Physics-Aware Fast Learning and Inference for Predicting Active Set of DC-OPF

Hossein Khazaei
Department of Electrical and Computer Engineering, Stony Brook University
Stony Brook, NY 11794
Email: hossein.khazaei@stonybrook.edu

Yue Zhao
Department of Electrical and Computer Engineering, Stony Brook University
Stony Brook, NY 11794
Email: yue.zhao.2@stonybrook.edu

Abstract*—DC-OPF stands as the cornerstone for efficient and secure operations of power systems. The grid operators need to solve DC-OPF repeatedly and in large numbers to maintain the balance of electricity supply and demand, especially under high penetration of renewable energies. Recently, research efforts have been made in predicting the optimal active sets as a key component in learning-based solvers for DC-OPF. In this paper, we investigate the classifiers that inherently exploit a key physical property of the optimal solutions of DC-OPF: the input space corresponding to an optimal active set is a polyhedron, and the classes of different active sets are linearly separable. In particular, we investigate the effectiveness of linear discriminant analysis (LDA) classifiers for predicting the optimal active sets for DC-OPF. This is because LDA, as a natural multi-class classifier, by definition guarantees that the decision regions for all the classes are polyhedrons. Simulations are conducted on the IEEE–162 bus test case with a 50% renewable penetration level provided by 37 renewable power producers. We examine LDA as well as other classifier candidates, namely support vector machines, neural networks, and gradient boosted decision trees. The numerical results suggest that LDA a) achieves a *testing* performance in accuracy and in run-time similar to carefully trained neural networks, and b) is also much faster and easier to *train* than the other more complicated algorithms compared. Given the highly competitive testing accuracy, extremely fast *training and testing*, and the straightforward application to any problem setting without the need of algorithm tuning, we advocate that LDA is a top choice of learning-based algorithm for predicting the optimal active set for DC-OPF.

I. Introduction

Solving optimization problems are at the core of many decision making processes in power system operations. In some cases, the decision maker needs to repeatedly solve optimization problems that are differ only in some input parameters. For example, the optimal power flow (OPF) problem that is an optimization problem for scheduling energy resources is solved as often as every 5 minutes. The high frequency of these problems along with the short time available to reach the solution leads to billions of dollars in loss due to the suboptimality of the solutions [1]. With the transition of power systems toward higher penetrations of intermittent renewable energies, the necessity for fast and reliable solvers become even more prominent: Not only the near-real-time OPF problems may see much higher variability in the net-loads, but also there is an increasing need for the hour-ahead/day ahead OPF problems to be solved for a large number of renewable scenarios due to their high uncertainty.

This work is supported by the National Science Foundation under Grant ECCS-2025152.

Recently, researchers have investigated the potential of using learning-based methods for solving optimization problems including those in power systems. These methods try to find the complex mapping from the input parameters (e.g., nodal net-loads) to the optimal solutions (e.g., the optimal generation dispatch). The learning-based solvers can be categorized into two classes: *end-to-end* solvers, and *two-step* solvers. With end-to-end solvers the trained predictors perform a direct mapping from the input parameters to the optimal solution. An example of end-to-end predictors can be found in [2] where a deep neural network is used as an end-to-end model for solving DC-OPF. To address the ensuing feasibility issues, a post-processing procedure is used. Two-step solvers leverage the idea that if we know the binding inequalities of the optimization problem at the optimal solution, finding the optimal solution would become a much less complex task. As such, two-step solvers break down the optimization problem into two steps: first, predicting the binding constraints, termed *active set*, at the *optimal* solution, and second, given the predicted active set, solving or predicting the optimal solution of the overall optimization problem.

In the following we review the related literature. In [3] the authors explore the idea of a two-step learning model for solving DC-OPF where they use a neural network for the classification of the active sets. [4] employs a similar idea and proposes a two-step model where it first uses a neural network to predict the active sets, and then it solves a reduced OPF problem where the non-binding constraints are removed from the problem. The loss function used for training the neural network is a meta-loss objective. The neural network in this model acts as an initialization method for the reduced OPF solver. In [5] a similar idea has been investigated where multi-candidates for the active set are used to increase the accuracy of the model. [6] study the use of active sets in solving optimization problems. The idea is to use the active sets, predicted using the neural networks, as additional features to the learning models that predicts the final optimal solutions. In [7] authors improve the generalization of the model in [3] by using the Input Convex Neural Network (ICNN).

In this paper we take a new look at predicting the optimal active set in solving DC-OPF problems. The key property we exploit for improving the learning efficiency is that (under reasonable problem modeling) the set of input parameters that lead to the same optimal active set are *polyhedrons, i.e., linearly separable*. This prompts us to investigate predictor models that *by definition guarantees* that the boundaries between the convex decision regions of different classes are linear: Linear discriminant analysis (LDA) and support vector machines (SVM). In particular, we show that LDA achieves a *testing* performance — both in accuracy and in run-time — similar to much more complicated predictors such as carefully tuned and trained neural networks. At the same time, the simplicity of LDA allows it to be *trained* a) many orders of magnitude faster than neural networks, and b) without the need of hyper-parameter tuning and straightforwardly applicable to any problem setting.

II. System Model and Properties

In this section we first study the general convex quadratic program with parametric inputs. We show that the set of input parameters corresponding to the same optimal active set is a polyhedron. We then focus the rest of the paper on the direct current optimal power flow (DC-OPF) problem. DC-OPF is an application of the convex quadratic program with parametric inputs which needs to be solved frequently and in large numbers by power system operators.

A. Convex quadratic program with parametric input

The general form of the convex quadratic optimization problem with parametric input is as follows:

$$\min_{x} \frac{1}{2} x^{\intercal} H x + g^{\intercal} x \tag{1}$$

$$\text{s.t.} \quad Cx + Dw = 0, \quad (\nu), \tag{2}$$

$$Ax + Bw \leq 0, \quad (\lambda), \tag{3}$$

In this formulation, w is the input parameter of the problem, and λ and ν are dual variables of the constraints.

Any solution x needs to be feasible to be a valid candidate for the *optimal solution*, meaning that it needs to satisfy the equality and inequality constraints in (2) and (3). For a solution, some of the inequality constraints in (3) may be binding with the rest non-binding. We define *active sets* as follows to include the binding constraints.

Definition II.1. ***Active Set:*** *For a feasible solution of* (1)-(3)*, the corresponding active set is the set of equality constraints in* (2) *as well as the set of binding constraints in* (3)*.*

Definition II.2. ***Optimal Active Set:*** *An optimal active set is the active set corresponding to an optimal solution of the optimization problem.*

We now have the following theorem.

Theorem 1. *In convex quadratic program with parametric input w in* (1)-(3)*, the input space of w corresponding to different optimal active sets are convex sets.*

Proof. We assume that the matrix H is a positive definite matrix, making (1)-(3) a strictly convex optimization problem. Assume that w_1 and w_2 are two input vectors corresponding to the same optimal active set, meaning that if we solve (1)-(3) for w_1 and for w_2, we get the same optimal active set for both of them, which we indicate it as $\mathcal{AS}^+$. Theorem 1 states that, if we pick any convex combination of w_1 and w_2, the corresponding optimal solution of (1)-(3) has the same active set, *i.e.* $\mathcal{AS}^+$.
To be precise, it is clear that for the input vectors w_1 and w_2, if (x_1, λ_1, ν_1) and (x_2, λ_2, ν_2) are the solutions of Karush–Kuhn–Tucker (KKT) conditions in (4) for w_1 and w_2, then, $\forall \alpha \in [0,1]$, for the input vector $w_3 = \alpha w_1 + (1-\alpha) w_2$, $\big\{x_3 = \alpha x_1 + (1-\alpha) x_2, \lambda_3 = \alpha \lambda_1 + (1-\alpha)\lambda_2, \nu_3 = \alpha \nu_1 + (1-\alpha)\nu_2\big\}$ with the same active set, *i.e.* $\mathcal{AS}^3 = \mathcal{AS}^+$, is also a solution to the KKT conditions in (4):

$$\begin{cases} (Ax+Bw)_i = 0, i \in \mathcal{AS}; & \lambda_i > 0, i \in \mathcal{AS}; \\ (Ax+Bw)_i < 0, i \notin \mathcal{AS}; & \lambda_i = 0, i \notin \mathcal{AS}; \\ Hx + g + A^{\intercal}\lambda + C^{\intercal}\nu = 0; & Cx + Dw = 0. \end{cases} \tag{4}$$

This completes the proof. □

Remark II.1. *The uniqueness of the optimal solution of* (1)-(3) *and the separating hyperplane theorem, cf. [8], indicate that, given that the input spaces corresponding to different optimal active sets of* (1)-(3) *are convex regions, their disjoint interior are linearly separable. This implies that these convex sets are all polyhedrons.*

B. DC-OPF Formulation

Optimal power flow is a dispatch scheduling problem that power system operators solve frequently to keep the supply and demand for electricity balanced. We now review the DC-OPF formulations which is an application of the convex quadratic program with parametric input in (1)-(3).

The DC-OPF problem which is solved for economic dispatch is formulated as follows:

$$\min_{\{p_g, g\in\mathcal{G}\}} \sum_{g\in\mathcal{G}} C_g(p_g) \tag{5}$$

$$\text{s.t.} \quad \boldsymbol{T} = [\mathbf{PTDF}] \cdot (\boldsymbol{w} - \mathbf{U} \cdot \boldsymbol{p}) \tag{6}$$

$$|T_{i,j}| \leq \overline{T}_{i,j}, \quad \forall (i,j) \in \mathcal{T}, \tag{7}$$

$$p_g^{min} \leq p_g \leq p_g^{max}, \quad \forall g \in \mathcal{G}, \tag{8}$$

$$\sum_{(i,j)\in\mathcal{T}} T_{i,j} + w_i - \sum_{g\in\mathcal{G}_i} p_g = 0, \quad \forall i \in \mathcal{N}, \tag{9}$$

The generator's cost functions are quadratic functions in the form of $C_g(p_g) = a_g p_g^2 + b_g p_g + c_g$, where p_g is the generation amount of generator g, and a_g, b_g and c_g are its cost coefficients. $T_{i,j}$ and $\overline{T}_{i,j}$ are the power flow on the line (i,j) and the capacity of the line respectively. p_g, p_g^{min} and p_g^{max} are the generator g's dispatch and its lower and upper bounds. $\mathcal{N}$, $\mathcal{G}$ and $\mathcal{T}$ are the sets of buses, generators and lines, respectively. $\mathcal{G}_i$ is the set of generators located on bus i. The matrix $\mathbf{U}$ maps the generators to the buses they are located on. $[\mathbf{PTDF}]$ is the matrix of Power Transfer Distribution Factors mapping the net nodal power injections, *i.e.* $(\boldsymbol{w}-\mathbf{U}\cdot\boldsymbol{p})$, to line flows. The input parameter of this problem is the vector of net demand on the buses, $\boldsymbol{w}$, which is the total demand net of any renewable power generation on the buses. The total number of *possible* active sets (among the inequality constraints of line flow capacities and generator bounds) for a DC-OPF problem is $3^{|\mathcal{G}|+|\mathcal{T}|}$, where $|\cdot|$ is the cardinality of a set, as each generator or line can either be at its upper/lower bound or be strictly in between. In practice, however, the *observed* optimal active sets of DC-OPF problem are limited [6].

Next, we note that given the optimal active set of (5)-(9), solving the DC-OPF problem reduces to solving a system of linear equations corresponding to the KKT equations, for which its solution is moreover a *linear function* of the net nodal demands $\boldsymbol{w}$. The coefficients of the linear function, denoted by L_k for the k^{th} active set, can be computed directly based on the power system parameters.

Notably, if one can correctly predict the optimal active set for the DC-OPF problem, finding the optimal solution reduces to a simple task of calling the appropriate linear function corresponding to that active set. Hence, the main focus in the two-step learning-based solvers is on predicting the optimal active set. For the rest of the paper we focus on the DC-OPF problem in (5)-(9). Without loss of generality, the proposed approaches can be applied to the general quadratic program in (1)-(3) as well.

III. Fast Learning and Inference of Optimal Active Set in DC-OPF

In this paper, similarly to [3] and [5], we use a two-step learning-based approach for solving DC-OPF. At the first step we use a classification model to return active set candidates for the optimal active set. In the second step, for each active set candidate, we find the corresponding solution to the DC-OPF, check for feasibility, and finally pick the best solution among them. The general structure of the two-step learning-based approach for solving DC-OPF is depicted in Fig. 1.

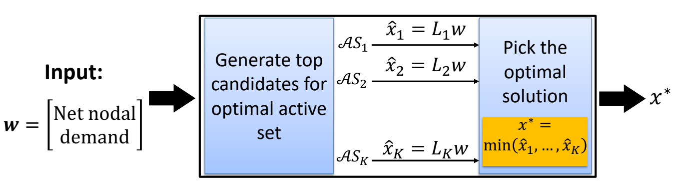

Fig. 1: Diagram of the proposed learning-based method for finding optimal solution of DC-OPF.

The input of the predictor is the net nodal injections. By *"net"* we mean the nodal demand net of any renewable power generation on a bus. We then predict the optimal active set, and find the optimal solution corresponding to that active set. As detailed in Section II-B, once an active set is predicted in step 1, the step 2 of computing the full optimal dispatch solution is simply calling a pre-determined linear function of the input parameter (i.e., the net demands). Accordingly, the focus of this paper is on the first step of the solution process, *i.e.* predicting the optimal active set, which is a multi-class classification task.

Among different potential classification methods, existing works have mainly focused on using *neural networks* for predicting optimal active set. Undoubtedly, neural network is a powerful tool for classification tasks, and our simulations (cf. Section IV) confirms its high performance in predicting optimal active sets. Nonetheless, the linear separability of classes of active sets in the input space (cf. Theorem 1) motivates us to explore other predictor models that can capture this special property of DC-OPF.

In particular, we investigate LDA and SVM as the predictor models for this problem, because the classification decision boundaries for both of these models are by design linear. Furthermore, instead of returning a single active set candidate, we let the predictor return multiple candidates for the optimal active set. As will be shown in Section IV, this strategy can significantly increase the prediction accuracy at a modest increase in computation time. In what follows we review SVM and LDA. We then discuss the benefits of returning multiple candidates of active set rather than one candidate.

Support Vector Machine: Theorem 1 and Remark II.1 show that the classes of optimal active sets in the input parameter space are linearly separable. As a result, SVM is an intuitive model choice for the classifier. Its effectiveness is confirmed by the numerical results to be shown later in Section IV as SVM has the best performance when returning only one candidate of active set. However, there are some major problems in using SVM for active set classification that severely limit its practicality for the DC-OPF problem:

- **Quadratic number of classifiers and computation time:** Given that every pair of active set classes are linearly separable, the best *multi-class* classification performance using SVM can be achieved with *a number of* "one-vs-one" SVM classifiers, one for every pair of classes. Now, assuming that there exists Γ classes, we need to train $\frac{\Gamma*(\Gamma-1)}{2}$ SVM classifiers. While this complexity is not much of a critical issue for training as the training phase is offline, this computation complexity that grows quadratically with the number of active sets is indeed a practical challenge during online testing/usage of multi-class SVM. We will provide a detailed cost-benefit analysis in the simulation study (cf. Section IV).
- **Inefficient extension to multi-candidate prediction:** SVM is a maximum margin method for classification, and hence does not directly return probabilities for different classes. When we would like multiple candidates for the optimal class to be returned, we need a ranking of the classes. While there are techniques to obtain approximate posterior probabilities from SVM, such as Platt Calibration [9], however, their performance is sometimes not as desirable. Moreover, this extra step of calculating the probabilities for SVM can add considerable computational time during online testing.

Linear Discriminative Analysis: LDA is an alternative choice that is by design tailored to classification tasks with linearly separated decision regions, and is especially convenient for *multi-class* classification. The intuition of LDA is to maximize the between-class variances while minimizing the within-class variances [10]. The advantages of LDA in comparison to SVM for our problem are:

- **Linear test time and fast training:** The computational time for testing a new data sample is *linear* in the number of the active set classes. Furthermore, in the training phase, LDA only needs to estimate the population parameters from the samples (whereas SVM, on the other hand, solves $\frac{\Gamma*(\Gamma-1)}{2}$ optimization problems). As such, not only LDA's testing speed is much faster than multi-class SVM, the speed advantage is even greater in training as *LDA's training is extremely fast*.
- **Built-in extension to multi-candidate prediction:** Since LDA works with the posterior probabilities, it is straightforward to return multiple candidates.

Next we briefly discuss the idea of multi-candidate predictions for active set classification.

Multi-Candidate Prediction: Classification algorithms usually assign scores to different classes, and pick the class with the highest score as the predicted label of the data sample. Alternatively, more than one candidate who have the highest scores may be returned. This is particularly useful when subsequent procedures exist to verify which of the multiple candidates returned is the best among them. This is exactly the case for the two-step approach we employ for solving DC-OPF. After the classification algorithm assigns scores to the classes, we pick the top-K active set candidates. We can then calculate the full dispatch solution for each active set candidate, check the solutions for feasibility, and compute the corresponding system cost, i.e., the value of the objective function in (5) for this solution. *Only the feasible solution with the lowest system cost can possibly be the correct optimal solution*, and should be the one we predict. Since computing the optimal solution and the system cost for each active set is just one pass of the second step and is computationally very fast, returning multiple candidates can be very advantageous as accuracy may be significantly improved with only a small increase in run-time.

IV. Simulations

We use the IEEE-162-dtc (DC) system available at [11] for the simulations. In the following simulations we placed 37 renewable power producers (RPPs) on the following buses 5, 6, 9, 17, 17, 18, 18, 18, 22, 27, 27, 28, 30, 36, 36, 38, 39, 47, 50, 57, 73, 74, 79, 82, 84, 87, 90, 91, 103, 108, 111, 114, 120, 125, 129, 151, and 160. We assume that there is a 50% renewable penetration in the system, meaning that the average total generation of the RPPs is 50% of the average total load. We also simulate RPPs' generation with i.i.d. normal distributions. *We assume that the coefficient of variation (COV) of an RPP's probability distribution equals 30%, which is the ratio of the standard deviation to the mean of the generation distribution*. The 30% COV of renewable energies represents typical day ahead renewable energy (e.g., wind) forecast error. We further assume that the demand at each bus is independent of that on other buses, follows a normal distribution with the mean equal to the load data provided in the IEEE-162-dtc (DC) benchmark case in [11], and *a COV equal to* 6%. The 6% COV of demands is found to represent the day ahead forecast error of large and aggregated demand [12]. We generated 55276 samples with *319 distinctive classes of active sets*. 80% of data is used for training and the rest for testing. Figure 2 shows the cumulative probability of the classes in the data set where the classes of active sets are sorted from the most likely to the least likely. In other words, the most frequently appeared class appears for just under 20% of the time, and so on. The input for each classifier in the simulations is the net nodal demand where RPPs' generation are considered as negative loads, and hence deducted from the nodal demand on the buses.

TABLE I: Accuracy for different classification methods

	LDA%	NN%	XGBoost-71 %	SVM-134 %
Top 1-Candidate	62.44	68.69	54.14	85.71
Top 2-Candidate	76.05	84.41	74.02	92.71
Top 3-Candidate	83.60	89.45	82.88	94.92
Top 4-Candidate	88.40	91.97	87.69	96.08
Top 5-Candidate	91.51	93.44	90.39	96.72
Top 6-Candidate	93.51	94.55	92.13	97.16
Top 7-Candidate	94.80	95.32	93.42	97.49
Top 8-Candidate	95.62	95.89	94.10	97.77
Top 9-Candidate	96.18	96.36	94.84	97.92
Top 10-Candidate	96.61	96.81	95.34	98.13
Top 11-Candidate	97.03	97.10	95.74	98.32
Top 12-Candidate	97.24	97.33	96.03	98.38
Top 13-Candidate	97.44	97.45	96.28	98.48
Top 14-Candidate	97.61	97.75	96.57	98.56
Top 15-Candidate	97.80	97.92	96.78	98.59

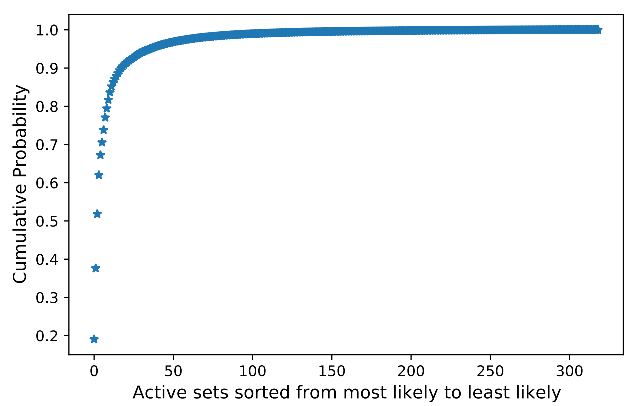

Fig. 2: Cumulative probability distribution of the active sets.

We begin by comparing the accuracy of different classification algorithms for classifying the active sets. These algorithms include LDA, Neural Networks (NN), XGBoost-71 where the 71 most frequently appeared classes are considered [13], and SVM-124 where the 124 most frequently appeared classes are considered. The corresponding accuracy of these algorithms is shown in Table I. The NN used in Table I is a 3-layer neural network with 810 neurons in the first two layers and

Relu activation functions. XGBoost-71 is a gradient tree boosting classifier which considers the top 71 classes. We note that we have also investigated many other algorithms and the above presented are the top performing ones that are the most competitive.

From Table I, we see that *solely from the accuracy's perspective*, the best performance consistently belongs to SVM. The intuitive reason is that SVM performs very well on linearly separable data. The performance of the LDA and NN are close, and the performance of the XGBoost algorithm closely follows them.

However, the above results did not present an important element of the algorithms — run-time. In learning-based approaches (and in fact in all algorithms), there is always a trade-off between accuracy and run-time: The longer time an algorithm is allowed to run, the better accuracy can often result from it. As such, to compare algorithms *fairly*, one needs to compare the accuracy achieved under the same run-time constraints. We now look into the accuracy/run-time comparisons of the algorithms. Figure 3 depicts the accuracy of the classification algorithms versus the actual total run-time they take for evaluating a set of 11056 testing samples[1].

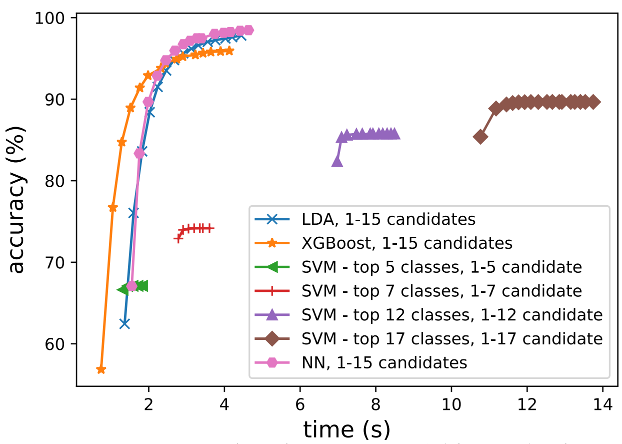

Fig. 3: Accuracy vs testing time (over 11056 samples in total) for different classification methods.

It is clear that, as expected, SVM requires significantly higher run-times than the other tested algorithms to achieve a similar accuracy. Note that, to maintain good readability of the figure we only plot SVM with top 5, 7, 12, and 17 classes, while the performance of the SVM in Table I is achieved by SVM with top 134 classes. This reveals that, in practice, SVM is in fact the least desired among the four algorithms. In comparison, LDA, NN and XGBoost have much faster computation in testing.

In particular, we would like to highlight the unique competitive advantage of LDA. As a very simple algorithm, LDA's testing performance is as good as carefully trained neural networks. This is quite remarkable because a) the *training time* of LDA is many orders of magnitude shorter (at least 1000 times faster) than that of neural networks, and b) there's almost no hyper-parameter to tune for LDA's training, whereas training a good neural network often requires huge manual efforts of tuning hyper-parameters. Moreover, LDA's computation in testing can be trivially parallelized, which would further reduce the test time significantly. Similar comparison remarks can be made between LDA and XGBoost. The extremely fast and simple *training and testing*, together with its very competitive performance compared with much more complicated algorithms, make LDA a very attractive active set predictor model in practice. The key to such a success of LDA is that it inherently captures the underlying physical property of the problem — the input space for different active set classes are polyhedrons and linearly separable.

[1] All testing times are measured on a laptop with an Intel Core i7 2.6-GHz CPU with 16 GB of RAM.

V. Conclusion

This paper investigates the idea of using linear discriminant analysis (LDA) classifier for predicting the optimal active sets in DC-OPF, a critical step in computing the optimal solution of DC-OPF. We show that LDA is particularly suitable for this task because (a) it inherently captures the linear separability of the classes of optimal active sets in the input space, leading to highly competitive testing accuracy, and (b) it is extremely fast in *both training and testing*, making it quickly and easily applicable to new problem settings.

References

[1] M. B. Cain, R. P. O'neill, and A. Castillo, "History of optimal power flow and formulations," *Federal Energy Regulatory Commission*, vol. 1, pp. 1–36, 2012.

[2] X. Pan, T. Zhao, and M. Chen, "Deepopf: Deep neural network for dc optimal power flow," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019, pp. 1–6.

[3] Y. Chen and B. Zhang, "Learning to solve network flow problems via neural decoding," 2020. [Online]. Available: https://arxiv.org/abs/2002.04091

[4] A. Robson, M. Jamei, C. Ududec, and L. Mones, "Learning an optimally reduced formulation of opf through meta-optimization," 2019. [Online]. Available: https://arxiv.org/abs/1911.06784

[5] D. Deka and S. Misra, "Learning for dc-opf: Classifying active sets using neural nets," 2019. [Online]. Available: https://arxiv.org/abs/1902.05607

[6] S. Misra, L. Roald, and Y. Ng, "Learning for constrained optimization: Identifying optimal active constraint sets," 2019. [Online]. Available: https://arxiv.org/abs/1802.09639

[7] L. Zhang, Y. Chen, and B. Zhang, "A convex neural network solver for dcopf with generalization guarantees," 2020. [Online]. Available: https://arxiv.org/abs/2009.09109

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Press, 2004.

[9] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 625–632.

[10] S. Theodoridis, *Machine Learning; A Bayesian and Optimization Perspective*. Academic Press, 2015.

[11] C. Coffrin. A library of ieee pes power grid benchmarks. [Online]. Available: https://github.com/power-grid-lib/pglib-opf

[12] R. Sevlian and R. Rajagopal, "A scaling law for short term load forecasting on varying levels of aggregation," *International Journal of Electrical Power & Energy Systems*, vol. 98, pp. 350–361, 2018.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794.