Optimal Power Flow Estimation Using One-Dimensional Convolutional Neural Network

Kun Yang, Wei Gao, Rui Fan
Electrical and Computer Engineering
University of Denver
Denver, USA
kun.yang@du.edu

***Abstract*—Optimal power flow (OPF) is an important research topic in power system operation and control decisions. Traditional OPF problems are solved through dynamic optimization with nonlinear programming techniques. For a large power system with large amounts of variables and constraints, the solving process would take a long time. This paper presents a new method to quickly estimate the OPF results using a one-dimensional convolutional neural network (1D-CNN). The OPF problem is treated as a high-dimensional mapping between the load inputs and the generator dispatch decisions. Therefore, through training the neural network to learn the mapping between loads and generator outputs, we can directly predict the OPF results with the load information of a system. In this paper, we built and trained a 1D-CNN to learn the mappings between system loads and generator outputs, and the 1D-CNN model was tested using IEEE 30, 57, 118, and 300 Bus systems. Extensive test and sensitivity study results have validated the effectiveness of using the 1D-CNN to estimate the OPF results.**

***Index Terms*—Optimal power flow, convolutional neural network, machine learning, IEEE test system**

I. INTRODUCTION

Power flow (PF) is one of the most important topics in the power system, as it determines the voltage magnitude and phase angle of each bus under steady-state conditions. In the electricity market, the goal is even more comprehensive with minimizing the total operating cost while maintaining the system generation and demand balance, which can be further utilized by system operators to dispatch the corresponding resources. Nowadays, with the worldwide trend toward deregulation of the electricity market, the objective function of this optimization is becoming more complicated as it involves a lot of nonlinear constraints, such as generation capacity and transmission line limits, etc. Therefore, the optimal power flow (OPF) problem is becoming extremely complicated.

In recent years, a number of dedicated methods have been proposed to solve the OPF problem [1]–[3]. Since the OPF is a non-convex, nonlinear and high-dimensional optimization problem, it is very difficult to be solved directly. Therefore, most methods focus on how to solve the OPF more efficiently in an iterative way. The linear programming (LP) method is a common optimization algorithm and it is widely used in many areas [1]. The LP method can solve an economic dispatch problem efficiently where only linear constraints exist. However, an OPF problem consists lots of nonlinear constraints, which make it difficult for the LP method. Another disadvantage of the LP method is that it may lead to a locally optimal solution [4]. Nonlinear programming (NLP) and quadratic programming (QP) were then proposed as they can deal with the nonlinear constraints [5], [6]. A classical algorithm called primal-dual interior-point was proposed in [7] and utilized maturely in MATPOWER software [8]. However, all these methods are time-consuming for convergence especially under a large-scale system, leading to a limited application in the real power market. The other famous iterative approach - Newton method in PF calculation was also proposed and utilized in [9] and an improved Newton-Raphson algorithm in a multi-energy system was proposed in [10], [11] due to its fast convergence speed, but the convergence is not always guaranteed. As a result, it is challenging to obtain an efficient real-time OPF result using existing optimization methods.

The goal of the OPF problem is to obtain the real and reactive power flows of each branch while minimizing the total operating cost under some system constraints. This indicates that there should be a relationship or mapping between the given system status and the OPF results. If this pattern can be learned efficiently, then the OPF problem becomes a straight-forward prediction rather than solving a complex optimization problem. Machine learning is usually used to learn the mapping between inputs and outputs. In [12], Guha *et al.* proposed to use a machine learning approach to solve the AC-OPF problem. The inputs were the real and reactive loads at each bus, and the outputs were the real power and voltage settings of each generator. Through generating tens of thousands training data, the author successfully trained a multi-layer perceptron (MLP) model to estimate the OPF results. Similar work was also conducted in [13]–[15]. The machine learning methods used in this work are based on traditional neural networks such as the MLP, which does not preserve the translation variance for the input data and may not work well for complex systems. In many research areas, the convolutional neural network (CNN) has shown its huge power in solving some formidable problems which are complex or even impossible using conventional methods, which will be continuously innovated with more state-of-the-art structures. With enough data trained by CNN, the mapping between network input and output can be learned effectively without considerations of the complicated character of the original problem.

In this paper, we propose a one-dimensional CNN (1D-CNN) based OPF estimation approach to solve the issue of time-consuming in conventional methods. The input of the 1D-CNN model is the load information at each branch, and the outputs are the generator's real and reactive power. The 1D-CNN consisted of three convolutional layers, where convolutional function, pooling function, and nonlinear activation functions were engaged. The output layer of the 1D-CNN is a dense layer that was used to predict the estimated generator power. To train the 1D-CNN model, we automatically generated tens of thousands of training data. The proposed 1D-CNN approach was tested on IEEE 30, 57, 118, and 300 Bus test systems. Extensive simulation results have validated the effectiveness of using the 1D-CNN to estimate the OPF results.

The rest of the paper is organized as follows. Section II briefly reviews the optimal power flow problems. Section III, the proposed 1D-CNN approach is explained. Section IV shows the experimental results. Finally, the conclusions of this paper are given in Section V.

II. Optimal Power Flow Problem

As early as 1962, J. Carpentier introduced a nonlinear programming method to solve the OPF problem [4]. Voltage constraints and other operational constraints were first introduced for the OPF problem.

The general optimal power flow problem has formulation:

$$min\ f(u,x) \tag{1}$$

$$s.t.\ \ g_p(u,x) = 0, p = 1,2,...,n_p \tag{2}$$

$$s.t.\ \ h_m(u,x) \leq 0, m = 1,2,...,n_m \tag{3}$$

where $f(x,u)$ is the objective function for minimizing the overall cost. x and u are the states and control variable of the system. To solve the OPF problem, the traditional method is to set up the Lagrangian:

$$L(x,u,\lambda) = f(u,x) + \sum_{i=1}^{m} \lambda_i g_i(x,u) \tag{4}$$

Where λ is the vector of the Lagrange multipliers. Then we make use of the fact that a necessary condition for a minimum is that:

$$\bigtriangledown L(x,u,\lambda) = 0 \tag{5}$$

For simplicity define

$$z = [x,u,\lambda]^T \tag{6}$$

The goal then is to solve for a z^* that minimizes the Lagrangian and hence solves:

$$\bigtriangledown L(z^*) = 0 \tag{7}$$

To solve this equation, first express $\bigtriangledown L(z^*)$ with is Taylor expansion about some known(and presumably non-optimal) point z:

$$\bigtriangledown L(z^*) = \bigtriangledown L(z) + \bigtriangledown^2 L(z) \triangle z + higher\ order\ terms \tag{8}$$

with $\triangle z = z^* - z$. If we ignore the higher order terms, we can directly solve for $\triangle z$

$$\triangle z = -\left[\bigtriangledown^2 L(z)\right]^{-1} \bigtriangledown L(z) \tag{9}$$

Since the higher order terms were ignored, $z + \triangle z$ is only an approximation of z^*. Hence we need to slove iteratively:

$$z^{(k+1)} = z^{(k)} - \left[\bigtriangledown^2 L\left(z^{(k)}\right)^{-1}\right] \bigtriangledown L\left(z^{(k)}\right) \tag{10}$$

To avoid having to continually write the gradient symbol, define:

$$h(z) = \bigtriangledown L(z) \tag{11}$$

$$W(z) = \bigtriangledown^2 L(z) \tag{12}$$

III. Proposed Approach

In the traditional calculation method of optimal power flow, an initial guess is made for state variables and decision variables, and then the optimal solution is obtained through iteration. The disadvantage of this solution method is that with the increase of the number of buses and generators, the amount of calculation will become huge. In the real world, grid operators must solve OPF for the entire electric grid every 5 minutes, considering the changes in the power generation and loads. Therefore, it is very challenging to fulfill the computing requirement in such a short time.

Therefore, in this paper, we propose to use a machine learning based approach to directly predict the OPF results without solving the optimization problem. The objective is to train a neural network to learn the mapping between the system information (as inputs) and the OPF results (as outputs). The machine learning method used in this study is the 1D-CNN approach.

A. 1D-CNN Model

CNN is a deep neural network model that is widely used in the fields of image and natural language processing. In 1999, Lecun [16] proposed a gradient-based back propagation algorithm for document recognition. In that neural network, the convolutional layer played a crucial role. CNN is a typical supervised learning method. It is suitable for identifying simple patterns in the data, and then use these simple patterns to generate more complex patterns at a higher level With the increasing computing power, some large CNN networks began to show great advantages in the field of images, therefore most of the existing applications of CNN are two-dimensional. However, CNN can also be applied to one-dimensional data, such as time-series waveform or vector inputs. In the OPF problem, because the power system load information can be treated as a vector, using 1D-CNN could be a feasible approach.

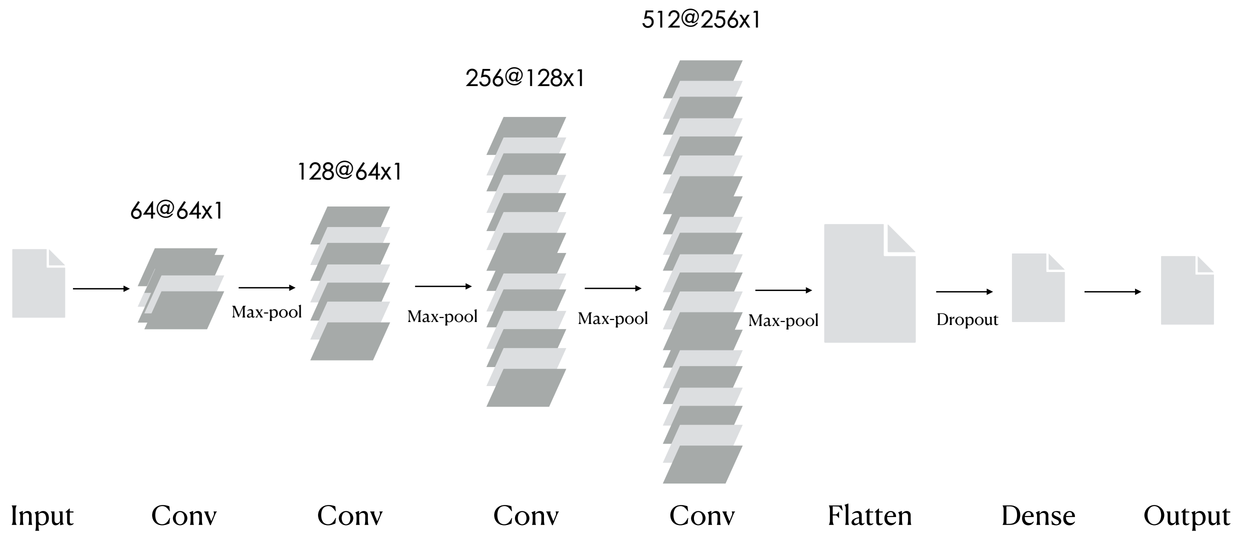

Fig. 1: Proposed 1D-CNN model for OPF estimation.

The proposed 1D-CNN model is shown in Fig. 1. As is known to all, CNN is often used for two-dimensional or higher-dimensional data processing, for the one-dimensional convolution layer, the biggest difference between the main convolution layer and the two-dimensional convolution layer is that the convolution kernel of the one-dimensional convolution layer becomes one-dimensional. That is to say, local one-dimensional sequence segments(i.e., sub-sequences) are extracted from the sequence according to a certain size window, and then dot product with weights, then output is a part of the new sequence. Then one-dimensional pooling layer was added to the model, one-dimensional pooling layer extracts one-dimensional sequence segments from input and outputs their maximum value or average value to reduce the length of one-dimensional input. The purpose of the pooling layer is to blur the result of convolution, summarize the statistical features in the local region and the over-fitting is avoided by dimensionality reduction. The "he uniform" initialization was adopted in every convolutional layer, which helps find a good variance for the distribution from the initially drawn parameters. This variance is adapted to the activation function used and is derived without explicitly considering the type of the distribution [17]. In the process of network training, the gradient is easy to disappear (the gradient is extremely close to 0) and the gradient explosion (the gradient is extremely large), resulting in most of the gradients obtained by back propagation are ineffective or counterproductive. The researchers hope to have a good weight initialization method: when the network is propagated forward or back, the output of the convolution and the gradient of the forward transmission are relatively stable.

Using the proposed 1D-CNN model in Fig. 1, the OPF problem was turned into a regression task, where a mapping between the input and output is the key issue. In other words, we were estimating the OPF results (generator output $P_{G,i}$ and $Q_{G,i}$) from load information ($P_{L,i}$ and $Q_{L,i}$). The remaining voltage magnitude and angles can be directly solved with the estimated OPF results, thus they are not included here. In conclusion, the input to the 1D-CNN model are the real and reactive loads at each bus $\boldsymbol{X} = [P_{L,1}, ..., P_{L,N}, Q_{L,1}, ...Q_{L,N}]$, and the output are the generator real and reactive power $\boldsymbol{Y} = [P_{G,1}, ..., P_{G,M}, Q_{G,1}, ...Q_{G,M}]$. The objective of the 1D-CNN is to learn the mapping $\boldsymbol{f} : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ that minimize the mean-squared error between the predicted generator output $\hat{\boldsymbol{Y}}$ and the optimal generator setting (true value) $\boldsymbol{Y}$.

B. Test IEEE Systems and Data

To train and test the proposed 1D-CNN approach, we have chosen several test systems with small, medium, and large scales. Specifically, the IEEE 30, 57, 118, and 300 Bus test systems were used in this study. For each IEEE test system with a load distribution of x, we randomly sampled the load distribution x' with a range of $[(1-\delta)x, (1+\delta)x]$. To test the impact of data variation range (δ) on the accuracy of OPF estimation, we specifically selected two different δ to be 10% and 30%. In other words, there were two data-sets for each IEEE test system, one has a range of 90% to 110% of its original load distribution, the other has a range of 70% to 130% of its original load distribution. In addition, for each of the IEEE test systems (and each 10% and 30% data variation), we generated around 200,000 solved OPF cases, which gave us enough data for training and testing the 1D-CNN performance. Note that not all the data were used to train the 1D-CNN model, and the training data-set and testing data-set were separate. In fact, when training the 1D-CNN model for each IEEE test system, we tested the impact of different numbers of training data on the accuracy of the proposed approach. Details were included in the following section.

IV. Results and Discussions

To test the performance of the proposed 1D-CNN approach, a traditional machine learning method was also implemented for comparison. The traditional machine learning method used a three-layer MLP model with 128, 64, 32 neurons for the respective layers. The average absolute mismatches of the OPF results were used as the criterion for evaluating the performances of the proposed 1D-CNN and transitional machine learning methods.

A. Test Result of 1D-CNN and MLP

In the first study, we trained the 1D-CNN and MLP models with 180,000 data for each IEEE test system. These data had a variation range δ of 10%. An example of the training losses for the 1D-CNN and MLP for the IEEE 57-Bus system is shown in Fig. 2. From the figure we could see that at the beginning the losses were very high for both models, however, the losses decreased as the training epochs increased. After a few hundred of training, the losses stabilized, which meant the training was completed. It was noticed that the 1D-CNN had a lower loss than the MLP method. Therefore, the 1D-CNN method provided better OPF estimation results than the MLP method.

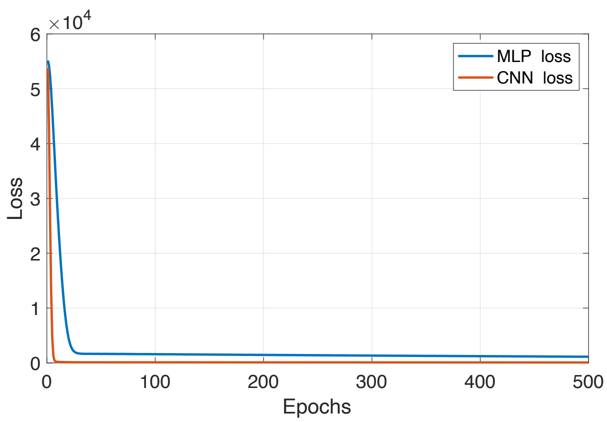

Fig. 2: Training losses for the 1D-CNN and MLP methods.

For each IEEE test system, separate 2,000 data were used to test the performance of the well-trained 1D-CNN and MLP models. The average absolute mismatches of the OPF results for the 1D-CNN and MLP predictions were summarized in Table I. From the comparison, we could see that the 1D-CNN had a higher accuracy in predicting the OPF results than the MLP method, which was in consistent with the training results showed in Fig. 2. The MLP method worked well for small systems, but the mismatches were high for the large systems. In contrast, the average mismatches for the 1D-CNN method stayed low regardless of the size of the system. Therefore, the proposed 1D-CNN approach could effectively predict the OPF results.

B. 10% Versus 30% Variation

The previous results were based on OPF results with a 10% variation range of the IEEE test system. In this study, we would like to test the impact of load variation ranges on the performance of the 1D-CNN and MLP methods. Specifically, we compared the OPF prediction results of a 10% variation range with a 30% variation range for each IEEE test system. The number of training data for each IEEE test system and each variation range was still 180,000 pieces. Similarly, separate 2,000 data were used to test the performance of the well-trained 1D-CNN and MLP models for each case. The results of the comparison are shown in Table II.

TABLE I: 1D-CNN and MLP result comparison

Case ID	CNN MR	MLP MR
case30	1.16%	2.25%
case57	0.29%	2.57%
case118	0.91%	3.15%
case300	1.81%	7.26%

TABLE II: Results of 10% vs 30% variation using 1D-CNN and MLP methods

Case ID	CNN MR	MLP MR
case30_10%	1.16%	2.25%
case30_30%	2.46%	3.57%
case57_10%	0.29%	2.57%
case57_30%	0.79%	5.14%
case118_10%	0.91%	3.15%
case118_30%	1.25%	4.25%
case300_10%	1.81%	7.26%
case300_30%	2.55%	11.79%

Note that the Case ID contained the detailed information of the test case. For example, "case118_30%" means the test system is IEEE 118-Bus, and the load variation is from 70% to 130%. From Table II we could see that the load variation range indeed had an impact on the accuracy of the OPF prediction results. The average mismatch of a 30% variation range was higher than the mismatch of a 10% variation range for all the IEEE test systems, no matter whether the machine learning methods were used. The results were as expected because both 1D-CNN and MLP methods had to learn a wider range of outputs for the 30% variation situation. However, compared with the MLP results, the 1D-CNN still presented a much higher OPF prediction accuracy. The average mismatches for the 1D-CNN method were below 3% in all the cases. Therefore, the proposed 1D-CNN method could be used to accurately predict OPF results.

C. Sensitivity Study on Training Data Size

As is known to everyone, machine learning methods require a large amount of training data to achieve a good performance. In this study, we investigated the impact of training data size on the performance of the proposed 1D-CNN method. Specifically, for each IEEE test system, we randomly selected 20,000, 40,000, 60,000, 80,000, 100,000, 120,000, 140,000, 160,000, and 180,000 pieces of training data to train the 1D-CNN model. After the 1D-CNN model was well trained, similarly, separate 2,000 data were used to test the performance. Again, we used the average mismatch to evaluate the performance. The results of the 1D-CNN prediction mismatches for IEEE 30-, 57-, 118-, and 300-Bus systems were shown in Fig. 3 to Fig. 6, respectively.

For either 10% or 30% load variation, we could see an obvious decline of the mismatch when the training data size increased for any IEEE test system. The result was reasonable as the increase of training data size would enhance the performance of machine learning methods. We also noticed that the mismatch of a 30% load variation was still higher than the 10% load variation situation, which was consistent with the conclusions we found in Table II. We also noted that the results of the IEEE 30-Bus system were interesting, as its mismatch was very high (around 10%) when the training data size is small, while the mismatch could be significantly decreased with more training data. If we set a 2% mismatch as the goal, the required training data sizes were 150,000 for the 30-Bus system, 20,000 for the 57-Bus system, 100,000 for the 118-Bus system, and 140,000 for the 300-Bus system. The general trend was that the large the test system, the more training data were required to achieve good performance for the 1D-CNN method.

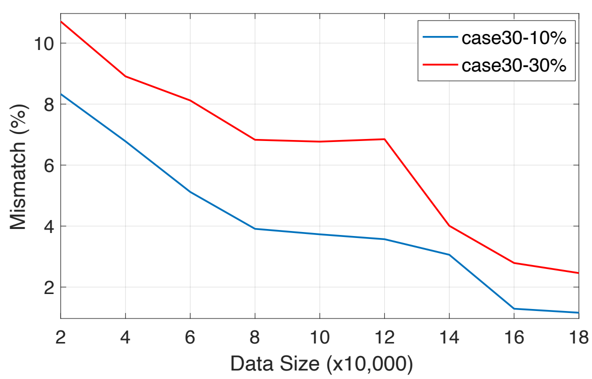

Fig. 3: Sensitivity studies for IEEE 30-Bus system with different training data size.

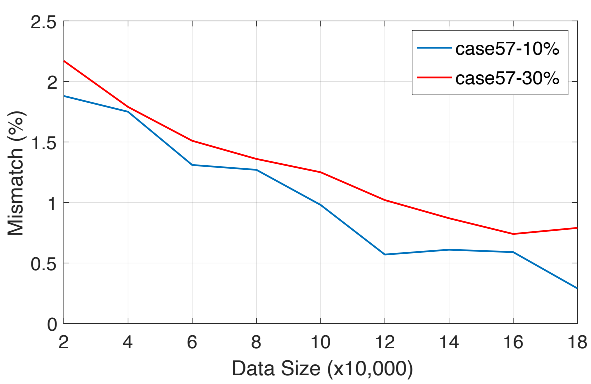

Fig. 4: Sensitivity studies for IEEE 57-Bus system with different training data size.

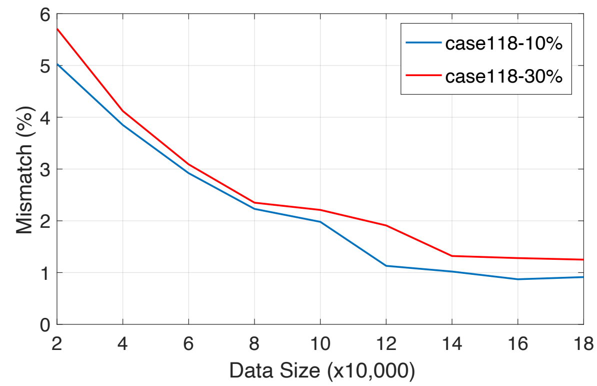

Fig. 5: Sensitivity studies for IEEE 118-Bus system with different training data size.

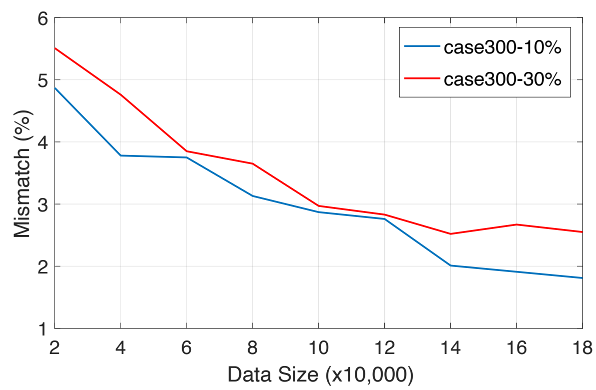

Fig. 6: Sensitivity studies for IEEE 300-Bus system with different training data size.

V. Conclusions

In this paper, we proposed a novel 1D-CNN based approach for predicting the OPF results. Compared with traditional machine learning method such as the MLP, the proposed 1D-CNN approach have been proven to be more accurate. The superiority of the 1D-CNN method was validated using IEEE 30-, 57-, 118-, and 300-Bus with different load variations. Test results find that the increase in load variation will decrease the 1D-CNN performance. A sensitive study has also shown that the accuracy of predicting OPF would significantly increase with more training data. The general trend was that the large the test system, the more training data were required to achieve good performance for the 1D-CNN method.

In future work, we would explore methods to increase the performance of the 1D-CNN approach in situations with larger load variation and fewer training data.

References

[1] O Alsac, J Bright, M Prais, and B Stott. Further developments in lp-based optimal power flow. *IEEE Transactions on Power Systems*, 5(3):697–711, 1990.

[2] David I Sun, Bruce Ashley, Brian Brewer, Art Hughes, and William F Tinney. Optimal power flow by newton approach. *IEEE Transactions on Power Apparatus and systems*, (10):2864–2880, 1984.

[3] Mohammad A Abido. Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems*, 24(7):563–571, 2002.

[4] Line Roald and Göran Andersson. Chance-constrained ac optimal power flow: Reformulations and efficient algorithms. *IEEE Transactions on Power Systems*, 33(3):2906–2918, 2017.

[5] D Pudjianto, S Ahmed, and G Strbac. Allocation of var support using lp and nlp based optimal power flows. *IEE Proceedings-Generation, Transmission and Distribution*, 149(4):377–383, 2002.

[6] Gerald F Reid and Lawrence Hasdorff. Economic dispatch using quadratic programming. *IEEE Transactions on Power Apparatus and Systems*, (6):2015–2023, 1973.

[7] Rabih A Jabr. A primal-dual interior-point method to solve the optimal power flow dispatching problem. *Optimization and Engineering*, 4(4):309–336, 2003.

[8] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on power systems*, 26(1):12–19, 2010.

[9] Mohamed Ebeed, Salah Kamel, and Francisco Jurado. Optimal power flow using recent optimization techniques. In *Classical and recent aspects of power system optimization*, pages 157–183. Elsevier, 2018.

[10] Yushuai Li, David Wenzhong Gao, Wei Gao, Huaguang Zhang, and Jianguo Zhou. Double-mode energy management for multi-energy system via distributed dynamic event-triggered newton-raphson algorithm. *IEEE Transactions on Smart Grid*, 11(6):5339–5356, 2020.

[11] Yushuai Li, David Wenzhong Gao, Wei Gao, Huaguang Zhang, and Jianguo Zhou. A distributed double-newton descent algorithm for cooperative energy management of multiple energy bodies in energy internet. *IEEE Transactions on Industrial Informatics*, 17(9):5993–6003, 2020.

[12] Neel Guha, Zhecheng Wang, Matt Wytock, and Arun Majumdar. Machine learning for ac optimal power flow. *arXiv preprint arXiv:1910.08842*, 2019.

[13] Jubeyer Rahman, Cong Feng, and Jie Zhang. Machine learning-aided security constrained optimal power flow. In *2020 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2020.

[14] Ferdinando Fioretto, Terrence WK Mak, and Pascal Van Hentenryck. Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 630–637, 2020.

[15] Xiang Pan, Minghua Chen, Tianyu Zhao, and Steven H Low. Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems. *arXiv preprint arXiv:2007.01002*, 2020.

[16] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.