# Machine Learning-Aided Security Constrained Optimal Power Flow

Jubeyer Rahman, *Student Member, IEEE*, Cong Feng, *Student Member, IEEE*, and Jie Zhang, *Senior Member, IEEE*
The University of Texas at Dallas
Richardson, TX, 75080 USA
{Jubeyer.Rahman, Cong.Feng1, jiezhang}@utdallas.edu

***Abstract*—Though many approaches have been proposed in recent decades to solve the full AC optimal power flow (OPF) problem, efficiently finding the solution still remains challenging due to its highly non-linear and non-convex nature, especially for large scale networks. Machine learning has proven to significantly improve the computational efficiency in many problems. Thus in this paper, a learning augmented optimization approach is developed to solve the security-constrained optimal power flow (SCOPF) problem. More specifically, a multi-input multi-output random forest model is developed to first solve network voltage magnitudes and angles of buses. Then, physics-based network equations are employed to determine the current injection and complex/real power injection at different buses. To evaluate the efficiency of the proposed machine learning-aided algorithm, two benchmark models are adopted: (i) one with the conventional MATPOWER Interior Point Solver, and (ii) the other one with an end-to-end pure machine learning approach. Test results on a 500-bus network show that the proposed machine learning-aided approach has significantly improved the computational efficiency compared to the MATPOWER solver, while all network constraints are successfully satisfied.**

***Index Terms*—OPF, random forest, supervised learning, constraint violation**

## I. Introduction

In current deregulated electricity markets, system operators need to produce precise dispatches and market signals to better utilize existing resources and preserve the interests of all market stakeholders [1]. Independent System Operators (ISO) run optimal power flow (OPF) algorithms to get their dispatch decisions. Due to the uncertain and variable nature of renewable sources, these decision parameters need to be solved more frequently to avoid curtailment. It has been reported that alone in the U.S., a 5% increase in market efficiency, mostly depending on OPF solvers, can save 6 billion dollars annually [2]. Security constraints in OPF often introduce integer decision variables, which makes the optimization problem more challenging to solve.

Machine learning has proven to significantly improve the computational efficiency in many problems, including OPF. In general, the utilization of machine learning approaches to solve OPF can be broadly divided into two categories: 1) an end-to-end approach purely based on machine learning, and 2) a hybrid approach of machine learning and physics-based solver. For example, a collection of supervised learning algorithms, like Gaussian naive Bayes, logistic regression, decision tree, random forest, extra-tree, and neural network were used to solve OPF in [3]. In [4], system real power load and costs of generation were used to predict generators real power output and voltage of P-V buses using several machine learning models, including multi-layer perceptron regression, gradient boosting regression, and support vector machine with a radial basis function kernel. Though the machine learning approach has a fairly high degree of testing accuracy, constraint violation was present in around 60% cases. Active OPF constraints prediction was studied in [5]–[7] to improve the computational efficiency of traditional algorithms. An end-to-end generator setting prediction was performed in [7] through a constraint violation penalty incorporated neural network approach. Similar to other work, there exist a large number of equality constraints violations. A deep learning technique was exploited in [8] to solve the generation real power in DC OPF, and post-processing was performed to ensure that the generation values are within lower and upper bounds.

Hybrid approaches were also attempted in the literature to reduce the computational cost, focusing primarily on yielding better warm-start points for physics-based solvers. Similar to [5]–[7], a neural network model was used in [9] to represent the system security boundary and add to the list of constraints in the form of a differentiable mapping function, which was then solved by a physics-based solver. A multi-input multi-output random forest (RF) regression method was used in [10] to predict the real power set-point of generators and bus voltage magnitudes, which were then passed to a physics-based solver. An artificial neural network model was used in [11] to only solve generator voltage and real power outputs, which were then passed to a physics-based solver to solve the reactive power.

The main drawback of using machine learning to obtain direct end-to-end solutions is that there might exist a large number of constraints violations, especially for equality constraints. It is noted in the literature that, a majority of end-to-end machine learning-based OPF methods solve real power generation of the generators and buses voltage magnitudes, which may lead to an infeasible solution space. This is due to that both the voltage magnitude and angle determine the amount of power flow in branches, and the power losses of lines are also dependent on voltage parameters. Thus the real power generation obtained from the machine learning model may not be useful, since it may contradict the injection values calculated from predicted voltage parameters based on network equations. While hybrid approaches can ensure the convergence, they still need physics-based solvers. Thus the computational efficiency improvement of hybrid approaches is still limited.

To address the challenges in both machine learning-based end-to-end approaches and hybrid approaches, this paper develops a machine learning-aided approach that does not need to run any physics-based OPF solvers. To address the issues of constraints violation, our machine learning model solves the voltage magnitudes and angles, instead of real power generation. Once the voltage magnitudes and angles are obtained, physics-based power flow equations are used to calculate other operating parameters, such as power generation settings.

The rest of the paper is organized as follows. In Section II, the proposed machine learning-aided approach is discussed along with AC OPF formulations. Section III describes the multi-input multi-output RF model for solving OPF. In Section IV, descriptions on a 500-bus network case study and dataset generation are presented, followed by detailed results discussion. Finally, the paper is concluded in Section V.

## II. Methodology

### A. The OPF Problem

The OPF problem deals with determining the optimum dispatch of generators to minimize the cost of generation, while satisfying the engineering and physical constraints. In this work, the OPF formulation of the 'Grid Optimization (GO)' competition [12] is adopted and modified, where switched shunt related security constraints are incorporated. A security constrained AC OPF problem is formulated as follows.

$$
\begin{aligned}
&\min_{v,\theta,p,q,b^{cs}} \sum_{g=1}^{G} c_g \\
&\text{s.t.} \\
&\underline{v}_i \leq v_i \leq \overline{v}_i,\ \forall i \in N \\
&\underline{\theta}_i \leq \theta_i \leq \overline{\theta}_i,\ \forall i \in N \\
&\underline{p}_g \leq p_g \leq \overline{p}_g,\ \forall g \in G \\
&\underline{q}_g \leq q_g \leq \overline{q}_g,\ \forall g \in G \\
&\underline{b}_k^{cs} \leq b_k^{cs} \leq \overline{b}_k^{cs},\ \forall k \in B \\
&p_{inj}^i - p_d^i = \sum_{(i,j)\in\mathcal{E}} p_{i,j}(v,\theta),\ \forall i \in N, \forall (i,j) \in \mathcal{E} \\
&q_{inj}^i - q_d^i = \sum_{(i,j)\in\mathcal{E}} q_{i,j}(v,\theta),\ \forall i \in N, \forall (i,j) \in \mathcal{E} \\
&\sqrt{p_{i,j}^2 + q_{i,j}^2} \leq \overline{s_f},\ \forall (i,j) \in \mathcal{E}
\end{aligned}
\tag{1}
$$

where $N$, $\mathcal{E}$, $B$, and $G$ are the set of bus, branch (both line and transformer), switched shunt, and generator, respectively. $c_g$ represents the generation cost of generator g, which is calculated from an individual piece-wise linear cost function. $v_i$ and $\theta_i$ are the voltage magnitude and angle at bus $i$, respectively. Generator active and reactive power is denoted by $p_g$ and $q_g$, respectively. The shunt susceptance is denoted by $b^{cs}$ and limited by their MVAR limits. $p_{inj}^i$ and $q_{inj}^i$ stand for the real and reactive power injection at bus $i$, whereas, $p_d^i$ and $q_d^i$ are the real and reactive power demand at that bus, respectively. The net real and reactive power injection $p_{i,j}$ and $q_{i,j}$ to a branch can be represented by the difference between injection and demand at the connected buses. At every branch, the real and reactive power flow is limited by its rating. A feasible solution ensures that these constraints are not violated. Constraints (e.g., equality) are derived from physical laws like Ohm's law and Kirchhoff's law, which have zero tolerance for violation. The constraints that are related to engineering practice could be relaxed during contingency events.

### B. Machine Learning-Aided OPF Methodology

Figure 2 shows overall framework of the proposed machine learning-aided OPF algorithm. With a given load condition (i.e, real and reactive power load), we first solve voltage magnitude and angles of buses using a multi-input multi-output RF regression model. The slack bus angle is excluded from the response variables since it is regarded as a reference point in the OPF problem. The collected sample dataset is then split into training and testing dataset. The RF model is trained based on the training dataset and later used for making voltage magnitude and angle predictions.

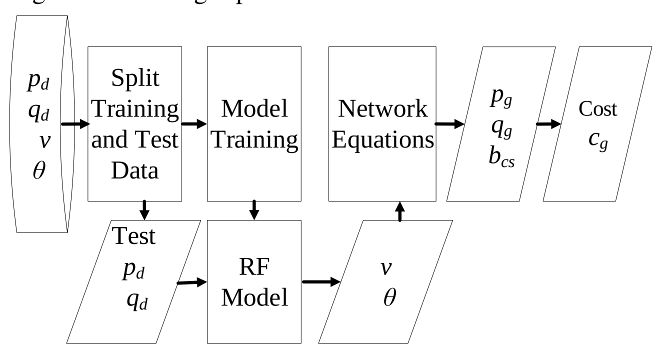



Figure 1. The proposed machine learning-aided OPF algorithm

After solving the voltage parameters, network equations are employed to determine the current injection $I^{inj}$ at different buses, based on a bus admittance matrix $Y$ that is usually calculated from line parameters.

$$I^{inj} = Y * v \tag{2}$$

Once current injections are known, the apparent power $S$ is calculated from the product of bus voltage and current conjugate.

$$S = v \times I^{inj^*} \tag{3}$$

Finally, the complex bus power injection $S_{inj}$ is calculated by subtracting the bus power demand $S_d$ from the apparent power. This is inspired from the repetitive adjustment of system voltage parameters to produce a zero mismatch in complex bus power injection, which is typically used in traditional power flow algorithms.

$$S_{inj} = S - S_d \tag{4}$$

The real part of the complex bus power injection at the generator bus is then fed into the piece-wise linear cost functions of individual generators to calculate the overall cost of production. The integration of machine learning and network equations ensures that all the constraints related to physical laws are satisfied.

## III. Multi-Input Multi-Output Random Forest Model

RF is adopted in this study due to its multi-input multi-output regression capability. RF is a combination of bagging [13] and random feature subspacing [14], both of which are ensemble techniques trained with a particular prediction scheme [15]. There exist many decision trees in a random forest model to make their predictions separately. Then these predictions are averaged to minimize the error. The trees are base learners and near-independent, which reduces the risk of biased decision or overfitting.

### A. Training Random Forest Model

To train the RF model, bootstrap samples are drawn from the input dataset (i.e., the system load $P_d$ and $Q_d$), and for each sample, an unpruned regression tree is grown. At every node, input variables are sampled randomly and splitting is done in the variable space. The best split is ensured by minimizing the residual sum of squares (RSS), given by:

$$RSS = \sum_{l=1}^{L} \sum_{m \in R_l} y_m - \hat{y}_{R_l} \tag{5}$$

where, $\hat{y}_{R_l}$ is the average of the observations in the $l^{th}$ non-overlapping region, $L$ is the total number of regions of the predictor space, and $y_m$ is the predicted value.

The training process continues until the number of grown trees is equal to the preset number. The model outputs ($v$ and $\theta$) are generated by aggregating the prediction of all the grown trees. Here, the average solution of all trees are considered as the final solution, which ensures zero random prediction errors of the trees (forest) to zero and preserves the true relationship between predictors and responses.

Unlike other machine learning algorithms, cross-validation is not required in RF since out-of-bag (OOB) error estimation is used during the tree construction [13].

### B. Parameter Setting

Main hyperparameters in the RF algorithm are the number of trees, the minimum number of samples for splitting, and the maximum depth of the tree. An asymptotic relation exists between the number of trees and prediction accuracy. The number of trees need to be optimized to balance the accuracy and computational cost.

Trees with high-depth often raise the risk of overfitting. In fact, the depth is directly related to the level of significance of the independence between inputs and outputs. The maximum tree depth needs to be carefully selected.

**Algorithm 1** Multi-input multi-output RF-based OPF Algorithm

**Require:**
RF training and test dataset: ($X_{tr}$, $Y_{tr}$) , ($X_{test}$,$Y_{test}$)
RF model parameter (no. of trees $B$, maximum depth $D$, min sample no. $n_s$)
Bus admittance matrix Y
**Ensure:** OPF variables are determined

```
procedure RandomForest( (X_tr, Y_tr), D)
    Initialize Q = φ
    for e = 1 to B do
        I_e ← a bootstrap sample from training set
        q_e ← get the learned tree from the sample with a cutpoint to minimize RSS
        Q ← Q ∪ q_e
    end for
    return Q
end procedure
Predict with Q on (X_test, Y_test) to get v and θ
Determine other OPF variables by using Eq. (2,3, & 4)
```

## IV. Case Study

### A. Experimental Setup

To demonstrate the efficacy of the proposed approach, a 500-bus transmission network is selected from the ‘GO’ competition repository (case 1, network 1, offline) [16]. The test system has 51 active generators, 17 controllable shunts, 200 load buses, and 593 branches. To generate a large number of training points, the given base case load data is used and extended by following the hourly load profile of the year 2011 of of the Electric Reliability Council of Texas (ERCOT) system. The load distribution among different buses is kept same as the base case, and the ERCOT load profile is scaled down to produce reasonable load patterns for the test case. This scheme yields 8,760 load samples which are then fed to MATPOWER [17] for solving the security constrainted OPF. Since controllable shunts cannot be modeled directly in MATPOWER, they are modeled as controllable synchronous condensers. The MATPOWER Interior Point Solver (MIPS) is adopted for solving the OPF problem. Only the solutions with convergence are used for RF model training and testing. Real and reactive power losses at each sample are also recorded for further checking of the nodal power balance constraints.

Out of all the converged cases, 80% data are used for training and the rest are left for testing. Shuffling is activated to ensure uniform variance throughout the training and testing dataset. The tree number and maximum depth are selected empirically and set to 100 and 30, respectively. The minimum number of samples required to split an internal node is set to 2. Hyperparameters that yield the lowest mean squared error are chosen to be the optimum settings for the RF model. For model building and testing, the *RandomForestRegressor* package of scikit-learn is used [18]. To better evaluate the effectiveness of the machine learning-aided method, another end-to-end case is tested by solving the OPF problem purely based on machine learning without considering the network equations (Case II). To this end, a holistic random forest model is trained on the same dataset for comparison. Since the whole set of OPF variables is determined from the ML model, a larger number of trees (i.e., 250) and higher maximum depth (i.e., 50) are chosen to support this comparatively higher dimensional mapping task. The *GridSearch* option of scikit-learn is utilized for tuning the parameters in RF.

### B. Results & Discussion

For the considered test case, the lower and upper limits of the voltage magnitude are 0.9 and 1.1 per unit, respectively; the lower and upper limits of angles are -180 and 180 degree, respectively. The solutions obtained directly from MATPOWER are considered as actuals for comparison. Figs. 2 and 3 compare the machine learning-aided solutions and MIPS solutions for voltage magnitude and angle, respectively. It is observed that the machine-learning aided solutions are close to the MIPS solutions and within their bounds. This could prevent outliers that may lead to severe constraints violation.

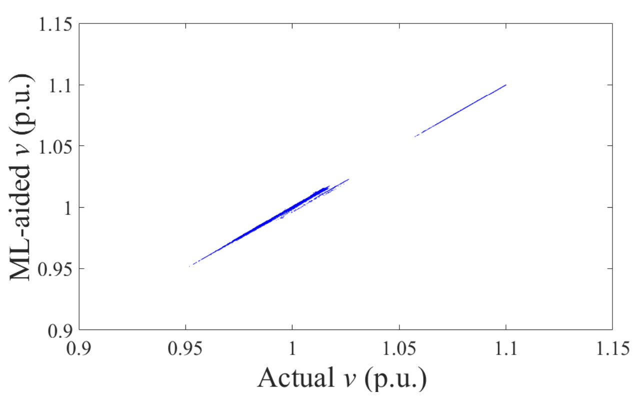



Figure 2. Voltage magnitude profile comparison: machine learning-aided vs. MIPS (ML stands for machine learning)

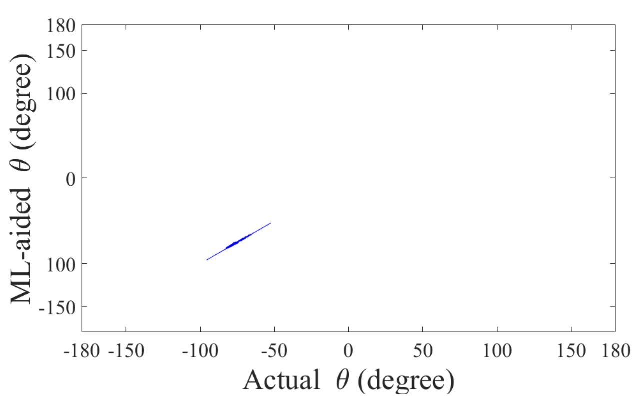



Figure 3. Voltage angle profile comparison: machine learning-aided vs. MIPS

After solving the voltage magnitudes and angles from the machine learning model, branch power flows are calculated and shown in Fig. 4. It is observed that branch power flows are well within their limits. Regarding the power injection from generator buses, no negative real power injections are found whereas no instance of positive real power injections are found in load buses.

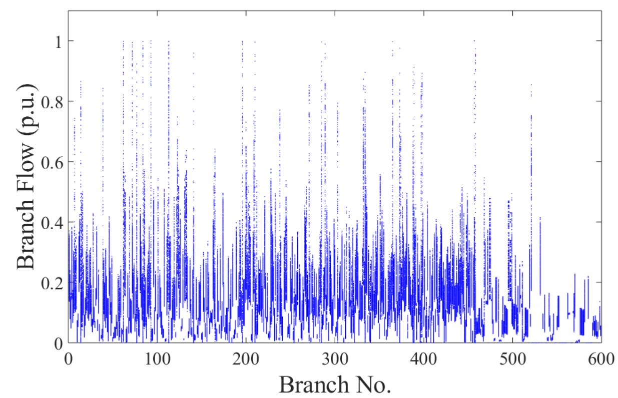
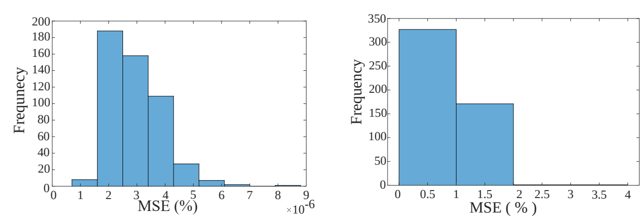



Figure 4. Branch flow profiles (each branch flow is scaled to its p.u. capacity)

Figure 5 shows the histograms of the deviations of voltage magnitudes and angles, respectively, between machine learning-aided and MIPS solutions. It is observed that the mean squared error (MSE) of the voltage magnitude is close to zero, and the MSE of the voltage angle in most buses is less than 2%.




(a) MSE of the voltage magnitude (b) MSE of the voltage angle

Figure 5. MSE of voltage parameters

The constraints violation profile for both of the considered cases are summarized in Table I. It is observed that the proposed ML-aided scheme that only gets voltage parameters from the ML model (Case I) performs better in terms of constraints violation, while the end-to-end scheme (Case II) that directly gets all the OPF variables from the ML model has significant number of equality constraints violation. Since for case II, all of the bus power injection mismatches are below 1 MW and 1 MVAR, they are considered as the bus power balance tolerance in Case II. Around 31.33% of cases are found to violate the power balance constraints when the aforementioned threshold limits are considered otherwise case II is found to be violating equality constraints in 100% cases.

Table I
PERFORMANCE COMPARISON BETWEEN THE PROPOSED MACHINE LEARNING-AIDED SOLUTIONS AND PURE MACHINE LEARNING SOLUTIONS

| Case | Solved variables by machine learning | Inequality constraints violation (%) | Equality constraints violation (%) | Training time (h) |
|---|---|---|---|---|
| Case I | $v$, $\theta$ | 0 | 0 | 25 |
| Case II | $v$, $\theta$, $p_g$, $q_g$, $b^{cs}$ | 0 | 100 | 34 |

To measure the optimality of the proposed approach, an optimality checking metric $O$ is adopted from [11].

$$O = \frac{1}{N} \sum_{n=1}^{N} \frac{c_g(Proposed) - c_g(Actual)}{c_g(Actual)} \tag{6}$$

Here, N is the total number of considered cases which is 1,080 (converged samples among the test dataset). The optimality of the proposed approach is $1.123 \times 10^{-4}$, which is acceptable compared with other methods reported in the literature [11].

To evaluate the computational efficiency of the proposed machine learning-aided method, the solving time between the machine learning-aided method and MIPS is compared in Table II. Both of the solving time of all 1,080 cases and the average solving time per case are reported. It is observed that the computational efficiency has been improved by 58.75 times with the machine learning-aided method, compared to MATPOWER MIPS.

Table II
SOLUTION TIME COMPARISON BETWEEN THE MACHINE LEARNING-AIDED SOLVER AND MIPS

| Solver | Total solution time (s) | Average solution time per case (s) |
|---|---|---|
| MIPS (MATPOWER) | 2030.4 | 1.88 |
| Machine learning-aided solver | 35.26 | 0.032 |

## V. CONCLUSION

This paper developed a machine learning-aided method for solving security constrained optimal power flow, which integrates both power network equations and a multi-input multi-output machine learning model to yield a near-optimal OPF result. The machine learning-aided approach has reduced the solution time significantly. The integration of network equations in the proposed method could ensure the feasibility of the OPF solutions. Results on the 500-bus system have shown zero constraints violation. However, it is challenging to satisfy equality constraints, when solving OPF purely based on machine learning models without network equations. The proposed approach could also be used in post-contingency analysis as well, where numerous network topologies with different load profiles and network parameters need to be solved within a very short time limit. Future work will test the machine learning-aided method on larger networks, and also explore other learning techniques including deep learning frameworks.

## ACKNOWLEDGMENT

This work was supported by the U.S. Department of Energy under Contract No. DE-AR0001075.

## REFERENCES

[1] M. B. Cain, R. P. O'neill, and A. Castillo, "History of optimal power flow and formulations optimal power flow paper 1," in *Optimal Power Flow and Formulation Papers*, Federal Energy Regulatory Commission (FERC), 2012.
[2] E. I. A. (US), *Annual Energy Outlook 2012: With Projections to 2035*. Government Printing Office, 2012.
[3] R. Canyasse, G. Dalal, and S. Mannor, "Supervised learning for optimal power flow as a real-time proxy," in *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, IEEE, 2017.
[4] T. Navidi, S. Bhooshan, and A. Garg, "Predicting solutions to the optimal power flow problem," in *Project Report*, Stanford University, 2016.
[5] D. Deka and S. Misra, "Learning for dc-opf: Classifying active sets using neural nets," *arXiv preprint arXiv:1902.05607*, 2019.
[6] Y. Ng, S. Misra, L. A. Roald, and S. Backhaus, "Statistical learning for dc optimal power flow," in *2018 Power Systems Computation Conference (PSCC)*, pp. 1–7, IEEE, 2018.
[7] N. Guha, Z. Wang, M. Wytock, and A. Majumdar, "Machine learning for ac optimal power flow," *arXiv preprint arXiv:1910.08842*, 2019.
[8] X. Pan, T. Zhao, and M. Chen, "Deepopf: Deep neural network for dc optimal power flow," *arXiv preprint arXiv:1905.04479*, 2019.
[9] V. J. Gutierrez-Martinez, C. A. Cañizares, C. R. Fuerte-Esquivel, A. Pizano-Martinez, and X. Gu, "Neural-network security-boundary constrained optimal power flow," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 63–72, 2010.
[10] K. Baker, "Learning warm-start points for ac optimal power flow," *arXiv preprint arXiv:1905.08860*, 2019.
[11] A. Zamzam and K. Baker, "Learning optimal solutions for extremely fast ac optimal power flow," *arXiv preprint arXiv:1910.01213*, 2019.
[12] ARPA-E, "Challenge 1 problem formulation." [Online] Available at: https://gocompetition.energy.gov/challenges/challenge-1/formulation. [Accessed: 30 Oct. 2019].
[13] L. Breiman, "Random forests," vol. 45, pp. 5–32, Springer, 2001.
[14] T. K. Ho, "Nearest neighbors in random subspaces," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 640–648, Springer, 1998.
[15] H. Linusson, "Multi-output random forests," in *Master's Thesis*, University of Borås, School of Business and IT, 2013.
[16] ARPA-E, "Datasets." [Online] Available at: https://gocompetition.energy.gov/challenges/22/datasets. [Accessed: 30 Oct. 2019].
[17] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2010.
[18] F. Pedregosa *et al.*, "sklearn.ensemble.randomforestregressor." [Online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. [Accessed: 30 Oct. 2019].