IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 April 2024, accepted 22 May 2024, date of publication 28 May 2024, date of current version 30 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406471

RESEARCH ARTICLE

Physics-Informed Neural Networks for Power Systems Warm-Start Optimization

ÀLEX TUDORAS-MIRAVET[1], ESTEBAN GONZÁLEZ-IAKL[1], ORIOL GOMIS-BELLMUNT[1,2], (Fellow, IEEE), AND EDUARDO PRIETO-ARAUJO[1,2], (Senior Member, IEEE)

[1]Centre d'Innovació Tecnològica en Convertidors Estàtics i Accionaments, Departament d'Enginyeria Elèctrica, Universitat Politècnica de Catalunya, 08028 Barcelona, Spain
[2]eRoots Analytics, 08039 Barcelona, Spain

Corresponding author: Àlex Tudoras-Miravet (alex.tudoras.miravet@gmail.com)

Research project TED2021-130351B-C21 funded by MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. This work was supported by the ACCIÓ-Generalitat de Catalunya through the Project OPTIWIND J-04143, under Grant ACE034/21/000071. The work of Oriol Gomis-Bellmunt was supported by the ICREA Academia Program. The work of Eduardo Prieto-Araujo was supported by the Serra Húnter Program.

ABSTRACT Several studies have demonstrated the potential of machine learning methods to solve optimal power flow problems. However, designing a scalable physics-informed neural network (PINN) model that can improve its performance being trained in diverse scenarios by considering the significance of its several elements remains a challenging task. Here, we propose an approach that leverages the inclusion of physical constraints into the loss function using a penalty factor and the utilization of bounds of optimization variables in the activation functions to enhance the generalization performance of tuned neural networks. The results indicate that this method significantly improves the success rate and computational speed gains of AC-optimal power flow (AC-OPF) calculations, especially when forward predictions are employed as warm-start points. Our PINN models are trained using accurate AC-OPF solutions from slow high-precision interior-point solvers across several power system scenarios. Furthermore, we examine and demonstrate the critical role of adjusting PINN's hyperparameters and architecture design in achieving the optimal tradeoff between empirical error and constraint violation to make accurate and feasible predictions. A combination of stochastic methods and grid search is utilized to establish a reliable and efficient way of performing optimization calculations for a wide range of power systems using collected data. The proposed PINN model offers a promising solution for adapting neural networks to diverse scenarios of a physical problem. Furthermore, it offers a robust methodology for successfully addressing optimal power flow (OPF) problems in power systems.

INDEX TERMS Hyperparameters, optimal power flow, physics-informed neural networks, warm start.

I. INTRODUCTION

Modern power systems are experiencing a deep transformation due to the high penetration of renewable generation [1]. As power systems continue to grow in complexity, the need to improve computational power and adopt innovative strategies to coordinate multiple energy sources, storage units, and network assets becomes crucial to optimize their performance and ensure a stable and efficient energy supply [2], [3]. The growing sophistication of power systems highlights the significance of Optimal Power Flow (OPF) solutions in effectively managing power flow in modern power networks, making it a challenge to compute them without compromising the conflicting objectives of robustness and accuracy.

The current state-of-the-art optimization methods for non-convex large systems include trust region methods, stochastic gradient descent, and derivative-free optimization algorithms, which have been shown to provide robust and efficient solutions for a variety of problems [4], [5]. The interior

The associate editor coordinating the review of this manuscript and approving it for publication was Ton Duc Do.

point method has been shown to be a robust optimization method, able to handle a wide range of systems with good performance, and being the most widely used method for resolving OPF [6], [7], [8]. Despite its high efficiency as an iterative algorithm, it requires the computation of the second derivatives of the Lagrangian of the system, at each iteration step. Moreover, the non-convex nature of the power flow equations appearing in the equality constraints can make this method prohibitively slow for large-scale systems [9].

An innovative approach to achieving efficient OPF solutions is the use of artificial neural networks (ANN). In recent years, machine learning techniques have gained significant attention due to their ability to handle complex and nonlinear relationships with learning models inspired by the structure and function of the human brain [10], [11]. Additional developments in recent years have also made it possible to apply them to real-world problems in the power sector, such as state estimation, power system stability analysis, and voltage control; making them a promising tool for the operation of power systems [12], [13], [14], [15], [16].

In particular, warm-start methods have been extensively researched and employed in the optimization of power systems to improve the efficiency and speed of finding an optimal solution [17], [18], [19]. The idea behind these methods is to use an initial solution as a starting point rather than starting the optimization process from scratch, which can significantly reduce the time required to find a solution. Research in this area strives to improve the accuracy and efficiency of power systems optimization. Reference [20] presents the notion that traditional warm-start techniques, such as relying solely on the previous operating point, may not necessarily capture the dynamics introduced by new inputs in some problems, reinforcing the notion that there is a need to find more effective warm-start approaches. Furthermore, [21] proposes an alternative warm-start algorithm for interior point methods using previous solutions which reduces the number of iterations by around 80%, highlighting the potential of alternative warm-start approaches using statistical predictive techniques, which can lead to improved computational efficiency and solution quality in a wider range of problems. Along similar lines, one trend is to use neural networks and decision trees to model a power grid and produce an initial solution to warm start the optimization [17], [18], [19], [22].

The existing methods can make fast regressions with ANNs which can be trained with any data size. However, this type of work focuses on attempting to match, individually, variables' optimal values, but the predicted values provided by these end-to-end methods are not guaranteed to belong to the feasible operating region [23], [24]. Violating physical constraints and laws may result in substantially longer solve times or even convergence failure; and, additionally, severe security issues for real power grids.

To tackle this issue, Physics Informed Neural Networks (PINNs) have also gained attention as they incorporate physical constraints and laws into machine learning models, resulting in high accuracy levels with simpler neural network architectures, especially shallow neural networks with a single hidden layer [25], and requiring considerably less training data [26]. In addition, some studies refer to similar concepts as Physics-Aware Neural Networks [27] and Physics-Guided Deep Neural Networks [28] by exploiting the topology and physical laws governing the power grid to estimate the state of the system and perform a power flow analysis.

PINNs have been proven to significantly improve the accuracy and reliability of predictions while aiming for physically feasible solutions [29], [30], [31], [32], [33]. Moreover, recent research suggests that to ensure the monitoring of grid operating conditions, the estimation of the state of the power system in real time using deep neural networks requires only offline training and minimal tuning effort to outperform traditional energy system estimation schemes [34]. In fact, numerical results demonstrate that the proposed methods outperform other alternatives and can be easily extended to address different test systems by adjusting only a few hyperparameters [35].

The development of such methods is important to ensure that ANNs are appropriately designed and optimized for the given problem and data, and provide more accurate and reliable predictions for power systems optimization.

Although PINNs have shown great potential in solving problems in power systems; indicating a potential for greater accuracy and faster convergence than regular ANNs, there are still some gaps in the existing knowledge about using them for optimality in power systems.

Although it is common to see statements in research papers on PINNs or neural networks in general that suggest that hyperparameters, activation functions, architecture, and other design choices should be made based on the nature of the problem and the data, these statements often lack sufficient detail and guidance for practitioners. In reality, the process of selecting appropriate hyperparameters and architecture can be highly complex and involve a significant amount of trial and error. Furthermore, many researchers simply rely on common practices or intuition to guide their decisions, rather than conducting rigorous experimentation to determine the best design choices. As a result, researchers need to provide more detailed and nuanced guidance on the selection of hyperparameters and architecture, backed by empirical evidence and thorough experimentation. Only then can we be confident that our models are appropriately designed and optimized for the given problem and data.

While ANNs can be designed, parametrized, and regularized in a methodical way to fit the input data and desired output, the generalization ability of PINNs is still not well understood. Although the literature provides several examples of the efficacy of PINNs for specific cases and the approach is theoretically attractive, there is limited consensus and knowledge on its applicability. To address these challenges, we present a multilevel optimization method for determining the essential hyperparameters and architecture of the networks, to reduce violations generally for different scales and grid topologies, while regularizing the learning procedure.

Furthermore, the results predicted by current approaches are slightly more feasible but still do not guarantee them to be feasible points, unlike the claimed advantage of PINNs. Additionally, previous studies do not provide a detailed examination of integration with other optimization techniques. Therefore, the proposed research aims to compare the performance of shallow PINNs with a single hidden layer against other numerical solvers to understand their relative strengths and limitations and to evaluate the stabilization and regularization of the results that the physical constraints in the model provide.

In summary, this paper presents a novel approach leveraging physics-informed neural networks (PINNs) to enhance the efficiency and accuracy of optimal power flow (OPF) solutions in power systems. The key contributions include the incorporation of physical constraints into the neural network architecture, hyperparameter tuning for optimal tradeoffs between empirical error and constraint violations, and the demonstration of significant improvements in computational efficiency and success rates for OPF calculations. These advancements address critical challenges in power system optimization, particularly in the context of increasing complexity and renewable energy integration. Moreover, the proposed methodology offers a promising solution for adapting neural networks to diverse scenarios and improving the robustness of OPF solutions. By comparing and contrasting our findings with related research, such as the work presented in the NSF papers on machine-learning approaches for predicting AC-OPF solutions [36], we assert the unique contributions of our study to the field of power systems optimization. Specifically, our approach offers a complementary perspective, emphasizing the integration of physics-based constraints and the optimization of hyperparameters to achieve more accurate and feasible OPF solutions efficiently. Through empirical validation and computational characterization, we demonstrate the effectiveness and practical applicability of our proposed methodology, highlighting its potential to advance the state-of-the-art in power system optimization.

In this study, we show that:

- Incorporating physical constraints into the loss function, and bounds in the activation function to make a physics-informed design of architecture can improve the generalization performance of neural networks, and the classic regularization techniques in neural networks become irrelevant when implementing PINNs.
- Hyperparameter tuning is critical for selecting the best tradeoff point between empirical error and constraint violations when using penalty factors to weight constraint violations. By staying within the stable learning region, where the network produces accurate predictions consistent with the physical behavior of the system, it is possible to optimize the performance of the physics-informed neural network.
- There exists a critical balancing region of the penalty factor. In this region, PINNs face challenges in balancing the fit to the data and constraint violations, which results in an increase in empirical error.
- It is possible to identify a maximum value of the penalty factor, beyond which the network becomes unstable and experiences numerical instabilities during training, making it impossible to learn.
- The proposed approach can significantly improve the computational efficiency and success rate of the optimization algorithm for AC-OPF in power systems, compared to starting from scratch, significantly reducing computational time and failure.

The rest of this paper is structured as follows. Section II presents the used model of a generic electrical power system. Section III explains our physics-informed neural network architecture and training design. Section IV describes our data collection, simulations, and evaluation methodology for enhancing the network performance and choosing hyperparameters. Section V presents a detailed analysis of our computational characterization and validation results. Section V discusses the implications of the study and conclusions along with future research directions.

II. POWER NETWORK MODEL

Consider an electrical power system consisting of n buses denoted by $i \in \mathcal{N} = \{1, \dots, n\}$, where each bus represents a location within the power grid. Among these buses, there are n_G generator buses denoted by $k \in \mathcal{G} \subseteq \mathcal{N}$, which are responsible for injecting power into the system. The power flows through n_L transmission or distribution lines corresponding to directed edges $e_{ik} \in \mathcal{E} \subset \mathcal{N} \times \mathcal{N}$, which form the backbone of the system.

To model the power system per-unit, we introduce the optimal power flow variables, including the three types of nodes: load, generator (PV), and voltage-controlled (PQ) buses. The state of the power grid, denoted by y, determines the current operating condition of the system and is represented by a vector y including the voltage magnitude and angle, active and reactive power injections from generators, and reactive power injection from voltage-controlled buses. Specifically, we represent the voltage magnitude and angle of bus i as $\vec{v}_i = v_i \cdot e^{j\delta_i}$, where v_i is the voltage magnitude and δ_i is the angle. The active and reactive power injections from generator bus k are denoted by p_k and q_k, respectively. The reactive power injection from voltage-controlled bus m is denoted by q_m. Hence, to represent the state of the system, the following vector $\boldsymbol{y}$ is used:

$$\boldsymbol{y} = \begin{bmatrix} \delta & \boldsymbol{v} & \boldsymbol{p} & \boldsymbol{q} \end{bmatrix}^T \in \mathbb{R}^{N_y}, \tag{1}$$

where $\delta, \boldsymbol{v} \in \mathbb{R}^n$, $\boldsymbol{p} \in \mathbb{R}^{n_G}$, $\boldsymbol{q} \in \mathbb{R}^{n_G+n_{PQ}}$, and n_{PQ} is the number of voltage-controlled buses. In addition to the state variables, the model takes into account the active p_i^d and reactive q_i^d power loads at each bus i, as well as time-varying parameters that represent demand response or available capacity, represented by the measurable input vector $\boldsymbol{x} \in \mathbb{R}^{N_x}$, which influence the behavior of the system.

A. AC-OPF OPERATOR FORMULATION

The operation of an electrical grid involves ascertaining the state y with optimal operating conditions of the power system while taking into account the cost-benefit relationship and restrictions of the entire system. This optimization problem can be formulated as an alternating current optimal power flow (AC-OPF) problem [37], where the goal is to minimize an objective function $f(\cdot) : \mathbb{R}^{N_y} \rightarrow \mathbb{R}$ that quantifies the production cost of all generators.

$$\begin{cases} \min\limits_{\boldsymbol{y}} \ f(\boldsymbol{y}) \\ \text{ s.t. } \ h(\boldsymbol{x}, \boldsymbol{y}) = 0 \\ g(\boldsymbol{x}, \boldsymbol{y}) \leq 0 \\ \underline{\boldsymbol{y}} \leq \boldsymbol{y} \leq \bar{\boldsymbol{y}} \end{cases} \tag{2}$$

This formulation gives the possibility of looking at the optimization problem as an operator [38] that maps the grid input $\boldsymbol{x}$ to the optimal value $\boldsymbol{y}^*$ of the optimization variables $\boldsymbol{y} \in \mathbb{R}^{N_L}$, establishing a non-linear relationship between the two vectors as an AC-OPF operator $\psi(\cdot)$. To simplify the notation used throughout the rest of the paper, the optimal value will be referred to as $\boldsymbol{y}$, rather than $\boldsymbol{y}^*$.

$$\boldsymbol{y} = \psi(\boldsymbol{x}) \tag{3}$$

B. CONSTRAINTS FOR INTERCONNECTED POWER GRIDS

The efficient operation of interconnected power grids requires balancing power flow under various operating conditions and satisfying the AC power flow equations (AC-PF) that describe power injections at each node. The AC-PF equations can be written in terms of the optimal power flow variables [39], [40], [41].

$$p_i^g - p_i^d = \boldsymbol{v}_i \sum_{k=1}^{n} \boldsymbol{v}_k \left(g_{ik} \cos \delta_{ik} + b_{ik} \sin \delta_{ik}\right) \quad \forall i \in \mathcal{N} \tag{4}$$

$$q_i^g - q_i^d = \boldsymbol{v}_i \sum_{k=1}^{n} \boldsymbol{v}_k \left(g_{ik} \sin \delta_{ik} - b_{ik} \cos \delta_{ik}\right) \quad \forall i \in \mathcal{N} \tag{5}$$

Here, p_i^g and p_i^d are the active power generated and consumed at node i, respectively, while q_i^g and q_i^d are the corresponding reactive power values. The angle difference between node i and node k is denoted by $\delta_{ik} = \delta_i - \delta_k$, and g_{ik} and b_{ik} are the entries of the real and imaginary parts of the bus admittance matrix $Y = G + jB$, in our per-unit system. Collectively, these equations describe a set of $N_E = 2\ n$ fundamental nonlinear mathematical relationships governing the behavior of the power system, which can be compactly represented as a function $h(\cdot)$.

$$\begin{aligned} &h(\cdot) : \mathbb{R}^{N_x} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}^{N_E} \\ &h(\boldsymbol{x}, \boldsymbol{y}) = 0 \end{aligned} \tag{6}$$

In addition to the AC-PF equations, an effective model must incorporate the generation capacities and operational limits of the system, such as transmission line limitations. If P_{ik} and Q_{ik} are the active and reactive power flowing from bus i to bus k over a specific line, then the following equations can be used to calculate them:

$$p_{ik} = \boldsymbol{v}_i \boldsymbol{v}_k \left(g_{ik} \cos \delta_{ik} + b_{ik} \sin \delta_{ik}\right) - \boldsymbol{v}_i^2 g_{ik} \tag{7}$$

$$q_{ik} = \boldsymbol{v}_i \boldsymbol{v}_k \left(g_{ik} \sin \delta_{ik} - b_{ik} \cos \delta_{ik}\right) + \boldsymbol{v}_i^2 \left(b_{ik} - b_{ik}^{\text{sh}}\right) \tag{8}$$

Here, b_{ik}^{sh} is the shunt susceptance of the line connecting bus i to bus k, and the apparent power flow is constrained to remain within line ratings established by specific values $|\bar{s}_{ik}|$. Note that the inequality constraints in optimal power flow models can also be expressed in terms of current; however, have chosen to express them in terms of power flow for simplicity and consistency with the previous equations.

$$\sqrt{p_{ik}^2 + q_{ik}^2} \leq |\bar{s}_{ik}| \quad \forall e_{ik} \in \mathcal{E} \tag{9}$$

Specifically, these constraints can be represented as a compact inequality described by $g(\cdot)$.

$$\begin{aligned} &g(\cdot) : \mathbb{R}^{N_x} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}^{N_I} \\ &g(\boldsymbol{x}, \boldsymbol{y}) \leq 0 \end{aligned} \tag{10}$$

Finally, it is also assumed that the physical characteristics and technical specifications impose lower and upper bounds on the individual variables of the system state for all nodes, such as generation and voltage limits.

$$\underline{\boldsymbol{y}} \leq \boldsymbol{y} \leq \bar{\boldsymbol{y}} \tag{11}$$

Known values, such as the reference angle, are also taken into account by utilizing the same value for both bounds; for example, $0 \leq \delta_1 \leq 0$.

III. PHYSICS-INFORMED NEURAL NETWORK FOR REGRESSION

In this section, we present a physics-informed neural network (PINN) model $\mathcal{M}_{[\omega]}$ that combines a feedforward neural network architecture with parameters $\omega \in \Omega$, consisting of weights and biases, a set of physical constraints, and is trained using carefully chosen hyperparameters $\theta \in \Theta$ and regularization techniques, so that it approximates the optimal power flow operator $\psi(\cdot)$, which brings any input power demand to the AC-OPF solution.

A. FORWARD REGRESSOR

In regression techniques, the sophisticated relationship given by $\psi(\cdot)$ can be approximated by the operations of a neural network, treating $\boldsymbol{x}$ as the input and the optimal values of $\boldsymbol{y}$ as the output of a feedforward neural network (FNN) model, as in Fig. 2, with $L - 1$ hidden layers, each consisting of N_l neurons, $l = 0, 1, \ldots, L$, from $N_0 = N_x$ until $N_L = N_y$. During forward propagation, the input vector $\boldsymbol{x} = \boldsymbol{a}^{[0]}$ is transformed into a series of weighted outputs $z^{[l]}$ through the application of weights $\boldsymbol{w}^{[l]}$ and biases $\boldsymbol{b}^{[l]}$

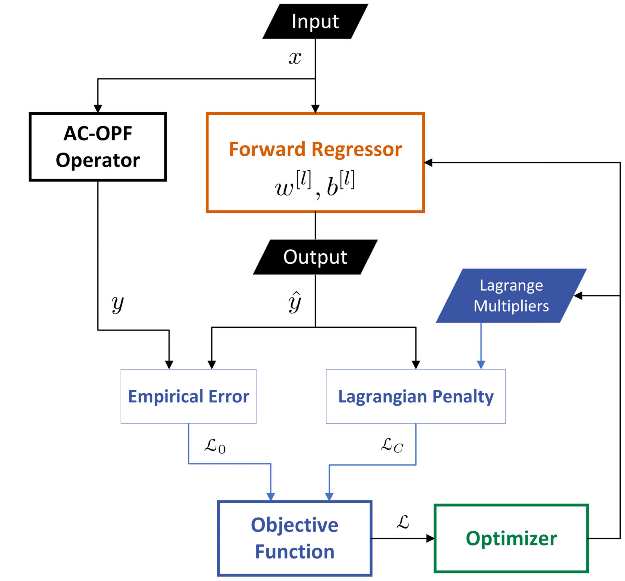

FIGURE 1. Flowchart of the physics-informed neural network training model operations model. The model combines a feedforward neural network architecture with parameters $\omega \in \Omega$, physical constraints, and carefully chosen hyperparameters $\theta \in \Theta$ to approximate the optimal power flow operator $\psi(\cdot)$. The training process involves minimizing the distance between the predicted output and the ground data, incorporating data-driven and physics-informed loss terms. The optimizer updates the weights, biases, and Lagrange multipliers of the neural network to minimize the objective function (22).

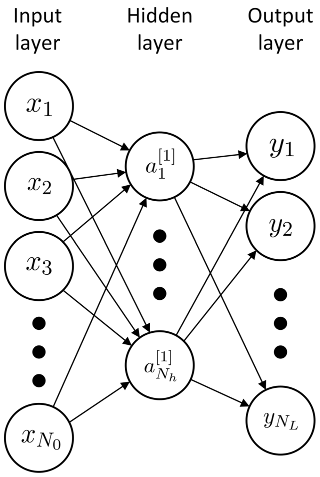

FIGURE 2. Schematic representation of a Feedforward Neural Network (FNN) architecture and associated activations $a^{[l]}$, consisting of an input layer with N_0 neurons where $x = a^{[0]}$, a hidden layer with N_h neurons, and an output layer with N_L neurons where $y = a^{[2]}$.

in each layer [42]. These weighted outputs are then passed sequentially through activation functions $\phi^{[l]}$ to introduce non-linearity and produce the final output vector $\hat{\boldsymbol{y}} = F(\boldsymbol{x})$ as the activation of the last layer, which is an estimate of the optimal state.

$$\boldsymbol{z}^{[l]} = \boldsymbol{w}^{[l]}\boldsymbol{a}^{[l-1]} + \boldsymbol{b}^{[l]} \tag{12}$$

$$\boldsymbol{a}^{[l]} = \phi^{[l]}\left(\boldsymbol{z}^{[l]}\right), \quad l = 1, \ldots, L \tag{13}$$

Therefore, for a network with 1 hidden layer, the Forward Regressor (Fig. 1) is based on the operator $F(\cdot)$.

$$F(\boldsymbol{x}) = \phi^{[2]}\left(\boldsymbol{w}^{[2]} \cdot \phi^{[1]}\left(\boldsymbol{w}^{[1]} \cdot \boldsymbol{x} + \boldsymbol{b}^{[1]}\right) + \boldsymbol{b}^{[2]}\right) \tag{14}$$

The voltage and power bounds are typically non-negotiable and must be satisfied for the power system to operate safely and reliably. Therefore, these bounds are treated as hard constraints and are included in the output activation function of the artificial neural network. In fact, the output activation function is responsible for transforming the output data of the last layer of neurons in the network into a format that is appropriate for the problem being solved. To ensure that the output data falls within the allowed bounds, $\phi^{[L]}(\cdot)$ must be carefully selected during the design of the artificial neural network.

Effective regularization plays a pivotal role in training neural networks that can generalize well to unseen data [43]. To achieve this, careful consideration must be given to the selection of $\underline{\boldsymbol{y}}$ and $\bar{\boldsymbol{y}}$. These bounds define the permissible range of target values for the network during training. If the chosen bounds are too wide or excessively large, the network can encounter numerical instability and difficulty in converging to an optimal solution. This can make it challenging for the PINN learning algorithm to navigate and make precise updates to the model parameters, resulting in poor regularization. To address this issue, it is necessary to accurately capture the range $\left[\underline{\boldsymbol{y}}, \bar{\boldsymbol{y}}\right]$ within a compact and minimal space that appropriately encompass the plausible range of target values expected in the problem domain, without extreme or unrealistic values.

Since the grid state variables are often subject to different bounds, they are treated as boundary constraints, and a vector-valued bounded activation function must be used, consisting of distinct N_y components for each output variable.

$$\underline{\boldsymbol{y}}_i \leq \phi_i^{[L]}\left(z_i^{[L]}\right) \leq \bar{\boldsymbol{y}}_i \quad \forall z_i^{[L]} \in \mathbb{R}, \forall i = 1, \ldots, N_y \tag{15}$$

In a neural network, it is crucial to include non-linear activation functions within the hidden layers [44], [45]. The reason for this is that linear functions cannot capture complex patterns and relationships within the data. Non-linear activation functions introduce non-linearity to the network, enabling it to learn and model these complex patterns.

The best choice for the activation function in the hidden layer is the modified sigmoid or the Swish function, which have been shown to outperform other activation functions for deep neural networks [46], [47], [48], [49], [50].

For the output layer, a linear activation function is often used. This is because regression problems involve predicting a continuous value, and a linear activation function allows the network to output any real number as its prediction. Nevertheless, in our problem, the output values are bounded by a known range, then, it is proposed that a scaled sigmoid or hyperbolic tangent might be the preferred option to consider to ensure the variables lie within the specified values.

B. OBJECTIVE FUNCTION BLOCK

To train the PINN model, we use a supervised learning approach to minimize the distance between the predicted output and the ground truth data, where we minimize a loss function to improve the accuracy of its predictions [51]. A data-driven loss term $\mathcal{L}_0$ measures the error between the predicted and true output values, obtained from the AC-OPF Operator (Fig. 1), while a physics-informed loss term $\mathcal{L}_C$ enforces the constraints imposed by the underlying physics of the problem being solved. By combining these two terms, we ensure that the network learns both the data and the underlying physics, leading to more accurate and robust predictions.

1) EMPIRICAL ERROR

Selecting an appropriate data-driven loss function is critical to a model's behavior and its ability to generalize to new data. However, there is no standardized loss function for physics-informed supervised neural network training in power systems, with researchers often choosing different functions based on their specific problem requirements and preferences.

Power system data is prone to noise and uncertainty from various sources, including sensor errors, communication delays, and environmental factors [52], [53]. This can result in imprecise data that is difficult to accurately model and predict, especially when dealing with variables such as Optimal Power Flow (OPF). To address this, it is crucial to select a loss function that is robust to noise and can handle outliers in the data. The Mean Absolute Error (MAE) is the preferred choice in such situations, as it considers the absolute magnitude of errors rather than their squared values [54]. In this case, using MAE can help to regularize the model and ensure better performance, even with limited or noisy training data. The function is shown next:

$$\mathcal{L}_0(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{N_y} \sum_{i=1}^{N_y} |\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i| \tag{16}$$

2) REGULARIZATION

Regularization techniques are essential for preventing overfitting or high variance during ANN training [55]. L1 regularization, also known as Lasso regression, promotes sparsity in weight matrices, leading to simpler models and better generalization performance [56]. L2 regularization, also known as Ridge regression, encourages small values in weight matrices, effectively minimizing the risk of overfitting by reducing the magnitude of weights [57]. These techniques are particularly useful in scenarios where the data is noisy or contains other sources of variability (Fig. 3). To implement these techniques, a regularization term, denoted by $\mathcal{L}_R$, is added to the loss function.

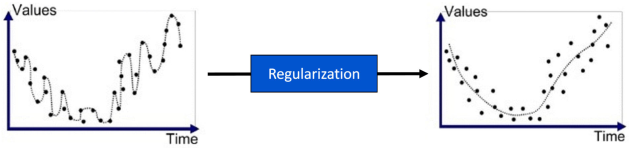

FIGURE 3. Effect of L1 and L2 regularization on regression results in neural network time series predictions [58].

$$\mathcal{L}_{R_{L1}} = \sum_{l=1}^{L} \|\boldsymbol{w}^{[l]}\|_1 = \sum_{l=1}^{L} \sum_{i,j} |\boldsymbol{w}_{i,j}^{[l]}| \tag{17}$$

$$\mathcal{L}_{R_{L2}} = \sum_{l=1}^{L} \|\boldsymbol{w}^{[l]}\|_2 = \sum_{l=1}^{L} \sum_{i,j} \boldsymbol{w}_{i,j}^{[l]\ 2} \tag{18}$$

To determine the impact of regularization on the model, it is crucial to evaluate its contribution. Choosing either L1 or L2 regularization, or a weighted combination of the two requires exploration to strike the right balance between the model's simplicity and the accuracy of its predictions.

3) AUGMENTED LAGRANGIAN PENALTY

It is worth noting that in the literature, there is often mention of the use of feasibility mapping or projection methods to incorporate physical constraints into the loss function. However, in practice, the application of these approaches to machine learning solutions typically involves approximations of the problem or, if not combined with penalty methods, iterative schemes that minimize the projection distance, which can complicate the problem with additional conflicting goals and increase computational complexity unnecessarily. In contrast, in practice, stochastic gradient descent (SGD)-like schemes can often reach better solutions with fewer iterations.

To effectively address the satisfaction of physical constraints while avoiding unnecessary computational complexity, a Lagrangian-based approach is a powerful strategy that can be utilized. By incorporating prior knowledge about the problem into the loss function, we can constrain the solution space and guide the model towards more physically meaningful solutions [59], [60].

In particular, we propose an augmented Lagrangian penalty method that involves adding a penalty term to penalize soft constraint violations and iteratively making the weights and biases produce feasible predictions. This method can achieve a high solution optimality and constraint compliance, even in real-time [61]. To compute its value, it is important to consider the type of each constraint to use ν_h, ν_g, whether it is equality or inequality. This can be achieved by utilizing the following equations, where the operators $(\cdot)^2$ and $(\max\{\cdot\})^2$ are applied element-wise and are chosen to enforce differentiability.

$$h(\boldsymbol{x}, \boldsymbol{y}) = 0 \Longrightarrow \nu_h(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \big(h(\boldsymbol{x}, \hat{\boldsymbol{y}})\big)^2 \tag{19}$$

$$g(\boldsymbol{x}, \boldsymbol{y}) \leq 0 \Longrightarrow \nu_g(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \big(\max\{0, g(\boldsymbol{x}, \hat{\boldsymbol{y}})\}\big)^2 \tag{20}$$

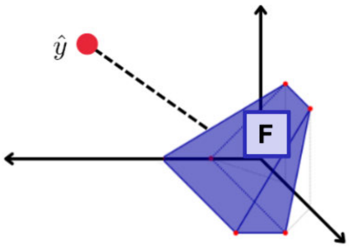

FIGURE 4. Schematic representation of the distance, in a dashed line, between the point $\hat{\boldsymbol{y}}$ predicted by a PINN, and the region F determined by the physical constraints.

The constraint violations are introduced into the loss function using Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{N_E}$, $\mu \in \mathbb{R}^{N_I}$ to capture the penalty that leads to more stable training, which is a measure of the distance between the predicted point and the feasible region defined by the physical constraints (Fig. 4).

$$\mathcal{L}_{C\,[\boldsymbol{\lambda},\mu]}(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \boldsymbol{\lambda}^T \cdot \nu_h(\boldsymbol{x}, \hat{\boldsymbol{y}}) + \mu^T \cdot \nu_g(\boldsymbol{x}, \hat{\boldsymbol{y}}) \tag{21}$$

4) BALANCE FACTORS

To design neural networks with multiple objective functions, we use the multi-objective optimization technique called multi-objective backpropagation (MOBP). MOBP is a variation of the traditional backpropagation algorithm that is designed to optimize multiple objectives simultaneously; such as the empirical loss and the constraints violation. This algorithm involves computing the gradients of each objective separately and then using a weighted combination of these gradients to update the network weights.

To apply MOBP, the different factors of the objective function are weighted by hyperparameters that balance the terms in the loss function, and they must be tuned to optimize the PINN's performance. In particular, a penalty factor α and a regularization rate γ are used. Additionally, due to the dependency $\hat{y} = F(x)$, we simplify our notation.

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{L}_0(\boldsymbol{y}, F(\boldsymbol{x})) + \alpha \cdot \mathcal{L}_{C\,[\boldsymbol{\lambda},\mu]}(\boldsymbol{x}, F(\boldsymbol{x})) + \gamma \cdot \mathcal{L}_R \tag{22}$$

When setting the parameters of the augmented Lagrangian penalty in Equation (22), it is essential to determine suitable values for the penalty factor α, and the regularization rate γ. The α parameter is system-specific and plays a crucial role in balancing the trade-off between minimizing the empirical error and penalizing constraint violations during the model's learning process. Detailed explanations of hyperparameter tuning for alpha are provided in the Methodology (Section IV), where we outline the process for systematically determining the optimal values tailored to each specific system under consideration.

On the other hand, the γ parameter represents the regularization strength and controls the impact of regularization on the model's training process. Unlike α, its values are fixed and remain invariant across different systems. These optimal values are determined through extensive experimentation and analysis presented in the Results (Section V), constituting one of the key contributions of this paper.

C. OPTIMIZER

The Optimizer block in 1 is responsible for updating the weights, biases, and Lagrange multipliers of a neural network during the training process (Fig. 1). It achieves this by minimizing the objective function with a learning rate η, which is a hyperparameter that requires careful selection. At every learning step t, the optimizer computes the update on the network's parameter ω, with value ω_t that is required to minimize the objective function, based on the current input/output pairs $(\boldsymbol{x}_t, \boldsymbol{y}_t)$.

$$\omega_{t+1} = \omega_t + D\left(\omega_t, \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t)\right) \tag{23}$$

Several algorithms can be used for optimization, with stochastic gradient descent (SGD) being the most popular. SGD computes the gradient of the loss function for each data point in the training set and updates the network parameters by taking a step in the opposite direction of the gradient to minimize the loss. However, it may lead to convergence issues or unstable behavior for some learning rates η. To address these issues, the variant Adaptive Moment Estimation (Adam) was proposed, which has been shown to converge faster and achieve better performance with the original hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ [62], and an exponential learning rate decay, which helps to speed up the convergence and makes the Adam optimizer more stable and less sensitive to fluctuations in the gradients.

$$\eta_t = \eta_0 \cdot \xi^{t/100} \tag{24}$$

Every particular parameter ω has first and second moment estimates m_ω, v_ω which are updated at each training step by receiving the current values for the loss function (Fig. 1), and are used to update the network's parameter,

$$\boldsymbol{m}_\omega(t) = \beta_1 \boldsymbol{m}_\omega(t-1) + (1-\beta_1)\nabla_\omega \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t) \tag{25}$$

$$\boldsymbol{v}_\omega(t) = \beta_2 \boldsymbol{v}_\omega(t-1) + (1-\beta_2)\left(\nabla_\omega \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t)\right)^2$$

$$\Longrightarrow D\left(\omega, \mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t)\right) = -\eta_t \cdot \frac{\boldsymbol{m}_\omega(t)}{1-\beta_1^t} \cdot \sqrt{\frac{1-\beta_2^t}{\boldsymbol{v}_\omega(t)+\epsilon}} \tag{26}$$

where η_t is the learning rate at step t.

Adam is particularly well-suited for training large neural networks with large datasets, as it requires relatively little memory compared to other stochastic optimization algorithms. It is used for both the neural network and the Lagrange multipliers. However, the learning rate stepsize for updating the Lagrange multipliers is weighted by α, depending on the relative importance of the constraints. Therefore, a criterion for selecting optimal hyperparameters is necessary.

IV. METHODOLOGY

A. DATA COLLECTION

1) GENERATION OF INPUTS

We collected a diverse set of training data by varying the parameters of several power system scenarios, listed in Table 1.

TABLE 1. Description of power system scenarios used to test PINN's performance to predict AC-OPF solutions with their corresponding number of buses n, generators n_G, and lines n_L.

Scenario	n	n_G	n_L	Description
A	5	5	6	PJM 5-bus system [63]
B	9	3	9	9-bus system [64]
C	14	5	20	IEEE 14-bus test [65]
D	22	1	21	22-bus RDS [66]
E	30	6	41	IEEE 30-bus test [67]
F	39	10	46	NE 39-bus system [68]
G	69	1	68	69-bus RDS [69]
H	70	2	78	70-bus system [70]
I	74	1	73	74-bus RDS [71]
J	94	1	93	Portugal 94-bus RDS [72]
K	118	54	186	IEEE 118-bus test [73]
L	141	1	140	141-bus RDS [74]

For each scenario $i \in A, \ldots, L$, we kept its grid topology and admittances as constant parameters; and generated diverse inputs to create N_S random sample cases. To accomplish this, we used a random uniform distribution to generate each input $\boldsymbol{x}_j^i$ for the case $j \in 1, 2, \ldots, N_S$, based on the original input vector $\boldsymbol{x}^i \in \mathbb{R}^{N_0^i}$. The range of the input values was set from 0% to 200%, allowing for considerable fluctuations in power demand and generation, thus enhancing the model's ability to generalize and more accurately predict new data.

In particular, these scenarios optimize their states to minimize a quadratic cost function parametrized by the diagonal matrix of costs $\left[C\right]$ obtained from the vector $\boldsymbol{c}$.

$$f(\boldsymbol{y}) = \boldsymbol{p}^{g\,T} \cdot \left[C\right] \cdot \boldsymbol{p}^{g} = \sum_{i=1}^{n_G} \boldsymbol{c}_i \cdot \boldsymbol{p}_i^{g\,2} \tag{27}$$

Therefore, the input vectors $\boldsymbol{x}_j^i$ contain active and reactive power demands in the buses $(\boldsymbol{p}^d, \boldsymbol{q}^d)$, as well as the costs of the objective function $\boldsymbol{c}$. However, the objective function is a choice of the particular problem in question, and the proposed PINN methodology applies to any user-defined objective function and a set of input parameters.

2) ACCURATE AC-OPF SOLUTIONS

To obtain accurate and precise solutions for AC OPF problems, we applied the `IPOPT` interior point solver [75]. It is a nonlinear optimization technique that iteratively solves a sequence of linearized subproblems to move towards the optimal solution while satisfying the constraints on the power system, known for its ability to provide more accurate and precise solutions than the quasi-Newton or linear-programming solvers for this type of problems, even though it can be computationally expensive. In particular, we adjusted the solver parameters to improve the accuracy and precision of the solution at the cost of increasing the computation cost, with convergence tolerance for optimization $\epsilon = 10^{-12}$, maximum number of iterations 10^6, and computing the exact Hessian matrix of the Lagrangian at every step, rather than an approximation. The maximum number of iterations was chosen by monitoring the solver solution during its run for the original case of each one of the presented scenarios and setting the minimum number of iterations that ensure that the objective function was minimized to the desired level of accuracy.

The solution of this solver $\boldsymbol{y}_j^i$ is the vector of optimal values of the output variables that satisfy the constraints on the power system that we want the PINN model to predict. Hence, the PINN will try to approximate the behavior of these solutions by using the pairs of input-output data $\mathcal{D} = \left(\boldsymbol{x}_j^i\ \boldsymbol{y}_j^i\right)$

$$\boldsymbol{y}_j^i = \psi(\boldsymbol{x}_j^i) \tag{28}$$

B. PINN TRAINING

We train the PINN using a set of hyperparameters 2 $= (\eta, N_h, \alpha)$. To improve the consistency of the results, the decay parameter ξ was explored as an additional hyperparameter to tune. However, it was observed that the variation between different cross-validation tests was greater than the variation between cases with different ξ values. Since values of ξ around 0.97 consistently produced good results and the focus of this study was on other hyperparameters, it was deemed unnecessary to explore further variations in ξ and fixed to this value.

We include the set of physical equality and inequality constraints in the computation of the loss function terms. The network architecture consists of a single hidden layer with N_h neurons, $L = 2$, and a scaled sigmoid output activation function, which takes into account the output variable's bounds.

$$\phi^{[1]}(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{29}$$

$$\phi^{[2]}(z) = \underline{y} + (\bar{y} - \underline{y}) \cdot \sigma\left(\frac{z}{\bar{y} - \underline{y}}\right) \tag{30}$$

We choose a single hidden layer because it has been shown to provide satisfactory results for a variety of problems, while also being computationally efficient [76], which will help us determine the validation of the proposed methodology.

The input to the network is the vector $\boldsymbol{x}$, and the output is the predicted value $\hat{\boldsymbol{y}}$ of the solution $\boldsymbol{y} = \psi(\boldsymbol{x})$ at that point. Those values are drawn from a dataset $\mathcal{D}$ previously generated, which contains 1000 pairs $(\boldsymbol{x}, \boldsymbol{y})$ for a specific scenario.

We train the network using the Adam optimizer with a learning rate of η and a batch size of 800 pairs randomly chosen to form the training dataset $\mathcal{D}_T$. During training, we evaluate the network performance on a validation set every 100 iterations, and stop training if the validation loss does not improve for 100 consecutive evaluations. We use the mean absolute error loss to compute the empirical loss term, and the violation of the physical constraints is computed using the sum of the constraint violation terms described in the PINN Lagrangian Penalty step.

C. PERFORMANCE EVALUATION

1) REGRESSION USING PINN

Given a set of hyperparameters Θ and a training dataset $\mathcal{D}_V$, we train our PINN for a specific arbitrary scenario and want to evaluate its performance in (1) learning to predict the optimal power flow state and (2) serving as a warm start method for optimization.

Once training is complete, we evaluate the trained PINN on a test set of 200 data samples that were not used during training or validation, by setting the learning rate $\eta = 0$. We are not interested in a singular case minimum error, but rather an overall good performance of our estimator. For this reason, the minimum error metric is not desired, but rather an average of the results from the validation dataset $\mathcal{D}_V$. We compare the predicted values with the true values of the test set along with the physical constraints violation and compute the average of the loss functions $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$ to determine the error of this PINN with $\mathcal{D}_V$ and hyperparameters Θ.

$$\mathcal{L}[\mathcal{D}_V, \Theta] = \frac{1}{|\mathcal{D}_V|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_V} \mathcal{L}_{[\mathcal{D}_V,\Theta]}(\boldsymbol{x}, \boldsymbol{y}) \tag{31}$$

The error can be split into two terms to indicate the result of the validation and testing.

$$\mathcal{L}[\mathcal{D}_V, \Theta] = \mathcal{L}_0[\mathcal{D}_V, \Theta] + \alpha \cdot \mathcal{L}_C[\mathcal{D}_V, \Theta] \tag{32}$$

2) WARM-START OPTIMIZATION

We conducted a series of experiments to evaluate the performance of the proposed warm start optimization approach PINN. The experiments compared the standard initialization approach against the warm start obtained from the PINN model by measuring several metrics.

At each iteration t, we fed the input x_t to both the AC-OPF operator and the PINN's forward regressor. The PINN's forward regressor used the weights, biases, and activation functions to generate an estimated output $\hat{y}$, which served as a warm start-point for the AC-OPF operator.

We then performed a second computation at the AC-OPF operator by setting the voltage magnitudes to $v_i = 1$ and initializing the phase angles to $\delta_i = 0 \; \forall i \in \mathcal{N}$. Additionally, we set the real and reactive power injections p_i^g and q_i^g $\forall i \in \mathcal{G}$ to their corresponding values obtained from Equations (4)-(5). This served as the starting point to reach the objective y_t from the direct input x_t and also enabled us to update the PINN model with the proposed learning rules.

If additional requirements, restrictions, or variable bounds were needed, the PINN was updated accordingly by modifying $\underline{\boldsymbol{y}}$, $\bar{\boldsymbol{y}}$, g, or h at each time step.

We compared the average computational time using the PINN model $\bar{t}_P$ against the standard solution $\bar{t}_S$ using the interior point method for different scenarios. The computational time of the interior point method without the PINN warm start was used as the baseline.

Furthermore, we analyzed the performance of the AC-OPF solver by measuring the success rate r_S and r_P of the optimization method, which indicated the percentage of times the optimization algorithm was able to find a feasible and optimal solution. The success rate was calculated for both the standard initialization approach and the PINN warm start approach.

To assess the effectiveness of the optimization ensemble, which combined the PINN predictions with the interior point method to warm start the optimization process, we measured the difference in the score of the objective function achieved in both cases Δf, as there could be multiple minima.

All these evaluation metrics were carefully chosen to provide a comprehensive and accurate analysis of the performance of the proposed warm start optimization approach. This AC-OPF operation was performed simulating a real-time situation where inputs $\boldsymbol{x}_t$ are received one at a time for each scenario of our power grid system model.

D. HYPERPARAMETER TUNING

To achieve the best performance of our physics-informed neural network (PINN), we need to carefully tune its hyperparameters. In particular, we need to balance the empirical loss (which measures the discrepancy between the predicted and true solution) with the constraint violation (which enforces the physical laws). A higher hyperparameter value α would increase the influence of the constraint violation term, minimizing violations while maintaining a low empirical loss. However, we must avoid selecting an excessively high value of α, which would cause the PINN to lose its generalizability. In such cases, the empirical loss would persistently increase even as the constraint violation diminishes with continued training.

We propose a four-step procedure to tune the hyperparameters of our PINN for a specific scenario, and examine the behavior of the PINN with their variation:

1) PROMISING CONFIGURATIONS

We first perform a random search to explore the hyperparameter search space $\eta \in [10^{-8}, 10] \subset \mathbb{R}$, $N_h \in [1, 100] \subset \mathbb{N}$. The goal is to identify regions that minimize the loss function for the learning rate and the number of hidden neurons while fixing $\alpha = 1$, a reasonable starting point for most PINN models. We use Latin Hypercube Sampling (LHS) with 10 strata to generate hyperparameter values that cover the entire range of values while ensuring a uniform random distribution of samples and use fewer iterations than an exhaustive search with every possible combination of hyperparameters [77]. After generating samples using LHS, we then generated a preprocessed dataset containing the cross-validated average score of 10 random samplings of the data collection of size $N_S = 1000$, with a splitting of 80% for the training set and 20% for the validation set, for every combination.

To identify areas where the loss function values are consistently low and narrow down the search space, we have successfully combined multiple approaches to tackle the problem. Then, we visualized the results of the random search

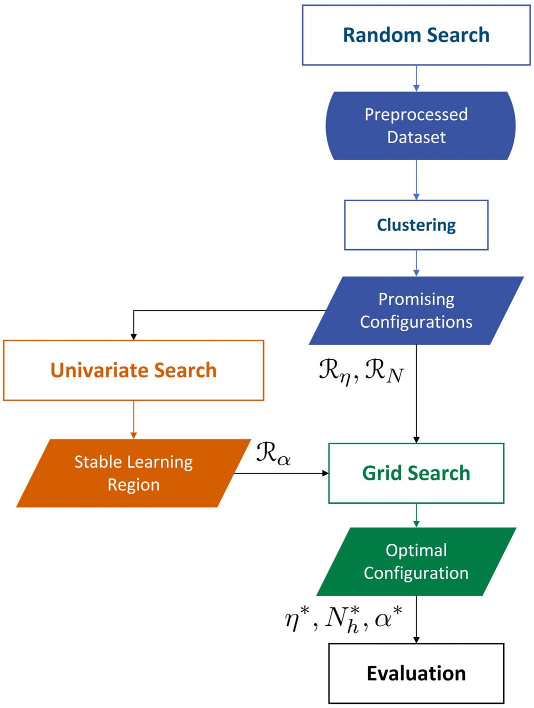

FIGURE 5. Scheme of physics-informed neural network hyperparameters tuning for a particular power systems scenario. Four-step procedure proposed for tuning the hyperparameters of a physics-informed neural network (PINN) is tailored to a specific power systems scenario. It involves exploring the hyperparameter search space using random search and Latin Hypercube Sampling (LHS) to identify promising configurations, followed by a focused grid search to determine stable learning regions. Optimal configurations are then identified by minimizing the combined loss function, comprising both empirical and constraint violation terms (35). Finally, the PINN model is evaluated on a separate validation dataset to ensure generalizability.

using heatmaps, and also looked at the specific combinations with the lowest loss function value to validate that they belong to promising regions of learning rate $\mathcal{R}_\eta \subset \mathbb{R}^+$, and hidden nodes $\mathcal{R}_N \subset \mathbb{N}$. Then, to validate and refine the results, we apply a k-means clustering algorithm with the Silhouette method to group the combinations based on their loss function values, resulting in a set of clusters in the search space with similar loss function values [78]. This is useful to gain insights into this complex dataset that may not be apparent through manual examination. We then use these clusters to identify the region of interest with the lowest average loss function.

2) STABLE LEARNING REGION

Based on the results of the random search, we perform an exhaustive grid search for the penalty factor $\alpha \in [10^{-8}, 10]$, which plays a crucial role in balancing the physical constraints and the empirical loss function in the physics-informed neural network. We focus on the best combination of hyperparameters (η, N_h) identified in the previous step and search locally around it to identify the stable learning region $\mathcal{R}_\alpha \subset \mathbb{R}^+$ of α. To compute the stable learning region, we analyze the intersection of the α regions of robust learning for the empirical error $(\mathcal{R}_\alpha^0)$ and for the constraint violation $(\mathcal{R}_\alpha^\nu)$, as identified in the simulations results.

3) OPTIMAL CONFIGURATION

Given the potential optimal range for α obtained from Univariate Search (Fig. 5), we perform a local grid search in $\mathcal{R} = \mathcal{R}_\eta \times \mathcal{R}_N \times \mathcal{R}_\alpha$ to identify the hyperparameters that minimize the whole loss function, comprising both the empirical and constraint violation terms, with normalized values $L_0[\cdot], L_C[\cdot] \in [0, 1]$.

$$L_0 = \frac{\mathcal{L}_0 - \min\{\mathcal{L}_0\}}{\max\{\mathcal{L}_0\} - \min\{\mathcal{L}_0\}} \tag{33}$$

$$L_C = \frac{\mathcal{L}_C - \min\{\mathcal{L}_C\}}{\max\{\mathcal{L}_C\} - \min\{\mathcal{L}_C\}} \tag{34}$$

Normalizing the terms is a common practice in situations where they have different magnitudes and units, and it ensures that they are all on the same scale, which facilitates comparisons and enables effective optimization and finding the optimal hyperparameters $\Theta^* = \left(\eta^*, N_h^*, \alpha^*\right)$.

$$\Theta^* = \arg\min_{\Theta \in \mathcal{R}} L_0\left[\mathcal{D}_V, \Theta\right] + L_C\left[\mathcal{D}_V, \Theta\right] \tag{35}$$

4) EVALUATION

Finally, we evaluate the PINN model on a separate validation dataset in the Evaluation block (Fig. 5) to ensure that the model generalizes well by computing the error of normalized values $\bar{E}$.

$$\bar{E}\left(\mathcal{D}_V\right) = \min_{\Theta \in \mathcal{R}} L_0\left[\mathcal{D}_V, \Theta\right] + L_C\left[\mathcal{D}_V, \Theta\right] \tag{36}$$

We compute the deviation of the optimal hyperparameters as the relative difference between the score for the optimal hyperparameters against the mean score of all the combinations obtained from equally spaced subsets of the hyperparameter search space. By carefully tuning the hyperparameters of our PINN using this four-step procedure, we intend to achieve with our model the best possible performance while avoiding overfitting and numerical instability.

This approach puts the focus on the applicability of the hyperparameters across a wide range of cases and made the optimization method more robust and adaptable. The resulting optimized hyperparameters are expected to improve the accuracy and performance of the optimization results. In light of the hyperparameters obtained from each step of our scheme (Fig. 5), we comprehensively analyze and assess the impact of various components that can be integrated into the training process of our PINN model compared to a conventional ANN. Furthermore, we validate the performance of the PINN in the context of warm-start optimization.

V. RESULTS

A. REGULARIZATION IMPACT

First, to investigate the impact of the regularization term in the loss function on the performance of the PINN,

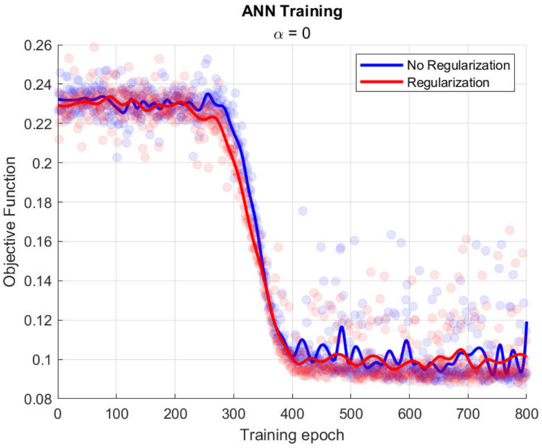

FIGURE 6. Comparison of training processes for the ANN model ($\alpha = 0$) of scenario E without and with L2-regularization with $\gamma = 1$, $\eta = 5 \cdot 10^{-5}$, $N_h = 20$. Line plots were computed using the average of the objective function, shown as a scatter plot, every 10 training steps.

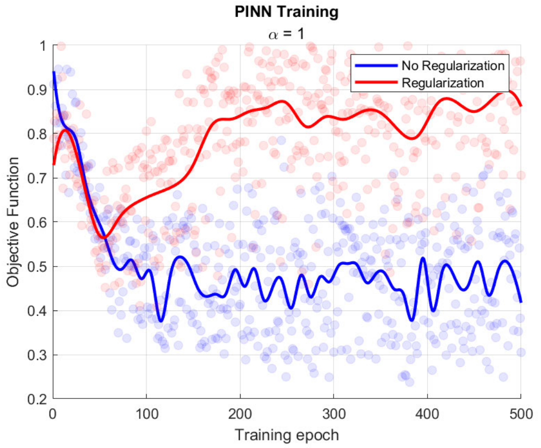

FIGURE 7. Comparison of training processes for the PINN model of scenario A without and with L2-regularization with $\gamma = 1$, $\eta = 10^{-2}$, $N_h = 25$, $\alpha = 1$. Line plots were computed using the average of the objective function, shown as a scatter plot, every 10 training steps.

we experimented with different values of the regularization factor, γ for both L1 and L2. We observed that for ANNs (Fig. 6), a moderate value of γ helped prevent overfitting and improved generalization performance. However, when implementing PINNs (Fig. 7), we found that the regularization term became irrelevant when $\alpha > 0$. This is due to the physics-informed constraints that are built into the loss function, which inherently provide regularization to the network. In fact, we observed that adding a large γ value to the loss function did not significantly affect the network's performance, and sometimes even hindered convergence for the considered scenarios. For this reason, the subsequent analyses have been used without including the regularization term in the loss function to avoid unpredictable instability and unnecessary computations, which is equivalent to setting $\gamma = 0$. Fig. 7 shows the training processes where the regularization term causes instability in the network's convergence.

Moreover, as a qualitative general appreciation, only including the regularization term $\gamma \cdot \mathcal{L}_R$ in the objective function, and not the augmented Lagrangian penalty $\alpha \cdot \mathcal{L}_C$, helped prevent some overfitting in a few cases, but only for hyperparameter values around the optimal ones (Fig. 5) determined for every scenario. In contrast, just including the term $\alpha \cdot \mathcal{L}_C$ is able to capture the regularization effect over a wider range of learning rate values, as seen in Fig. 7.

This reveals that the regularization term in the loss function becomes noise when the physics-informed constraints are incorporated into the loss function; while the empirical results showed that a moderate value of the regularization factor can prevent overfitting and improve generalization performance of artificial neural networks. The simulation results suggest that the constraint violation in the loss function inherently provides regularization to the network, and the regularization term is irrelevant when implementing PINNs.

B. PENALTY FACTOR ANALYSIS

In this study, we investigate the effect of varying the hyperparameter α on the performance of the physics-informed neural network (PINN). We observe that a very low value of α results in low empirical error but poor constraint violation satisfaction and a high $\mathcal{L}_C$ term. Intuitively, increasing α should prioritize constraint violation in the loss function and, after some learning iterations, reduce the $\mathcal{L}_C$ term but with a trade-off of increasing empirical error. However, we find that for some values of α within a middle range $\mathcal{R}_0^{\text{cr}}$, typically around 0.5, the PINN loses generalization, and empirical loss increases significantly after only a few training steps. Beyond a threshold value of α, the network is able to learn and predict optimal values accurately while reducing constraint violations and keeping the empirical loss low.

We analyze this behavior by plotting normalized error terms with respect to several α values and identifying four regions, in order of the parameter α.

1) **Stable Overfitting:** The first region corresponds to low α values with the lowest empirical error but high constraint violation. This suggests that the network is overfitting to the data and not constrained enough by the physical equations, leading to poor generalization and inaccurate predictions.
2) **Critical Balancing:** The second region corresponds to α values near the critical point. During the early stages of training, the network prioritizes fitting the data, thus decreasing the empirical error. However, as training progresses, the network begins to overemphasize the constraints, leading to an increase in the empirical error. This behavior indicates that the network struggles to balance the fit to the data and constraint violations.
3) **Stable Learning:** The stable learning region corresponds to α values that are slightly higher than the lowest values with a low empirical error and low constraint violation. This region indicates that the network produces accurate predictions that are consistent with the physical behavior of the system.

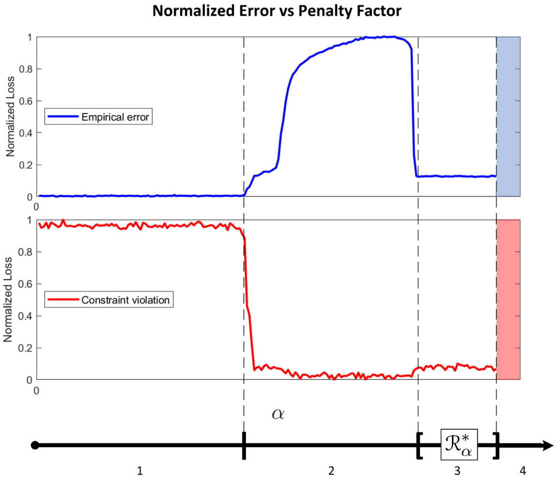

FIGURE 8. Normalized empirical error and constraint violation as a function of penalty factor α for the PINN model learning to predict values for scenario B with $\gamma = 0$, $\eta = 3.5 \cdot 10^{-3}$, and $N_h = 23$. The behavior of the model is analyzed in four different regions along the real line of α: stable overfitting (1), critical balancing (2), stable learning (3), and instability (4). The shaded background in both plots indicates the region of numerical divergence where the model experiences numerical instabilities during training.

4) **Instability:** The unstable region corresponds to high α values with numerical divergence in the network weights and biases' updates; suggesting that the network is unable to learn and experiencing numerical instabilities during training.

We note that the stable learning region presents an adequate trade-off between the conflicting objectives, and determining the optimal trade-off point requires hyperparameter tuning with restricted values of α.

By imposing a substantial penalty factor, the PINN is incentivized to perform a forward regression that restricts the range of predicted values to an approximate feasible space determined by the physical constraints of the model. The effectiveness of this approach is demonstrated in Figure 9, which illustrates that the PINN model exhibits a more precise convergence in the total objective function compared to the ANN model with no Lagrangian penalty ($\alpha = 0$).

Hence, very low values of α result in low empirical error but poor constraint violation satisfaction, while very high values caused numerical divergence in the network weights and biases' updates. The stable learning region presents an adequate trade-off between conflicting objectives, which required hyperparameter tuning with restricted values of α. Moreover, the findings indicate that hyperparameter tuning is essential for choosing the optimal tradeoff point between empirical error and constraint violations when using penalty factors in the PINN.

C. OPTIMAL HYPERPARAMETERS

Using the four-step procedure outlined in Section IV (Fig. 5), we identified the optimal hyperparameters for each scenario and evaluated their performance.

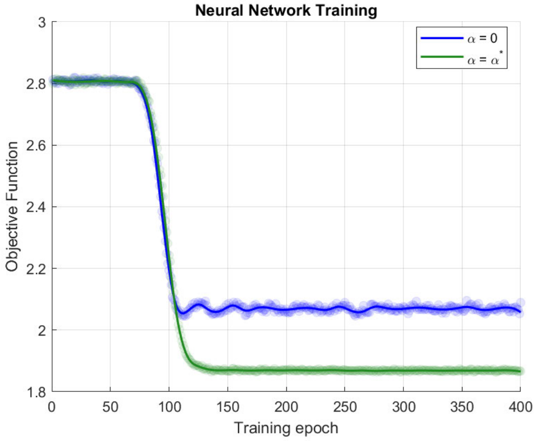

FIGURE 9. Comparison of training processes for null and optimal penalty factor $\alpha^* = 1.08$ for the PINN model of scenario H with $\eta = 3 \cdot 10^{-3}$, and $N_h = 14$, showing the advantage of the use of PINN over ANN.

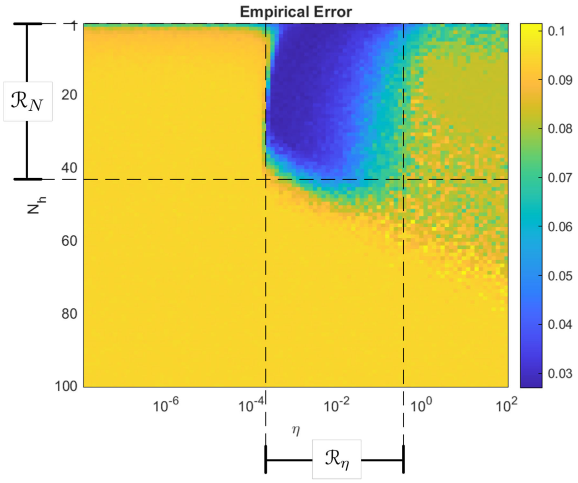

FIGURE 10. Heatmap of loss function values obtained from random search for scenario C. The best combinations of hyperparameters are located in a region of the search space with a learning rate of $0.0025 \leq \eta \leq 0.75$ and number of hidden nodes $1 \leq N_h \leq 42$.

Fig. 10 shows the heatmap of the loss function values obtained from the random search for scenario C, as an example. The heatmap shows that the best combinations of hyperparameters are located in a region of the search space with a learning rate of $0.0025 \leq \eta \leq 0.75$ and number of hidden nodes $1 \leq N_h \leq 42$. As it can be seen in Fig. 11, this region corresponds to the cluster with the lowest average loss function value, which is used to verify our results in the methodology.

Remarkably, we found that the optimal hyperparameters led to a significant reduction of around 85%-90% in the loss function with respect to the average in the hyperparameter space of the fine-tuned PINN model used for predicting AC-OPF with physical constraints. This highlights the effectiveness of our approach in accurately modeling the underlying physics and constraints of the power system, resulting in improved predictive performance.

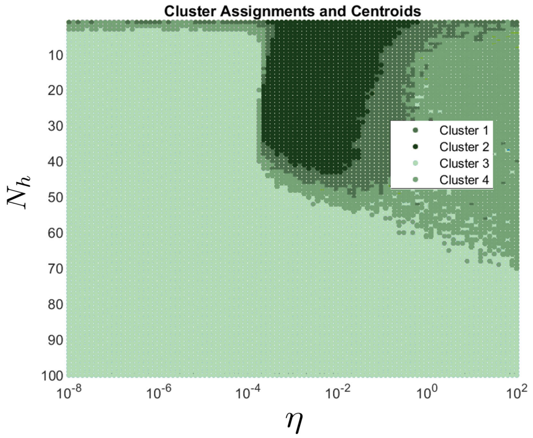

FIGURE 11. Clusters obtained from *k*-means clustering with the Silhouette method for scenario C, and the loss function as a score. The cluster with the lowest average loss function value (Cluster 2, highlighted in darkest green) corresponds to the region of the search space with the best combinations of hyperparameters.

TABLE 2. Optimal PINN hyperparameters that minimize the MAE for several grid topologies' OPF estimations.

Scenario	η^*	α^*	N_h^*	Error	Error reduction
A	0.0090	1.23	33	0.122	89.46%
B	0.0035	1.30	23	0.191	85.79%
C	0.0009	1.01	20	0.193	86.84%
D	0.0004	1.35	1	0.147	88.89%
E	0.0006	1.03	18	0.244	83.40%
F	0.0015	1.49	33	0.082	94.35%
G	0.0160	0.02	35	0.228	84.53%
H	0.0030	1.07	14	0.179	87.70%
I	0.1000	0.0005	1	0.104	92.64%
J	0.0011	1.37	20	0.222	84.73%
K	0.0040	1.16	23	0.094	93.97%
L	0.0155	1.4	35	0.098	93.42%

TABLE 3. Performance evaluation of the proposed warm start optimization approach using PINN for several scenarios. $\Delta\bar{t}$: Average computational time gain. Δr: Success rate gain. $\Delta\bar{f}$: Objective function value gain.

Scenario	$\Delta\bar{t}$ [%]	Δr [%]	$\Delta\bar{f}$ [%]
A	-9.86	-1	-9.6
B	-10.93	0	-0.97
C	-9.42	0	-1.97
D	-21.04	0	-0.25
E	-16.29	+27.61	-0.71
F	-30.22	+52.83	+8.58
G	-20.49	+7.41	-3.13
H	-43.57	+14.00	+0.16
I	-14.37	0	+0.30
J	-15.81	+490.36	-1.45
K	-23.39	+31.58	+1.77
L	-8.64	0	-1.05
Average	$\mathbf{-18.67}$	$\mathbf{51.90}$	$\mathbf{-0.69}$

Our hyperparameter tuning results indicate that the number of neurons in the hidden layer and the learning rate were the most critical hyperparameters for achieving optimal performance. In fact, we found that these hyperparameters had a greater impact on the model's performance than varying the regularization coefficient α, even when α was within the stable learning region and the hyperparameters were within their respective regions of promising configurations. Regularized flexible activation functions [79] were also explored; however they did not result in any performance gain, but rather an increase in the number of hyperparameters to tune, compromising the reduced complexity of the method. This highlights the importance of careful selection and fine-tuning of the number of neurons and learning rate for achieving optimal performance in the PINN model.

D. WARM-START OPTIMIZATION

We compare the optimization results obtained with and without the use of PINN. Specifically, we compute the percentage improvement in optimization performance achieved by using the PINN as a warm start compared to starting from scratch. The higher the improvement ratio of the metrics, the more significant the contribution of the PINN warm start.

The results of the simulations show that the Physics-Informed Neural Network (PINN) approach achieved comparable results to the benchmark solution, while also offering an average speedup of 18.7%. In fact, for case H, the PINN achieved a maximum speedup of 43.57%, indicating that it can reduce the computation time of AC-OPF by almost half compared to the robust traditional solver. Notably, the optimization algorithm's success rate was consistently increased or maintained, with an average increase of 51.9%. Only in scenario A's simulation did the success rate decrease slightly from 100% to 99%.Furthermore, the proposed method showcased exceptional effectiveness by significantly enhancing the success rate of the AC-OPF computation in case J, achieving an impressive improvement of 490.36%. Additionally, in 8 out of the 12 cases, the objective function value was slightly decreased at the end of the optimization, while it was slightly increased in the remaining 4.

The simulations conducted in this study demonstrate that the Physics-Informed Neural Network (PINN) approach yields comparable results to the benchmark solution, while providing an average speedup of 18.7%. Remarkably, for case H, the PINN achieves a substantial maximum speedup of 43.57%, effectively reducing the computation time of AC-OPF by nearly half compared to the robust traditional solver. It is worth noting that the optimization algorithm's success rate consistently increased or remained stable, exhibiting an average increase of 51.9%. The only exception was observed in scenario A, where the success rate experienced a slight decrease from 100% to 99%.

Furthermore, the proposed PINN method demonstrates exceptional effectiveness in enhancing the success rate of AC-OPF computation, as evident in case J, where it achieves an impressive improvement of 490.36%. Moreover, among the 12 cases analyzed, the objective function value slightly decreased at the end of the optimization in 8 cases. This outcome suggests that utilizing PINN predictions as a warm start allows the algorithm to converge towards improved local minima and, in some cases, even global minima.

It is important to consider alternative techniques such as relaxations, like Second-Order Cone Programming (SOCP), which offer a faster resolution of the problem [80]. Simulations indicate that SOCP provides an approximate solution to the original AC-OPF problem using only around 10% of the iterations required by the original AC-OPF formulation, resulting in a significant computational gain. However, utilizing the SOCP solution as a warm start does not offer any notable changes in the computational time or the success rate, compared to not using a warm start at all. This suggests that relaxations for optimal power flow are highly valuable for obtaining approximate solutions with considerable computational time savings. Nevertheless, they prove to be less effective than the PINN approach for warm starts to achieve precise AC-OPF solutions.

The evaluation presents evidence that the proposed PINN method is a promising approach for warm start optimization in power systems, as it is able to balance computational efficiency and solution accuracy in most cases. The optimization ensemble can further improve the computational efficiency and more remarkable the convergence of classical optimization AC-OPF algorithms, without sacrificing the accuracy of the solution.

VI. CONCLUSION

The study investigated the impact of several factors on the performance of physics-informed neural network for the estimation of AC-OPF results.

Existing studies favoring warm starting often rely on the previous operating point or an approximate solution as the initialization, assuming its adequacy in capturing the problem's characteristics. However, this approach overlooks the fact that each new AC-OPF case introduces new inputs that slightly modify the problem. Consequently, warm-starting with the previous operating point fails to align with the new problem, and multiple iterations are needed to converge to a well-centered solution. In contrast, the data yielded by this study provides evidence that warm-start initialization with PINN predictions is an alternative approach that not only improves the success rate but also enhances computational speed when solving the problem in an AC-OPF exact formulation. Therefore, it seems fair to assert that PINN exhibits a superior performance in finding warm-start points for exact solutions, while relaxation techniques like SOCP primarily excel in achieving faster approximate solutions.

Additionally, incorporating physics-informed constraints information can improve the generalization performance. Our results suggest that the PINN approach can significantly improve the efficiency and success rate of the AC-OPF optimization algorithm in power systems, compared to starting from scratch. The PINN method can reduce the computational time by almost half, while maintaining or increasing the success rate and achieving results comparable to the benchmark solution. The reduction in the value of the objective function in most cases indicates the effectiveness of the proposed approach. Overall, the PINN method can enhance the scalability and computational efficiency of optimization algorithms in power systems, leading to cost savings and improved system performance.

Future research can explore the potential of the PINN method for other power system optimization problems and investigate the benefits of combining it with other machine learning techniques. Firstly, it should be noted that in this study we performed the optimal design of a PINN with a shallow feedforward neural network architecture and a single hidden layer to optimize their advantages on grid operation, setting the groundwork for studying multiple hidden layers and larger scale problems, and quantifying their improvements in future analysis.

Our findings demonstrate a promising method to use PINNs to initialize OPF, which can be applied not only in offline simulations but also in real-time systems where there are frequent changes. This study can contribute to the development of more efficient and accurate optimization algorithms for power systems, thereby improving the reliability and sustainability of the grid.

REFERENCES

[1] J. Shair, H. Li, J. Hu, and X. Xie, "Power system stability issues, classifications and research prospects in the context of high-penetration of renewables and power electronics," *Renew. Sustain. Energy Rev.*, vol. 145, Jul. 2021, Art. no. 111111.

[2] M. Liu and L. Ochoa. (Nov. 2018). *Using OPF for Smart Grids: From Concept to Reality*. Accessed: Feb. 9, 2023.

[3] M. Fan, Z. Zhang, and C. Wang, "New algorithms related to power flow," in *Mathematical Models and Algorithms for Power System Optimization*. New York, NY, USA: Academic, 2019, ch. 4, pp. 81–119.

[4] R. J. Vanderbei and D. F. Shanno, "An interior-point algorithm for nonconvex nonlinear programming," *Comput. Optim. Appl.*, vol. 13, no. 1, pp. 231–252, 1999.

[5] D. Bertsimas, O. Nohadani, and K. M. Teo, "Nonconvex robust optimization for problems with constraints," *Informs J. Comput.*, vol. 22, no. 1, pp. 44–58, Feb. 2010.

[6] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[8] S. J. W. J. Nocedal, *Numerical Optimization*. Springer, 1999.

[9] Z. Yao, "Efficient second-order methods for non-convex optimization and machine learning," Ph.D. thesis, UC Berkeley, Berkeley, CA, USA, 2021.

[10] R. E. Uhrig, "Introduction to artificial neural networks," in *Proc. 21st Annu. Conf. IEEE Ind. Electron. (IECON)*, vol. 1, Sep. 1995, pp. 33–37.

[11] R. Setiono, W. Kheng Leow, and J. M. Zurada, "Extraction of rules from artificial neural networks for nonlinear regression," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 564–577, May 2002.

[12] M. R. AlRashidi and M. E. El-Hawary, "Applications of computational intelligence techniques for solving the revived optimal power flow problem," *Electr. Power Syst. Res.*, vol. 79, no. 4, pp. 694–702, Apr. 2009.

[13] F. Hasan, A. Kargarian, and A. Mohammadi, "A survey on applications of machine learning for optimal power flow," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, Feb. 2020, pp. 1–6.

[14] B. Donon, R. Clément, B. Donnot, A. Marot, I. Guyon, and M. Schoenauer, "Neural networks for power flow: Graph neural solver," *Electr. Power Syst. Res.*, vol. 189, Dec. 2020, Art. no. 106547.

[15] F. Li, "Successful applications and future challenges of machine learning for power systems: A summary of recent activities by the IEEE WG on machine learning for power systems [what's popular]," *IEEE Electrific. Mag.*, vol. 10, no. 4, pp. 90–96, Dec. 2022.

[16] L. Zhang, Y. Chen, and B. Zhang, "A convex neural network solver for DCOPF with generalization guarantees," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 2, pp. 719–730, Jun. 2022.

[17] F. Diehl, "Warm-starting AC optimal power flow with graph neural networks," in *Proc. NeurIPS Workshop Tackling Climate Change Mach. Learn.*, 2019.

[18] K. Baker, "Learning warm-start points for AC optimal power flow," in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Oct. 2019, pp. 1–6.

[19] Y. Kim and M. Anitescu, "A real-time optimization with warm-start of multiperiod AC optimal power flows," *Electr. Power Syst. Res.*, vol. 189, Dec. 2020, Art. no. 106721.

[20] A. Shahzad, E. C. Kerrigan, and G. A. Constantinides, "A warm-start interior-point method for predictive control," in *Proc. UKACC Int. Conf. Control*, Sep. 2010, pp. 1–6.

[21] Z. Zhang, Q. Zhao, and F.-A. Dai, "A warm-start strategy in interior point methods for shrinking horizon model predictive control with variable discretization step," *IEEE Trans. Autom. Control*, vol. 68, no. 6, pp. 3830–3837, 2023.

[22] M. Jamei, L. Mones, A. Robson, L. White, J. Requeima, and C. Ududec, "Meta-optimization of optimal power flow," in *Proc. ICML*, 2019.

[23] N. Guha, Z. Wang, M. Wytock, and A. Majumdar, "Machine learning for AC optimal power flow," 2019, *arXiv:1910.08842*.

[24] A. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," 2019, *arXiv:1910.01213*.

[25] M. Soltanolkotabi, A. Javanmard, and J. D. Lee, "Theoretical insights into the optimization landscape of over-parameterized shallow neural networks," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 742–769, Feb. 2019.

[26] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Aug. 2020, pp. 1–5.

[27] A. S. Zamzam and N. D. Sidiropoulos, "Physics-aware neural networks for distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4347–4356, Nov. 2020.

[28] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2082–2092, May 2021.

[29] R. Nellikkath and S. Chatzivasileiadis, "Physics-informed neural networks for AC optimal power flow," *Electr. Power Syst. Res.*, vol. 212, Nov. 2022, Art. no. 108412.

[30] Q.-H. Ngo, B. L. H. Nguyen, T. V. Vu, J. Zhang, and T. Ngo, "Physics-informed graphical neural network for power system state estimation," *Appl. Energy*, vol. 358, Mar. 2024, Art. no. 122602.

[31] B.-G. Risi, F. Riganti-Fulginei, and A. Laudani, "Modern techniques for the optimal power flow problem: State of the art," *Energies*, vol. 15, no. 17, p. 6387, Sep. 2022.

[32] O. Kundacina, M. Cosovic, and D. Vukobratovic, "State estimation in electric power systems leveraging graph neural networks," in *Proc. 17th Int. Conf. Probabilistic Methods Appl. Power Syst. (PMAPS)*, Jun. 2022, pp. 1–6.

[33] B. Huang and J. Wang, "Applications of physics-informed neural networks in power systems—A review," *IEEE Trans. Power Syst.*, vol. 38, no. 1, pp. 572–588, Jan. 2023.

[34] L. Zhang, G. Wang, and G. B. Giannakis, "Real-time power system state estimation and forecasting via deep unrolled neural networks," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4069–4077, Aug. 2019.

[35] X. Lei, Z. Yang, J. Yu, J. Zhao, Q. Gao, and H. Yu, "Data-driven optimal power flow: A physics-informed machine learning approach," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 346–354, Jan. 2021.

[36] M. Chatzos, T. W. K. Mak, and P. V. Hentenryck, "Spatial network decomposition for fast and scalable AC-OPF learning," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2601–2612, Jul. 2022.

[37] R. R. Shoults and D. T. Sun, "Optimal power flow based upon P-Q decomposition," *IEEE Power Eng. Rev.*, vol. PER-2, no. 2, pp. 25–26, Feb. 1982.

[38] F. Zhou, J. Anderson, and S. H. Low, "The optimal power flow operator: Theory and computation," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 2, pp. 1010–1022, Jun. 2021.

[39] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Trans. Power App. Syst.*, vol. PAS-86, no. 11, pp. 1449–1460, Nov. 1967.

[40] H. Dommel and W. Tinney, "Optimal power flow solutions," *IEEE Trans. Power App. Syst.*, vol. PAS-87, no. 10, pp. 1866–1876, Oct. 1968.

[41] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, "Optimal power flow by Newton approach," *IEEE Trans. Power App. Syst.*, vol. PAS-103, no. 10, pp. 2864–2880, Oct. 1984.

[42] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, and S. Rajan, "Comparison of feed-forward neural network training algorithms for oscillometric blood pressure estimation," in *Proc. 4th Int. Workshop Soft Comput. Appl.*, Jul. 2010, pp. 119–123.

[43] W. Hu, Z. Li, and D. Yu, "Simple and effective regularization methods for training on noisily labeled data with generalization guarantee," 2020, *arXiv:1905.11368*.

[44] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci.*, vol. 6, no. 12, pp. 310–316, 2017.

[45] G. Alcantara, "Empirical analysis of non-linear activation functions for deep neural networks in classification tasks," 2017, *arXiv:1710.11272*.

[46] B. L. Kalman and S. C. Kwasny, "Why tanh: Choosing a sigmoidal function," in *Proc. Int. Joint Conf. Neural Netw.*, 1992, pp. 578–581.

[47] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.

[48] M. A. Mercioni and S. Holban, "P-Swish: Activation function with learnable parameters based on swish activation function in deep learning," in *Proc. Int. Symp. Electron. Telecommun. (ISETC)*, Nov. 2020, pp. 1–4.

[49] G. C. Tripathi, M. Rawat, and K. Rawat, "Swish activation based deep neural network predistorter for RF-PA," in *Proc. IEEE Region 10 Conf. (TENCON)*, Oct. 2019, pp. 1239–1242.

[50] A. D. Rasamoelina, F. Adjailia, and P. Sincák, "A review of activation function for artificial neural network," in *Proc. IEEE 18th World Symp. Appl. Mach. Intell. Informat. (SAMI)*, Jan. 2020, pp. 281–286.

[51] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 97–116, Apr. 2017.

[52] T. Jin, S. Liu, R. C. C. Flesch, and W. Su, "A method for the identification of low frequency oscillation modes in power systems subjected to noise," *Appl. Energy*, vol. 206, pp. 1379–1392, Nov. 2017.

[53] J. Zhang, C. Y. Chung, and L. Guan, "Noise effect and noise-assisted ensemble regression in power system online sensitivity identification," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2302–2310, Oct. 2017.

[54] X. Wang, Y. Hua, E. Kodirov, D. A. Clifton, and N. M. Robertson, "IMAE for noise-robust learning: Mean absolute error does not treat examples equally and gradient Magnitude's variance matters," 2019, *arXiv:1903.12141*.

[55] F. Li, J. M. Zurada, Y. Liu, and W. Wu, "Input layer regularization of multilayer feedforward neural networks," *IEEE Access*, vol. 5, pp. 10979–10985, 2017.

[56] T. Sun and C.-H. Zhang, "Sparse matrix inversion with scaled Lasso," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 3385–3418, 2013.

[57] O. Demir-Kavuk, M. Kamada, T. Akutsu, and E.-W. Knapp, "Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features," *BMC Bioinf.*, vol. 12, no. 1, pp. 1–10, Dec. 2011.

[58] M. Dozmorov, *Biostatistics: Fundamentals of Deep Learning*, 2021.

[59] M. R. Hestenes, "Multiplier and gradient methods," *J. Optim. Theory Appl.*, vol. 4, no. 5, pp. 303–320, Nov. 1969.

[60] D. Fontaine, L. Michel, and P. Van Hentenryck, "Constraint-based Lagrangian relaxation," in *Principles and Practice of Constraint Programming*, B. O'Sullivan, Ed. Cham, Switzerland: Springer, 2014, pp. 324–339.

[61] Z. Yan and Y. Xu, "Real-time optimal power flow: A Lagrangian based deep reinforcement learning approach," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 3270–3273, Jul. 2020.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[63] F. Li and R. Bo, "Small test systems for power system economic studies," in *Proc. IEEE PES Gen. Meeting*, Jul. 2010, pp. 1–4.

[64] J. H. Chow and J. J. Sanchez-Gasca, *Power System Modeling, Computation, and Control*. Hoboken, NJ, USA: Wiley, 2020.

[65] *14 Bus Power Flow Test Case*, Univ. Washington, Washington, DC, USA, 2014.

[66] H. Bouchekara, "Comprehensive review of radial distribution test systems," *TechRxiv*, Jan. 2020.

[67] *30 Bus Power Flow Test Case*, Univ. Washington, Washington, DC, USA, 2014.

[68] G. W. Bills, *On-line Stability Analysis Study*, document RP 90-1, 1970.

[69] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 725–734, Jan. 1989.

[70] D. Das, "Reconfiguration of distribution system using fuzzy multi-objective approach," *Int. J. Electr. Power Energy Syst.*, vol. 28, no. 5, pp. 331–338, Jun. 2006.

[71] S. M. Myint and S. W. Naing, "Network reconfiguration for loss reduction and voltage stability improvement of 74-bus radial distribution system using particle swarm optimization algorithm," *Int. J. Electr., Electron. Data Commun.*, vol. 3, no. 6, pp. 32–38, Jun. 2015.

[72] D. F. Pires, C. H. Antunes, and A. G. Martins, "NSGA-II with local search for a multi-objective reactive power compensation problem," *Int. J. Electr. Power Energy Syst.*, vol. 43, no. 1, pp. 313–324, Dec. 2012.

[73] UW Archive. (1993). *Power Flow Data for IEEE 118 Bus Test Case*. Accessed: Oct. 15, 2014.

[74] H. M. Khodr, F. G. Olsina, P. M. D. O.-D. Jesus, and J. M. Yusta, "Maximum savings approach for location and sizing of capacitors in distribution systems," *Electr. Power Syst. Res.*, vol. 78, no. 7, pp. 1192–1203, Jul. 2008.

[75] L. T. Biegler, H. Pirnay, and R. López-Negrete. (2011). *sIPOPT Reference Manual*.

[76] M. Raissi, P. Perdikaris, and 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.*, vol. 378, pp. 686–707, Feb. 2019.

[77] M. D. Shields and J. Zhang, "The generalization of Latin hypercube sampling," *Rel. Eng. Syst. Saf.*, vol. 148, pp. 96–108, Apr. 2016.

[78] F. Wang, H.-H. Franco-Penya, J. D. Kelleher, J. Pugh, and R. Ross, "An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity," in *Proc. 13th Int. Conf. Mach. Learn. Data Mining Pattern Recognit.*, New York, NY, USA. Springer, 2017, pp. 291–305.

[79] R. Jie, J. Gao, A. Vasnev, and M.-N. Tran, "Regularized flexible activation function combination for deep neural networks," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 2001–2008.

[80] A. F. Soofi, S. D. Manshadi, G. Liu, and R. Dai, "A SOCP relaxation for cycle constraints in the optimal power flow problem," *IEEE Trans. Smart Grid*, vol. 12, no. 2, pp. 1663–1673, Mar. 2021.

ÀLEX TUDORAS-MIRAVET received the B.S. degrees in mathematics and in engineering physics from the Universitat Politècnica de Catalunya, Barcelona, Catalonia, Spain, in 2022. He is currently pursuing the Ph.D. degree in neuroscience with the University of California at San Francisco, San Francisco, CA, USA.

In 2019, he was a Project Engineer with the Space Division, GTD Systems and Software Engineering. In 2021, he was a Research Intern with Nostrum Biodiscovery. From 2021 to 2022, he was a Research Intern with the Center for Neural Science, New York University, New York City, NY, USA. From 2022 to 2023, he was a Junior Engineer with the Centre d'Innovació Tecnològica en Convertidors Estàtics i Accionaments, Electrical Engineering Department, Universitat Politècnica de Catalunya.

ESTEBAN GONZÁLEZ-IAKL received the B.S. degree in industrial engineering with specialization in electrical engineering and the Diploma degree in sustainable energies from the Pontifical Catholic University of Chile (PUC), in 2009 and 2016, respectively, and the M.S. degree in energy engineering from the School of Industrial Engineering, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2020. He is currently pursuing the Ph.D. degree as a member of the research institution CITCEA, UPC, specializing in the optimization of electrical power systems. Until 2018, he dedicated nearly a decade to promote the development of renewable energies in Chile, working as a Project Engineer with Aaktei Energía SpA.

ORIOL GOMIS-BELLMUNT (Fellow, IEEE) received the degree in industrial engineering from the School of Industrial Engineering of Barcelona (ETSEIB), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2001, and the Ph.D. degree in electrical engineering from UPC, in 2007. In 1999, he joined Engitrol S.L., where he was a Project Engineer in the automation and control industry. Since 2004, he has been with the Electrical Engineering Department, UPC, where he is currently a Professor and participates in the CITCEA-UPC Research Group. Since 2020, he has been an ICREA Academia Researcher. In 2022, he co-founded the start-up eRoots Analytics focused on the analysis of modern power systems. His research interests include the fields linked with power electronics, power systems, and renewable energy integration in power systems.

EDUARDO PRIETO-ARAUJO (Senior Member, IEEE) received the degree in industrial engineering from the School of Industrial Engineering of Barcelona (ETSEIB), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2011, and the Ph.D. degree in electrical engineering from UPC, in 2016. He joined the CITCEA-UPC Research Group, in 2010. In 2021, he was a Visiting Professor with the Automatic Control Laboratory, ETH Zürich. He is currently a Serra Húnter Associate Professor with the Electrical Engineering Department, UPC. His research interests include renewable generation systems, control of power converters for HVDC applications, interaction analysis between converters, and power electronics dominated power systems and data-driven tools for power systems.