Advanced convolutional neural network for optimizing AC power flow

Quan Tran[1], Joydeep Mitra[2], Nga Nguyen[1]
[1]Electrical Engineering and Computer Science, University of Wyoming, Laramie, WY, {qtran2, nga.nguyen}@uwyo.edu
[2]Electrical and Computer Engineering, Michigan State University, East Lansing, MI, mitraj@msu.edu

***Abstract*—Current methods for solving optimal power flow have difficulties obtaining real-time solutions due to the complexity and nonlinearity of large power systems. Taking advantage of machine learning (ML) algorithms, trained machine-learning models are viable alternatives to tackle the challenges of online OPF-solving. This paper proposes the application of neural network models with self-attention to reduce an inductive bias and overcome the drawbacks of traditional ML-based approaches in solving OPF. By incorporating all power system parameters into a multiple-channel matrix akin to an image, the proposed weighing-convolutional neural network (WCNN) can extract typical features from training data and learn the underlying patterns. The high computational burden is alleviated through the training phase, and an OPF solution can be produced instantly by inference of the trained model. The proposed method is implemented on three IEEE standard systems to prove its efficacy. The experiment results show that the proposed WCNN algorithm obtains a high level of accuracy, which facilitates its potential capability for deployment in the practical operation of complex power systems.**

***Index Terms*—Convolutional neural network, cost of generation, AC optimal power flow, self-attention.**

I. Introduction

To solve OPF problems in power systems, linearization, convex relaxation, and approximation techniques are popularly used [1]–[8]. However, achieving an OPF solution using these methods is often time-consuming and sometimes infeasible for large-scale and intricate power systems or online OPF calculation due to the system's nonlinearity and complexity [9]. A typical approximation technique utilizes distributed/decentralized methods, which split the main problem into numerous smaller problems, to solve OPF problems efficiently through parallel processes. A solution to solve semi-definite programming in a distributed manner by utilizing the alternating direction method of multipliers (ADMM) was proposed in [5], [10]. An alternative to ADMM called Augmented Lagrangian-based Alternating Direction Inexact Newton method (ALADIN) was used to handle non-convex OPF in a distributed manner in [6]. Notwithstanding, these approximation methods take advantage of the relaxation to obtain the possible optimal solution, resulting in an uncertainty of the globally optimal solution and unlikely feasibility in real-time operational conditions [11]. Despite reducing the colossal computational amount based on the relaxation, OPF-solving is complicated to handle, especially when the OPF problems become non-convex.

Multiple realms of technological fields have resorted to the advantages of machine learning to solve optimization problems. Numerous ML-based approaches are proposed to handle various emerging problems of electric power systems. In [12]–[17], deep-learning models are built to alleviate the arduous computation burden of classical OPF-solving. These ML-based models have shown a significant improvement in computational speed compared to traditional methods. Nevertheless, learning models, which are built on a background of multilayer perception (MLP) with each layer constructed by nodes (neurons), have a critical shortcoming of whopping trainable parameters to large-scale power systems [18]. To dodge that point, these learning-based end-to-end works take some power systems' information to extract features and learn mapping rules from given datasets. For instance, load demand or power generation is collected as input to feed to learning models for decreasing the number of trainable parameters [12], [13]. Consequently, the generalization of trained neural networks would be oppressed due to the lack of comprehensive input. In other ML-based hybrid frameworks, deep learning algorithms are utilized as alternative solvers for conventional OPF-solving through power flow equations [15]–[17]. The from-ML-achieved results would be a successive input to compute power systems' parameters.

By a convolution of a weighted small matrix over an image patch, a convolutional neural network (CNN) comprehensively captures all information, unlike earlier machine learning models that tend to selectively focus on a limited subset. CNN is emerging as a potential candidate to tackle the difficulty of vast trainable parameters. CNN excels at capturing spatial hierarchies and is particularly effective for image-related tasks due to the capability of learning relevant features from the data. This paper takes the apparent advantages of CNN to develop a learning model that could cover all essential information of electric power systems by arranging the input under an image-like shape to solve OPF. The proposed approach is conducted by excelling the conception of DenseNet [19], accompanied by a self-attention mechanism [20]. A decoder block comprised of fully connected layers helps transform extracted patterns into expected results. The main contribution of the paper is as follows:

- Propose an approachable solution of leveraging convolutional neural networks and a self-attention mechanism to find the solution of AC OPF efficiently without any constraint on the input size of a case study.
- Embed the entire electric power system information into the learning process to drive the proposed ML model to reflect the nature of the inner variation of elements such

as load demands, generation power, and topology changes straightforwardly.

The structure of the paper is as follows. Section II develops a weighing-convolution neural network for OPF and the process of transforming the OPF problem into a machine-learning model. Section III presents the simulation results with discussions. Finally, the conclusion is in Section IV.

II. A WEIGHING-CONVOLUTIONAL NEURAL NETWORK FOR OPF-SOLVING

Neural networks are structured to mimic the decision-making process of the human brain. It comprises multiple layers, each containing artificial neurons, arranged sequentially one after the other. Machine learning (ML) algorithms are mathematical conceptions that express the calculation between interconnected neurons. The ultimate goal is to figure out the mapping rules or mathematical functions to depict the most accurate relation between input and output. Considering the complexity of a problem, opting for an appropriate structure of a neural network is crucial to ensure the extraction is successful or that the most approximate mapping function is obtained.

A. The residual network and its variants

The proposed learning model is a renewal of predecessors of the CNN family, and it has all distinctive characters whose previous models were devised, such as Residual Networks - ResNet [21], and DenseNet [19].

The residual network has a significant impact on the field of computer vision when a deep learning process should not lead to lower accuracy due to the information transformation after feeding multiple layers. The core idea is to maintain a skip connection or interactive connection between the model input and the model output to ensure the accuracy of the entire model that is not far lower than the original one.

The operation of ResNet is abstracted as follows:

$$f(X) = X + g(X) \tag{1}$$

where X is a given input added to $g(X)$. $g(X)$ is the mapping rule of interactive connection to form $f(X)$, and $f(X)$ is the expected mapping function. It is shown in equation (1) that ResNet is a combination of two separate elements: an affine one and a non-linear one.

The distinction of ResNet is the addition operation arising in its forward feeding where the input is transmitted directly to the output via a shortcut connection, specifically shown in Fig. 1. This approach forms a kind of interactive connection to ensure the enlarging of the model not to lower the training accuracy [18].

The residual network resolves the risk of lower accuracy as expanding learning models deeper due to the additional task. Notwithstanding, some underlying patterns are dropped out or in other words, they are merged through the additional task of mapping function. Consequently, not all low-level features are extracted which leads to a high discrepancy in predictions and ‘ground truth’ values. To that end, the densely connected

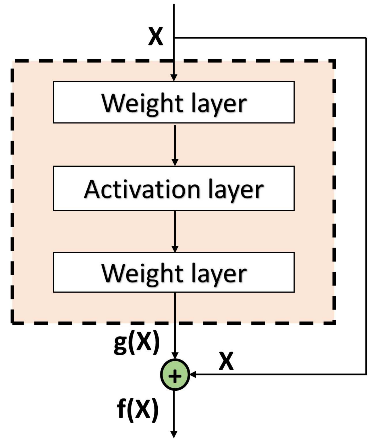

Fig. 1: The principle of the residual network ResNet.

network DenseNet - a variant of the residual network was introduced in [19] to fix the residual network’s drawback of missing out latent information.

Instead of the addition, the task of concatenation is used as an alternative for preserving the input information during the entire forward process. The principle of DenseNet is depicted in Fig. 2 and Equation (2).

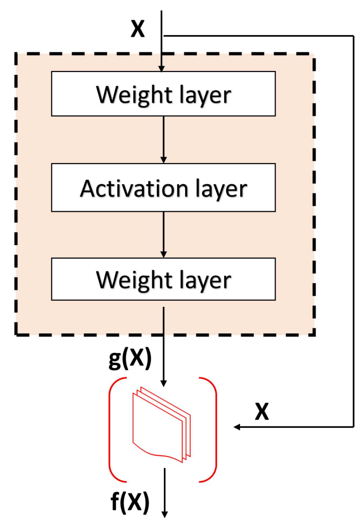

Fig. 2: The basic principle of a Dense Block. The concatenation operation is an alternative to ResNet’s addition one.

$$X \longrightarrow \{X, f_1(X), f_2(X, f_1), f_3(X, f_1, f_2), ...\} \tag{2}$$

Due to the concatenation task, DenseNet not only retains the patterns of input data but also filters the correlation of a variety of channels out during the process of enlarging the network architecture deeper. As a matter of fact, DenseNet outperforms the daunting tasks of image processing compared to ResNet [19]. Specifically, the densely connected network feeds forward an input X to the output, simultaneously gathering underlying patterns extracted at each dense block. As a result, the latent patterns inside of the data are accumulated gradually according to the model depth instead of being transformed under the addition task. Based on that, DenseNet leverages the exceptional attributes of CNNs for image processing efficiently rather than previous predecessors.

B. Application of a self-attention mechanism

A self-attention mechanism is a crucial part of natural language processing models, machine translation, text summarization, and sentiment analysis [20]. It helps to raise the efficiency compared to previous RNN models, such as Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM). It weighs all individual elements within an entire context to evaluate each element's significance. Noteworthily, the individual's weighing is helpful in capturing latent patterns and features hidden in spatial-positioned grids like electric power systems.

The triple of a self-attention mechanism includes:

- The $query$: $q = W_q x$ to set up the weights for its input.
- The key: $k = W_k x$ to set up the weights for its correlation.
- The $value$: $v = W_v x$ to set up the weights for its output.

The weight matrices W_q, W_k, and W_v are updated during the learning process to weigh the individual's significance. The output $y^{<t>}$ is a total of weighted vectors $v^{<1>}, ..., v^{<t>}$ expressed as follows [22]:

$$y^{<t>} = \sum_{u \leqslant t} \alpha_{tu} v^{<u>} \tag{3}$$

with $\alpha_{tu} = \frac{\exp\left(score(q^{<t>}, k^{<u>})\right)}{\sum_{v=1}^{t} \exp\left(score(q^{<t>}, k^{<v>})\right)} \quad \forall u \leqslant t$ and the score function is the dot-product function.

C. Optimizing AC power flow

AC optimal power flow is a process that allocates pre-calculated amounts of essential generated power to all participating generation units. In essence, AC OPF aims to reach the most efficient cost of allocation of power generation resources considering various factors such as fuel costs, operating constraints, and load demand. OPF is an optimization problem with an objective function of minimizing the operational cost. The formulation of the optimal AC power flow could be expressed as:

$$\min_x f(x) \tag{4}$$

subject to

$$g(x) = 0 \tag{5}$$

$$h(x) \leqslant 0 \tag{6}$$

$$x_{min} \leqslant x \leqslant x_{max} \tag{7}$$

where

$$f(P_g, Q_g) = \sum_{i}^{n_G} f_P^i(p_g^i) + f_Q^i(q_g^i) \tag{8}$$

The cost function $f(x)$ (8) is the polynomial equation of active generation power and reactive generation power. The inequality-constrained equations $h_i(x)$ encompass the constrained power flow of the transmission lines, and x_{max}, x_{min} are the upper and lower limits of the size of voltage magnitudes, voltage angles, and the generator dispatch capabilities [23]. The inequality-constrained equations can be the line's security power-transmitted capability [24]. The equality-constrained equations $g(x)$ consist of the nonlinear trigonometric power flow equations below:

$$P_i - |V_i| \sum_{k=1}^{n} [(G_{ik} \cos\theta_{ik} + B_{ik} \sin\theta_{ik})|V_k|] = 0 \tag{9}$$

$$Q_i - |V_i| \sum_{k=1}^{n} [(G_{ik} \sin\theta_{ik} - B_{ik} \cos\theta_{ik})|V_k|] = 0 \tag{10}$$

where $V_i = |V_i| \angle \theta_i, \theta_{ik} = \theta_i - \theta_k, Y_{ik} = G_{ik} + jB_{ik}$ and $i = 1, 2, ..., n$.

D. Weighing-convolutional neural network

The family of convolutional neural networks is powerful for grid-structured-relating tasks like image processing and computer vision. Convolutional computation enhances the capability of catching features such as edges, margins, and identities. Additionally, it reduces the number of trainable parameters in a neural network compared to previous MLP-based models due to the convolution process causing the update of trainable parameters to take place on a kernel or a filter which is a small weight matrix.

The convolutional calculation for two-dimensional matrices $f, g : \mathbb{R}^d \to \mathbb{R}$, is constructed as follows [25]:

$$(f * g)(i', j') = \sum_{i'', j''} f(i'', j'') g(i' - i'', j' - j'') \tag{11}$$

where the indices for F are indicated by a pair (i', j'), and likewise a pair (i'', j'') indicates for the indices for g. $\mathbf{w}$ is a weight matrix known as a filter or a kernel of the convolution layer. The indices (i'', j'') indicate the surrounding area where a convolution task is manipulated, specifically the vicinity of (i'', j'').

In mathematical terms, the output of a convolution layer can be expressed as follows [22]:

$$z_{i',j'} = \sum_{(i'', j'')} w_{i'', j''} x_{(i'+i''),(j'+j'')} + b \tag{12}$$

Equation (12) illustrates that the convolution computation can reduce the number of trainable parameters and the convolution layer is a proper alternative to the fully connected layer. Specifically, by a dot-product operation of a small weight matrix - a kernel $\mathbf{w}$ over each point of the input, the output $z_{i',j'}$ is harnessed by the summation of all dot product calculation results of a kernel and the respective input area that a kernel scans through. Consequently, it is not contingent on the input shape of a model that makes CNN models privileged in solving spatial-structured models.

Based on that, an integration of the self-attention layer right after convolutional structures is a potential approach to take advantage of the CNN family for OPF-solving and constrain its drawbacks, such as weakly reflecting the correlation and interdependence of patterns extracted by convolution layers. As mentioned in Subsection II-B, the self-attention layer can help weigh the significance of each neuron in a layer, therefore the spatial patterns like topology information and magnitude fluctuations of input variables like load demand can be recognized by the weighing task of the self-attention mechanism.

Hereafter, DenseNet is chosen as a CNN model to combine the self-attention mechanism to implement on three IEEE standard systems: IEEE-6, IEEE-9, and IEEE-14.

III. Simulation Results

In this section, the process for testing the proposed neural network on the IEEE standard systems is introduced to illustrate the data generation, setting up the learning model, and conducting experiment.

A. Dataset preparation

MATPOWER Interior Point Solver [23], [26] is exploited to generate training data for each case study. MATPOWER Interior Point Solver settings are kept by default and Monte Carlo simulation is applied to ensure the diversity of the dataset. A coefficient vector whose dimension is equal to the number of loads is drawn from the uniform distribution of the range $[0.8, 1.2]$, except for IEEE-6 with the range $[0.95, 1.05]$ due to the hardship in convergence. Preset load profiles and the cost of generators are multiplied by the distinct random vector for every sample to ensure the diversification of the training dataset. Additionally, to ensure that the model has the dataset sufficient for training, a set of $10,000$ samples for each case study is considered.

B. Structure of training dataset

The dataset structure of a sample consists of 2D matrices, which are concatenated as follows:

- The matrices of load demand $P_{N_B \times N_B}$/$Q_{N_B \times N_B}$ in $MW/MVAr$, where N_B is the number of buses.
- The matrices of each element (G, B) of the admittance Y_B in pu.
- $N_C \times N_G$ diagonal matrices of generation cost in $\$/MW, \$/MVAr$, with N_C indicates the maximum exponential index of the objective function (8), and N_G indicates the number of generators in each case study.

Generally, the dataset I has the shape: $[N_I, (4 + N_C \times N_G)_{channels}, N_B, N_B]$. The detailed dataset size of all case studies is empirically computed, as shown in Table. I. The output shape is generalized as $(2 \times N_B + 2 \times N_G)$, which encompasses all the optimal variables: angles and magnitudes of voltage, dispatched active and reactive power of generators. The error function that is used to train the WCNN is the mean squared error (MSE). For a given i-th sample of I, the MSE formulation is written as follows [13]:

$$\min_{\theta} \frac{1}{N_I} \sum_{i}^{N_I} \|\mathbf{y}_i - \mathcal{F}\left(\mathbf{x}_i; \theta\right)\|_2^2 \tag{13}$$

where N_I is the size of I; $\mathcal{F}\left(\cdot; \theta\right)$ is the desired mapping rule, with θ is symbolized for the optimal variable of the current model which is a set of trainable parameters. The generator dispatch optimization problem is illustrated via equations from (4) to (8), has become simple by finding a set of the WCNN parameter θ that satisfies the equation (13) through the WCNN training.

TABLE I: Case study information (including generators, loads, buses, and lines)

System	Configuration	Input	Output
IEEE-6	3, 3, 6, 11	$[N_I, 13, 6, 6]$	$[N_I, 18]$
IEEE-9	3, 3, 9, 9	$[N_I, 13, 9, 9]$	$[N_I, 24]$
IEEE-14	5, 11, 14, 20	$[N_I, 19, 14, 14]$	$[N_I, 38]$

C. Configuration of the weighing-convolutional neural network

The weighing-convolutional neural network (WCNN) is constructed on a base of an encoder block and a decoder block. The dense blocks of successive convolutional layers, followed by the self-attention layer, are grouped in an encoder block. In [27], the WCNN utilizes two stacked dense blocks whose each dense block has two convolution layers, and the number of convolution layer's channels is chosen as 8. The kernel matrix of the convolution layers has a size of 3×3, and the self-attention layer has multiple heads of 10. With such a unique structure, the WCNN can dig deeper into hidden patterns in data structures to extract mapping rules in terms of the fluctuation of input variables and the correlation of the like-grid system's nodes in a given spatial layout.

Specifically, the encoder block is depicted in Fig. 3. It processes the provided data input to extract the fundamental properties of data and hidden patterns of the data correlation. The encoder block effortlessly adapts and summarizes electric power network parameters, which are structured in an image-like format, using multiple stacked dense blocks. Following the dense blocks is the self-attention layer whose task is to weigh how important each element of the previous layer's output contributes to the contextual situation.

The decoder comprises multiple dense layers known as multilayer perceptron (MLP), as shown in Fig. 4. Herein, there are 7 hidden dense layers in the WCNN complying with the principle of a non-linear (relu) activation-function dense layer (1024 units) followed by a linear activation-function dense layer (512 units) as a regularization technique of a bottle-neck to reduce mutual connections for avoiding overfitting and enhancing the generalization. The WCNN is implemented by the Python-language programmed Tensorflow library and the optimization algorithm - Adam is used as an optimizer [28].

D. Advanced adjustment of the WCNN configuration

An advanced adjustment of the hyperparameters of the WCNN's encoder and decoder is proposed by increasing the number of convolutional layers in the encoder's dense blocks, simultaneously decreasing the number of dense layers' units compared to the original model in Section III-C. Particularly,

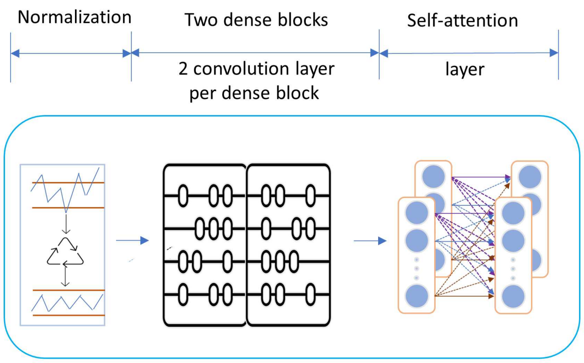

Fig. 3: The encoder block of the base WCNN includes three parts which are data normalization, making dense blocks, and implementing a self-attention mechanism. There are 2 convolution layers per dense block [27].

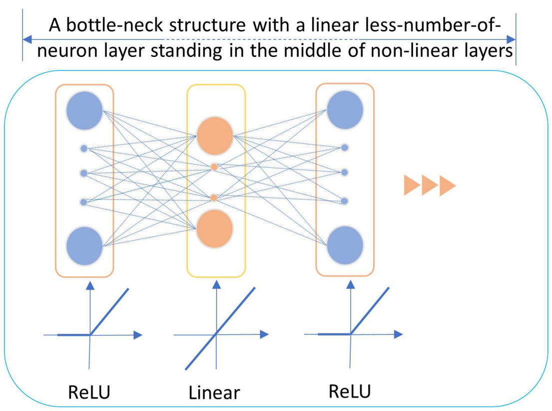

Fig. 4: The WCNN's decoder block is akin to a multilayer perceptron, exceptionally with an additional regularized technique - a bottle-neck technique.

the number of convolutional layers is shifted up to "8" per dense block from "2", while the number of units of the decoder's dense layers is reduced by half. As a result, the number of trainable parameters is reduced significantly in Table II which illustrates a considerable benefit obtained by the hyperparameter adjustment. Moreover, the hyperparameter-modified model outperforms its original version as in Table. III. It can be seen that an extension of the WCNN in the depth of dense blocks in the encoder might enhance the capability of mapping the rule lying in the complex data pattern of power systems' collecting information with lower consumption of resources or the smaller number of model's parameter, which is shown in Table. II for training the model compared to the original model.

E. Simulation Results

The proposed WCNN is conducted on three IEEE standard systems: IEEE-6, IEEE-9, and IEEE-14. For each case study, a training dataset of $10,000$ samples is provided to train the WCNN before testing on a $1,000$-sample testing dataset

TABLE II: The number of parameters of the original WCNN (base) [27] and the advanced WCNN (advance).

System	Type of Model	
	Base	Advance
IEEE-6	$3,001,590$	$1,125,554$
IEEE-9	$3,151,656$	$1,323,938$
IEEE-14	$3,582,034$	$1,906,658$

that has not ever been seen. All the datasets are generated following the procedure mentioned in Subsection. III-A. The mean absolute error function (MAE) of optimal variables such as angles/magnitudes of voltage, and active/reactive power yields the accuracy level of the estimated values from the WCNN. The discrepancy does not vary significantly between the case studies. However, there is a remarkable difference in the accuracy when the complex level of a case study grows. Particularly, there is the same tendency of rising for both values of voltage regarding the angle and magnitude, active/reactive power. Nevertheless, this phenomenon is natural in essence when the WCNN faces the increasingly intricate level of case studies but its hyperparameters and structure remain unchanged. It can be understood that the WCNN can learn well about different electric power systems if there is an infrastructure of hardware resources equivalent to the complicated issues it will encounter. The detailed MAE error is displayed in Table III.

By increasing the depth of dense blocks, the proposed model has captured easily latent features under different data patterns, thus leveraging the WCNN's decoder configuration which would make the model training resource-saving and faster. It merges the convolution properties and the attention mechanism, which are supplemented to extract features from the given dataset whose characteristics are intricated and especially untractable as to the large-scale power systems. With a structure arranged in a like-image shape, a convolution kernel could sweep through the entire context of power systems to obtain profound features such as topology changes, variation of load demand, and the generating cost, without concern about the number of the input dimension. Therefore, the proposed model may outperform MLPs (multilayer perceptrons) which are popular in recent ML-related works in an aspect of the scalability to complex power systems with a significant computing resource saving in an exponential reduction of trainable parameters [18]. Furthermore, the functional task of the attention mechanism supports the feature extraction capability of the proposed model by weighing the importance of each neuron in the neural network. As a result, the proposed model can overcome the disadvantages of MLPs in figuring out the patterns underlying the input data with a considerably smaller number of trainable parameters. Additionally, the computation time of machine learning inference is better than that of OPF problems solved by numerical methods, which are usually time-consuming [27]. The forward process of a neural network is simply the multiplication between parameter matrices. Thus it is faster to compute than the iterative loops done by numerical methods.

TABLE III: Mean absolute error (MAE) of the original model (Base) [27] and the proposed model (Advance) across the testing dataset.

System	IEEE-6		IEEE-9		IEEE-14	
Model Type	Base	Advance	Base	Advance	Base	Advance
σ (degree)	8.539×10^{-4}	2.161×10^{-3}	8.443×10^{-3}	8.785×10^{-3}	1.151×10^{-2}	1.080×10^{-2}
|V| (pu)	4.058×10^{-4}	1.090×10^{-3}	1.644×10^{-3}	1.857×10^{-3}	7.674×10^{-3}	4.716×10^{-3}
P_G (pu)	8.629×10^{-3}	9.593×10^{-3}	6.577×10^{-2}	5.788×10^{-2}	5.516×10^{-2}	5.445×10^{-2}
Q_G (pu)	3.546×10^{-3}	8.694×10^{-3}	3.058×10^{-2}	2.740×10^{-2}	1.436×10^{-2}	1.315×10^{-2}

TABLE IV: Mean absolute error (MAE) of the proposed model across the testing dataset with and without noise added.

System	IEEE-6		IEEE-9		IEEE-14	
	Without Noise	With Noise	Without Noise	With Noise	Without Noise	With Noise
σ (degree)	2.161×10^{-3}	1.391×10^{-2}	8.785×10^{-3}	6.008×10^{-3}	1.080×10^{-2}	1.087×10^{-2}
|V| (pu)	1.090×10^{-3}	2.214×10^{-3}	1.857×10^{-3}	1.711×10^{-3}	4.716×10^{-3}	4.461×10^{-3}
P_G (pu)	9.593×10^{-3}	1.529×10^{-1}	5.788×10^{-2}	7.892×10^{-2}	5.445×10^{-2}	5.646×10^{-2}
Q_G (pu)	8.694×10^{-3}	6.036×10^{-2}	2.740×10^{-2}	2.417×10^{-2}	1.315×10^{-2}	1.280×10^{-2}

TABLE V: Mean absolute error (MAE) of the proposed model (WCNN) compared to the multilayer-perceptron-based model (MLP) [13].

System	IEEE-6		IEEE-9		IEEE-14	
	WCNN	MLP	WCNN	MLP	WCNN	MLP
σ (degree)	4.962×10^{-3}	1.495×10^{-2}	9.062×10^{-3}	8.215×10^{-2}	1.737×10^{-2}	3.766×10^{-2}
|V| (pu)	2.148×10^{-3}	6.134×10^{-3}	3.707×10^{-3}	1.258×10^{-2}	6.550×10^{-3}	3.679×10^{-3}
P_G (pu)	2.076×10^{-2}	1.300×10^{-1}	4.042×10^{-2}	5.027×10^{-2}	6.615×10^{-2}	8.276×10^{-2}
Q_G (pu)	1.772×10^{-2}	1.154×10^{-1}	1.659×10^{-2}	1.794×10^{-1}	1.820×10^{-2}	2.239×10^{-2}

In addition, a multilayer-perceptron-based method (MLP) has recently become popular for solving AC-OPF. To reflect the efficacy of the WCNN and MLP, the MLP is configured identically to the decoder of WCNN, with the input only including active/reactive power at loads similar to the method in [13]. Both the MLP and WCNN are trained and tested on the identical dataset which includes numerous samples reflecting various scenarios such as power fluctuation at load buses, N-1 contingencies, and the variety of power generation prices. The results achieved after running experiments for three IEEE test cases are shown in Table. V. The trained MLP has a worse prediction than the trained WCNN on the same testing data due to the higher value of most MAE errors. It can indicate that the previous methods such as MLP-based models, which do not cover all power systems' information, do not have a comprehensive performance in intricate scenarios. Conversely, the WCNN, which has lower values of MAE error, proves its capability not only to synthesize all power systems' information without any concerns regarding the power system scalability but also to perform prediction reliably close to truth values in a diversified scope of scenarios.

Regarding the proposed model's robustness, multiple regularization techniques are applied to enhance the reliability when predicting unseen data. 'Dropout' layers are added into the model along with 'batch normalization' layers to prevent the model from overfitting or getting stuck at local optimal solutions. Furthermore, a non-Gaussian noise was added to the testing datasets, but not to the training datasets to inspect the variation of the prediction under noise-added circumstances representing the uncertainty of data or measurement errors. Measurement errors were emulated randomly, and a random noise of size $0-5\%$ of the original active/reactive load demand was generated and added to the testing datasets. In Table. IV, the results generally show that MAE errors are still within an accepted threshold when predicting on the noise-added testing dataset. Therefore, the proposed model is robust enough to encounter stochastic data or common errors in collecting data.

The simulation results show that the WCNN is adaptive and open to adjustments and fine-tuning for improving the model efficacy. Although being modified from the base model, the proposed model still inherits all of its original characteristics such as extracting fundamental correlation of spatial features such as the topology information, load demand, or grid structure. Hence, the proposed model is flexible and reliable to apply for various tasks in power system research fields.

IV. Conclusion

This paper has proved that the weighing-convolutional neural network has a strong potential application in electric power systems, particularly in OPF. Furthermore, the WCNN is adaptive, scalable, and potentially improved when it has utilized the pros of a powerful family of CNN whose cons are reduced by integrating the self-attention mechanism. The proposed model provides an advanced insight into the way how to enhance the efficiency of the WCNN based on making changes to its hyperparameters. The simulation results have proved that the proposed method obtains a high accuracy level and effective resource consumption compared to the original WCNN by making it deeper in regard to feature extraction of data patterns. Thus, it is promising that machine learning algorithms can handle intricate OPF problems with multiple objective functions and security constraints in the real world and support contingency screening and rapid decision-making.

The method will not only support the system operators in improving the efficiency of the complex power systems but also facilitate the development of intelligence resources and their applications in other aspects of power systems in the future.

REFERENCES

[1] J. Mitra, “Linear optimal power flow system and method,” May 21 2019. US Patent 10,296,988.

[2] R. A. Jabr, “Radial distribution load flow using conic programming,” *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1458–1459, 2006.

[3] W. Wei, J. Wang, N. Li, and S. Mei, “Optimal power flow of radial networks and its variations: A sequential convex optimization approach,” *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2974–2987, 2017.

[4] M. Farivar, C. R. Clarke, S. H. Low, and K. M. Chandy, “Inverter var control for distribution systems with renewables,” in *2011 IEEE international conference on smart grid communications (SmartGridComm)*, pp. 457–462, IEEE, 2011.

[5] E. Dall’Anese, H. Zhu, and G. B. Giannakis, “Distributed optimal power flow for smart microgrids,” *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1464–1475, 2013.

[6] A. Engelmann, Y. Jiang, T. Mühlpfordt, B. Houska, and T. Faulwasser, “Toward distributed opf using aladin,” *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 584–594, 2018.

[7] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, “The qc relaxation: A theoretical and computational study on optimal power flow,” *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3008–3018, 2015.

[8] A. Garces, “A quadratic approximation for the optimal power flow in power distribution systems,” *Electric Power Systems Research*, vol. 130, pp. 222–229, 2016.

[9] W. Liang, Y. Wang, Z. Zhao, B. Liu, and X. Li, “A data-driven ac optimal power flow using extreme learning machine,” in *Journal of Physics: Conference Series*, vol. 2418, p. 012105, IOP Publishing, 2023.

[10] S. S. Guggilam, E. Dall’Anese, Y. C. Chen, S. V. Dhople, and G. B. Giannakis, “Scalable optimization methods for distribution networks with high pv integration,” *IEEE Trans. Smart Grid*, vol. 7, no. 4, pp. 2061–2070, 2016.

[11] A. M. Nakiganda, S. Dehghan, and P. Aristidou, “Comparison of ac optimal power flow methods in low-voltage distribution networks,” in *IEEE PES Innovative Smart Grid Technologies Europe*, pp. 1–5, 2021.

[12] K. Baker, “A learning-boosted quasi-newton method for ac optimal power flow,” *arXiv preprint arXiv:2007.06074*, 2020.

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

[14] A. Velloso and P. Van Hentenryck, “Combining deep learning and optimization for preventive security-constrained dc optimal power flow,” *IEEE Trans. on Power Syst.*, vol. 36, no. 4, pp. 3618–3628, 2021.

[15] X. Pan, T. Zhao, M. Chen, and S. Zhang, “Deepopf: A deep neural network approach for security-constrained dc optimal power flow,” *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 1725–1735, 2020.

[16] X. Pan, “Deepopf: deep neural networks for optimal power flow,” in *The 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 250–251, 2021.

[17] X. Pan, M. Chen, T. Zhao, and S. H. Low, “Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems,” *IEEE Systems Journal*, vol. 17, no. 1, pp. 673–683, 2022.

[18] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, “Dive into deep learning,” *arXiv preprint arXiv:2106.11342*, 2021.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[22] B. Liquet, S. Moka, and Y. Nazarathy, “The mathematical engineering of deep learning,” 2023.

[23] R. D. Zimmerman and C. E. Murillo-Sánchez, “Matpower 6.0 user’s manual,” *Power Systems Engineering Research Center*, vol. 9, 2016.

[24] T. Viet and T. Quan, “An enhancement of multi-objective optimization method in unbalanced power distribution system integrated distributed energy resources,” in *IEEE 6th International Conference on Green Technology and Sustainable Development*, pp. 1017–1023, 2022.

[25] W. Rudin, “Functional analysis . macgraw-hill book company,” *International Edition*, 1973.

[26] R. D. Zimmerman and H. Wang, “Matpower interior point solver mips 1.3 user’s manual,” 2016.

[27] Q. Tran, J. Mitra, and N. Nguyen, “Learning model combining convolutional deep neural network with a self-attention mechanism for ac optimal power flow,” *Electric Power Systems Research*, vol. 231, p. 110327, 2024.

[28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.