

# Deep learning architectures for inference of AC-OPF solutions

Thomas Falconer<sup>1</sup> Letif Mones<sup>2</sup>

## Abstract

We present a systematic comparison between neural network (NN) architectures for inference of AC-OPF solutions. Using fully connected NNs as a baseline we demonstrate the efficacy of leveraging network topology in the models by constructing abstract representations of electrical grids in the graph domain, for both convolutional and graph NNs. The performance of the NN architectures is compared for regression (predicting optimal generator set-points) and classification (predicting the active set of constraints) settings. Computational gains for obtaining optimal solutions are also presented.

## 1. Introduction

Electricity market dynamics are generally governed by (some form of) Optimal Power Flow (OPF): the computation of (minimal cost) generation dispatch subject to reliability and security constraints of the grid. The classical OPF derivation (AC-OPF) is a non-linear, non-convex constrained optimization problem, which, when integrated with day-ahead unit commitment, forms a Mixed-Integer Program known to be NP-hard. Proliferation of renewable energy resources (e.g. wind and solar) has exacerbated uncertainties in modern power systems, thereby necessitating OPF solutions in near real-time to sustain accurate representation of the system state, preferably combined with probabilistic techniques for modelling uncertainty such as scenario-based Monte-Carlo simulation (Mezghani et al., 2019). This requires sequential OPF solutions at rates unattainable by conventional algorithms.

Typically, AC-OPF is solved using Interior-Point (IP) optimization methods (Wachter & Biegler, 2006), which are computationally expensive given the required evaluation of the Hessian (second-order derivative) of the Lagrangian at each iteration, rendering a time complexity which scales superlinearly with system size. To guarantee suitable rates of convergence, grid operators instead often resort to cheap

approximations of OPF, by refactoring the problem using convex relaxations or by utilising the inveterate DC-OPF – a Linear Program with considerably less control variables and constraints. However, such approximate frameworks are predisposed to sub-optimal estimations of locational marginal prices, engendering wasted emissions-intensive electricity generation – illustrated by the estimated 500 million metric tons of carbon dioxide emissions per year by virtue of global grid inefficiencies (Surana & Jordaan, 2019). Extending the OPF objective to incorporate generator-wise emissions costing furthers the computational burden, hence enabling fast computation of AC-OPF not only facilitates renewable energy generation, but is a direct method for climate change mitigation (Gholami et al., 2014).

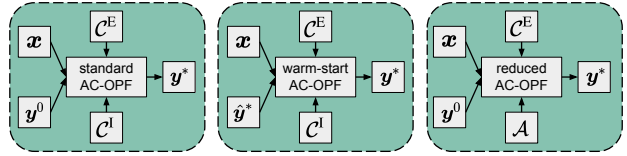


Figure 1. Strategies for solving AC-OF with IP methods: standard (left), warm-start (middle), and reduced (right).  $x$  denotes the vector of grid parameters,  $C^E$  represents the set of equality constraints,  $C^I$  and  $A$  represent the full and active sets of inequality constraints, respectively, and  $y^*$  and  $y^0$  are the optimal and initial vectors of the primal variables.

Applications of machine learning (ML) to OPF typically use regression techniques to directly infer the OPF solution from grid parameters, bypassing conventional solvers, thereby shifting the cost from online optimization to offline training (Guha et al., 2019). However, since the optimal point is not necessarily a smooth function of the inputs, this approach requires training on relatively large data sets to obtain acceptable accuracy. Further, there is no guarantee that the inferred optimal solution is feasible (i.e. satisfies all constraints), and violation of important constraints could be catastrophic in the context of power systems. It is more pragmatic therefore to ensure convergence to the optimal solution by instead initialising the IP solver with the inferred optimal solution, a so-called *warm-start* (Figure 1, middle panel) (Baker, 2019), whereby the regressor is parameterised using a (meta-)loss function corresponding directly to the time complexity to minimise computational burden (Jamei et al., 2019).

<sup>1</sup>UCL Energy Institute, London, United Kingdom <sup>2</sup>Invenia Labs, Cambridge, United Kingdom. Correspondence to: Thomas Falconer <thomas.falconer.19@ucl.ac.uk>.

Alternatively, the optimal active set (Misra et al., 2018) or the binding status of each constraint (Robson et al., 2020) can be inferred using classification. Although the number of constraints is exponential in system size (Deka & Misra, 2019), typically only a fraction constitutes the active set, thus a reduced problem (Figure 1, right panel) can be formulated whilst preserving the objective. Security risks associated with false negatives can be avoided by iteratively solving the reduced OPF and adding violated constraints until all those of the full problem are satisfied, hereafter referred to as the *iterative feasibility test*. Since the reduced OPF is much cheaper relative to the full problem, this procedure can in theory be rather efficient – the computational cost can also be minimised via meta-optimization by directly encoding the time complexity into the (meta-)loss function and optimizing the weights accordingly (Robson et al., 2020).

In each case, it is crucial to use model architectures that maximise predictive performance. The field typically employs fully connected models (FCNNs), however CNNs (Chen & Tate, 2020) and GNNs (Owerko et al., 2020) have recently been investigated to exploit spatial dependencies within the electrical grid. This paper addresses contentions throughout literature of the most suitable modelling frameworks for OPF by offering a systematic comparison between these architectures, and between the regression and classification methods. Two input domains are studied: (1) only load variables, to exclusively mimic fluctuations in demand and (2) the entire set of grid parameters, to reflect the seasonality of all grid components and to evaluate the utility of the architectures in (more realistic) high-dimensional scenarios. Experiments are carried out using synthetic grids from Power Grid Lib (Babaeinejadsarookolaee et al., 2019). We believe our work can facilitate the transition toward a clean and efficient power grid – an important contribution to tackling climate change.

## 2. Methods

The electrical grid can be represented as an undirected graph  $\mathbb{G} = (\mathcal{V}, \mathcal{E})$  with a set of  $|\mathcal{V}|$  number of nodes (buses) and  $|\mathcal{E}|$  number of edges (transmission lines) such that  $\mathcal{E} = \{(i, j) \mid i \text{ is connected to } j; i, j \in \mathcal{V}\}$  (e.g. Figure 2). We define  $\mathcal{X}$  as the set of *control variable types*, which can include the following: active and reactive loads, maximum active and reactive power outputs of generators, line thermal ratings and the reactance and resistance of each transmission line. Hereafter we refer to the two studied domains of the control variable types as: (1) only load variables  $\mathcal{X}_{\text{load}}$  and (2) all grid parameters  $\mathcal{X}_{\text{all}}$ . Further, the above grid variable types can be classified as node  $\mathcal{X}^{\mathcal{V}}$  or edge  $\mathcal{X}^{\mathcal{E}}$  types. The complete set of operational constraints  $\mathcal{C}$  is characterised by two distinct subsets: (1) the set of both convex and

non-convex equality constraints  $\mathcal{C}^E \subset \mathcal{C}$ , which enforce the non-linear AC power flow equations and (2) the set of inequality constraints  $\mathcal{C}^I \subset \mathcal{C}$ , which enforce the operational and physical limits of the system (e.g. lower and upper bounds of generator power injection).

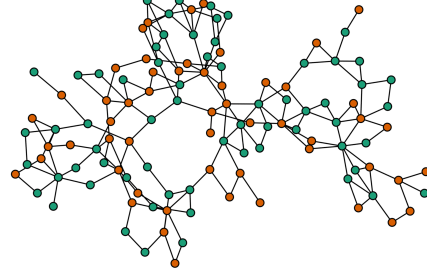


Figure 2. Graph representation of synthetic grid 118-ieee. Orange and green circles denote generator and load buses, respectively.

With this notation, AC-OPF can be expressed in the following concise form of mathematical programming:

$$\begin{aligned} \min_{\mathbf{y}} \quad & f(\mathbf{x}, \mathbf{y}) \\ \text{s. t.} \quad & c_i^E(\mathbf{x}, \mathbf{y}) = 0 \quad i = 1, \dots, |\mathcal{C}^E| \\ & c_j^I(\mathbf{x}, \mathbf{y}) \geq 0 \quad j = 1, \dots, |\mathcal{C}^I| \end{aligned} \quad (2.1)$$

Here,  $\mathbf{x}$  denotes the vector of grid parameters,  $\mathbf{y}$  is the vector of voltage magnitudes and active power injections of generators and  $f(\mathbf{x}, \mathbf{y})$  is the objective function – typically a quadratic or piecewise linear function of the (monotonically increasing) generator cost curves. This formulation (eq. 2.1) lets us view AC-OPF as an operator that maps the grid parameters to the optimal solution.

For an arbitrary input  $\mathbf{x}$  the regression objective is to infer the set of optimal primal variables  $\mathbf{y}^*$ , indicative of the minimum information required for the warm-start. Defining  $\mathcal{G} \subset \mathcal{V}$  as the set of generators, the regressor learns the *true* AC-OPF mapping defined by the operator  $\Omega : \mathbb{R}^{|\mathcal{X}|} \mapsto \mathbb{R}^{|\mathcal{V}|+|\mathcal{G}|}$  by parameterising a neural network  $\hat{\Omega}$  with parameters  $\Theta$  and minimising the mean-squared error (MSE) between the ground-truths  $\mathbf{y}^* = \Omega(\mathbf{x})$  and the inferred optimal solution  $\hat{\mathbf{y}}^* = \hat{\Omega}(\mathbf{x}, \Theta)$ .

The classification setting determines the binding status of each constraint in  $\mathcal{C}^{\text{NT}} \subset \mathcal{C}^I$ , the subset of inequality constraints, which change binding status at least once in the training set (hereafter referred to as non-trivial constraints). This facilitates construction of the active set  $\mathcal{A} \subseteq \mathcal{C}^I$ , the subset of inequality constraints binding at the optimum (such that  $\mathcal{C}^E \cup \mathcal{A}$  constitute the congestion regime). We define the classifier output as a binary vector representing an enumeration of the set of non-trivial constraints. The classifier

therefore learns the mapping  $\Psi : \mathbb{R}^{|\mathcal{X}|} \mapsto \{0, 1\}^{|\mathcal{C}^{\text{NT}}|}$  provided by the OPF by minimising the binary cross-entropy loss (BCE) between the ground-truths  $\Psi(\mathbf{x})$  and the predicted probabilities  $\hat{\Psi}(\mathbf{x}, \Theta)$ .

To maintain validity of the constructed data sets, we generated samples within prescribed boundaries around the Power Grid Lib defaults:  $\pm 15\%$  for nodal active load and  $\pm 10\%$  for all remaining parameters. On account of demonstrated computational efficiency relative to conventional Uniform scaling methods, we adopt a Random Walk Metropolis-Hastings-based sampling technique whereby steps are taken through normalised space subject to an isotropic Gaussian proposal distribution  $\mathcal{N}(\mathbf{x}^*, \alpha \mathbf{I})$ , centred on the current system state  $\mathbf{x}^*$  with step size  $\alpha$ , only accepting AC feasible candidates (Falconer, 2020). AC-OPF solutions were found using `PowerModels.jl` (Coffrin et al., 2018) in combination with the IPOPT solver (Wachter & Biegler, 2006).

We first construct a baseline FCNN designed to operate on data in vector format, hence lacking sufficient relational inductive bias to exploit any underlying structure. Network topology can be expressed in the graph domain using the (weighted) binary adjacency matrix. For CNNs with only load variables as input, each parameter can be represented by a vector of length  $|\mathcal{V}|$  and subsequently combined into a 3D tensor with dimensions  $\mathbb{R}^{|\mathcal{V}| \times 1 \times |\mathcal{X}|}$ . For all grid parameters, those relevant to nodes  $\mathcal{X}^{\mathcal{V}}$  or edges  $\mathcal{X}^{\mathcal{E}}$  can be constructed into diagonal or off-diagonal matrices, respectively, such that the input dimensions are  $\mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}| \times |\mathcal{X}|}$ .

For GNNs, we investigate both spectral and spatial convolutions, the latter of which are typically constrained to operate on node features and a 1D set of edge features. We overcome this limitation by encoding edge features as node features by concatenating structures akin to that mentioned above, engendering input dimensions of  $\mathbb{R}^{|\mathcal{V}| \times (|\mathcal{X}^{\mathcal{V}}| + |\mathcal{X}^{\mathcal{E}}| \|\mathcal{V}\|)}$ . Although spatial graph convolutions ordinarily permit multi-dimensional edge features, enhanced performance was observed empirically using an input structure akin to that defined here.

Spectral graph convolutions operate in the Fourier domain; input signals are passed through parameterised functions of the normalised Laplacian, exploiting its positive-semidefinite property. Given this procedure has  $\mathcal{O}(|\mathcal{V}|^3)$  time complexity, we investigate two spectral layers, ChebConv (Kipf & Welling, 2017) and GCNConv (Defferrard et al., 2017), which reduce the computational cost by approximating the kernel functions using Chebyshev polynomials of the eigenvalues up to the  $K$ -th order, avoiding expensive full eigendecomposition of the Laplacian. GCNConv constrains the layer-wise convolution to first-order neighbours ( $K = 1$ ), lessening overfitting to particular localities.

Spatial graph convolutions are instead directly performed

in the graph domain, reducing time complexity whilst minimising information loss. For a given node, SplineConv computes a linear combination of its features together with those of its  $K$ -th order neighbours, weighted by a kernel function using the product of parameterised B-spline basis functions (Fey et al., 2018). The local support property of B-splines reduces the number of parameters, enhancing the computational efficiency of the operator.

Each graph layer (ChebConv, GCNConv and SplineConv) was used to construct a unique GNN, hereafter referred to as CHNN, GCN and SNN, respectively. Parameters were optimized using ADAM (Kingma & Ba, 2014) with learning rate initialised at  $\eta = 10^{-4}$ . Hidden layers were applied with BatchNorm and a ReLU activation function; dropout (with probability of 0.4) was applied to fully-connected layers. CNNs were constructed using  $3 \times 3$  kernels,  $2 \times 2$  max-pooling layers, zero-padding and a stride length of 1. For CHNN and SNN,  $K$  was set to 5. Hyper-parameters were tuned iteratively, to construct models where for each case, the number of weights was in the same order of magnitude to facilitate a fair comparison of performance.

### 3. Results

We generated 10k samples for several synthetic grids (Table 1). Less unique active sets were uncovered using only load variables since without altering the other grid parameters, changes in system state are much less profound, rendering homogeneity amongst congestion regimes. For the 300-ieee case, this is capped at the number of samples, implying that even 10k samples restricts convergence to the *true* parameter distribution for larger grids. The additional complexity of using all grid parameters is highlighted by the larger input domains  $\dim(\mathbf{x})$  in addition to the greater cardinalities of non-trivial constraints  $|\mathcal{C}^{\text{NT}}|$ . Data sets were subsequently split into training, validation and test sets with a ratio of 80:10:10.

Table 1. Grid characteristics and number of unique active sets for different AC-OPF cases, using 10k samples.

Case	Only Load: $\mathbf{x} = \mathbf{x}_{\text{load}}$			All Parameters: $\mathbf{x} = \mathbf{x}_{\text{all}}$		
	$\dim(\mathbf{x})$	$ \mathcal{C}^{\text{NT}} $	# Active Sets	$\dim(\mathbf{x})$	$ \mathcal{C}^{\text{NT}} $	# Active Sets
73-ieee-rtts	146	28	1008	660	60	2902
118-ieee	236	31	437	864	66	6649
162-ieee-dtc	322	58	1805	1102	92	6026
300-ieee	600	143	9803	1773	201	10000

Regression performance is depicted by the average (test set) MSE for five independent runs (Table 2); each GNN outperformed FCNN and CNN, in some cases by an order of magnitude. Relative performance is more overt using only load given the lower input dimensionality, however, we still see performance enhancements in the more complex all parameter setting. SNN typically outperforms GCN and CHNN, possibly due to the information loss via spectral node em-

bedding. This, combined with reduced training times in virtue of the computational efficiency of the B-spline kernel functions, alludes to better scaling to larger systems. We observe no significant performance enhancements using CNN, which was expected, since the non-Euclidean data structures are deprived of the geometric priors on which the convolution operator relies. Anomalous cases of reduced error can be attributed to the coincidental unearthing of structural information within the receptive fields when convolving over the (weighted) adjacency matrices.

Table 2. Average test set MSE values of regression models.

Case ( $x = x_{\text{load}}$ )		FCNN	CNN	GCN	CHNN	SNN
73-ieee-rtts	$10^{-4}$	6.613	7.625	0.556	0.612	<b>0.527</b>
118-ieee	$10^{-4}$	2.171	3.042	<b>0.306</b>	0.334	0.329
162-ieee-dtc	$10^{-3}$	9.492	6.026	3.341	3.039	<b>2.145</b>
300-ieee	$10^{-2}$	3.654	5.973	2.283	2.156	<b>1.948</b>
Case ( $x = x_{\text{all}}$ )		FCNN	CNN	GCN	CHNN	SNN
73-ieee-rtts	$10^{-3}$	4.916	5.241	2.011	1.953	<b>1.247</b>
118-ieee	$10^{-3}$	2.621	3.487	0.396	0.450	<b>0.372</b>
162-ieee-dtc	$10^{-2}$	2.783	4.585	1.411	1.682	<b>1.229</b>
300-ieee	$10^{-1}$	1.293	1.466	0.723	0.711	<b>0.574</b>

Classification performance is reported in terms of average (test set) recall and precision, in addition to BCE, to see how each model balances the trade-off between false positives (FPs) and false negatives (FNs). GNNs again outperform FCNN and CNN in each case, subject to each metric (Table 3). Interestingly, we observe a greater precision than recall in virtually every instance, implying the BCE objective is more sensitive to FPs, which is unfavourable given the computational cost of omitting *true* binding constraints from the reduced problem (FNs). This implies the loss function is misaligned with the desired objective and should be altered to bias the reduction of FNs.

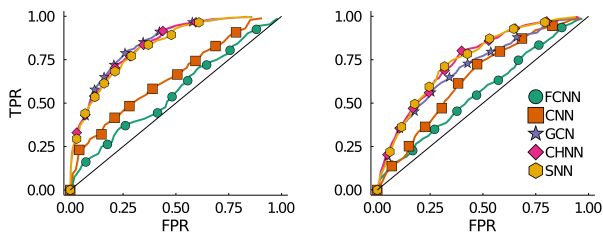


Figure 3. ROC curves for each classification model (118-ieee system) using only load (left) and all parameters (right) as input.

Using the 118-ieee system as a case study to visualise the ROC curves for each model (Figure 3), we observe that GNNs are superior irrespective of classification threshold. For many of the binary classifications, GNNs demonstrated AUCs surpassing 0.9, whereas FCNN and CNN often reported AUCs less than 0.5, i.e. worse than random.

Finally, we report the average computational gain of the

test set realised for the warm-start and reduced problem, using all grid parameters as a case study (Table 4). In both settings, the models that provide more accurate inference (GNNs) engender superior results. Moderate gains were expected since differentiating the non-convex equality constraints remains to greatest computational expense in both the warm-start approach and reduced formulation (Robson et al., 2020). Moreover, in the regression setting, only the primal variables are initialised, therefore a minimum number of iterations is still required for the duals to converge i.e. there is an empirical upper bound.

Given the complexity of the 162-ieee-dtc system, it is probable that the primals were near the feasible boundary, rendering an ill-conditioned system thus constraining the solver to small step sizes, hence the negative gain. The similar or greater performance of the reduced problem relative to the warm-start implies that this method is preferable overall, as we could achieve greater gains merely by reducing the number of FNs (thereby reducing the cost of the iterative feasibility test) with a more sophisticated objective function.

## 4. Conclusion

Our systematic comparison of NN architectures for direct and indirect inference of AC-OPF solutions, as well as methods for augmenting the IP solver, has highlighted the importance of predictive accuracy in reducing the computational cost of AC-OPF with ML, essential for minimising emissions in the electricity market. We demonstrated the utility of explicitly incorporating network topology in the learning process using GNNs and concluded the gains of the reduced problem allude to better scaling to larger grids. In future work, we will investigate more sophisticated objectives to bias the optimization towards a reduction in FNs, and leverage the capacity of GNNs to supplement the meta-optimization technique proposed by Robson et al. (2020).

## References

- Babaeinejadsarookolae, S., Birchfield, A., Christie, R. D., Coffrin, C., DeMarco, C., Diao, R., Ferris, M., Fliscounakis, S., Greene, S., Huang, R., Jozs, C., Korab, R., Lesieutre, B., Maeght, J., Molzahn, D., Overbye, T., Panciatici, P., Park, B., Snodgrass, J., and Zimmerman, R. The power grid library for benchmarking ac optimal power flow algorithms. 2019.
- Baker, K. Learning warm-start points for ac optimal power flow. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2019. doi: 10.1109/MLSP.2019.8918690.
- Chen, L. and Tate, J. E. Hot-starting the ac power flow

Table 3. Average BCE, recall and precision of classification models.

Case ( $\mathcal{X} = \mathcal{X}_{\text{load}}$ )	BCE					Recall					Precision				
	FCNN	CNN	GCN	CHNN	SNN	FCNN	CNN	GCN	CHNN	SNN	FCNN	CNN	GCN	CHNN	SNN
73-ieee-rtts	0.183	0.158	0.064	0.061	<b>0.057</b>	0.724	0.774	0.854	0.843	<b>0.865</b>	0.928	0.935	0.964	0.946	<b>0.976</b>
118-ieee	0.193	0.179	0.041	<b>0.039</b>	0.052	0.631	0.658	0.772	0.738	<b>0.792</b>	0.942	0.875	<b>0.957</b>	0.929	0.946
162-ieee-dtc	0.235	0.214	0.093	0.089	<b>0.084</b>	0.682	0.675	0.764	<b>0.767</b>	0.761	0.773	0.782	0.857	<b>0.871</b>	0.864
300-ieee	0.157	0.163	0.111	<b>0.107</b>	0.115	0.611	0.582	<b>0.683</b>	0.678	0.664	0.775	0.791	0.874	<b>0.882</b>	0.879

Case ( $\mathcal{X} = \mathcal{X}_{\text{all}}$ )	BCE					Recall					Precision				
	FCNN	CNN	GCN	CHNN	SNN	FCNN	CNN	GCN	CHNN	SNN	FCNN	CNN	GCN	CHNN	SNN
73-ieee-rtts	0.263	0.247	0.115	0.123	<b>0.111</b>	0.602	0.597	0.642	0.699	<b>0.724</b>	0.893	0.874	0.89	0.892	<b>0.921</b>
118-ieee	0.224	0.206	0.104	0.101	<b>0.098</b>	0.532	0.614	<b>0.685</b>	0.674	0.676	0.846	0.833	0.873	<b>0.891</b>	0.86
162-ieee-dtc	0.187	0.194	0.136	<b>0.127</b>	0.131	0.491	0.482	0.538	0.523	<b>0.552</b>	0.823	0.835	0.879	<b>0.886</b>	0.873
300-ieee	0.195	0.201	0.128	0.121	<b>0.119</b>	0.587	0.583	0.635	0.638	<b>0.642</b>	0.794	0.801	0.847	0.845	<b>0.859</b>

Table 4. Average computational gain for direct (warm-start) and indirect (reduced problem) approaches.

Case (direct)	FCNN	CNN	GCN	CHNN	SNN
73-ieee-rtts	17.58	15.76	<b>21.65</b>	21.18	21.43
118-ieee	14.38	15.92	<b>17.27</b>	16.57	17.09
162-ieee-dtc	-133.74	-125.79	-96.76	-93.79	<b>-90.64</b>
300-ieee	12.20	11.67	17.31	<b>17.56</b>	16.38

Case (indirect)	FCNN	CNN	GCN	CHNN	SNN
73-ieee-rtts	-13.82	-11.91	-4.26	-2.60	<b>-1.31</b>
118-ieee	21.17	22.91	<b>27.06</b>	26.44	25.75
162-ieee-dtc	-38.29	-25.08	-10.87	<b>-9.10</b>	-12.26
300-ieee	13.24	12.83	<b>17.84</b>	16.90	17.73

with convolutional neural networks. *ArXiv e-prints*, arXiv:2004.09342, 2020.

Coffrin, C., Bent, R., Sundar, K., Ng, Y., and Lubin, M. PowerModels.jl: An open-source framework for exploring power flow formulations. In *2018 Power Systems Computation Conference (PSCC)*, pp. 1–8. IEEE, 2018.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *ArXiv e-prints*, arXiv:1606.09375, 2017.

Deka, D. and Misra, S. Learning for DC-OPF: Classifying active sets using neural nets. *ArXiv e-prints*, arXiv:1902.05607, 2019.

Falconer, T. Reducing the computational cost of ac optimal power flow with geometric deep learning, 2020.

Fey, M., Lenssen, J. E., Weichert, F., and Müller, H. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. *ArXiv e-prints*, arXiv:1711.08920, 2018.

Gholami, A., Ansari, J., Jamei, M., and Kazemi, A. Environmental/economic dispatch incorporating renewable energy sources and plug-in vehicles. *IET Generation, Transmission & Distribution*, 8(12):2183–2198, 2014.

Guha, N., Wang, Z., and Majumdar, A. Machine learning for AC optimal power flow. ICML, Climate Change: How Can AI Help? Workshop, 2019.

Jamei, M., Mones, L., Robson, A., White, L., Requeima, J., and Ududec, C. Meta-optimization of optimal power flow. ICML, Climate Change: How Can AI Help? Workshop, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *ArXiv e-prints*, arXiv:1412.6980, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *ArXiv e-prints*, arXiv:1609.02907, 2017.

Mezghani, I., Misra, S., and Deka, D. Stochastic ac optimal power flow: A data-driven approach. *ArXiv e-prints*, arXiv:1910.09144, 2019.

Misra, S., Roald, L., and Ng, Y. Learning for Constrained Optimization: Identifying Optimal Active Constraint Sets. *ArXiv e-prints*, arXiv:1802.09639, 2018.

Owerko, D., Gama, F., and Ribeiro, A. Optimal power flow using graph neural networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5930–5934, 2020. doi: 10.1109/ICASSP40776.2020.9053140.

Robson, A., Jamei, M., Ududec, C., and Mones, L. Learning an optimally reduced formulation of opf through meta-optimization. *ArXiv e-prints*, arXiv:1911.06784, 2020.

Surana, K. and Jordaan, S. M. The climate mitigation opportunity behind global power transmission and distribution. *Nature Climate Change*, 9(9):660–665, 2019.

Wachter, A. and Biegler, L. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.