Ensemble Learning based Linear Power Flow

Ren Hu, Qifeng Li
Dept. of Electrical & Computer Engineering
Dept. of Statistics & Data Science
University of Central Florida
Orlando, USA

Shunbo Lei
Dept. of Electrical Engineering& Computer Science
University of Michigan
Ann Arbor, USA

***Abstract*—This paper develops an ensemble learning-based linearization approach for power flow with reactive power modeled, where the polynomial regression (PR) is first used as a basic learner to capture the linear relationships between the bus voltages as the independent variables and the active or reactive power as the dependent variable in rectangular coordinates. Then, gradient boosting (GB) and bagging as ensemble learning methods are introduced to combine all basic learners to boost the model performance. The inferred linear power flow model is applied to solve the well-known optimal power flow (OPF) problem. The simulation results on IEEE standard power systems indicate that (1) ensemble learning methods can significantly improve the efficiency of PR, and GB works better than bagging; (2) as for solving OPF, the data-driven model outperforms the DC model and the SDP relaxation in both accuracy, and computational efficiency.**

***Index Terms*-- Power flow, data-driven, ensemble learning, gradient boosting, bagging.**

I. INTRODUCTION

Power flow analysis is an indispensable part for power system planning and operation. However, the nonlinear and nonconvex properties of power flow equations restrict the convergency and computational speed of power flow solutions, especially in large scale power systems, which attracts many researchers. One of the typical measures is to linearize the nonlinear alternating current (AC) power flow model into linear models, such as the direct current (DC) power flow that maps the linear relationship between the active power and the voltage phase angle. Other extended linearized models of power flow can also be found in [1]- [7]. Linear approximations of the active and reactive power demands in distribution networks are proposed in [1], and the sufficient conditions for the existence of the power flow solution are also investigated. Similarly, in [2], a linear power flow for three-phase distribution systems is proposed considering both balanced and unbalanced conditions with ZIP load models. Other linearized models are presented in [3]-[7] through decoupling voltage magnitude and phase angle.

Although these approaches enhance the accuracy beyond the DC power flow, most of them are based on assumptions of simplifying network parameters to reach the linearization of power flow. In [8], variable-selection based methods are exploited to obtain the linearization of power flow, however, they indirectly change the bus number, even replace the original voltage variables by new variables and lead to the poor applicability in optimization and control problems. In the presence of massive databases and sufficient measurements related to power system operation, data-driven approaches attract more and more attentions, with applications in estimating Jacobian matrix [9] distribution factors [10], and the admittance matrix [11]. In this paper, our contributions are summarized as: 1) the polynomial regression [12], [13] is used as a basic learner to infer the linear map between the bus voltages as the independent variables and the active or reactive power as the dependent variable; gradient boosting [14], [15] and bagging [16], [17] as ensemble learning techniques are leveraged to assemble all basic learners to improve the accuracy of the learned model; 2) the resulting linear power flow model is used to construct a linear optimal power flow (OPF) model.

The rest of this paper is organized as below. Section II illustrates the existing problems and solutions for computing and fitting power flow. The data driven linear power flow model is inferred in section III through ensemble learning methods. Eventually, the simulations and conclusions on some IEEE cases are displayed in Section IV and V.

II. PROBLEM FORMULATION AND STATEMENT

This section depicts the problem formulation and existing solutions in computing optimal power flow. Some challenges and tactics in fitting power flow are also discussed from the perspective of data mining.

A. Problem Formulation and Solutions

In general, the nonlinear AC power flow (ACPF) formulations in an n-bus power system can be described by the following equations in (1)~(10):

$$P_i = e_i \sum_{j=1}^{n}(G_{ij}e_j - B_{ij}f_j) + f_i \sum_{j=1}^{n}(G_{ij}f_j + B_{ij}e_j) \quad (1)$$

$$Q_i = f_i \sum_{j=1}^{n}(G_{ij}e_j - B_{ij}f_j) - e_i \sum_{j=1}^{n}(G_{ij}f_j + B_{ij}e_j) \quad (2)$$

$$P_{ij} = G_{ij}e_ie_j - B_{ij}e_if_j + B_{ij}f_ie_j + G_{ij}f_if_j - G_{ij}(e_i^2 + f_i^2) \quad (3)$$

$$Q_{ij} = G_{ij}f_ie_j - B_{ij}e_ie_j - G_{ij}e_if_j - B_{ij}f_if_j + B_{ij}(e_i^2 + f_i^2) \quad (4)$$

This work is supported by the U.S. National Science Foundation under Grant #1808988.

978-1-7281-5508-1/20/$31.00 ©2020 IEEE

$$P_i = P_i^G - P_i^L \quad (5)$$

$$Q_i = Q_i^G - Q_i^L \quad (6)$$

$$P_i^{Gmin} \le P_i^G \le P_i^{Gmax} \quad (7)$$

$$Q_i^{Gmin} \le Q_i^G \le Q_i^{Gmax} \quad (8)$$

$$P_{ij}^2 + Q_{ij}^2 \le S_{ij}^{Gmax} \quad (9)$$

$$V_i^{min2} \le e_i^2 + f_i^2 \le V_i^{max2} \quad (10)$$

where P_i, Q_i, e_i, f_i denotes the active power injection, reactive power injection, the real and imaginary parts of voltage at bus i; $P_{ij}, Q_{ij}, G_{ij}, B_{ij}$ are the active and reactive line flow, the real and imaginary parts of the line admittance between bus i and bus j; G is the index set of generators; P_i^G, Q_i^G are the active and reactive power of the *i*-th generator; P_i^L, Q_i^L are the active and reactive power load at *i-th* bus; P_i^{Gmin}, P_i^{Gmax}, Q_i^{Gmin}, Q_i^{Gmax} are the lower and upper limits of active and reactive power of the *i*-th generator; V_i^{max}, S_{ij}^{Gmax} are the maximums of the *i*-th bus voltage and the *ij*-th branch transmission capacity.

From the aspect of data mining, in (1)~(4) we treat e_i and f_i as independent variables, while P_i ,Q_i, P_{ij}, Q_{ij} are treated as dependent variables. Though the linearization methods of power flow based on the assumptions of simplifying network parameters, to some extent, accelerate the power flow computation [1]- [8], their accuracy can be unsatisfactory. For advancing the computational accuracy and efficiency of algorithms and making the full use of big data techniques, the data mining method seems to be worth exploring with the accessible data pertaining to power system operation.

B. Fitting Challenges and Remedies

One of the most common problems in machine learning is overfitting. An overfitted model usually fits the training dataset perfectly, inversely, fails to predict the test dataset reliably. Correspondingly, ensemble learning [17], [18] approaches are recommended to avoid overfitting when learning the linear power flow model, through regularizing learning parameters. Another fitting issue is to handle the multicollinear correlations within the bus voltages as the independent variables, which is actually caused by insufficient data to reveal their true associations. Although many variable selection or shrinkage methods [19] have been suggested to meliorate the multicollinearity through reducing the number of independent variables, they may indirectly change the bus number and remove crucial variables for further optimization and control application in power system. Therefore, in this paper, we address the multicollinearity via increasing more datasets in model fitting to avoid removing variables.

III. ENSEMBLE LEARNING BASED LINEAR POWER FLOW

This section presents full details of the proposed approach, which includes: 1) the mapping rules of linear power flow, 2) ensemble learning methodologies, and 3) the formulation of OPF based on the learned linear models.

A. Linear Mapping Formulations

Based on the data mining techniques, the nonlinear quadratic forms of AC power flow in (1)~(4) can be approximated to linear formulations in rectangular coordinates as below.

$$P_i = A_i X + b_i \quad (11)$$

$$Q_i = C_i X + d_i \quad (12)$$

$$P_{ij} = A_{ij} X_{ij} + b_{ij} \quad (13)$$

$$Q_{ij} = C_{ij} X_{ij} + d_{ij} \quad (14)$$

where A_i, C_i, A_{ij}, C_{ij} are the corresponding linear coefficient vectors; b_i , d_i , b_{ij} , d_{ij} represent the constant terms; $X = [e_1\ f_1, .., e_n\ f_n]^T = [x_1\ x_2, .., x_{2n}]^T$, $X_{ij} = [e_i\ f_i\ e_j\ f_j]^T = \left[x_{2i-1}\ x_{2i}\ x_{2j-1}\ x_{2j}\right]^T$ are the bus voltage vectors.

B. Ensemble Learning Methodologies

Gradient boosting (GB) and bagging as two typical ensemble learning methods are leveraged in this paper to reinforce the polynomial regression (PR) as a basic learner and compute all linear coefficient vectors and constant terms in (2). Assume that there is a dataset containing M samples $\{(X_m, Y_m)\}_{m=1}^M$ where $X_m = [x_{m1}, x_{m2}, .., x_{m(2n)}]$ is the m-th sample of bus voltages as the independent variables; Y_m is the m-th sample of the active or reactive power at all buses or branches, for instance, $Y_m = [p_{m1}, p_{m2}, .., p_{mn}] = \{p_{mi}\}$ is the m-th sample of the active power at all buses. We take the dependent variable P_i as a general example to illustrate how to apply GB and bagging.

1. *Gradient Boosting*

Generally, GB tweaks learning parameters in an iterative fashion to find the minimum descending gradient through minimizing a loss function. We choose the mean squared error function as the specific loss function in (3)

$$L(p_{mi}, \widehat{p_{mi}}) = \frac{1}{2}(p_{mi} - \widehat{p_{mi}})^2 \quad (15)$$

where p_{mi} and $\widehat{p_{mi}} = p_{mi}(X)$ are the observed and estimated values of P_i. The procedure of GB is depicted as below.

Algorithm: gradient boosting

1. Initialize the model with a constant value $\boldsymbol{p_{mi}^0(X)}$.

$$\boldsymbol{p_{mi}^0(X) = arg\,min_\alpha \sum_{m=1}^M L(p_{mi}, \alpha)} \quad (16)$$

where $\boldsymbol{\alpha}$ is the initial constant vector.

2. For $\boldsymbol{t = 1}$ to **T** where **T** is the number of learners.

1) Compute the descending gradient $\boldsymbol{\gamma_t}$ by

$$\boldsymbol{\gamma_t = -\left[\frac{\partial L(p_{mi}, p_{mi}(X))}{\partial p_{mi}(X)}\right]_{p_{mi}(X) = p_{mi}^{t-1}(X)}} \quad (17)$$

2) Fit a base learner $\boldsymbol{\varphi_t(X;\rho)}$ by

$$\boldsymbol{\rho_t = arg\,min_\theta \sum_{m=1}^M L(\gamma_t, \varphi_t(X_m;\rho))} \quad (18)$$

where $\boldsymbol{\rho_t}$ is the coefficient vector of $\boldsymbol{\varphi_t(X;\rho)}$ by fitting $\boldsymbol{\gamma_t}$. Here the polynomial regression (PR) as a basic learner is used to fit the key parameters in equations (11)~(14).

3) Compute the learning rate $\boldsymbol{\mu}$ **by**

$$\boldsymbol{\mu_t = arg\,min_\mu \sum_{m=1}^M L(p_{mi}, p_{mi}^{t-1}(X_m) + \mu\varphi_t(X;\rho))} \quad (19)$$

It is also allowed to set a constant learning rate. The smaller $\boldsymbol{\mu}$ is applied, the better generalization is achieved.

4) Update the model:

$$\boldsymbol{p_{mi}^t(X) = p_{mi}^{t-1}(X) + \mu_t\varphi_t(X)} \quad (20)$$

3. Output $\boldsymbol{p_{mi}^T(X)}$

2. *Bagging*

Bagging as one of the model averaging approaches can reduce variances and avoid overfitting by adjusting the number of bootstraps. The key of bagging is to draw random samples with replacement and combine a basic learning method to train models. The algorithm is given as below.

Algorithm: bagging

1. For $\boldsymbol{bt} = \mathbf{1}$ to **BT** where **BT** is the number of bootstraps.
   1) At the *bt*-th bootstrap draw $\boldsymbol{M'}$ $(\boldsymbol{M'} \leq \boldsymbol{M})$ random samples with replacement.
   2) Fit a base learner $\boldsymbol{p_{mi}^{bt}(X;\rho)}$ by
   $$\boldsymbol{\rho_{bt} = arg\,min_{\rho} \sum_{m=1}^{M} L(p_{mi}^{bt}, p_{mi}^{bt}(X_m;\rho))} \quad (21)$$
   where $\boldsymbol{p_{mi}^{bt}}$ denotes the observed value of $\boldsymbol{P_i}$ at the *bt*-th bootstrap; $\boldsymbol{\rho_{bt}}$ is the coefficient vector of $\boldsymbol{p_{mi}^{bt}(X;\rho)}$ by fitting $\boldsymbol{p_{mi}^{bt}}$. Similarly, the PR as the basic learner is performed to compute all parameters in Section A.
2. Output $\boldsymbol{p_{mi}^{bag}(X)}$ by averaging all bootstrap results in
   $$\boldsymbol{p_{mi}^{bag}(X) = \frac{1}{BT}\sum_{bt=1}^{BT} p_{mi}^{bt}(X)} \quad (22)$$
   where $\boldsymbol{p_{mi}^{bag}(X)}$ is the predictive value of $\boldsymbol{P_i}$.

C. Convexifying the Optimal Power Flow

According to the fitted linear power flows, the data driven convex approximation (DDCA) for OPF can be rewritten as

$$\text{Minimize} \sum_{i\in G}(c_{i0} + c_{i1}P_i^G + c_{i2}P_i^{G2}) \quad (23)$$
$$A_iX + b_i = P_i = P_i^G - P_i^L \quad (24)$$
$$C_iX + d_i = Q_i = Q_i^G - Q_i^L \quad (25)$$
$$x_{2i-1}^2 + x_{2i}^2 \leq V_i^{max2} \quad (26)$$
$$P_i^{Gmin} \leq P_i^G \leq P_i^{Gmax} \quad (27)$$
$$Q_i^{Gmin} \leq Q_i^G \leq Q_i^{Gmax} \quad (28)$$
$$P_{ij}^2 + Q_{ij}^2 \leq S_{ij}^{Gmax} \quad (29)$$
$$A_{ij}X_{ij} + b_{ij} = P_{ij} \quad (30)$$
$$C_{ij}X_{ij} + d_{ij} = Q_{ij} \quad (31)$$

where c_{i0}, c_{i1}, c_{i2} are the *i*-th generator cost coefficients; x_{2i-1} and x_{2i} (e_i and f_i) are the real and imaginary part of voltage. The learned linear power flows transform (23)~(31) into a convex optimization problem which is more tractable than the originally nonconvex ACOPF problem. The DDCA for OPF tends to have much simpler formulations and calculations than the semidefinite programming (SDP) relaxation. As the proposed linear power flow model applies the data of power system operation without assumptions of the DC model and takes the reactive power into account, it seems to be much closer to the original ACPF and more accurate than the DC model.

IV. SIMULATION ANALYSIS

In this study, Monte Carlo method is adopted to generate random data samples of the bus voltages, the active and reactive power at each bus or branch, given in IEEE 5-, 57- and 118-bus systems [20]. The active and reactive power loads are stochastically fluctuating around 0.6~1.1 times of preset values. Generally, the empirical required minimum sample size is at least 2.4 times as many as the number of buses [8], [10], [19].

A. Comparing Performance of Predictive Accuracy

The simulation results through the polynomial regression (PR), gradient boosting (GB), and bagging, are obtained by the equal size of training and test datasets. The average root mean square error (RMSE) of the predicted dependent variable is used to measure the predictive accuracy, and the performance demonstration of all methods is characterized by comparing the test RMSEs, not the training RMSEs, shown in TABLE I.

TABLE I TEST AND TRAINING RMSES OF ALL METHODS

Case	Method	PR		GB		Bagging	
RMSE (10 e-05)		test	training	test	training	test	training
case 5 (size=175, T=200, BT=50)	P	9476.04	11.22	64.08	59.01	310.04	112.91
	Q	2247.02	556.30	395.06	353.26	1333.24	570.53
	P_{ij}	248.11	76.50	43.92	39.81	144.66	88.62
	Q_{ij}	1812.00	323.00	179.32	175.83	385.72	341.62
case 57 (size=250, T=200, BT=50)	P	67.30	4.63	17.27	6.50	26.43	5.90
	Q	237.99	18.14	59.36	22.55	88.93	19.77
	P_{ij}	23.52	17.20	17.27	6.51	19.73	18.04
	Q_{ij}	63.04	42.50	52.65	52.39	54.51	50.09
case 118 (size=400, T=200, BT=50)	P	100.33	15.87	41.65	17.12	82.17	16.07
	Q	180.27	30.45	79.06	32.22	161.17	31.31
	P_{ij}	56.97	20.80	20.64	20.71	25.90	21.56
	Q_{ij}	117.57	57.09	63.24	62.56	75.44	58.60

Note that the unit of above data is 10 e-05.

From TABLE I, some conclusions can be observed on all cases: 1) ensemble learning methods work better than the single learner PR; 2) GB outperforms bagging and PR; 3) there is no pronounced overfitting as the training RMSE is consistently smaller than the test RMSE for any dependent variable.

B. Tuning Parameters

As the model performance described by the test and training RMSEs is directly associated with a set of regularization parameters to handle overfitting, this subsection presents the tuning processes of GB and bagging, respectively, with regard to the number of learners T and the number of bootstraps BT.

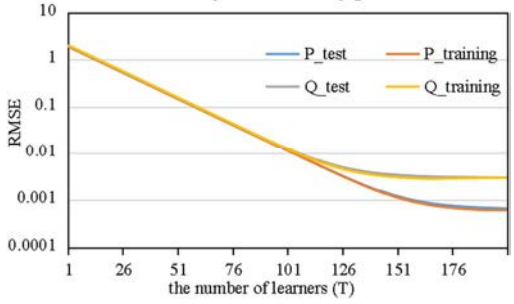
1. *Tuning the Number of Learners T in GB*

The model performances of the active and reactive power injections *P* and *Q* on all cases are plotted in the logarithms of RMSEs in Fig. 1. (a)~(c). Each plot depicts how the RMSEs change as the number of learners T increases. From Fig. 1, we can observe that:

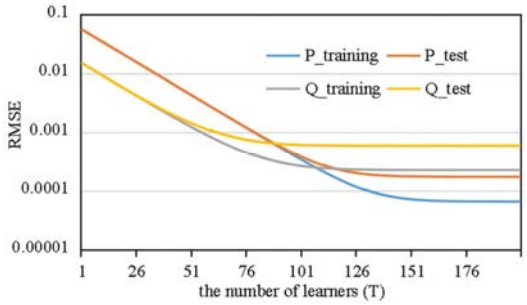
(1) For all cases, the test and training RMSEs of *P* and *Q* gradually decrease to stable levels with the increase of the number of learners T;

(2) Each case reaches the balance points at different numbers of learners. For case 5, after T=150 and T=180 the test and training RMSEs of *Q* and *P* tend to be constant, separately. The RMSEs of case 57 become hardly changeable when T=100 for *Q* and T=150 for *P*. Similarly, the RMSEs of case 118 stop descending when T=140 for *Q* and T=180 for *P*;

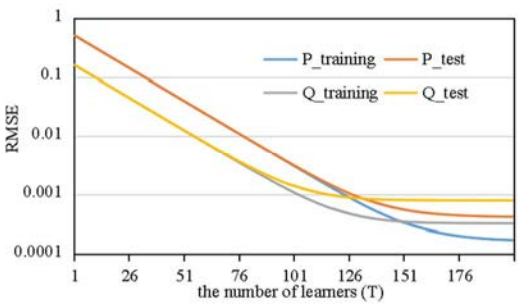
(3) No evident overfitting or underfitting problems are observed on all cases through the tuning process.



(a) case 5: RMSEs of *P*, *Q* by boosting(log)

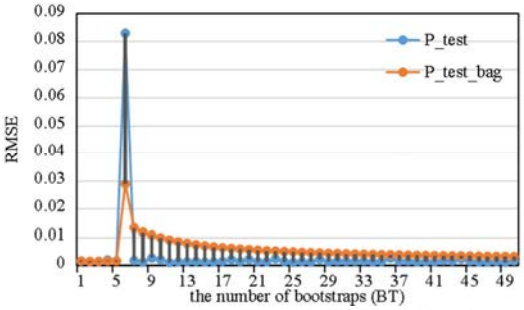


(b) case 57: RMSEs of P, Q by boosting(log)

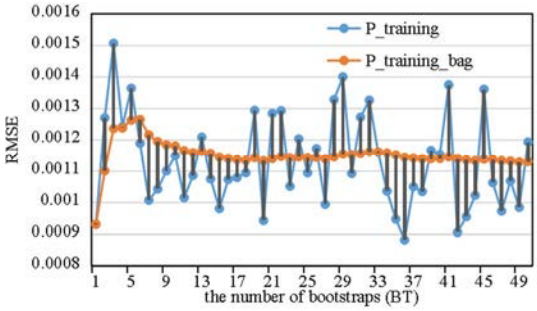


(c) case 118: RMSEs of P, Q by boosting(log)

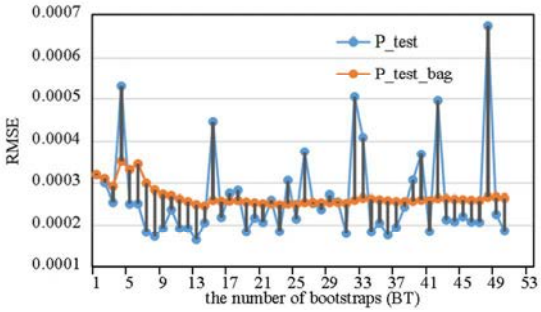
Fig. 1. The test and training RMSEs of P, Q with increasing T



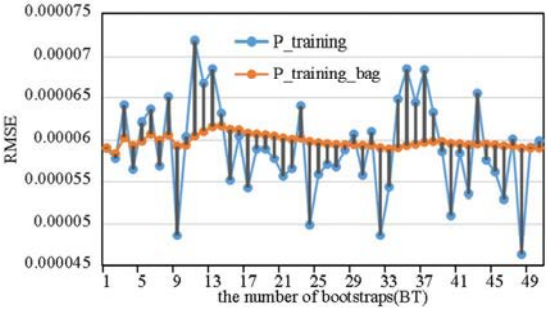
(a) case 5: test RMSEs of *P* by PR and bagging



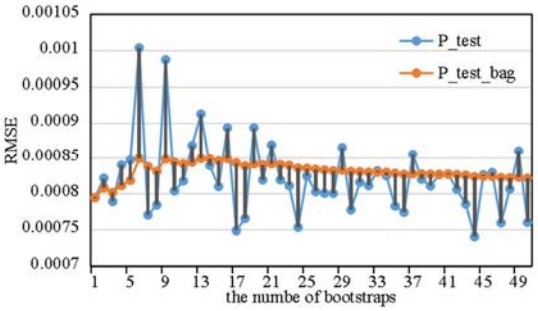
(b) case 5: training RMSEs of *P* by PR and bagging



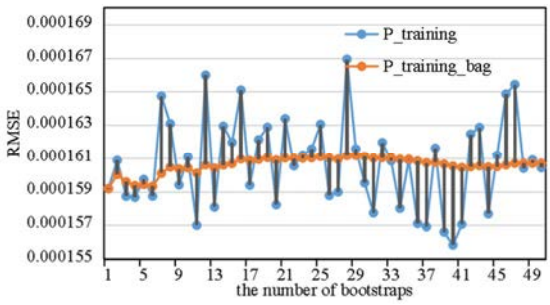
(c) case 57: test RMSEs of *P* by PR and bagging



(d) case 57: training RMSEs of *P* by PR and bagging



(e) case 118: test RMSEs of *P* by PR and bagging



(f) case 118: training RMSEs of *P* by PR and bagging

Fig. 2. Comparison of the test or training RMSE of P on all cases with increasing BT

2. *Tuning the Number of Bootstraps BT in Bagging*

In Fig. 2. (a)~(f), comparing the RMSEs of the active power injection *P* on all cases before and after bagging is incorporated. Each plot represents the results of each bootstrap (blue broke curve) and bagging (red curve) as the number of bootstraps BT goes up. According to Fig. 2, we can conclude that:

(1) The test or training RMSE of bagging becomes very small with slight fluctuations after sufficient times of bootstraps. When the number of bootstraps is 20 or more, both the test and training RMSEs seem to work well on all cases.

(2) The result of every single bootstrap fluctuates stochastically around the red curve, indicating that the single learner has its unstable weakness.

(3) Bagging can average the variances of all single learners and avoid overfitting.

C. *Comparing OPF Results*

As GB exhibits its superiority over others, the linear models fitted by GB are chosen to develop a data-driven convex approximation (DDCA) method and compute the optimal objective values (OOV) shown in TABLE II (unit: $/hr), compared with the original nonconvex ACOPF, the DCOPF and the semidefinite programming relaxation of OPF (SDPOPF). The results of ACOPF, DCOPF and SDPOPF can

be computed by Matpower 7.0 on all cases. Note that the results of ACOPF are set as the benchmarks and and the optimality gap g displayed in TABLE III is defined as [20]:

$$g = \frac{(g_1 - g_2)}{g_1} \times 100\%$$

where g_1 denotes the OOV of ACOPF; g_2 represents one of the OOVs of DDCAOPF, DCOPF and SDPOPF. Additionally, the runtime (unit: s) of each approach is recorded in TABLE IV.

TABLE II COMPARING OBJECTIVE VALUES OF OPF

case	ACOPF	DDCAOPF	DCOPF	SDPOPF
case 5	17551.89	17547.4	17479.9	16635.78
case 57	12100.86	12096.04	10211.99	10458.06
case 118	129660.70	129680.13	125947.88	129713.07

TABLE III COMPARING OPTIMALITY GAPS

case	ACOPF	DDCAOPF	DCOPF	SDPOPF
case 5	0%	0.025%	0.41%	5.22%
case 57	0%	0.040%	15.61%	13.57%
case 118	0%	-0.014%	2.86%	-0.040%

TABLE IV COMPARING RUNTIMES OF DIFFERENT METHODS

case	ACOPF	DDCAOPF	DCOPF	SDPOPF
case 5	2.95	1.57	2.30	23.82
case 57	2.83	2.15	1.77	31.99
case 118	3.94	2.91	2.11	39.08

From TABLE II~ IV, they reveal that:

(1) For the accuracy of OOV, DDCA performs better than the DC method and the SDP relaxation on all cases.

(2) The table of optimality gaps proves that DDCA (-0.014%~0.040%) works more robustly than the SDP relaxation (-0.040%~13.57%) and the DC method (0.41%~15.61%) on the computational accuracy.

(3) TABLE IV indicates on all cases DDCAOPF is more computationally tractable than ACOPF and SDPOPF, and its runtimes even can match DCOPF.

V. CONCLUSIONS AND FUTURE WORK

In this paper, an ensemble learning based linearization of power flow is proposed and applied in computing the OPF problem. As for the performance of learning methods, the application of ensemble learning methods in fitting power flow suggests their superiority over single learning method. In terms of solving the OPF, the proposed data-driven linear model of power flow outperforms the DC method and the SDP relaxation on the computational accuracy, and works better than ACOPF and SDPOPF on the computational efficiency.

However, the research results mentioned above are only based on simulating datasets without considering uncertainties of power load and distributed generators (DG). Instead of using the simulating datasets, our future work will focus on sampling operation data from the real power systems, inferring the complexity and sensitivity of learned models, and introducing the uncertainties of power load and DGs by quantitative approaches.

REFERENCES

[1] S. Bolognani, S. Zampieri, "On the Existence and Linear Approximation of the Power Flow Solution in Power Distribution Networks," *IEEE Trans. Power Syst*, vol. 31, no. 1, pp. 740–741, Jan. 2016.

[2] A. Garces, "A Linear Three-Phase Load Flow for Power Distribution Systems," *IEEE Trans. Power Syst*, vol. 31, no. 1, pp. 827–828, Jan. 2016.

[3] J. R. Martí, H. Ahmadi, L. Bashualdo, "Linear Power-Flow Formulation Based on a Voltage-Dependent Load Model," *IEEE Trans. Power Delivery*, vol. 28, no. 3, pp.1682-1689, July. 2013.

[4] S. M. Fatemi, S. Abedi, G. B. Gharehpetian, et al, "Introducing a Novel DC Power Flow Method with Reactive Power Considerations," *IEEE Trans. Power Syst*., vol. 30, pp. 3012- 3023, 2015.

[5] Z. Yang, H. Zhong, A. Bose, et al, "A Linearized OPF Model with Reactive Power and Voltage Magnitude: A Pathway to Improve the MW-Only DC OPF," *IEEE Trans. Power Syst*., vol. 32, no. 3, pp. 1734-1735, March. 2017.

[6] J. Yang, N. Zhang, C. Kang, et al, "A State-Independent Linear Power Flow Model with Accurate Estimation of Voltage Magnitude," *IEEE Trans. Power Syst*., vol. 32, no. 5, pp. 3607-3617, Sept. 2017.

[7] Z. Li, J. Yu, Q. H. Wu, "Approximate Linear Power Flow Using Logarithmic Transform of Voltage Magnitudes with Reactive Power and Transmission Loss Consideration," *IEEE Trans. Power Syst*., vol. 33, no. 4, pp. 4593-4603, July. 2017.

[8] Y. X. Liu, N. Zhang, "Data-Driven Power Flow Linearization: A Regression Approach", *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2569 - 2580, May. 2019.

[9] Y. C. Chen, J. Wang, A. D. Domínguez-García, and P. W. Sauer, "Measurement-based estimation of the power flow Jacobian matrix," *IEEE Trans. Power Syst*., vol. 7, no. 5, pp. 2507-2515, Sept. 2016.

[10] Y. C. Chen, A. D. Dominguez-Garcia and P. W. Sauer, "Measurement-based estimation of linear sensitivity distribution factors and applications," *IEEE Trans. Power Syst*., vol. 29, no. 3, pp. 1372-1382, May. 2014.

[11] Y. Yuan, O. Ardakanian, S. Low, et al, "On the inverse power flow problem," arXiv preprint arXiv:1610.06631, 2016.

[12] V. Kekatos, G. B. Giannakis, "Sparse Volterra and Polynomial Regression Models: Recoverability and Estimation", *IEEE Trans. Signal Process,* vol. 59, no. 12, pp. 5907-5920, Dec. 2011.

[13] Y. Wi, S. Joo, et al, "Holiday Load Forecasting Using Fuzzy Polynomial Regression with Weather Feature Selection and Adjustment", *IEEE Trans. Power Syst*., vol. 27, no. 2, pp. 596-603, May. 2012.

[14] R. Punmiya, S.H. Choe, "Energy Theft Detection Using Gradient Boosting Theft Detector With Feature Engineering-Based Preprocessing", *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2326-2329, Mar. 2019.

[15] J. H., Friedman, "Greedy Function Approximation: A Gradient Boosting Machine", Stanford University, Feb. 1999.

[16] B. Leo, "Bagging predictors. Machine Learning", 1996, 24 (2): 123–140. doi:10.1007/ BF00058655. CiteSeerX: 10.1.1.32.9399.

[17] E.T. Iorkyase, C. Tachtatzis, et al, "Improving RF-based Partial Discharge Localization via Machine Learning Ensemble Method", *IEEE Trans. Power Del*., to be published. DOI 10.1109/TPWRD.2019.2907154.

[18] J. Zhang, M. Wu, V. S. Sheng, "Ensemble Learning from Crowds", *IEEE Trans. Knowl. Data Eng*., vol. 31, no. 8, pp. 1506-1509, Aug. 2018.

[19] G. Casella, S. Fienberg, I. Olkin, "An Introduction to Statistical Learning with application in R". New York, NY, USA: Springer, 2013.

[20] B. Kocuk, S. S. Dey, X. Sun, "Inexactness of SDP Relaxation and Valid Inequalities for Optimal Power Flow", *IEEE Trans. Power Syst*., pp. 642-651, vol. 31, no. 1, Jan. 2016.