This article has been accepted for publication in IEEE Transactions on Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIA.2024.3514992

Newton-Raphson AC Power Flow Convergence based on Deep Learning Initialization and Homotopy Continuation

Samuel N. Okhuegbe, Member, IEEE, Adedasola A. Ademola, Member, IEEE, and Yilu Liu, Fellow, IEEE

***Abstract*—Power flow forms the basis of many power system studies. With the increased penetration of renewable energy, grid planners tend to perform multiple power flow simulations under various operating conditions and not just selected snapshots at peak or light load conditions. Getting a converged AC power flow (ACPF) case remains a significant challenge for grid planners especially in large power grid networks. This paper proposes a two-stage approach to improve Newton-Raphson ACPF convergence and was applied to a 6102 bus Electric Reliability Council of Texas (ERCOT) system. The first stage utilizes a deep learning-based initializer with data re-training. Here a deep neural network (DNN) initializer is developed to provide better initial voltage magnitude and angle guesses to aid in power flow convergence. This is because Newton-Raphson ACPF is quite sensitive to the initial conditions and bad initialization could lead to divergence. The DNN initializer includes a data re-training framework that improves the initializer's performance when faced with limited training data. The DNN initializer successfully solved 3,285 cases out of 3,899 non-converging dispatch and performed better than random forest and DC power flow initialization methods. ACPF cases not solved in this first stage are then passed through a hot-starting algorithm based on homotopy continuation with switched shunt control. The hot-starting algorithm successfully converged 416 cases out of the remaining 614 non-converging ACPF dispatch. The combined two-stage approach achieved a 94.9% success rate, by converging a total of 3,701 cases out of the initial 3,899 unsolved cases.**

***Index Terms*—Deep Learning, Homotopy Continuation, Neural Network, Newton-Raphson, Power flow convergence, Power flow initialization.**

I. Introduction

SOLVING the AC power flow (ACPF) problem is of utmost importance to power system planners as it provides insights into the steady-state operating condition of the grid. It involves calculating the voltage magnitude and angles at each bus. From these solutions, the system losses, thermal and voltage violations, and line loading condition can be evaluated [1], [2]. Power flow also forms the basis for performing many power system studies such as expansion planning, interconnection studies and contingency analysis [3]. From existing literature, several power flow solution methods have been developed which includes non-iterative methods such as holomorphic embedding, DC power flow (DCPF), and iterative methods such as Gauss-Seidel, Newton-Raphson, and Fast Decoupled power flow [4][14]. Other methods include continuation power flow developed in [14] and the Iwamoto power flow.

Among the various solution techniques available, the Newton-Raphson method is the most commonly used in industry because it exhibits significant robustness and quadratic convergence compared to other methods [5]. As such, it is incorporated into many industrial simulation software programs, such as PSS/E and Powerworld. Due to the nonlinear nature of the power flow equations, achieving a converged ACPF case can be quite difficult, especially in large power grids. In fact, under certain operating conditions, a power flow case may be unsolvable [7]. This presents a peculiar challenge for researchers on how to improve ACPF convergence. The convergence of Newton-Raphson ACPF is affected by factors such as the initial guess of the voltage magnitude and angle, generator and load dispatch, reactive power, and the control devices present in the ACPF case. Addressing these factors could improve convergence. Given the niche nature of ACPF convergence, both foundational and recent advancements in this specialized field will be reviewed.

Because the Newton-Raphson method is sensitive to the initial conditions [1], better initializations tend to aid in convergence. Based on this fact, researchers in [15] solved non-converging ACPF models using DCPF initialization. In this approach, the power flow model is first solved using DCPF, and the voltage angle results obtained from the DCPF are then used as the starting point for the Newton-Raphson ACPF. Although this technique is regularly used in industry, a major limitation is that DCPF does not consider voltage magnitude, losses, or reactive power in the model. Initialization using only voltage angles obtained from DCPF tends to reduce the success rate of converging previously non-converging ACPF cases, as observed in [8]. To further address the initial guess problem, researchers in [13] introduced an iterative analytical approach that starts with a random initial guess for the voltage magnitude and angles. After the first iteration, the Jacobian matrix is analyzed, and a convergence parameter

This work is supported the National Science Foundation under award number 2243204 and the Center for Ultra-Wide-Area Resilient Electric Network. (Corresponding author, S.N Okhuegbe).

S.N Okhuegbe is with the Bredesen Center for Interdisciplinary Research and Graduate Education, and the Center for Ultra-Wide-Area Resilient Electric Network, University of Tennessee, Knoxville, USA. (email: sokhuegb@vols.utk.edu)

A.A Ademola is with Dominion Energy Virginia. He was with Bredesen Center for Interdisciplinary Research and Graduate Education, and the Center for Ultra-Wide-Area Resilient Electric Network, University of Tennessee, Knoxville, USA.

Y. Liu is with the Oak Ridge National Laboratory, Bredesen Center for Interdisciplinary Research and Graduate Education, and the Center for Ultra-Wide-Area Resilient Electric Network, University of Tennessee, Knoxville, USA.

This article has been accepted for publication in IEEE Transactions on Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIA.2024.3514992

ρ is computed based on the Jacobian. If ρ is greater than a particular threshold, the initial guess is updated using a developed affine matrix. However, the effectiveness of this approach was not tested on relatively large-scale ACPF cases under varying operating conditions.

Apart from addressing initialization, researchers regularly apply various forms of homotopy continuation to converge ACPF cases. The general concept of homotopy involves replacing an original difficult dispatch case with an easily solvable dispatch case. The parameters in this easy-to-solve case are then slowly changed and solved until they match the original dispatch case. Researchers in [11] developed a methodology known as ”Power Flow Homotopy” that slowly converts an easy DC into ACPF and tested it on a relatively small 300-bus ACPF model. Recent applications of homotopy continuation can be observed in [9] and [10]. In [9], an incremental hot-starting algorithm based on homotopy continuation was developed and applied to a reduced 243-bus Western Electricity Coordinating Council (WECC) system. The developed algorithm was able to successfully solve 77.8% of the created AC dispatch cases. However, the effectiveness of this algorithm was not tested on large-scale ACPF cases regularly encountered in industry. Researchers in [10] published a recent study that applies homotopy continuation to a large-scale ACPF case. The homotopy algorithm in [10] incorporates fictitious shunt admittances to solve the ACPF model. Additionally, the algorithm was not tested under varying operating conditions. From the literature on homotopy, it can be observed that the role of control devices, such as switched shunts, has not been incorporated into the various homotopy-based algorithms currently available.

In addition to addressing initialization and the application of homotopy, recent studies have solved non-converging ACPF cases by applying a reactive power planning methodology developed by researchers in [7]. This technique involves using fictitious generators with unlimited reactive power. Researchers at the National Renewable Energy Laboratory (NREL) [4] recently applied this methodology to a reduced 243-bus WECC power flow model. In this approach, numerous fictitious generators are added to the ACPF case to increase the solvability region. These fictitious generators provide zero active power and unlimited reactive power. The assumption is that the initial ACPF cases were not solved due to insufficient reactive power; therefore, adding generators would provide the necessary reactive support. Once the case is solved, these fictitious generators are gradually removed. A limitation of this approach is the lack of a standardized method for determining the number of fictitious generators to add to an ACPF case, as well as for selecting the buses to which these fictitious generators should be connected. Adding too many fictitious generators could cause the case to diverge, so this method requires some heuristic based on prior knowledge of the specific ACPF case being solved.

Another method for solving non-converging ACPF cases was recently published by researchers at the Pacific Northwest National Laboratory [6]. In this method, ACPF cases are solved by gradually reducing the load in the system until the case finally converges. This approach assumes that the loads in a dispatch are variable and is not suitable for scenarios where the load dispatch is fixed by external factors, such as a load forecast [9]. Shedding load presents a limitation to this approach. Practically, it is easier to control generation than to control load, and load shedding is only done under extremely necessary conditions.

From the reviewed literature, most solution techniques for converging ACPF cases in practical large grid systems include reducing system load, DCPF initialization, homotopy continuation, and reactive power planning. However, researchers in [8] introduced a new method based on artificial intelligence for solving non-converging Newton-Raphson ACPF cases. These non-converging cases could not be solved from a flat start or initialization with a reference voltage solution. In this method, a random forest (RF) machine learning initializer was used to predict the initial voltage magnitude and angle guesses for a 6102-bus Electric Reliability Council of Texas (ERCOT) system. The algorithm achieved a 54% success rate by converging 2106 cases out of 3,899 non-converging cases. Although machine learning has been regularly used to predict power flow solutions for solvable cases where the actual solutions are already known, such as those in [16], which used a convolutional neural network (CNN); [17], which applied physics-informed stacked extreme machine learning; [18], which applied physics-informed deep learning; and [19], which used RF. The algorithm proposed by researchers in [8] presents a new approach for solving non-converging ACPF cases for which the solutions are not known.

This paper aims to address the Newton-Raphson ACPF convergence problem by improving on the methodology developed in [8]. While the work in [8] used only a RF initializer, we propose a two-stage approach that combines deep learning power flow initialization with homotopy continuation using a hot-starting algorithm. The deep learning initialization is based on deep neural networks (DNN) and incorporates a data-retraining framework that enhances the initializer's performance when faced with limited training data. The DNN initializer provides better initial guesses for the voltage magnitude and angles, aiding Newton-Raphson ACPF convergence. ACPF cases not solved by the DNN initializer are then processed through a hot-starting algorithm based on homotopy continuation. This hot-starting method incorporates switched shunt control to further improve convergence.

The main contribution of this work compared with literature can be itemized as follows:

- A proposed two-stage method that avoids the addition of fictitious generators and load shedding. It is also applied to a realistic large scale ACPF case under multiple operating conditions.
- A deep learning initializer based on neural networks to provide better initial voltage magnitude and angle guesses for unfamiliar ACPF cases when compared with other initialization methods such as flat start,RF initialization and DCPF initialization.
- A data retraining framework that improves the performance of the deep-learning initializer which is useful in cases with limited training data.
- Monitoring and adjustment of oscillating switched shunts.

The proposed two-stage methodology is applied to the same dispatch data used in [8] and shows significant improvement in results. This work was developed using PSS/E and Python. The rest of the paper is structured as follows. Section II and III discuss the power flow problem formulation and the data generation respectively. Section IV presents the methodology and technical approach of the proposed method. While Section V and VI present the results and conclusion respectively.

II. The Power Flow Problem Formulation

The power flow problem involves solving a set of nonlinear equations, as shown in (1) and (2) [20]. The Newton-Raphson solution method is a recursive iterative method that solves the power flow equations until the active and reactive mismatch tolerance is met. This mismatch tolerance is the stopping criterion and signifies that the solution has converged. Although the solutions to (1) and (2) are the voltage phasors at each bus, an initial guess close to the actual solutions (voltage phasors) is needed for the Newton-Raphson method to solve and converge efficiently. From the voltage phasor solutions (voltage magnitude and angles), other parameters can be calculated, such as the power flowing through each line and system losses. The nonlinear algebraic power flow equations simply map the bus active and reactive power injections to the voltage phasor at each bus [8].

$$P_i^{inj} = V_i \sum_{k=1}^{N} V_k (G_{ik} Cos\Theta_{ik} + B_{ik} Sin\Theta_{ik} \quad (1)$$

$$Q_i^{inj} = V_i \sum_{k=1}^{N} V_k (G_{ik} Sin\Theta_{ik} - B_{ik} Cos\Theta_{ik} \quad (2)$$

From (1) and (2), P_i^{inj} and Q_i^{inj} represent the net active and reactive power injection at bus i. The Y_{bus} admittance matrix has a real part G_{ik} and an imaginary part B_{ik}. The bus angle difference between bus i and k is given as Θ_{ik}. For N buses in the system V_i and V_k represent the bus voltage magnitude at bus i and k. The inverse of the Jacobian matrix, $J(\Theta, V)$, expressed in (3) is basically the derivatives of the active power, P, and reactive power, Q, with respect to the angle, Θ , and voltage magnitude, V, and it is computed at each iteration.

$$J(\Theta, V) = \begin{bmatrix} \frac{\partial P}{\partial \Theta} & \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \Theta} & \frac{\partial Q}{\partial V} \end{bmatrix} \quad (3)$$

Assuming an initial guess of V and Θ at iteration t = 0, the voltage magnitude and angle at the next iteration t+1 is given by (4)

$$\begin{bmatrix} \Theta^{t+1} \\ V^{t+1} \end{bmatrix} = \begin{bmatrix} \Theta^{t} \\ V^{t} \end{bmatrix} + [J(\Theta, V)^t]^{-1} \begin{bmatrix} \Delta P^{t+1} \\ \Delta Q^{t+1} \end{bmatrix} \quad (4)$$

Where ΔP^t and ΔQ^t are the active and reactive power mismatch at iteration t. From the power flow problem formulation, it can be observed that a bad initial guess could cause divergence. Machine/deep learning can be used to establish key mapping between the active and reactive powers and the bus voltage solutions.

TABLE I
Generated ACPF Dispatch Cases.

Parameter	Number of ACPF Cases
Total Generated Dispatch	8,761 Cases
Solved Dispatch	4,862 Cases
Unsolved (non-converging)	3,899 Cases

III. Data Generation

The data used in this work consist of 8,761 power flow dispatch cases generated by [8] for a 6102-bus ERCOT system. These dispatch cases were generated using data from the U.S. Energy Information Administration (EIA), which includes hourly generation and load data for ERCOT for the year 2022 and the first hour of 2023 [21]. The ACPF cases represent the ERCOT system over a wide range of operating conditions spanning a year. The minimum load in the dispatch is 31.9 GW while the peak load is 79.8 GW [8]. The data generation process is documented in [8]. The dispatch cases were solved using the full Newton-Raphson method and the voltage solutions of the reference ERCOT PSS/E ACPF case. From this process, 3,899 ACPF cases did not converge. These 3,899 cases could not be solved from a flat start or initialization with the voltage solutions of the reference case. Table I shows the number of converged and non-converging ACPF cases. The solved cases form the training and validation data (based on a 90/10 split), while the non-converging cases with unknown solutions form the testing data.

IV. Methodology

To solve the non-converging ACPF cases, a framework that combines deep-learning initialization with incremental hot-start homotopy and oscillating switched-shunt control is developed. First, the non-converging case is initialized using a DNN initializer. If the case does not converge, an incremental hot-starting algorithm is applied, while the actions of control devices, such as the switched shunt, are monitored.

A. Deep Learning Initializer with Data Re-training

The general idea is to train a model to learn the relationship between active and reactive powers and the bus voltage phasor solutions (voltage magnitude and angle). Once the model has been successfully trained, it is applied to the non-converging ACPF cases. The predicted voltage magnitudes and angles are then used to initialize their respective non-converging ACPF cases, and the full Newton-Raphson method is applied to solve the cases.

Although researchers in [8] proposed a similar framework using an RF-based initializer, the algorithm was only able to achieve a 54% convergence rate due to the limited training data. The initial data is limited because the training data (4,376 cases) is not significantly greater than the testing data (3,899 cases). Additionally, the training data is significantly less than the total number of features. This paper uses the same limited data for the exact same ERCOT system but significantly improves the convergence rate by using a neural

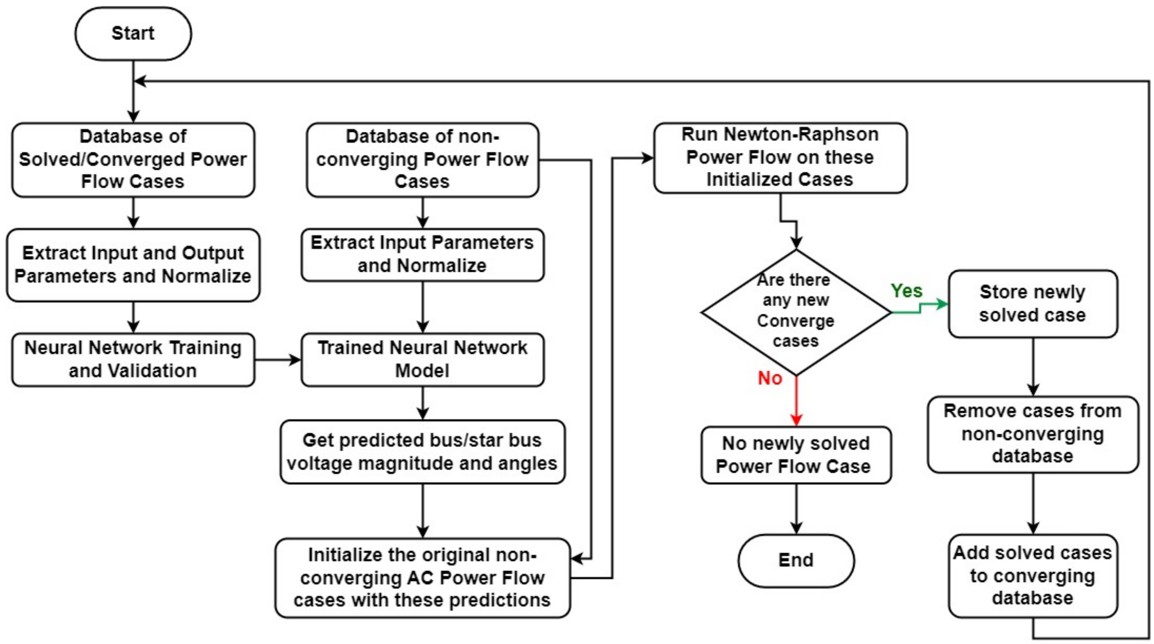

Fig. 1. Framework of the Deep learning Initializer with data re-training for Newton-Raphson ACPF Convergence.

network-based initializer and incorporating an iterative data re-training process into the algorithm, as shown in Fig. 1

From Fig. 1, the deep learning initializer framework starts by extracting the input and output parameters of the converged ACPF cases stored in a computer folder that acts as the database/storage. The input parameters are the generator active power at each bus, from bus 1 to the last bus N $(P_{(gen,1...N)})$, the active power load at each bus $(P_{(load,1...N)})$ and the reactive power load at each bus $(Q_{(load,1...N)})$. If a bus does not have a generator attached to it, the P_{gen} value is recorded as zero. The same applies to the P_{load} and Q_{load} at each bus. The outputs are the bus voltage magnitude, bus angles, star bus voltage magnitudes and star bus angles. The star buses are fictitious buses used in modelling three-winding transformers in many standard software like PSS/E and Powerworld [22].

The input data is then further separated into training and validation data based on a 90/10 split. Four separate neural network models are trained for 500 epochs for each output category. The training setup is shown in Fig.2. Since this is a regression task the loss function used is the mean square error and each neural network model is validated using the Root Mean Squared Error (RMSE) metric shown in (5), where n is the total data points, X_i and $\hat{X}_i$ are the true value and predicted value respectively. Regardless of the value of the RMSE the main task is to investigate whether the DNN initializer would assist in Newton-Raphson convergence.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2} \quad (5)$$

The input parameters of the non-converging ACPF cases are then extracted and normalized. For these non-converging cases, we do not know the actual solutions. The trained neural network model is applied to these non-converging ACPF cases, and the predicted voltage magnitudes and angles are obtained. The neural network regressor learns the mappings between the power injections and the power flow solutions. Although there is no guarantee that the neural network will learn the exact power flow equations, the hypothesis is that the predicted voltage solutions are close to the actual solutions and therefore within the convergence region [8]. This hypothesis is supported by [1], which highlights the importance of choosing initial guesses within the convergence boundary to ensure that an ACPF case converges. Additionally, [16] successfully used CNNs to initialize previously solved ACPF cases with the aim of reducing iteration time.

To test this hypothesis, the predicted voltage solutions of the non-converging cases are used as the starting points for Newton-Raphson power flow. Basically the predicted solutions are used to initialize their respective ACPF cases in PSS/E, after which full Newton-Raphson power flow is performed. ACPF cases that converge are then saved and added back to the training data. An ACPF case is considered converged or solved when the mismatch tolerance of the active and reactive power is met. A max tolerance of 0.5 MW/MVAR is used.

After each training iteration, the newly solved or converged cases (which were previously non-converging) are added to the training database, and the neural network is retrained, thus improving performance. By adding new solved cases to the training database, the neural network can learn from new information. This retraining process continues until no new cases are solved. Finally, all solved cases are checked for voltage violations by adding switched shunts using the QV analysis method [6]. It is important to note that the same hyperparameters and architecture are used for all four models, except for the output layer, which is determined by the size of the outputs (6,102 neurons for the bus voltage magnitudes/angles and 134 neurons for the star bus voltages/angles).

This article has been accepted for publication in IEEE Transactions on Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIA.2024.3514992

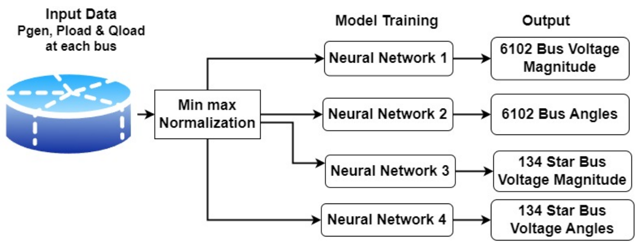

Fig. 2. Deep Neural Network Training Setup.

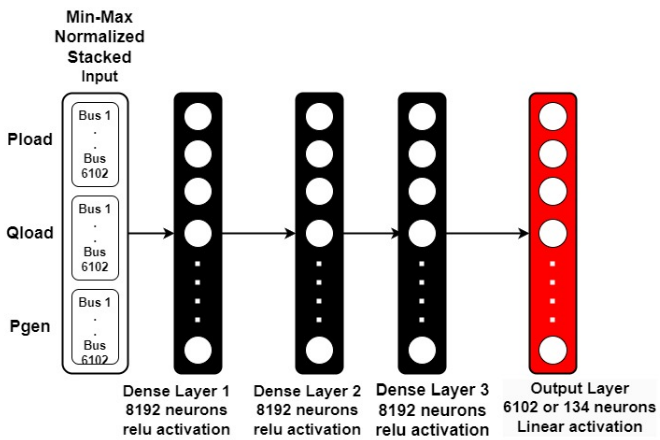

Fig. 3. Deep Neural Network Architecture.

Each neural network model consists of four fully connected layers, hence the term "deep neural network." Fig. 3 shows the architecture of the DNN, including the number of layers, the number of neurons in each layer, and the activation function used in each layer.

B. Incremental Hot-Starting Algorithm with Switched Shunt Control Based on Homotopy Continuation

ACPF cases that do not converge after being initialized with the predicted solutions from the DNN initializer are passed through an incremental hot-starting algorithm. This algorithm, based on homotopy continuation, incorporates switched shunt control, as shown in Fig. 4. The developed incremental hot-starting algorithm is significantly different from other methods in the literature because it avoids the optimal power flow technique [9] and the injection of generators and load technique used in [3]. Additionally, the developed algorithm includes a simple yet effective switched shunt monitoring and control mechanism.

The non-converging ACPF cases are referred to as the target dispatch, while the well-solved ACPF cases are the candidate dispatch. The concept is to select a candidate dispatch as a starting base case and gradually adjust its P_{gen} , P_{load} and Q_{load} values until they match those of the target dispatch. Essentially, P_{gen}, P_{load} and Q_{load} are considered homotopy parameters. It is crucial to select a candidate case that is as close as possible to the target dispatch. To determine the candidate case to select as the starting base case, the unit commitment similarity score (UCSS) metric [9] and a modified mean squared difference (MMSD) metric are used. UCSS assesses the similarity of the unit commitment of the generators between the candidate and the target cases. The candidate case with the highest UCSS score is selected as the starting base case. USCC is expressed in (6) where u_k represents the commitment of generator k and takes a value of either 0 or 1. If generator k is switched on in the target dispatch, then $u_{k,target} = 1$ otherwise $u_{k,target} = 0$. The same applies for the candidate dispatch. $\oplus$ is an Ex-NOR operator and N_g is the number of generators [9].

$$UCSS = \sum_{k=1}^{N_g} \overline{u_{k,target} \oplus u_{k,candidate}} \tag{6}$$

When there are multiple candidate cases with the same largest UCSS score, then MMSD is used. MMSD is a modification of the Mean Square Difference (MSD) Metric developed in [9]. The key difference is that MMSD considers both the generators and loads unlike MSD that considers only generator. MMSD looks at how close the generator and load dispatch are between the candidate and target case. The candidate case with the smallest MMSD is selected as the starting base case. MMSD is expressed in (7). N_l is the number of loads. $P_{load,j,t}$ and $P_{load,j,c}$ are the active power values of the load in the target and candidate case respectively. $Q_{load,j,t}$ and $Q_{load,j,c}$ are the reactive load values in the target and candidate case likewise, $P_{k,t}$ and $P_{k,c}$ are the generator active powers in the target and candidate case respectively.

$$\begin{aligned} MMSD = & \frac{1}{N_g}\sum_{k=1}^{N_g}(P_{k,t} - P_{k,c})^2 \\ & + \frac{1}{N_l}\sum_{j=1}^{N_l}(P_{load,j,t} - P_{load,j,c})^2 \\ & + \frac{1}{N_l}\sum_{j=1}^{N_l}(Q_{load,j,t} - Q_{load,j,c})^2 \end{aligned} \tag{7}$$

From Fig. 4, after selecting the candidate case, which now serves as the starting base case, the generator and load dispatch at each bus are slowly adjusted using an arbitrarily small initial step size S (1 MW/MVAR). After each adjustment iteration, the intermediate dispatch is compared with the target dispatch. If the intermediate dispatch matches the target dispatch, the hot-starting process is stopped. At this point, the target dispatch is considered solved, and quality checks are performed before adding this solved target case to the database. If the latest intermediate dispatch does not converge, the last solvable/converged intermediate dispatch is checked for oscillating switched shunts. If oscillating switched shunts are present, they are locked, and the step size is reduced. The intermediate case is then further adjusted and solved. If there are no oscillating switched shunts in the last solvable intermediate case, only the step size is reduced. The step size is an important parameter; a large step size could cause the intermediate case to diverge. It is advisable to use a small step size when adjusting the intermediate case, although this may increase the computational time of the algorithm. If the current intermediate case does not solve with $S \leq S_{min}$ (where S_{min}

This article has been accepted for publication in IEEE Transactions on Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIA.2024.3514992

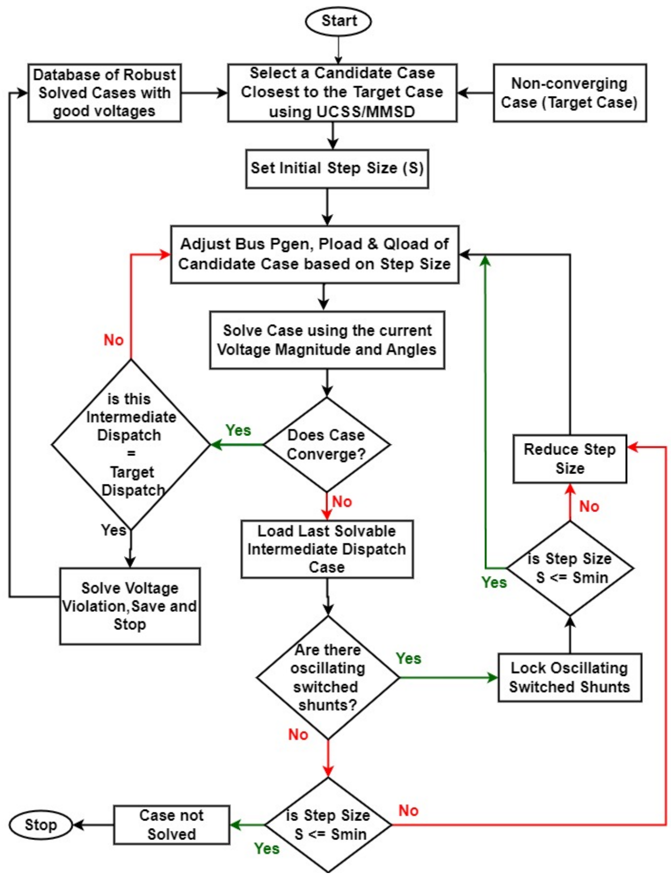

Fig. 4. Incremental Hot-Starting Algorithm with Switched Shunts Control.

Algorithm 1 Combined Two-Stage Convergence Framework

1 Start Stage 1
2 Get all stored solved/converged ACPF cases
3 Get all stored non-Converged ACPF cases
4 Train DNN Initilizer
5 Pass non-converging ACPF cases through DNN Initilizer
6 Obtain voltage and angle prediction
7 Initialize non-converging ACPF Cases
8 Run Full Newton-Raphson
9 **if** *New Cases Converge* **then**
10   Update database in 2 and 3
11   Jump to 4
12 **else**
13   Stop Stage 1
14 Check voltage violation in all converged cases
15 Solve any violation and store
16 **if** *there are still non-converging cases in 3* **then**
17   Start Stage 2
18   Run Hot-Start Algorithm
19   **if** *Case Converges* **then**
20   Check and solve voltage violation
21   Update database in 2 and 3
22   **else**
23   Two-Stage algorithm failed to converge ACPF Case
24 Stop Stage 2
25 End

is the minimum step size) and the last solvable intermediate case does not have any control issues with the switched shunt, then it means the algorithm cannot solve this case. At this point, both the DNN initializer and the hot-starting algorithm have failed to solve this case, and these cases need to be further investigated manually. Although manual investigation of unsolved cases is outside the scope of this paper.

C. Combined Framework for the Newton-Raphson AC Power flow Convergence Algorithm

The developed Newton-Raphson ACPF convergence algorithm is essentially a two-stage process. In the first stage, a deep learning-based initializer with data retraining is used to address the non-converging ACPF dispatches. The cases that converge at this stage are checked for voltage quality. Voltage violations are addressed using an automated process that adds switched shunts to the violating buses. This voltage violation solver is based on the reactive power planning framework developed in [6]. However, this automated voltage violation solver is not guaranteed to solve all voltage violations, as the algorithm relies on QV analysis, which has the potential to diverge. The ACPF dispatches that do not converge at this stage are then processed using the second stage, which involves the incremental hot-starting algorithm. The entire framework is detailed in Algorithm 1.

V. Results

A. Results from Deep Learning Initializer with Data Retraining

1) Results After the First Training Iteration: The performance of the deep learning initializer after the first iteration is shown in Table II. The RMSE values between the Deep Neural Network (DNN) initializer and the RF initializer used in [8] are quite close. The DNN initializer had a lower RMSE for the Bus/Star Bus angle predictions, while the RF initializer had slightly better Bus/Star bus voltage magnitude predictions. However, regardless of the RMSE values, the major concern is how well the initializer aids in solving the non-converging ACPF cases. Generally, training a deep neural network for a large system such as the 6102-bus ERCOT system takes significant computational time. To address this, a Graphics Processing Unit (GPU) with an NVIDIA GeForce RTX 2080 Ti was used for training the initializer, which significantly reduced the training time to about an hour for each output category. The DNN initializer successfully converged 2,316 ACPF cases out of the initial 3,899 non-converging cases after the first training iteration. As shown in Table III, even at the first training iteration, the DNN initializer outperformed both RF and DCPF initialization. The DNN had a 59.39% convergence success rate, representing a 5.38% increase in performance compared to RF initialization, which had a 54.01% convergence rate [8]

This article has been accepted for publication in IEEE Transactions on Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIA.2024.3514992

TABLE II
RMSE PERFORMANCE COMPARISON BETWEEN INITIALIZATION METHODS AFTER FIRST TRAINING ITERATION

Category	RMSE on Validation Validation Data from DNN	RMSE on Validation Data from RF [8]
Bus Voltage Magnitude	0.01892	0.01629
Bus Voltage Angle	0.0114	0.0138
Star Bus Voltage Magnitude	0.01016	0.0086
Star Bus Voltage Angle	0.0059	0.01024

TABLE III
CONVERGENCE PERFORMANCE COMPARISON BETWEEN INITIALIZATION METHODS AFTER FIRST ITERATION

Parameter	DNN Initializer	RF Initializer[8]	DCPF Initializer[8]
Total (Initial non-Converged Power flow Cases)	3,899 Cases	3,899 Cases	3,899 Cases
Power Flow Cases Converged by Initialization	2,316 Cases	2,106 Cases	758 Cases
Percentage (%) of Cases Solved by Initialization	59.39%	54.01%	19.44%
Remaining Non-Converged Power Flow Dispatch Cases after 1^{st} Training Iteration	1,583 Cases	1,793 Cases	4,141 Cases

2) Results with Data Re-Training: Although the DNN results are promising after the first training iteration, incorporating a data re-training methodology helps improve the convergence rate. This means that the initial 2,316 solved cases are added to the training database, and the DNN is re-trained. The added data provides the DNN with more information, which further aids convergence. This re-training framework continues until no new cases are solved. During re-training, the model architecture and hyperparameters were not changed. Fig. 5 shows the number of ACPF cases that successfully converged and the number of ACPF cases left unsolved after each re-training iteration. From Fig. 5, it can be observed that the number of new cases solved decreases after each re-training iteration. After the first training iteration, 1,583 ACPF cases did not converge. In the second training iteration, 359 cases were solved out of the 1,583 cases, leaving 1,224 unsolved ACPF cases. This process continues until the sixth training iteration. By the sixth iteration, only 5 ACPF cases were solved. Adding these 5 cases to the training database did not offer any significant improvement.

The DNN was able to successfully converge a total of 3,285 ACPF cases out of the initial 3,899 non-converging ACPF cases. This shows that the DNN initializer with data re-training aided in converging about 84.25% of the initial 3,899 non-converging ACPF cases. The DNN's ability to re-learn provides an advantage over non-iterative convergence methods like DCPF initialization. This re-learning capability makes artificial intelligence-based methods very useful when large number of cases need to be solved.

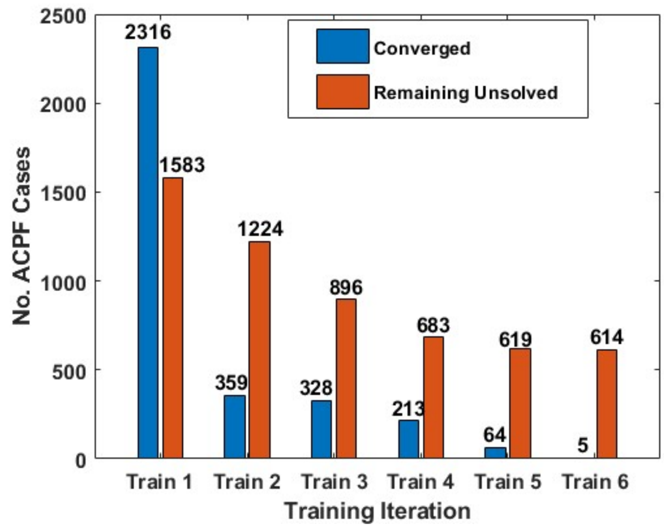

Fig. 5. ACPF Cases Converged at Each Re-Training Iteration .

3) Quality Checks and Voltage Violations in Converged ACPF Cases: The 3,285 ACPF cases solved by the DNN initializer are checked for voltage violations. From Table IV, a typical converged ACPF case has about 288 buses with voltages above 1.1 pu or below 0.9 pu. With 6,102 buses in the ERCOT system, this means about 4.7% of the total buses are violating voltage limits for a typical dispatch case. To address this, switched shunt control devices are added to the violating buses using the QV analysis method described in [6]. The switched shunts supply or absorb reactive power (MVARs) to keep the voltage at the violating buses within a specified range. The QV analysis is used to determine the size of MVARs to add on each violating bus. When solving for a particular bus, if the QV analysis causes the case to diverge, then that bus is skipped as described in [6]. While this automated voltage solver is not guaranteed to solve all voltage violations, it significantly reduces the number of violations to an average of 8 buses per case. The automated voltage solver reduced the number of violating buses (below 0.90 pu or above 1.1 pu) from an average of 4.7% (288 buses) to 0.13% (8 buses) for a typical dispatch in the 6,102-bus ERCOT system.

Fig. 6 shows the voltage of a typical ACPF case solved using the DNN initializer before switched shunts are added. It can be observed that multiple buses were below 0.9 pu for this particular dispatch case. Fig. 7 shows the improved voltage profile of the ACPF case after automated switched shunts were added to the various violating buses. The solved cases are also checked to ensure that their load dispatch values match the original specified values. This is because PSS/E automatically reduces loads when there is a significant voltage drop ($<$ 0.7 pu) [22]. After solving the voltage violations, cases that initially had reduced loads are rescaled back to their original values and solved. This ensures that the final solved case is of relatively good quality in terms of voltage and matches the specified dispatch without any load shedding. These 3,285 ACPF cases are then added to the database of solved cases.

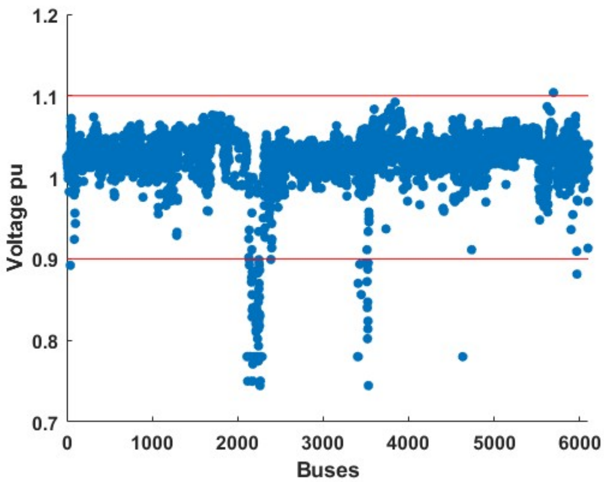

Fig. 6. Bus Voltage Magnitude of a typical ACPF case before applying automated switched shunts.

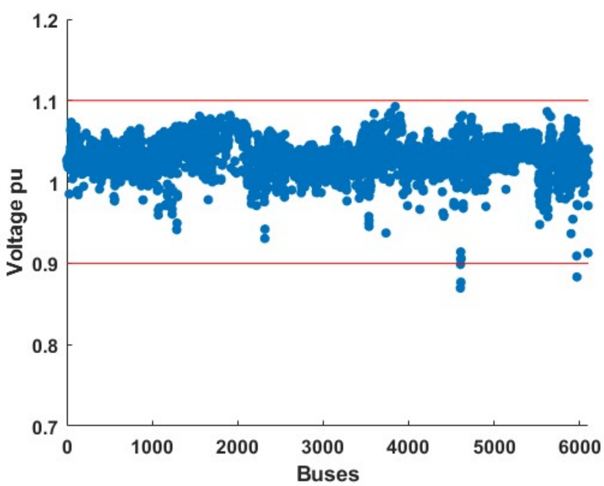

Fig. 7. Bus Voltage Magnitude of a typical ACPF case after applying automated switched shunts.

B. Results from the Hot-Starting Homotopy Algorithm with Switched Shunt Control

From the initial 3,899 non-converging ACPF cases, 3,285 ACPF cases have been successfully solved with relatively good quality using the DNN Initializer with reactive support added. The remaining 614 unsolved ACPF cases are then passed through the hot-starting algorithm based on homotopy continuation and switched shunt control. The hot-starting algorithm successfully converged 416 cases out of the initial 614 non-converging cases, as shown in Table V. The hot-starting algorithm gradually changes the P_{gen} , P_{load} and Q_{load} to fit the target unsolved dispatch cases. Of the 416 solved cases, 74 ACPF cases were solved by locking oscillating switched shunt devices and then adjusting the dispatch, while the remaining 342 ACPF cases were solved by adjusting the dispatch without locking any switched shunt devices. After solving the 416 ACPF cases via the hot-start algorithm, quality checks are performed to improve the voltage profile. Table VI shows the

TABLE IV
VOLTAGE VIOLATION STATISTICS IN 3,285 ACPF CASES SOLVED WITH DNN INITIALIZER

Voltage Magnitude Ranges (pu)	Average No. of Violating Buses per Converged Testing Case(Without Extra) Voltage) Support)	Average No. of Violating Buses per Converged Testing Case (With Automated Voltage) Support)
$<$ 0.90pu or $>$ 1.1pu	288	8
$<$ 0.85pu or $>$ 1.12pu	192	2
$<$ 0.80pu or $>$ 1.2pu	146	0

TABLE V
CASES SOLVED USING HOT-STARTING ALGORITHM WITH SWITCHED SHUNT CONTROL

Parameter	ACPF Cases
Initial Non-Converging Cases	614 Cases
Solved Cases with Hot-Starting	416 Cases
Remaining Unsolved	198 Cases

average number of buses with voltage violations for a typical case before and after adding the automated voltage support using switched shunt devices.

C. Combined Results from the Newton-Raphson ACPF Convergence Algorithm

The developed two-stage Newton-Raphson ACPF convergence framework successfully converged 3,701 ACPF cases out of the initial 3,899 non-converging ACPF cases. This represents a 94.9% convergence success rate. Fig. 8 shows an overview of the results. Of the initial 3,899 cases, 3,285 were solved in the first stage using the DNN initializer with data retraining, while 614 cases remained unsolved. The 614 unsolved cases were then passed through the hot-starting algorithm with switched shunt control. The hot-starting algorithm solved 416 of these 614 cases, leaving 198 cases unsolved. Further investigation of these 198 unsolved cases revealed that 125 could be adjusted to their load dispatch but not their generator dispatch, while 73 cases could not be adjusted to either their load or generator dispatch. From the voltage violation analysis of the total 3,701 converged cases, shown

TABLE VI
VOLTAGE VIOLATION STATISTICS IN 416 ACPF CASES SOLVED VIA HOT-STARTING

Voltage Magnitude Magnitude Ranges (pu)	Average No. of Violating Buses per Converged Testing Case(Without Extra) Voltage) Support)	Average No. of Violating Buses per Converged Testing Case (With Automated Voltage) Support)
$<$ 0.90pu or $>$ 1.1pu	36	8
$<$ 0.85pu or $>$ 1.12pu	5	1
$<$ 0.80pu or $>$ 1.2pu	1	0

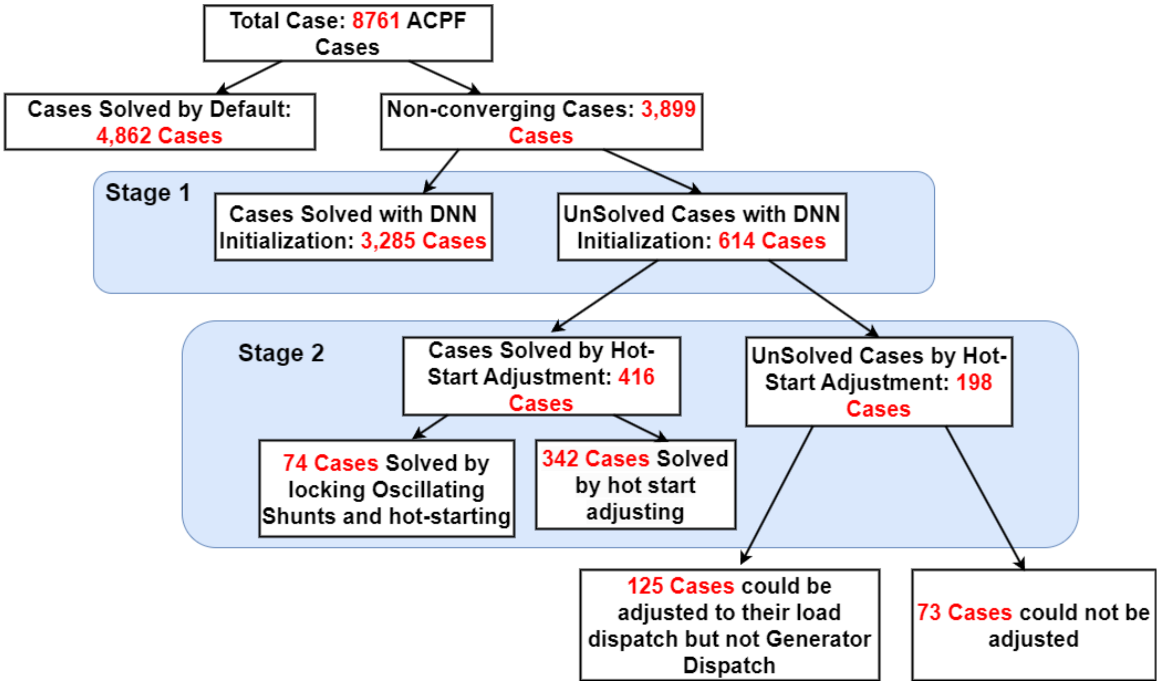

Fig. 8. Result Overview of the Combined Two-Stage Algorithm for Newton-Raphson ACPF Convergence.

TABLE VII
VOLTAGE VIOLATION STATISTICS IN 3,701 ACPF CASES

Voltage Magnitude Ranges (pu)	Average No. of Violating Buses per Converged Case
< 0.90pu or > 1.1pu	8
< 0.85pu or > 1.12pu	2
< 0.80pu or > 1.2pu	0

in Table VII, a typical case has about 8 buses with voltage violations out of the 6,102 buses in the system.

D. Result Comparison with Existing Literature

The result obtained from our two-stage approach is compared with other machine learning initialization methods and the standard DCPF initialization method currently used in the industry during power flow base case development. From Table VIII, the two-stage method described in this paper performed better than DCPF and other machine learning initializers [8] by achieving a 94.9% success rate.

Compared to other literature, the load reduction convergence method recently developed by researchers at the Pacific Northwest National Laboratory [6] and applied to a 22,000-bus WECC case successfully converged 89.4% of the ACPF dispatch cases. This technique, however, is not suitable when the load levels in a dispatch need to be maintained. NREL also recently applied the fictitious generator convergence method on a reduced 243-bus WECC case, which successfully converged 100% of the test cases [4]. However, it is important to note that only 17 non-converging dispatch cases of the WECC system were tested. Researchers in [9] achieved a 77.8% success rate, while researchers in [10] and [13] did not vary the operating conditions of their test ACPF cases.

Based on the percentage (%) of converged cases, our proposed two-stage method performs on par or better than the results obtained in other literature.

VI. CONCLUSION

In this paper, a two-stage automated framework that combines deep learning initialization with a hot-starting algorithm based on homotopy continuation and switched shunt control was designed. This framework achieved ACPF convergence in 3,701 cases out of 3,899 previously non-converging cases. The proposed method was applied to a 6,102-bus ERCOT system.

The developed method uses a two-stage approach, with the DNN initializer as the first stage. The DNN initializer addresses the issue of poor initial voltage guesses. It predicts initial voltage magnitudes and angles within the convergence region. Since Newton-Raphson is very sensitive to the initial voltage guess; the closer the initial guesses are to the actual solution, the higher the chance of convergence. The DNN initializer includes a re-training mechanism that enhances convergence performance, especially when faced with limited training data. It performed better than both the random forest initialization and the DCPF initialization commonly used in the industry.

The second stage involves a hot-starting algorithm with switched shunt control, based on homotopy continuation. Although the general concept of hot-starting power flow cases is well-established, incorporating switched shunt control improves performance. The proposed framework, which combines DNN initialization with the hot-starting algorithm, is of significant relevance to power system planners. Achieving a converged ACPF case in large power grids remains a challenge

TABLE VIII
RESULT COMPARISON ON ERCOT 6102 BUS CASE

ACPF Convergence Method	DNN Initializer + Hot-Starting and Switched Shunt Control	DCPF Initialization	RF Initializer	Decision Tree Initializer	Linear Regression Initializer
No. of Initial non-converging dispatch ACPF cases	3,899 Cases	3,899 Cases	3,899 Cases	3,899 Cases	3,899 Cases
No. of Converged ACPF Cases	3,701 Cases	758 Cases	2,106 Cases	1,783 Cases	246 Cases
Percentage.(%) of Converged ACPF Cases	94.9%	19.44%	54.01%	45.73%	6.31%

regularly faced by grid planners, and attaining convergence is a non-trivial task, especially in large grids. While the method developed in this paper shows promise by achieving a 94.9% success rate, it does have some limitations that provide opportunities for improvement in future studies. Some of the limitations and ideas for future work include:

1) The developed DNN initializer does not guarantee convergence without voltage violations or congestion. To address this, reactive support using switched shunts was added. Machine/deep learning simply maps input to outputs, and this raises a question on how effective it is in achieving power flow convergence for previously non-converging cases.
2) The results also show that machine/deep learning cannot be totally relied upon for solving power flow. As we observed several cases where the DNN initializer failed to assist in ACPF convergence. This area can be further explored in future.
3) The developed DNN initializer does not consider changes in grid topology. A framework for addressing grids with different topologies could be investigated in the future to further improve this method.
4) Although this paper is focused solely on steady-state ACPF convergence, future work would incorporate methods for evaluating the dynamic performance of all newly converged cases.

The main idea of the proposed two-stage algorithm is to automate the process for developing power flow cases with relatively good quality, and it is not a substitute for engineering judgement. The methodology described in this paper significantly reduces the arduous effort spent by grid planners in developing various solved power flow cases at different operating conditions.

REFERENCES

[1] J.J. Deng and H. D. Chiang. "Convergence region of Newton iterative power flow method: Numerical studies," *Journal of Applied Math*, 2013,doi: 10.1155/2013/509496.

[2] X. Dong et al., "Power Flow Analysis Considering Automatic Generation Control for Multi-Area Interconnection Power Networks," *IEEE Transaction on Industry Application*, vol. 53, no. 6, pp. 5200–5208, Nov. 2017, doi: 10.1109/TIA.2017.2738618.

[3] W. Murray, T. T. De Rubira, and A. Wigington, "Improving the robustness of Newton-based power flow methods to cope with poor initial points," *45th North American Power Symposium, NAPS* , 2013. doi: 10.1109/NAPS.2013.6666905.

[4] B. Wang and J. Tan, "DC-AC Tool: Fully Automating the Acquisition of the AC Power Flow Solution," *National Renewable Energy Laboratory* , 2022. [Online]. Available: www.nrel.gov/publications

[5] S. Dutto, G. Masetti, S. Chiaradonna, and F. Di Giandomenico, "On Extending and Comparing Newton-Raphson Variants for Solving Power-Flow Equations," *IEEE Transactions on Power Systems* , vol. 34, no. 4, pp. 2577–2587, Jul. 2019, doi: 10.1109/TPWRS.2019.2897640.

[6] B. Vyakaranam, Q. H. Nguyen, T. B. Nguyen, N. A. Samaan, and R. Huang, "Automated Tool to Create Chronological AC Power Flow Cases for Large Interconnected Systems," *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 166–174, 2021, doi: 10.1109/OAJPE.2021.3075659.

[7] A. B. Birchfield, T. Xu, and T. J. Overbye, "Power flow convergence and reactive power planning in the creation of large synthetic grids," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6667–6674, Nov. 2018, doi: 10.1109/TPWRS.2018.2813525

[8] S. N. Okhuegbe, A. A. Ademola, and Y. Liu, "A Machine Learning Initializer for Newton-Raphson AC Power Flow Convergence," *IEEE Texas Power and Energy Conference (TPEC)*, 2024

[9] A. A. Ademola, S. You, and Y. Liu, "Automated Two-stage Conversion of Hourly Optimal DC Flow Solution to AC Power Flow," *CSEE Journal of Power and Energy Systems*, vol. 9, no. 1, pp. 28–36, Jan. 2023, doi: 10.17775/CSEEJPES.2021.08870

[10] F. D. Freitas and A. L. Silva, "Flat start guess homotopy-based power flow method guided by fictitious network compensation control," *International Journal of Electrical Power and Energy Systems*, , vvol. 142, Nov. 2022, doi: 10.1016/j.ijepes.2022.108311.

[11] S. Cvijic, P. Feldmann, and M. Hie, "Applications of homotopy for solving AC power flow and AC optimal power flow," *IEEE Power and Energy Society General Meeting*, 2012. doi: 10.1109/PESGM.2012.6345453.

[12] R. Kalaba and Tesfatsion L, "Solving Nonlinear Equations by Adaptive Homotopy Continuation," *Applied Mathematics and Computation*, pp. 99–115, 1991.

[13] Q. Sun, L. Liu, D. Ma, and H. Zhang, , "The initial guess estimation Newton method for power flow in distribution systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 2, pp. 231–242, Apr. 2017, doi: 10.1109/JAS.2017.7510514

[14] F. Milano, , "Continuous Newton's method for power flow analysis," *IEEE Transactions on Power Systems*,, vol. 24, no. 1, pp. 50–57, 2009, doi: 10.1109/TPWRS.2008.2004820

[15] B. Stott, , "Effective Starting Process for Newton-Raphson Load Flows," *Proceedings of The Institution of Electrical Engineers*vol. 118, no. 8, pp. 1–2, 1971.

[16] L. Chen and J. E. Tate, , "Hot-Starting the Ac Power Flow with Convolutional Neural Networks," ,Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.09342

[17] 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 Transactions on Power Systems* vol. 36, no. 1, pp. 346–354, Jan. 2021, doi: 10.1109/TPWRS.2020.3001919.

[18] X. Hu, H. Hu, S. Verma, and Z. L. Zhang, , "Physics-Guided Deep Neural Networks for Power Flow Analysis," *IEEE Transactions on Power Systems* vol. 36, no. 3, pp. 2082–2092, 2021, doi: 10.1109/TPWRS.2020.3029557.

[19] K. Baker, , "Learning Warm-Start Points for AC Optimal Power Flow," *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)* 2019

[20] F. Casella and B. Bachmann, , "On the choice of initial guesses for the Newton-Raphson algorithm," *Applied Mathematics Computation* vol. 398, Jun. 2021, doi: 10.1016/j.amc.2021.125991.

[21] EIA, , "Hourly Electric Grid Monitor," *EIA* Accessed: Jun. 05, 2023. [Online].Available $https://www.eia.gov/electricity/gridmonitor$

[22] Siemens, , "Program Operation Manual - PSS®E 34.8.1 ," *Siemens* 2020.