

## I. OPTIMAL POWER FLOW

Optimal power flow (OPF) is one of the most important optimization problems for the energy industry. In its simplest form, it is the problem of finding the optimal power outputs for several generators on the grid with respect to some objective function - usually a second order polynomial,  $c(\mathbf{p}^G, \mathbf{p}^R)$ , where  $\mathbf{p}^G, \mathbf{p}^R$  is the generated active and reactive power.

This problem is at the heart of daily electricity grid operation. However, it is a very difficult problem to solve. Since electrical systems are predominantly AC, this introduces several non-linearities in the form of trigonometric functions in the voltage angle constraint equations (7). Moreover, the problem is non-convex due to non-linearities in the power flow equations (2, 3) [1], [2] and has been proven to be NP hard [3]. This makes it one of the most difficult optimization problems [4]. Therefore, in practice linear DC approximations are used to solve the problem, but which exhibit significant inaccuracy [2]. There is a lot of active research. Recently a lot of works attempt convex approximations using techniques like semi-definite programming [2]. In this paper we are concerned with applying recently developed [x] graph neural networks to approximate the AC OPF problem, which can be formulated as follows.

$$\min_{\mathbf{p}_i^G} \sum_{i \in \mathcal{G}} c_i(\mathbf{p}_i^G), \text{ subject to} \quad (1)$$

$$\mathbf{p}_i(\mathbf{v}, \boldsymbol{\delta}) = \mathbf{p}_i^G - \mathbf{p}_i^L \quad \forall i \in \mathcal{N} \quad (2)$$

$$\mathbf{q}_i(\mathbf{v}, \boldsymbol{\delta}) = \mathbf{q}_i^G - \mathbf{q}_i^L \quad \forall i \in \mathcal{N} \quad (3)$$

$$\mathbf{p}_i^{G, \min} \leq \mathbf{p}_i^G \leq \mathbf{p}_i^{G, \max} \quad \forall i \in \mathcal{G} \quad (4)$$

$$\mathbf{q}_i^{G, \min} \leq \mathbf{q}_i^G \leq \mathbf{q}_i^{G, \max} \quad \forall i \in \mathcal{G} \quad (5)$$

$$\mathbf{v}_i^{\min} \leq \mathbf{v}_i \leq \mathbf{v}_i^{\max} \quad \forall i \in \mathcal{N} \quad (6)$$

$$\boldsymbol{\delta}_i^{\min} \leq \boldsymbol{\delta}_i \leq \boldsymbol{\delta}_i^{\max} \quad \forall i \in \mathcal{N} \quad (7)$$

$$(8)$$

In the OPF equations,  $\mathcal{N}$  is the set of all nodes and  $\mathcal{G}$  is the set of all generators.  $\mathbf{p}_i(\mathbf{v}, \boldsymbol{\delta}), \mathbf{q}_i(\mathbf{v}, \boldsymbol{\delta})$  are the net active and reactive power flow into node  $i$ . They are non-linear functions of the voltages and voltage angles of all other nodes.  $\mathbf{p}_i^G, \mathbf{q}_i^G$  is the net power generated and  $\mathbf{p}_i^L, \mathbf{q}_i^L$  is the net load at node  $i$ . Therefore, (2) and (3) are the power flow constraints, which are derived from Kirchhoff's current law. Equations (4) and (5) represent the operational constraints on generator output, where  $\mathbf{p}_i^G, \mathbf{q}_i^G$  are the active and reactive power generated. The voltage,  $\mathbf{v}_i$  and voltage angle  $\boldsymbol{\delta}_i$  at node  $i$  are also constrained by equations (6) and (7) respectively.

## II. MACHINE LEARNING FOR OPF

In the past there have been several works on applying evolutionary programming, evolutionary strategies, genetic algorithms, artificial neural networks, simulated annealing, fuzzy set theory, ant colony optimization and particle swarm optimization [7]. Genetic algorithms have been shown to effectively solve the OPF problem for small and medium networks [8], but fail to converge to feasible solutions for networks with more than 100 nodes [9]. Similarly, particle swarm algorithms have been demonstrated to work for the IEEE 5 node case [10] and a hybrid between particle swarm and Newton-Rapson was demonstrated for a 30 node network [11].

Nevertheless, there has been limited work on applying machine learning for optimal power flow. More specifically, few authors explore using machine learning to perform OPF end to end as we do. There are two notable exceptions to this. Dobbe et al. [12] propose a system that uses machine learning for decentralized OPF [12].

For each node in the network, they decide to which other node to communicate with using a greedy algorithm – note that this node need not be adjacent. They use a multiple stepwise regression algorithm implemented in a previous paper [13] to estimate the output of an OPF – the optimal power generated. Specifically, they use load and slack bus voltage angle as inputs. Their numerical experiments on a 129 node network demonstrate performance near optimum. The predicted solution deviates by 0.15% on average as compared to the objective value achieved by the OPF. The authors also measure how many constraints are violated. Their solution also learned local constraints from the data itself, on average violating only 0.00025% of them.

Guha et al. [5] demonstrate using deep learning for end-to-end prediction of AC OPF and constraint prediction - i.e. predicting which constraints are bounding, which can speed up traditional solver performance. They create synthetic data for the IEEE 30 and IEEE 118 test cases by randomly perturbing the load distribution provided by the cases. Their dataset for IEE30 and IEEE118 has 812888 and 95000 samples, respectively. They achieve an average cost deviation of 0.2% and satisfy constraints for 51% and 70% of the 30 bus and 118 bus case. They find that the optimal model for the 30 bus case was a two layer NN with 512 hidden units and tanh activations, while the best model for the 118 bus case was a three layer NN with 512 hidden units and ReLU activation.

## III. GRAPH NEURAL NETWORKS

## IV. NUMERICAL EXPERIMENTS

### A. Problem Description

We consider a graph  $\mathcal{G}$  which represents the electrical network of a test case with  $|\mathcal{N}|$  nodes.  $\mathcal{G}$  is represented by an adjacency matrix  $A \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{N}|}$  as defined in equation (9).

$$A_{ij} = \begin{cases} 0 & w_{ij} < \omega \\ w_{ij} & \text{otherwise} \end{cases} \quad (9)$$

$$w_{ij} = \exp(-|z_{ij}| \sigma) \quad (10)$$

$|z_{ij}|$  is the magnitude of impedance in ohms between bus  $i$  and  $j$ . It is defined as  $|z_{ij}| = |r_{ij} + ix_{ij}|$  where  $r_{ij}, x_{ij}$  is the resistance and reactance of the line between nodes  $i$  and  $j$ . The lower the impedance the more coupled the two nodes are to each other. The values are obtained from the test case.  $\sigma = 0.001$  is a scaling parameter, and  $\omega = 0.01$  is a threshold parameter - these are chosen by trial and error such that  $A$  has only one connected component but is maximally sparse.

$\mathbf{p}_i^G \in \mathbb{R}^{|\mathcal{G}|}$  is the active power generated by each generator is the controlled parameter. Note that for generators there is no variable  $Q^G$  for reactive power, generator output is fully defined by its active power and voltage. The latter is constant and specified by the test case. The uncontrolled parameters are the active and reactive components of the load at each node,  $L \in \mathbb{R}^{|\mathcal{N}| \times 2}$ . We define  $L$  as,

$$L = [\mathbf{p}^L, \mathbf{q}^L]. \quad (11)$$

For any physically feasible  $L$ , and  $\mathbf{p}^G$  we can run a power flow to find the voltage, voltage angle, net active power, and net reactive power for each node at the steady state of the system. These quantities are physically measurable at each node. We represent them by a matrix  $S$ .

$$S = [\mathbf{v}, \boldsymbol{\delta}, \mathbf{p}, \mathbf{q}] \in \mathbb{R}^{N \times 4} \quad (12)$$

where  $\mathbf{v}$  is the voltage,  $\boldsymbol{\delta}$  is the voltage angle,  $\mathbf{p}$  is the net active power, and  $\mathbf{q}$  is the net reactive power at each node.

Our goal is to imitate optimal power flow as follows. For each load  $L$  there exists a  $\mathbf{p}^G$  that can be calculated using an exact AC

	GNN with MLP	Local GNN
IEEE 39	0.0036	0.0138
Iceland (118 nodes)	0.0030	0.0507

Fig. 1. Normalized RMSE for different architectures and test cases.

OPF algorithm - we use the interior point implementation provided by Pandapower [14]. There is also a suboptimal solution,  $\mathbf{p}_{dc}^G$ , obtained by running a DC approximation to the OPF problem. Using  $\mathbf{p}_{dc}^G$  and  $L$  we run a power flow to find  $S$ , the steady state of the system for the suboptimal generator output. We train a graph neural network to approximate  $\mathbf{p}^G$  with  $\hat{\mathbf{p}}^G$  as a function of  $S$ .

$$\hat{\mathbf{p}}^G = H(S). \quad (13)$$

### B. Data Generation and Training

We use Pandapower [14] to construct the case models and perform OPF calculations. Pandapower also provides a substantial set of electrical system test cases, with reference load distributions. For supervised learning we generate synthetic loads using the reference loads provided by the test case,  $L_{ref}$ . To generate each load we sample from a uniform distribution around the reference load. This is the same methodology as used in literature [5].

$$L \sim \text{Uniform}(0.9L_{ref}, 1.1L_{ref}) \quad (14)$$

For each test case we generate 10000 samples which are split 80/10/10 between the train, test and validation sets. For each sample of  $L$  we also pre-compute the sub-optimal steady state  $S$  and the optimal  $\mathbf{p}^G$ . We train the GNN using Adam optimizer and minimize the mean squared error between  $\mathbf{p}^G$  and  $\hat{\mathbf{p}}^G$ .

### C. Results

#### REFERENCES

- [1] B. C. Lesieutre and I. A. Hiskens, "Convexity of the set of feasible injections and revenue adequacy in ftr markets," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1790–1798, Nov. 2005. DOI: 10.1109/TPWRS.2005.857268.
- [2] D. K. Molzahn and I. A. Hiskens, "Convex relaxations of optimal power flow problems: An illustrative example," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 5, pp. 650–660, May 2016. DOI: 10.1109/TCSI.2016.2529281.
- [3] D. Bienstock and A. Verma, *Strong np-hardness of ac power flows feasibility*, 2015. arXiv: 1512.07315 [math.OC].
- [4] M. B. Cain, R. P. O'Neil, and C. Anya, "History of optimal power flow and formulations," pp. 1–31, 2012. [Online]. Available: <https://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-1-history-formulation-testing.pdf>.
- [5] N. Guha, Z. Wang, and A. Majumdar, "Machine learning for ac optimal power flow," in *ICML 2019*, 2019.
- [6] S. Frank and S. Rebennack, "An introduction to optimal power flow: Theory, formulation, and examples," *IIE Transactions*, vol. 48, no. 12, pp. 1172–1197, 2016. DOI: 10.1080/0740817X.2016.1189626. eprint: <https://doi.org/10.1080/0740817X.2016.1189626>. [Online]. Available: <https://doi.org/10.1080/0740817X.2016.1189626>.
- [7] M. R. AlRashidi and M. E. El-Hawary, *Applications of computational intelligence techniques for solving the revived optimal power flow problem*, ID: 271091, 2009. DOI: //doi.org/10.1016/j.ejpsr.2008.10.004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378779608002757>.
- [8] A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas, and V. Petridis, "Optimal power flow by enhanced genetic algorithm," *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 229–236, May 2002. DOI: 10.1109/TPWRS.2002.1007886.
- [9] M. Todorovski and D. Rajcic, "A power flow method suitable for solving opf problems using genetic algorithms," in *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, vol. 2, Sep. 2003, 215–219 vol.2. DOI: 10.1109/EURCON.2003.1248186.
- [10] C.-R. Wang, H.-J. Yuan, Z.-Q. Huang, J.-W. Zhang, and C.-J. Sun, "A modified particle swarm optimization algorithm and its application in optimal power flow problem," vol. 5, 2005, 288–2889 Vol. 5. DOI: 10.1109/ICMLC.2005.1527435.
- [11] M. R. AlRashidi and M. E. El-Hawary, "Hybrid particle swarm optimization approach for solving the discrete opf problem considering the valve loading effects," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2030–2038, Nov. 2007. DOI: 10.1109/TPWRS.2007.907375.
- [12] R. Dobbe, O. Sondermeijer, D. Fridovich-Keil, D. Arnold, D. Callaway, and C. Tomlin, "Data-driven decentralized optimal power flow," *CoRR*, vol. abs/1806.06790, 2018. arXiv: 1806.06790. [Online]. Available: <http://arxiv.org/abs/1806.06790>.
- [13] O. Sondermeijer, R. Dobbe, D. Arnold, C. Tomlin, and T. Keviczky, "Regression-based inverter control for decentralized optimal power flow and voltage regulation," *CoRR*, vol. abs/1902.08594, 2019. arXiv: 1902.08594. [Online]. Available: <http://arxiv.org/abs/1902.08594>.
- [14] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "Pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018. DOI: 10.1109/TPWRS.2018.2829021.
- [15] E. Wilson, *Commercial and residential hourly load profiles for all tmy3 locations in the united states*, en, 2015. [Online]. Available: <https://openet.org/doe-opendata/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-united-states>.