improve the NN prediction accuracy by using physics-informed neural networks while using considerably fewer input data points. Second, we propose a method to extract and minimize the worst-case generation and line constraints violations for the AC-OPF problem. In the future, the work will be extended to include a multilevel optimization algorithm to determine the key physics-informed neural network hyperparameters which minimize worst-case constraint violations.

CRediT authorship contribution statement

Rahul Nellikkath: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft. **Spyros Chatzivasileiadis:** Supervision, Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D.K. Molzahn, I.A. Hiskens, A survey of relaxations and approximations of the power flow equations, Foundations and Trends® in Electric Energy Systems 4 (1–2) (2019) 1–221, http://dx.doi.org/10.1561/3100000012.
[2] B. Stott, J. Jardim, O. Alsac, DC power flow revisited, IEEE Trans. Power Syst. 24 (3) (2009) 1290–1300, http://dx.doi.org/10.1109/TPWRS.2009.2021235.
[3] D. Deka, S. Misra, Learning for DC-OPF: Classifying active sets using neural nets, in: 2019 IEEE Milan PowerTech, 2019, pp. 1–6, http://dx.doi.org/10.1109/PTC.2019.8810819.
[4] L. Duchesne, E. Karangelos, L. Wehenkel, Recent developments in machine learning for energy systems reliability management, Proc. IEEE 108 (9) (2020) 1656–1676, http://dx.doi.org/10.1109/JPROC.2020.2988715.
[5] F. Thams, A. Venzke, R. Eriksson, S. Chatzivasileiadis, Efficient database generation for data-driven security assessment of power systems, IEEE Trans. Power Syst. 35 (1) (2020) 30–41, http://dx.doi.org/10.1109/TPWRS.2018.2890769.
[6] A. Venzke, D.K. Molzahn, S. Chatzivasileiadis, Efficient creation of datasets for data-driven power system applications, Electr. Power Syst. Res. 190 (2021) 106614, http://dx.doi.org/10.1016/j.epsr.2020.106614.
[7] A. Venzke, S. Chatzivasileiadis, Verification of neural network behaviour: Formal guarantees for power system applications, IEEE Trans. Smart Grid 12 (1) (2021) 383–397, http://dx.doi.org/10.1109/TSG.2020.3009401.
[8] F. Fioretto, T.W.K. Mak, 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, (01) 2020, pp. 630–637.
[9] P.L. Donti, D. Rolnick, J.Z. Kolter, DC3: A learning method for optimization with hard constraints, 2021, arXiv preprint arXiv:2104.12225.
[10] X. Pan, T. Zhao, 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.
[11] A. Venzke, G. Qu, S. Low, 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.
[12] R. Nellikkath, S. Chatzivasileiadis, Physics-informed neural networks for minimising worst-case violations in DC optimal power flow, in: 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm, IEEE, 2021, pp. 1–7.

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

[14] V. Kekatos, G. Wang, H. Zhu, G.B. Giannakis, PSSE redux: Convex relaxation, decentralized, robust, and dynamic solvers, in: Advances in Electric Power and Energy: Static State Estimation, Wiley Online Library, 2020, pp. 171–208.

[15] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, http://dx.doi.org/10.1016/j.jcp.2018.10.045, URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[16] R. Nellikkath, S. Chatzivasileiadis, Physics-informed neural networks for minimising worst-case violations in DC optimal power flow, 2021, arXiv preprint arXiv:2107.00465.

[17] 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, 2019, arXiv preprint arXiv:1908.02788.

[18] S.S. Khonde, S.S. Dhamse, A.G. Thosar, Power quality enhancement of standard IEEE 14 bus system using unified power flow controller, Int. J. Eng. Sci. Innov. Technol. 3 (5) (2014).

[19] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.

[20] 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, 2016, arXiv preprint arXiv:1603.04467.

[21] 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.

[22] R. Nellikkath, S. Chatzivasileiadis, Supplementary data and code: Physics-informed neural networks for AC optimal power flow, 2021, https://github.com/RahulNellikkath/Physics-Informed-Neural-Networks-for-AC-Optimal-Power-Flow.