Data-Driven Optimal Solver for Coordinating a Sustainable and Stable Power Grid

Junfei Wang
Dept. of Electrical Engineering and Computer Science
York University
Toronto, ON
jwang@yorku.ca

Pirathayini Srikantha
Dept. of Electrical Engineering and Computer Science
York University
psrikan@yorku.ca

Abstract

With today's pressing climate change concerns, the widespread integration of low-carbon technologies such as sustainable generation systems (e.g. photovoltaics, wind turbines, etc.) and flexible consumer devices (e.g. storage, electric vehicles, smart appliances, etc.) into the electric grid is vital. Although these power entities can be deployed at large, these are highly variable in nature and must interact with the existing grid infrastructure without violating electrical limits so that the system continues to operate in a stable manner at all times. In order to ensure the integrity of grid operations while also being economical, system operators will need to rapidly solve the optimal power flow (OPF) problem in order to adapt to these fluctuations. Inherent non-convexities in the OPF problem do not allow traditional model-based optimization techniques to offer guarantees on optimality, feasibility and convergence. In this paper, we propose a data-driven OPF solver built on information-theoretic and semi-supervised machine learning constructs. We show that this solver is able to rapidly compute solutions (i.e. in sub-second range) that are within 3% of optimality with guarantees on feasibility on a benchmark IEEE 118-bus system.

1 Introduction

The modern grid is undergoing unprecedented changes that are mainly motivated by climate change concerns. In an effort to significantly reduce carbon footprint from the electricity sector, system operators aim to widely integrate sustainable generation systems and accommodate smart consumer appliances (1; 2). However, as these are highly variable in nature and if not properly coordinated, inefficiencies are inevitable and the integrity of grid operations may be compromised (3). The system operator will need rapidly and frequently solve the optimal power flow (OPF) problem to account for the changing grid parameters so that these power entities operate economically while maintaining system limits. The main challenge in achieving this lies in the non-convexities introduced by power flow constraints that capture the electrical interdependencies and limits of the grid in the OPF formulation. It is well known that solving a non-convex problem directly is NP-hard (4).

As such, existing literature on solving the OPF problem in a tractable manner can be loosely categorized into model-based and data-driven methods. With model-based techniques, one approach is to employ convex relaxations (5) to eliminate non-convex constraints and then solve the relaxed problem

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2021.

directly. As the grid is operating close to its limits, these relaxations can result in infeasibilities and violations of grid constraints. Heuristic approaches such as (6; 7; 8) are commonly employed to solve the original non-convex OPF. However, these cannot guarantee optimality, convergence and/or feasibility of the computed solutions. On the other hand, data-driven techniques such as (9; 10; 11; 12; 13) have been proposed in the literature. One subset of these approaches utilize supervised learning constructs (9; 10) where labelled datasets containing inputs (i.e. grid parameters) and corresponding outputs (i.e. optimal setpoints of active power entities) are utilized to train the proposed solvers. The main difficulty with this approach is curating the labelled datasets as generating the optimal outputs for various input combinations via traditional solvers is time-consuming, not tractable and may not be actually optimal or feasible. Another set of data-driven approaches (11; 12; 13) utilize both machine learning constructs and traditional solvers where partial solutions are computed by machine learning models and the remaining solutions are "completed" by traditional solvers in order to guarantee feasibility. Since traditional solvers are looped into the computational process, these can lead to computational delays and non-convergence issues.

In this paper, our contributions are four-fold: 1) We propose a data-driven approach based on semi-supervised learning to build an OPF solver where the training dataset needs to contain only feasible outputs - not optimal outputs (this data is readily available as the grid always operates in feasible regions that are not necessarily optimal); 2) We combine information theoretic constructs with machine learning model to implicitly extract information about the costs of various solutions; 3) We leverage on the generator and discriminator modules in the generative adversarial networks (GANs) to synthesize feasible outputs and check the feasibility of these; and 4) We demonstrate the efficacy of our proposal on a benchmark IEEE 118 bus system. With a fast OPF solver like ours, once trained, inferencing can take place very quickly and this is conducive to allowing for system operators to rapidly and economically react to changes in the electric grid introduced by highly variable sustainable energy systems.

2 Methodology

The proposed machine learning model is composed of two main modules: 1) Synthesis of feasible solutions for specific inputs; and 2) Implicit learning of solution costs. Various components of the proposed model are presented in Fig. 1. In the following, these components will be detailed.

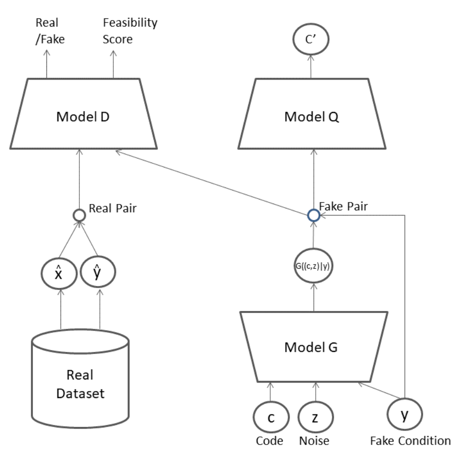

Figure 1: Latent code c and cost of solutions.

In order to synthesize feasible solutions for various grid parameters that will serve as inputs, the continuous conditional GAN (CCGAN) will be utilized. The reasons for using this machine learning model are two-fold. First, as the grid parameters are continuous inputs and the original GAN model (14) does not accept data inputs, we treat the inputs of the OPF problem as continuous conditions. The second reason is that the general architecture of the GAN consists of a discriminator component D that can be leveraged to verify the feasibility of the solutions synthesized. Training datasets containing various grid parameters and corresponding *feasible* operational setpoints and grid states are utilized. After training, the generator G in the CCGAN will learn the conditional distribution of the feasible outputs for the corresponding inputs and the discriminator aims to distinguish outputs from the actual training dataset versus the synthetic ones generated by the generator. During training, the generator and discriminator will aim to maximize the losses of one another. The loss function $\mathcal{L}_{\mathcal{CCGAN}}$ pertaining to the CCGAN is a combination of the loss function presented in reference (15) and Wasserstein loss with gradient penalty(16) that allows for the stable training of the system.

$$\mathcal{L}_{CCGAN} = \mathop{\mathbb{E}}_{y\sim P_d(y|x)}[D(\tilde{y}|x)] - \mathop{\mathbb{E}}_{z\sim P_n}[D(G(z,c|x))] + \delta \mathop{\mathbb{E}}_{z\sim P_n}(||\nabla_{G(z,c)|x} D(G((z,c)|x))|| - 1)^2 + \lambda \mathop{\mathbb{E}}_{x,z,c} \nabla_x ||G((z,c)|x)||$$

where z is an M dimensional random variable drawn from the Gaussian distribution P_n (this allows for variability in the outputs synthesized by G), input x is the input to the OPF problem, y is the output that is feasible for the grid under the conditions imposed by x, c is a single-dimensional latent code correlated to the cost of y, G synthesizes feasible outputs to the OPF problem, D outputs a probability corresponding to whether its input (i.e. y) is real or synthesized for the conditions imposed by x along with feasibility scores for each constraint in the OPF problem, P_d is the real conditional distribution of the output y given x, $G((c,z)|x)$ is y, $\tilde{y}$ is set to either the real $\hat{y}$ or synthetic y output with equal probability, and δ and λ are non-negative weights. The first, second term and third terms reflect the Wasserstein loss and the last term reflects the gradient penalty. Once this module is trained, a feasible output will be synthesized by G for inputs $z,\ c$ and x.

To further improve the feasibility of the outputs from G, a physical loss term $\mathcal{L}_{ph}$ is added.

$$\mathcal{L}_{Ph} = \mathbb{E}[||H(x,\tilde{y}) - K||^2] + \mathop{\mathbb{E}}_{i\in B}[\max(0, R_i(x,\tilde{y}) - P_i)] + \nu(\mathbb{E}[|\ ||H(x,\tilde{y}) - K||^2 - D_1(\tilde{y}|x)|] + \sum_{i\in B} \mathbb{E}[|\ ||R_i(x,\tilde{y}) - P_i||^2 - D_i(\tilde{y}|x)|])$$

where $\tilde{y}$ is the output from G given x. $H(x,\tilde{y}) = K$ are equality constraints (e.g. power balance constraints) and $R_i(x,\tilde{y}) \leq P_i$ are B inequality constraints (e.g. current limits, voltage limits, etc.) in the OPF problem. Detailed formulation of the OPF problem is listed in the Appendix. The first and second terms impose penalties for violating these constraints. The third and fourth term ensure that the feasibility scores outputted by D captures the degree of violations of equality and inequality constraints. ν is a positive weight term.

The final component $\mathcal{L}_C$ of the loss function aims to linearly correlate the latent code c with the cost of solutions generated by G.

$$\mathcal{L}_C = \mathop{\mathbb{E}}_{c\sim P_c, z\sim P_{noise}} ||(Q(G(c,z|x))) - c||^2$$

where Q is a module that is added to the CCGAN in the proposed ML model. This loss term encapsulates the learning of mutual information pertaining to the cost of the solution with the latent code c. This is an approximation of mutual information as outlined in reference (17). P_c is a uniform distribution used to sample c.

Now, all loss terms are combined as follows to form the complete loss function $\mathcal{L}$.

$$\mathcal{L} = \mathcal{L}_{CCGAN} + \mu\mathcal{L}_{Ph} + \eta\mathcal{L}_C$$

μ and η are non-negative weights assigned to various components of the loss function. Implementation details of the components G, D and Q are presented in the Appendix.

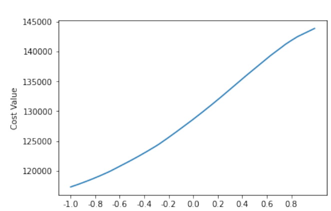

Figure 2: Latent code c and cost of solutions.

Once the proposed ML model is trained, a systematic search of samples from G using the latent code c is conducted to find solutions that are associated with the low costs while maintaining constraint feasibility as depicted by the feasibility scores from D. Since Q is designed to correlate actual cost of solution $\tilde{y}$ with c in a nearly linear manner where c takes values in the interval $[-1, 1]$, the search for c that will result in low cost will take place in the lower end of the interval $[-1, -\epsilon]$ where ϵ is a small positive value. Fig. 2 illustrates the relationship between c and solutions generated. The search is designed to select low values of c and randomly sampled z that result in $\tilde{y}$ that are feasible as indicated by D. With our proposal, knowledge of grid structure and cost of power generation will not be necessary during inferencing.

3 Result

The proposed model was trained and tested on benchmark IEEE 118-bus power system. The training dataset has been generated using MatPower's load flow analysis that computes *feasible* outputs. In order to compare the degree of optimality of our proposal, we use MatPower's OPF solver which uses the interior point method (IPM) and semi-definite programming (SDP) from Sedumi (18) for solving the original OPF. The average of time entailed in computing the solutions with the IPM solver was around 10 seconds. With the SDP solver there were convergence issues and required almost 100 seconds on average for the cases that did converge. Table 1 lists the results obtained with our proposal. The first column lists the number of samples queried from the generator. The second column quantifies the gap of the computed solution from the optimal solution determined by the SDP solver from Sedumi. The next column lists the gap of the computed solution from the optimal solution computed by the IPM solver from Matpower. The solutions of the solutions computed by IPM and SDP were comparable. The fourth column lists the average time entailed in computing the solution with our proposed model after training. The last column indicates whether or not the final solution is feasible from the perspective of load flow analysis from Matpower.

It is clear from these results that the time entailed in computing a solution with our model is a fraction of a second. Moreover, the optimality gap is very small (within 3%). All solutions computed are also feasible. These are superior qualities of an OPF solver. With the speed supported by the proposed algorithm, perturbations and unexpected fluctuations from renewable and sustainable energy systems can be quickly accounted for by system operators. The outputs can be readily utilized by the system operators to ensure that the grid remains operational within stable regimes while maintaining efficient operations.

Table 1: Comparison with traditional solvers for IEEE 118-bus system.

M	Gap to SDP(%)	Gap to IPM(%)	Time(s)	Feasibility
50	3.97±0.49	3.79±0.47	0.12	✓
100	3.87±0.48	3.27±0.39	0.21	✓
200	3.63±0.45	3.44±0.42	0.27	✓
500	3.52±0.44	2.97±0.38	0.29	✓
1000	3.18±0.38	3.00±0.37	0.34	✓
3000	2.90±0.35	2.70±0.33	0.56	✓
5000	2.92±0.35	2.71±0.35	0.65	✓

4 Conclusions

In this paper, we present a novel OPF solver that is based on machine learning constructs, information theoretic constructs and domain knowledge. This solver has been demonstrated to be extremely fast especially when compared with traditional solvers like SDP and IPM while guaranteeing feasibility. Furthermore, the solutions are tuned to be near-optimal with the strategic selection of the latent code that implicitly encodes details about the cost of the solutions synthesized. This solver does not require datasets containing input to optimal output pairs. This allows system operators to use abundantly available datasets containing feasible grid states and measurements to train the OPF solver. The speed of inferencing allows grid operators to practically account for the fluctuations introduced by sustainable energy sources and thus the high proliferation of these devices. This directly supports climate change goals set by policy makers, system operators and the public in general.

References

[1] Farhat, M., Kamel, S., Atallah, A. M., & Khan, B. (2021). Optimal power flow solution based on jellyfish search optimization considering uncertainty of renewable energy sources. IEEE Access, 9, 100911-100933.

[2] Khan, B., & Singh, P. (2017). Optimal power flow techniques under characterization of conventional and renewable energy sources: A comprehensive analysis. Journal of Engineering, 2017.

[3] Chamanbaz, M., Dabbene, F., & Lagoa, C. AC optimal power flow in the presence of renewable sources and uncertain loads, 2017.

[4] Lavaei, J., & Low, S. H. (2011). Zero duality gap in optimal power flow problem. IEEE Transactions on Power systems, 27(1), 92-107.

[5] Low, S. H. (2014). Convex relaxation of optimal power flow—Part I: Formulations and equivalence. IEEE Transactions on Control of Network Systems, 1(1), 15-27.

[6] Osman, M. S., Abo-Sinna, M. A., & Mousa, A. A. (2004). A solution to the optimal power flow using genetic algorithm. Applied mathematics and computation, 155(2), 391-405.

[7] Bakirtzis, A. G., Biskas, P. N., Zoumas, C. E., & Petridis, V. (2002). Optimal power flow by enhanced genetic algorithm. IEEE Transactions on power Systems, 17(2), 229-236.

[8] Abido, M. A. (2002). Optimal power flow using particle swarm optimization. International Journal of Electrical Power & Energy Systems, 24(7), 563-571.

[9] Fioretto, F., Mak, T. W., & Van Hentenryck, P. (2020, April). Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 01, pp. 630-637).

[10] Mahto, D. K., Saini, V. K., Mathur, A., Kumar, R., & Saxena, A. (2021, October). Data Driven Approach for Optimal Power Flow in Distribution Network. In 2021 5th International Conference on Information Systems and Computer Networks (ISCON) (pp. 1-6). IEEE.

[11] Jia, Y., & Bai, X. (2021, August). A CNN approach for optimal power flow problem for distribution network. In 2021 Power System and Green Energy Conference (PSGEC) (pp. 35-39). IEEE.

[12] Pan, X., Chen, M., Zhao, T., & Low, S. H. (2020). Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems.

[13] Donti, P. L., Rolnick, D., & Kolter, J. Z. (2021). Dc3: A learning method for optimization with hard constraints.

[14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. Generative adversarial networks. Communications of the ACM, 63(11), 139-144, 2020

[15] Zheng, Y., Zhang, Y., & Zheng, Z. (2021). Continuous Conditional Generative Adversarial Networks (cGAN) with Generator Regularization.

[16] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

[17] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems, 29.

[18] Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization methods and software, 11(1-4), 625-653.

[19] Ghofrani, M., Hassanzadeh, M., Etezadi-Amoli, M., & Fadali, M. S. (2011, August). Smart meter based short-term load forecasting for residential customers. In 2011 North American Power Symposium (pp. 1-5). IEEE.

[20] Dong, X., Qian, L., & Huang, L. (2017, February). Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach. In 2017 IEEE international conference on big data and smart computing (BigComp) (pp. 119-125). IEEE.

[21] Xia, M., Shao, H., Ma, X., & de Silva, C. W. (2021). A stacked GRU-RNN-based approach for predicting renewable energy and electricity load for smart grid operation. IEEE Transactions on Industrial Informatics, 17(10), 7050-7059.

[22] Wang, H., Murillo-Sanchez, C. E., Zimmerman, R. D., & Thomas, R. J. (2007). On computational issues of market-based optimal power flow. IEEE Transactions on Power Systems, 22(3), 1185-1193.

[23] Huang, W., & Chen, M. DeepOPF-NGT: Fast No Ground Truth Deep Learning-Based Approach for AC-OPF Problems.

[24] Grainger, J. J. (1999). Power system analysis. McGraw-Hill.

[25] https://www.iit.edu/ [Accessed: 27-July-2022]

[26] https://www.eia.gov/ [Accessed: 27-Jan-2022]

[27] Parajuli, A. (2016). A statistical analysis of wind speed and power density based on Weibull and Rayleigh models of Jumla, Nepal. Energy and Power Engineering, 8(7), 271-282.

[28] Afzaal, M. U., Sajjad, I. A., Awan, A. B., Paracha, K. N., Khan, M. F. N., Bhatti, A. R.,& Tlili, I. (2020). Probabilistic generation model of solar irradiance for grid connected photovoltaic systems using weibull distribution. Sustainability, 12(6), 2241.

[29] Reddy, S. S. (2017). Optimal power flow with renewable energy resources including storage. Electrical Engineering, 99(2), 685-695.

[30] Syed, M. S., Chintalapudi, S. V., & Sirigiri, S. (2021). Optimal power flow solution in the presence of renewable energy sources. Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 45(1), 61-79.

Appendix

A. Description of Optimal Power Flow

The OPF problem formulation $\mathcal{P}_{OPF}$ based on the bus injection model is presented here.

$$\mathcal{P}_{OPF} : \min_{P_{G_i}} \sum_{i \in \mathcal{N}} C_g(P_{G_i})$$

$$\text{s.t. } \forall\, i \in \mathcal{N} : \sum_{j \in \mathcal{N}} Re\{V_i(V_i^* - V_j^*)y_{ij}\} = P_{G_i} - P_{D_i} \quad \textbf{[C1]}$$

$$\sum_{j \in \mathcal{N}} Im\{V_i(V_i^* - V_j^*)y_{ij}\} = Q_{G_i} - Q_{D_i} \quad \textbf{[C2]}$$

$$P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max} \quad \textbf{[C3]}$$

$$Q_{G_i}^{min} \leq Q_{G_i} \leq Q_{G_i}^{max} \quad \textbf{[C4]}$$

$$V^{min} \leq |V_i| \leq V^{max} \quad \textbf{[C5]}$$

$$\phi^{min} \leq \phi_i \leq \phi^{max} \quad \textbf{[C6]}$$

$$|V_i(V_i^* - V_j^*)y_{ij}^*| \leq S_{i,j}^{max} \; \forall\, j \in \mathcal{N}_i \quad \textbf{[C7]}$$

[C1] and [C2] reflect the real and reactive power balance on bus i. [C3]-[C7] reflect limits on real and reactive power generation, bus voltage magnitude, phase angles, and branch flows. Constraints $[C1]$, $[C2]$, $[C5]$, and $[C7]$ are non-convex.

Table 2: Parameters and Variables in OPF Problem

Name	Notation	Dimensions
Total number of buses and generator	N and G	$\mathbb{R}$
Set of all buses	$\mathcal{N}$	$\mathbb{R}^N$
Set of buses that are incident to bus i	N_i	$\mathbb{R}^{\lvert N_i \rvert}$
Objective function	C_g (Non-decreasing)	$\mathbb{R}$
Real and reactive power generated by generator i	P_{G_i}, Q_{G_i}	$\mathbb{R}$
Real and reactive power demand on bus i	P_{D_i}, Q_{D_i}	$\mathbb{R}$
Nodal admittance between bus i and j	y_{ij}	$\mathbb{C}$
Voltage at bus i	V_i	$\mathbb{C}$
Lower and upper limits of real and reactive power on bus i	$P_{G_i}^{min}, P_{G_i}^{max}, Q_{G_i}^{min}, Q_{G_i}^{max}$	$\mathbb{R}$
Lower and upper limits of bus voltage magnitude	V^{min}, V^{max}	$\mathbb{R}$
Lower and upper limits of bus angles	ϕ^{min}, ϕ^{max}	$\mathbb{R}$
Maximum limit on magnitude of complex power flowing on line (i, j)	S_{ij}	
Active Power Demand	P_{D_i}	N
reactive Power Demand	Q_{D_i}	N
Upper and lower limits on power generation		

B. Implementation of the Proposed Model

The proposed machine learning model is implemented using Tensorflow and Keras 2.7.0. To accelerate the training process, models are trained and tested on Google Colaboratory, where Graphical Processing Units (GPUs) and Tensor Processing Units (TPUs) computing resources are available. Table 3 summarizes the model's architecture for IEEE 118-bus system. As discussed in Section 2, the proposed algorithm contains three neural networks: G, D and Q. Their input and output dimensions, hidden layers, activation, etc., are listed. Model D executes two tasks: one is to assign a feasibility score, and the other is to identify real/synthetic inputs. The two tasks share 4 hidden layers (HL) and separately calculate their outputs in the last 3 layers. Model Q has a much simpler structure for the expectation of extracting a linear-like correlation between code c and output y, so the cost value. c is sampled uniformly in the range $[-1, 1]$, and $z \in \mathbb{R}^{200}$ is sampled from the normal distribution.

Table 3: Architecture of the Proposal(118-bus system)

Model G			
Input Layer x:236 nodes c:1 nodes z:200 nodes	Hidden Layer 256 nodes 128 nodes 64 nodes 32 nodes	Output Layer y:344 nodes 108 for P/Q generation 236 for $\|V\|$ and ϕ	Activations LeakyRelu for 4 HL Sigmoid for output
Model D			
Input Layer x:236 nodes y:344 nodes	Shared Hidden Layer 512 nodes 256 nodes 128 nodes 64 nodes	Seperate Hidden&Output Layer 32 nodes 16 nodes 8 nodes Output of D_r:1 Output of D_f:2	Activations LeakyRelu for 7 HL Linear for output
Model Q			
Input Layer y:54 nodes	Hidden Layer 4 nodes	Output Layer $\hat{c}$: 1	Activations LeakyRelu for HL tanh for output

D. Proposed Dataset for Power Flow Studies

In the literature, supervised machine learning-based algorithms (9; 12) require a dataset with power demand and optimal solutions produced by existing OPF solvers. However, this work relaxes the assumptions of having this optimal dataset, and the data with only feasible solutions are sufficient. We are releasing the dataset PFFDS (Power Flow Feasible Dataset) with this paper. As the IEEE 118-bus system only provides a snapshot of the grid, we follow the common practice in existing literature (9; 12; 23) for generating power demand P_{D_i} and Q_{D_i} in the [80%,120%] range. Also, the branch flow limits are not included in the IEEE bus-118 benchmark, so a testing constraint in (25) is used. A total number of 70,000 scenarios of demand/solution pairs are included in this dataset. Furthermore, the stochasticity of renewable energy resources(wind and photovoltaic) is modeled in the data generation process based on their ratio in the United States in 2020 (26). These can be seen as random perturbations integrated into constraint [C1] in the formulation of OPF in Appendix. A. The dataset can be accessed on the link [placeholder].