Learning the Optimal Strategy of Power System Operation with Varying Renewable Generations

Mingxuan Li, Wei Wei *Senior Member, IEEE*, Yue Chen, Ming-Feng Ge *Member, IEEE*, João P. S. Catalão, *Senior Member, IEEE*

***Abstract*—Optimal dispatch of modern power systems often entails efficiently solving large-scale optimization problems, especially when generators have to respond to the fast fluctuation of renewable generation. This paper develops a method to learn the optimal strategy from a mixed-integer quadratic program with time-varying parameters, which can model many power system operation problems such as unit commitment and optimal power flow. Different from existing machine learning methods that learn a map from the parameter to the optimal action, the proposed method learns the map from the parameter to the optimal integer solution and the optimal basis, forming a discrete pattern. Such a framework naturally gives rise to a classification problem: the parameter set is partitioned into polyhedral regions; in each region, the optimal 0-1 variable and the set of active constraints remain unchanged, and the optimal continuous variables are affine functions in the parameter. The outcome of classification is compared with analytical results derived from multi-parametric programming theory, showing interesting connections between traditional mathematical programming theory and the interpretability of the learning-based method. Tests on a small-scale problem demonstrate the partition of the parameter set learned from data meets the theoretical outcome. More tests on the IEEE 57-bus system and a real-world 1881-bus system validate the performance of the proposed method with a high-dimensional parameter for which the analytical method is intractable.**

***Index Terms*—power system operation, interpretability, machine learning, renewable generation, uncertainty**

I. Introduction

As modern power systems continue to expand, some large-scale systems can possess thousands of buses and generators [1]. The optimal operation of power systems often entails solving a large mixed-integer optimization problem, which is a challenging task. Meanwhile, the increasing penetration of volatile renewable resources brings notable uncertainty in system operation, and generators must rapidly respond to the fluctuation of renewable power and load in real-time, requiring high computational efficiency.

This work is supported by the National Natural Science Foundation of China under grant 51807101.

M. Li and W. Wei are with the State Key Laboratory of Power Systems, Department of Electrical Engineering, Tsinghua University, 100084, Beijing, China. (e-mail: wei-wei04@mails.tsinghua.edu.cn).

Y. Chen is with the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong, Hong Kong SAR, China. (e-mail: yuechen@mae.cuhk.edu.hk).

M. Ge is with School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074 China (e-mail: gemf@cug.edu.cn).

J. P. S. Catalão is with the Faculty of Engineering of the University of Porto, 4200-465 Porto, Portugal, and also with INESC TEC, 4200-465 Porto, Portugal (e-mail: catalao@fe.up.pt).

For the optimal power flow (OPF) and economic dispatch (ED) problems, some online algorithms have been proposed, such as the quasi-Newton method [2], dynamic alternating direction method of multipliers based on suitable linearization [3], and hierarchical decentralized algorithm [4]. However, these methods can only handle continuous variables. Optimization problems with integer variables are also of great importance in power system operation, such as in unit commitment (UC) and transmission switching. Although mixed-integer programming solvers have been significantly improved during the past decades, solving a large-scale mixed-integer program may still be time-consuming due to non-convexity.

With the quick development of artificial intelligence, machine learning (ML) methods are increasingly popular in power system problems. One of the main applications is renewable generation forecast. For wind generation, ref. [5] applies deep Boltzmann machine to wind speed forecast, extracting high-level features of data that are informative for inference. Ref. [6] predicts wind speed intervals with an ensemble model composed of gated recurrent unit, variational mode decomposition, and error correction. Refs. [7] and [8] combine the convolutional neural network (CNN) and long short-term memory (LSTM) to predict wind speed, where CNN extracts the spatial features and LSTM learns the temporal information. Ref. [9] combines a bagging neural network with K-means clustering method for wind power forecast. For solar generation, ref. [10] adopts unsupervised clustering methods and support vector machine for solar power forecast. Refs. [11] and [12] ensemble several ML methods for solar output forecast. Ref. [13] tackles probabilistic solar irradiance forecasting using convolutional graph autoencoder, which can effectively estimate the distribution of future irradiance given historical data. Ref. [14] employs Generative Adversarial Networks for the imputation of solar data.

In the field of power system operation, several learning-assisted optimization methods have also been found. The majority of them employ surrogate models, i.e., the optimal solution is directly predicted from ML models. Ref. [15] applies the stacked extreme learning machine method to the OPF problem. The mapping from load demands to optimal generation schedules is divided into several sequential learning tasks. Ref. [16] combines artificial neural network (ANN) and simulated annealing algorithm for the UC problem; ANN predicts the on-off status of generators, and simulated annealing optimizes the generator output. Ref. [17] applies enhanced augmented Lagrange Hopfield network to the ED problem. The network is trained by regarding augmented Lagrangian function as the

energy function. Ref. [18] uses a Gaussian-process-emulator as a nonparametric surrogate model for the stochastic economic dispatch with wind penetration, where Isomap, a manifold learning method, is utilized for reducing the dimension and improving the efficiency of gaussian-process-emulator.

In recent years, reinforcement learning (RL) becomes a flourishing method for power system operation and control. Refs. [19]–[21] use RL to solve large-scale OPF and ED problems. These RL models are trained with the rewards defined as the objective function with Lagrangian penalty terms, while decision variables such as generator outputs are discretized to present discrete actions. Ref. [22] applies the Q-learning method to UC problems with renewables; the actions are the startup/shutdown behaviors of generators; the Q values are defined as generation costs and are approximated by a linear ANN. Ref. [23] integrates particle swarm algorithm with RL to solve UC problems under uncertainties, where RL is used to avoid local convergence of the meta-heuristic algorithm. Ref. [24] learns the multi-agent trading policy of renewable energy via Q-learning. Ref. [25] solves the automatic generation control problem with the Deep Double Q-Network algorithm. Ref. [26] uses a state-action-reward-state-action algorithm to schedule electric vehicle charging stations. A linear function is used to approximate the function between features and rewards, and the weights in the linear function are regarded as the training target.

The aforementioned ML methods can improve the efficiency of solving large-scale optimization problems, but there are also some shortcomings. First, unlike traditional mathematical programming model that explicitly cope with constraints, ML models usually treat constraint violation as a penalty; it is sometimes difficult to find the exact security boundary due to the lack of data. So security may become an issue because the loss for deploying an infeasible strategy is unaffordable. Second, some ML methods, such as neural network, lack interpretability and resemble a black box. The decision maker can hardly know why such a strategy is found and how large the optimality gap is. Third, ML methods are typically model-free, which is attractive in tasks like computer vision and natural language processing. However, the classic models of the power system contain important information for making a dispatch decision, but are not fully utilized.

In view of the ability of ML techniques and its potential in helping power system operators making quick decisions to tackle uncertainty, such as in the online and rolling-horizon optimization problems, this paper aims to develop a method to learn the optimal solution of UC, ED, and OPF problems with time-varying parameters, which are mathematically modeled as a quadratic program or a mixed-integer quadratic program (MIQP). The contributions are twofold:

1) A model-based ML method for power system operation problems that are formulated as MIQPs. Unlike existing ML methods that learn a map from parameters to the optimal strategy, the proposed method learns the map from parameters to the optimal pattern, including the integer solution and the set of active constraints. The optimal pattern is naturally discrete and the learning task gives rise to a classification problem. The outcome is a polyhedral partition of the parameter set. In each region, the optimal integer variables and the set of active constraints remain unchanged, and the optimal continuous variables are affine functions in the parameters. Such a framework has two advantages. First, it makes full use of the power system model and explicitly accounts for security constraints. Second, the computational burden is encapsulated in the offline training process. During the online stage, no optimization problem is solved; only algebraic operation is needed, saving computational resources. This means the commercial solver does not need to be integrated with the dispatch platform, and there is no need to transfer real-time data between the energy management system and the commercial solver, simplifying the architecture of the energy management system. We notice the same idea in a recent literature [27]. Nevertheless, the intrinsic structure of the optimal solution as well as the interpretability is not discussed.
2) A method to interpret the machine learning results. The classification can be implemented via either a neural network or a classification tree. When the tree method is used for classification, the result naturally interprets how and why the decision is made. It is shown that both neural networks and classification trees provide consistent partition. We reveal interesting connections between the machine learning results and multi-parametric programming theory. It is found that the criteria of classification coincide with the boundary of critical regions which can be analytically derived if the dimension of parameters is low. Such discussions bridge the traditional optimization theory and ML methods. On the one hand, parametric MIQP provides a unique perspective to explain and validate the ML method; on the other hand, as it is rather difficult to analytically solve a parametric MIQP with a high-dimensional parameter vector, the ML method can procure at least an approximate solution which contains structured information of power system optimal operation policy.

The rest of this paper is organized as follows. Section II introduces the properties of MIQPs and the proposed method. Section III reveals the connection between the classification results and multi-parametric programming theory. Section IV discusses the special case in which the objective function is linear and the application on locational marginal price (LMP) calculation. Section V presents case studies on the IEEE 57-bus system and a real-world 1881-bus system. Finally, Section VI summarizes the conclusion.

II. Proposed method

The motivation of the ML method is presented first, and then the method is formalized via a classification problem.

A. Motivation of the Learning Method

The canonical form of an MIQP can be expressed as:

$$\begin{aligned} \min \quad & x^\top Q x/2 + c_1^\top x + c_2^\top y \\ \text{s.t.} \quad & Ax + Ey \le b + B\theta \\ & x \in \mathbb{R}^{n_c}, y \in \{0,1\}^{n_b} \end{aligned} \tag{1}$$

where x and y represent continuous and binary variables, respectively; Q is a positive semidefinite matrix; A, B, E, c_1, c_2, and b are constant coefficients with compatible dimensions. We assume $Q = Q^\top$, otherwise, replacing Q with a symmetric matrix $(Q + Q^\top)/2$ does not alter the objective function. θ is the parameter representing renewable generation. For instance, Formulation (1) encompasses several power system operation problems, such as OPF with $n_b = 0$ and UC with $n_b > 0$.

Given the value of θ, MIQP (1) can be solved by commercial solvers. Due to the existence of binary variables, the computation can take some time, depending on the problem size. As the renewable generation is volatile, the value of θ changes rapidly. To maintain power system security and achieve an economic operation, a straightforward mean is to solve problem (1) more frequently with the latest update or forecast of θ, such as the rolling horizon scheme [28]–[30] and the online operation scheme [2]–[4], putting forward the need of highly efficient methods to solve a single instance of MIQP (1) with a fixed θ. ML technique encapsulates computational efforts in the offline training process and is able to provide quick decisions within moderate time in the online stage. The task comes down to learning the optimal value $v(\theta)$ and optimal solution $(x(\theta), y(\theta))$ as functions in the parameter θ. However, model-free learning methods do not fully utilize the mathematical model (1) and its specific solution structure.

When the parameter θ changes in a certain region, the number of optimal integer variables and active constraints are typically much fewer than the number of possible candidates. Take the UC problem for an example. Many units are always on because of lower costs; some constraints are always active while some others never becomes binding. The above observations motivates learning the map from θ to the **optimal binary variables** and **active constraints**, constituting an optimal pattern which is discrete. Fixing the optimal integer variable $y^\star$ and set of active constraints, MIQP (1) reduces to

$$\begin{aligned} \min \quad & x^\top Q x/2 + c_1^\top x \\ \text{s.t.} \quad & Ax \le b - Ey^\star + B\theta \end{aligned} \tag{2}$$

As the set of active constraints have been learned, the optimal continuous variable $x^\star$ can be determined from the following Karush-Kuhn-Tucker (KKT) optimality condition

$$\begin{aligned} Qx^\star + c_1 + \bar{A}^\top \mu = 0 \\ \bar{A}x^\star + \bar{E}y^\star - \bar{B}\theta = \bar{b} \end{aligned} \tag{3}$$

where $\bar{A}$, $\bar{B}$, $\bar{E}$, and $\bar{b}$ are sub-matrices corresponding to the active constraints; μ is the dual variable corresponding to the active constraints. A momentous observation is that when θ varies in some certain set, the optimal pattern remains unchanged. If we regard (3) as a set of linear equations with $(x^\star, \mu)$ being the variable and θ the observed parameter, the optimal continuous variable and dual variable can be easily obtained by solving a linear equation set which is very efficient, as optimization is no longer required. Therefore, to learn the optimal value function $v(\theta)$ and optimal solution function $(x(\theta), y(\theta))$, we have to identify all the optimal patterns and their corresponding invariant set of parameter θ.

B. Learn to Solve MIQP via Classification

Now, the core of the learning method is to identify the mapping from the parameter θ to an optimal pattern. The ML method consists of two main parts: offline learning and online prediction, as detailed in the following.

1) Offline Learning: Offline learning mainly includes learning the mapping from the parameter to the optimal patterns and storing the coefficient matrices associated with each pattern. The procedures are as follows.

(a) *Problem setup*: Formulate the problem as the canonical form stand form (1), and store the coefficient matrices. Retrieve the lower and upper bounds of parameters θ_{min} and θ_{max}, and define the parameter set $\Theta = \{\theta | \theta_{min} \le \theta \le \theta_{max}\}$. Initialize optimal pattern set $S = \emptyset$.

(b) *Sampling*: Get historical observations of θ and optimal binary variable; if unavailable, generate samples θ_1, θ_2, $\cdots$ from parameter set Θ that are likely to appear. For each sample, solve MIQP (1) under parameters θ_i, and record the optimal binary solution and active constraints in pattern s_i; If $s_i \notin S$, update $S \leftarrow S \cup s_i$. With a proper sampling set, the set S basically covers all possible patterns that are likely to appear in practice. More details about the sampling procedure are provided in the discussion.

(c) *Matrix Storage*: For each pattern s_i, Record the coefficients of KKT condition (3), i.e., the optimal binary variable $y^\star$ and coefficient matrices $\bar{A}$, $\bar{B}$, $\bar{E}$, and $\bar{b}$ of active constraints.

(d) *Training*: Learning the mapping from parameters to the optimal patterns comes down to a multi-class classification problem. The input is the sampled parameters, and the output is the label of the corresponding pattern. Samples are divided into a training set and a testing set. The classification model is trained on the training set.

Some technical details are provided as follows.

1. The number of samples should be sufficiently large, such that most optimal patterns have been explored. A method called the Good-Turing estimator was provided in [31] to determine the proper number of samples through estimating the probability for the non-existence of undiscovered patterns. Let N be the current number of samples, and N_1 the number of samples corresponding to the patterns appeared only once. Then the probability p_0 of there still existing undiscovered patterns with a confidence level of δ is bounded by

$$p_0 \le \frac{N_1}{N} + (2\sqrt{2} + \sqrt{3}) \sqrt{\frac{1}{N} \ln\left(\frac{3}{1-\delta}\right)} \tag{4}$$

Equation (4) provides an upper bound of p_0. In order to ensure that p_0 is below a certain threshold α, we can let the probability be less than a threshold, namely

$$\frac{N_1}{N} + (2\sqrt{2} + \sqrt{3}) \sqrt{\frac{1}{N} \ln\left(\frac{3}{1-\delta}\right)} \le \alpha \tag{5}$$

If we have enough data so that N_1 eventually approaches 0, then from (5) we have

$$N > \frac{20.8}{\alpha^2} \ln\left(\frac{3}{1-\delta}\right) \tag{6}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2021.3088951, IEEE Transactions on Sustainable Energy

In practice, when the classification model is trained with more data, both N and N_1 grows. We can terminate the training algorithm once (5) is met. Assuming $\delta = 0.99$, $\alpha = 0.05$, which means p_0 is guaranteed to be less than 0.05 with a confidence level of 0.99, then the stopping criterion is

$$0.0025N^2 - 118.6N - 0.1N_1N + N_1^2 \geq 0, N_1 \leq 0.05N \quad (7)$$

Although the above Good-Turing estimator only provides a probabilistic guarantee, it works well in practice, which can be seen from the case studies in Section V.

2. Two methods can implement the classification: ANN and oblique decision trees (ODT) [32]. We will demonstrate that both of them are capable of learning the optimal patterns. From the application perspective, ANN has better computational performance thus can effectively handle complicated problems, while ODT prevails in interpretability which helps the system operator better understand how the decision is made. Nevertheless, the classification can be implemented by other approaches, e.g. SVM and gradient boosting trees, depending on the specific need.

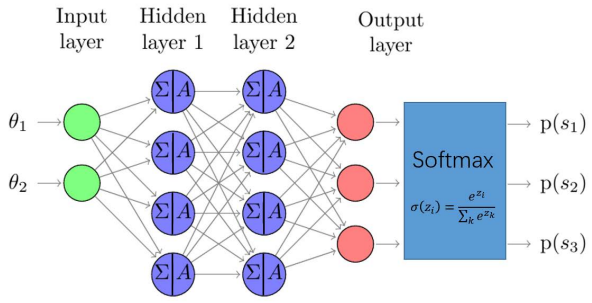
For the ANN method, we choose the 3-layer structure with ReLU activation, and the output layer is activated by softmax so that the probability of each class can be estimated, as shown in Fig. 1(a). Such a structure strikes a proper balance between representation capability and ease of training. 2-layer ANN with 1 hidden layer requires a significant number of neurons to maintain the representation performance, while the ANN with more than 3 layers is unnecessary for the simple classification task. For the ODT method, a binary tree is applied where each node is split by a multivariate linear expression, as shown in Fig. 1(b). Coefficients $\mathbf{a_i}, \mathbf{b_i}$ are iteratively optimized via a heuristic algorithm [32], ensuring that the tree is optimal or near-optimal.

2) Online Prediction: The online prediction procedure for the two classification methods will be introduced separately.

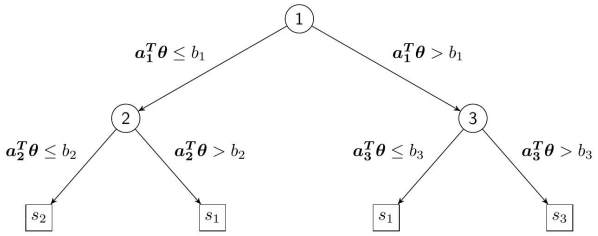
The online algorithm for ODT is as follows:

(a) *Retrieving the optimizer*: The label of the optimal pattern is predicted by ODT, thus the optimal binary variable $y^\star$ and the set of active constraints are clear. The optimal continuous variable is optimal in problem (2), and is retrieved by solving linear equation set (3).

(b) *Feasibility Check*: Substitute the optimal solution into the remaining inequality constraints of the original problem (1). If the constraint violation is small, an alternative pattern with a feasible solution needs to be picked out as the predicted solution. It is likely that the prediction error comes from only a few node splitting procedures, so we can backtrack each parent node and iteratively examine the patterns in the other branch until a feasible solution is acquired. If the constraint violation is large, it means no similar scenario has been explored in the training process. A remedy is to first deploy a feasible strategy that can be quickly found, and retrain the classification once time permits. The feasible strategy is acquired by traversing the existing patterns, which is efficient because obtaining a solution under each pattern only entails matrix operations. Thus a near-optimal solution can be quickly found. Afterwards, the ODT should be retrained to prevent infeasibility from occurring in the future. Before retraining the model, new samples are generated near the infeasible point in the parameter space, the number of which can be approximated as the average number of samples in the existing patterns. For each new sample, the i-th parameter θ_i is randomly picked within the interval $[\hat{\theta}_i - \frac{\theta_i^u - \theta_i^l}{2m^{1/D}}, \hat{\theta}_i + \frac{\theta_i^u - \theta_i^l}{2m^{1/D}}]$, where $\hat{\theta}_i$ is the i-th parameter of the infeasible point, θ_i^u and θ_i^l represent the upper bound and lower bound of the i-th parameter, m is the number of existing patterns, and D is the dimension of the parameter space. The above interval is determined based on the approximation that the undiscovered critical region is centered at the infeasible point and occupies a volume of V/m, where V is the total volume of the parameter space which is typically a hypercube. After the new samples are added to the sample set, the ODT is modified by finding out the leaf nodes which the new samples fall into, and then training an additional branching for the ones with false predictions using the new samples. As only minor changes to the model are required, the above retraining process is computationally efficient.



(a) ANN structure



(b) ODT structure

Fig. 1: Structures of ML methods

The online algorithm for ANN differs a little. As the output layer of ANN provides the probability of each pattern rather than a single optimal one. We can verify all the candidate patterns in order to obtain the actual optimal one. The online algorithm for ANN is as follows:

(a) *Retrieving candidate optimizers*: The possibilities of each pattern being optimal are predicted by ANN. Find out the first k candidate patterns with the highest possi-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2021.3088951, IEEE Transactions on Sustainable Energy

bilities, where k is pre-specified. Each candidate pattern corresponds to a reduced problem in (2). Solve the linear equation set (3) associated with each candidate pattern.

(b) *Feasibility and Optimality Check*: Check the feasibility of each candidate solution by substituting them into the remaining inequality constraints of the original problem (1). Find out the feasible solution with the best objective value, and accept it as the final optimizer. If no feasible solution is found, the same strategy explained in the ODT case is performed, i.e., new samples are generated and the ANN is fine-tuned with these new samples. The retraining procedure is also efficient because the current weights provide a good initiation.

The flowchart of the proposed method is shown in Fig. 2.

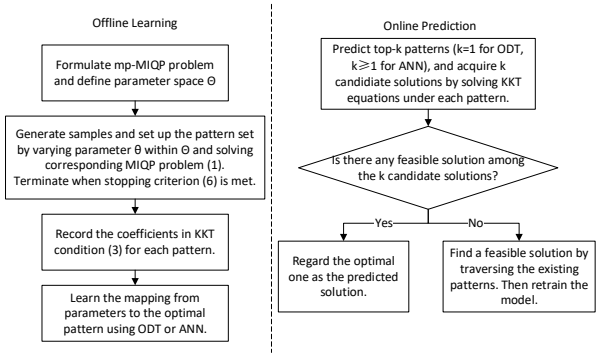


Fig. 2: Flowchart of the proposed method

By learning the optimal patterns offline, the task during the online stage just entails solving linear equations, which is easy to implement and efficient; no optimization problem is solved, and thus the proposed method is very fast. As we will show in the next section, ANN and ODT give very similar predictions, and the mappings learned from the two methods match the exact one which is analytically derived from multi-parametric programming theory. On this account, the ODT clearly explains why a solution is chosen. Such interpretability is highly desired by the power system operator.

III. Explaining the Result

This section discusses the true mapping from the parameter θ to the optimal solution $(x^\star, y^\star)$. For small-scale problems, such a mapping can be analytically computed and used to validate the result of the learning method.

Under a given binary variable $y^\star$, the optimal continuous variable solves the following convex quadratic problem

$$\begin{aligned} \min \quad & x^\top Q x/2 + c_1^\top x \\ \text{s.t.} \quad & Ax \le b^\star + B\theta \end{aligned} \tag{8}$$

where $b^\star = b - Ey^\star$. How the value of θ influence x will be investigated. For a given θ, suppose the set of active constraints is $\bar{A}x^\star = \bar{b}^\star + \bar{B}\theta$, then there must be a multiplier $\lambda^\star$ satisfying

$$\begin{aligned} Qx^\star + c_1 + \bar{A}^\top \lambda^\star = 0 \\ \bar{A}x^\star - \bar{b}^\star - \bar{B}\theta = 0 \end{aligned} \tag{9}$$

The primal-dual pair $(x^\star, \lambda^\star)$ can be expressed as

$$\begin{bmatrix} x^\star \\ \lambda^\star \end{bmatrix} = \begin{bmatrix} Q & \bar{A}^\top \\ \bar{A} & 0 \end{bmatrix}^{-1} \begin{bmatrix} -c_1 \\ \bar{b}^\star + \bar{B}\theta \end{bmatrix} \tag{10}$$

When the problem is non-degenerated, which is a mild condition, the inverse matrix exists [33]. Then, we can solve $(x^\star)$ and $\lambda^\star$ from (10)

$$\begin{aligned} x^\star &= M\theta + m \\ \lambda^\star &= N\theta + n \end{aligned} \tag{11}$$

where M, N, m, and n are constant coefficients. Although (11) is derived based on a given θ, it helps understand how θ affects $x^\star$ and $\lambda^\star$. To this end, recall the remaining constraints

$$\hat{A}x^\star < \hat{b}^\star + \hat{B}\theta \tag{12}$$

where $\hat{A}$, $\hat{B}$, and $\hat{b}$ correspond to the inactive constraints. Their corresponding dual variables must be 0 due to complementarity and slackness condition. In addition,

$$\lambda^\star \ge 0 \tag{13}$$

must hold in the KKT condition of problem (8).

Substituting (11) into (12) and (13), we obtain a set

$$\mathcal{CR} = \left\{ \theta \middle| \begin{array}{c} N\theta + n \ge 0 \\ (\hat{A}M - \hat{B})\theta \le b^\star - \hat{A}m \end{array} \right\} \tag{14}$$

In the theory of multi-parametric programming [33], set $\mathcal{CR}$ in (14) is called a critical region. When θ varies in $\mathcal{CR}$, the set of active constraints does not change, so the optimal solution can always be recovered from (11), regardless of the value of θ. Therefore, in a critical region, the optimal solution $x^\star$ and dual variable $\lambda^\star$ are affine functions of θ. If θ steps outside one critical region, the set of active constraints change, and another critical region can be found. From (14) we can see critical regions are polyhedral sets.

The solution properties of problem (8) under a given binary variable $y^\star$ are summarized below:

- The parameter set Θ is partitioned into critical regions which are polyhedral sets.
- In each critical region, the optimal continuous variable $x^\star$ is an affine function in parameter θ
- The optimal objective is a quadratic function in θ

In the ODT method, each tree node is split by a hyperplane $\mathbf{a_i^T}\theta = \mathbf{b_i}$. Such a hyperplane is the boundary of some critical regions in (14). In this sense, the result of ODT is interpretable and coincides with the theoretical analysis.

In order to show the interpretability more intuitively, we apply both analytical methods and the proposed learning method to the direct-current OPF problem

$$\min \quad \sum_i a_i p_{gi}^2 + b_i p_{gi} \tag{15a}$$

$$\text{s.t.} \quad p_{gi} + p_{wi} - p_{di} = \sum_{k \in \Lambda_i} \frac{\delta_i - \delta_k}{X_{ik}}, \forall i \tag{15b}$$

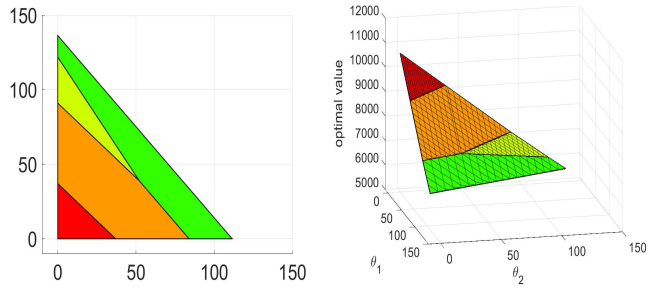
$$-P_{ik}^{Lm} \le p_{ik}^L \le P_{ik}^{Lm}, \forall i, \forall k \in \Lambda_i \tag{15c}$$

$$P_{gi}^l \le p_{gi} \le P_{gi}^m, \forall i \tag{15d}$$

$$-\pi \le \delta_i \le \pi, \forall i \tag{15e}$$

$$\Theta = \left\{ p_w \middle| p_{w,i}^{min} \le p_{w,i} \le p_{w,i}^{max} \right\} \tag{15f}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2021.3088951, IEEE Transactions on Sustainable Energy



(a) Critical regions by (14) (b) Optimal value function

Fig. 3: Critical regions and the optimal value function

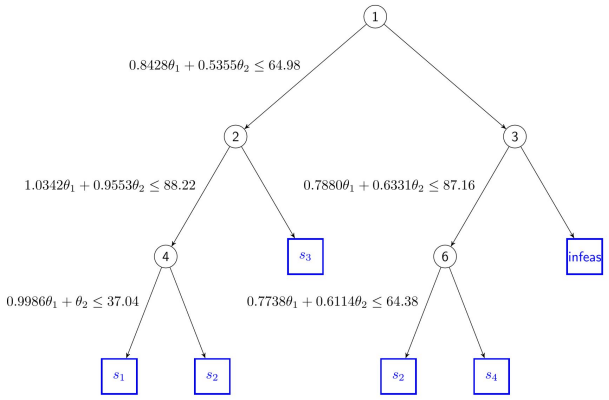
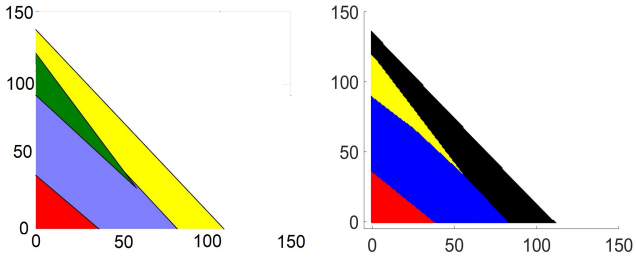


Fig. 4: Training result of ODT



(a) Critical regions by ODT (b) Critical regions by ANN

Fig. 5: Critical regions learned by ML methods

Decision variables are the generation outputs $p_{g,i}$, bus angles δ_i and line flows p_{ik}^L. Parameters are the outputs of wind farms p_{wi}. Load demand p_{di} is assumed to be fixed for ease of exhibition. a_i, b_i are the coefficients of generation cost functions. X_{ik} is the reactance of the branch connecting buses i and k; P_{ik}^{Lm} is the power flow capacity. P_{gi}^l/P_{gi}^m and $p_{w,i}^{min}/p_{w,i}^{max}$ are the minimum/maximum output of thermal units and wind power at bus i. The objective function (15a) is to minimize the total fuel cost of thermal units. Constraint (15b) is the nodal power balance equation, while the remaining constraints in (15c)-(15f) define lower and upper bounds of decision variables and parameters.

A test case is constructed based on a modified IEEE 30-bus system with two wind farms connected at bus 26 and bus 29. The output range of the two wind farms are the parameters and vary in the interval $[0, 150]$MW. The critical regions obtained from the theoretical analysis are displayed in Fig. 3, where the grey area is the infeasible region, i.e., for any θ in this region, the OPF problem (15) is infeasible. The hyperplane expressions of boundaries in Fig. 3 are calculated by (14) and summarized in (16a)-(16e).

$$\theta_1 + \theta_2 = 37.05 \tag{16a}$$
$$\theta_1 + \theta_2 = 91.04 \tag{16b}$$
$$0.7741\theta_1 + 0.6331\theta_2 = 64.98 \tag{16c}$$
$$0.8469\theta_1 + 0.5318\theta_2 = 64.83 \tag{16d}$$
$$0.7741\theta_1 + 0.6331\theta_2 = 86.32 \tag{16e}$$

Equalities (16a) and (16b) imply that some generators reach the minimum output when the total wind generation increases to certain level. Equalities (16c), (16d), and (16e) originate from the limitations of line flow capacity (15c).

Then, the ANN and ODT methods are performed to learn the set of active constraints, as there is no binary variable in the OPF problem (15). For the ODT method, the decision tree after training is displayed in Fig. 4; the splitting criteria are shown beside each branch. It is observed that the splitting criteria in ODT are very close to the exact partition of critical regions. The criteria of nodes 1-5 match (16d), (16b), (16e), (16a), and (16c) respectively. Such coincidence explains that the true optimal dispatch policy can be learned with high accuracy.

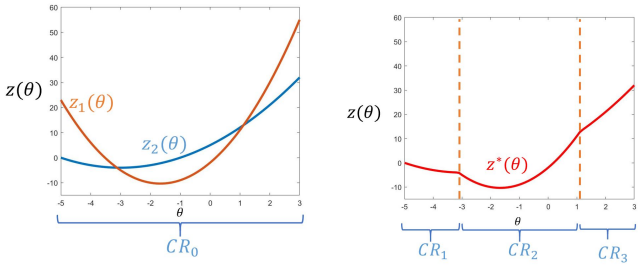
The ANN method is also conducted, but the ANN does not offer a directly interpretable result. Nevertheless, since ANN and ODT identify the set of active constraints, the classification results are examined point-by-point and exhibited in Fig. 5. Compared to Fig. 3(a), we can see two ML methods provide almost exact partitions of critical regions, and the results are well interpreted via the tree in Fig. 4.

The above example considers a continuous optimization problem. When binary variables are taken into account, the situation is more complicated. We can think that the MIQP can be solved at all possible values of binary variables. Each binary variable corresponds to an optimal value function, then the optimal value function of the MIQP is the pointwise minimum of all the optimal value functions with fixed binary variables[1]. For example, Fig. 6(a) shows the optimal value functions $z_1(\theta)$ and $z_2(\theta)$ under different binary variables. The pointwise minimum $z(\theta) = \min\{z_1(\theta), z_2(\theta)\}$ is piecewise quadratic and drawn in Fig. 6(b). The region CR_0 is divided into three critical regions CR_1, CR_2, and CR_3. In each new critical region, the optimal continuous variable remains an affine function in the parameter θ.

Although the solution structure of MIQP has been analyzed, solving a parametric MIQP is much more difficult than solving a single MIQP, especially when the dimension of the parameter is high. The proposed learning method is scalable to the dimension of parameter θ.

[1]Among all possible combinations of 0-1 variables, only a few are candidate optimal solutions when θ varies in a certain range. In the offline training, a sufficient number of optimal patterns are explored by solving (1) with sampled data. There is no need to traverse binary values or solving any optimization problem during the online phase, which saves computation time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2021.3088951, IEEE Transactions on Sustainable Energy



(a) Objectives and CRs before comparison

(b) Objectives and CRs after comparison

Fig. 6: Optimal value function and critical regions.

IV. Extension of the Proposed Method

As the cost of fossil-fuel generators is a quadratic function, the discussions in Section II rely on MIQP. In some power markets, the independent system operators ask for a linear cost bid, yielding an MILP, a special case of MIQP in which $Q = 0$. Hence, the optimal primal and dual variables can be analyzed separately. The cases with a linear objective function and line contingency constraints are also discussed in this section.

A. Suitability for Linear Case

For MILP, the offline learning process remains unchanged. For a given optimal pattern with a binary variable $y^\star$ and a set of active constraints $\bar{A}x + \bar{E}y^\star = \bar{b} + \bar{B}\theta$, the optimal continuous variable solves

$$\begin{aligned} \min \quad & c_1^\top x \\ \text{s.t.} \quad & \bar{A}x = \bar{b}^\star + \bar{B}\theta \end{aligned} \tag{17}$$

where $\bar{b}^\star = \bar{b} - \bar{E}y^\star$. The optimality condition of (17) is

$$\begin{aligned} \bar{A}^\top \lambda^\star &= c_1 \\ \bar{A}x^\star &= \bar{b}^\star + \bar{B}\theta \end{aligned} \tag{18}$$

where $\lambda^\star$ is the dual variable corresponding to the active constraints. When the problem is non-degenerated, matrix $\bar{A}^\top$ is invertible, and $(x^\star, \lambda^\star)$ can be expressed as

$$\begin{aligned} x^\star &= \bar{A}^{-1}\bar{b}^\star + \bar{A}^{-1}\bar{B}\theta \\ \lambda^\star &= (\bar{A}^\top)^{-1}c_1 \end{aligned} \tag{19}$$

The remaining dual variables $\hat{\lambda}$ for inactive constraints are 0 according to complementary and slackness condition.

The inactive constraints are

$$\hat{A}x^\star < \hat{b}^\star + \hat{B}\theta \tag{20}$$

Substituting (19) into (20), the critical region can be obtained

$$\mathcal{CR} = \left\{ \theta \middle| (\hat{A}\bar{A}^{-1}\bar{B} - \hat{B})\theta < \hat{b}^\star - \hat{A}\bar{A}^{-1}\bar{b}^\star \right\} \tag{21}$$

The set of active constraints keeps invariant in a critical region, i.e., $(\bar{A}, \bar{B}, \bar{b}^\star)$ does not change. So $x^\star$ is an affine function of θ, and $\lambda^\star$ is constant in a critical region according to (19). In fact, (19) and (21) are special cases of (11) and (14). When $Q = 0$, the conditions on primal variable $x^\star$ and dual variable $\lambda^\star$ are decoupled.

In summary, the proposed method can be applied to MILP-based problems with minor adjustments on the calculation of continuous optimizer.

B. Implication on LMP

The above discussion inspires learning the map from parameter θ to locational marginal prices (LMPs). Indeed, a multi-parametric linear programming method for LMP forecast was proposed in [34]. In OPF problem (15) with a linear objective function, the LMP at each bus can be extracted from the dual variable γ associated with each nodal power balancing constraint (15b). By the expression in (19), in each critical region LMP remains constant. Therefore, we just need to learn the critical regions and the corresponding value of LMP. The procedures are as follows:

Offline Training:

(a) Formulate the DCOPF problem as an mp-LP problem.
(b) Retrieve data samples and generate the pattern set. Each pattern corresponds to a set of active constraints.
(c) For each pattern, record the coefficients $(\bar{A}, \bar{B}, \bar{b}^\star)$, $(\hat{A}, \hat{B}, \hat{b}^\star)$ and $\lambda^\star$ in (19)-(20).
(d) Train the ML model for the classification task.

Online Prediction:

(a) Predict top-k candidate patterns for the sample.
(b) Retrieve k candidate solutions $(x^1, ..., x^k)$ for the D-COPF problem, and find out the optimal one $x^\star$ by comparing the objective value. Regard the corresponding $\lambda^\star$ as the predicted LMP.

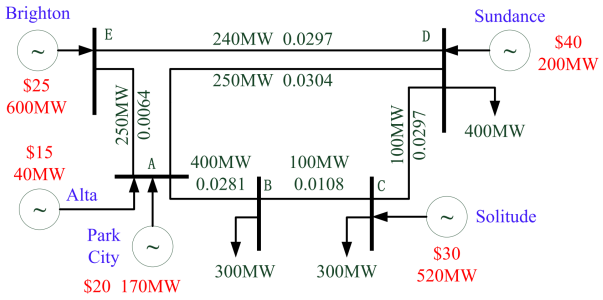
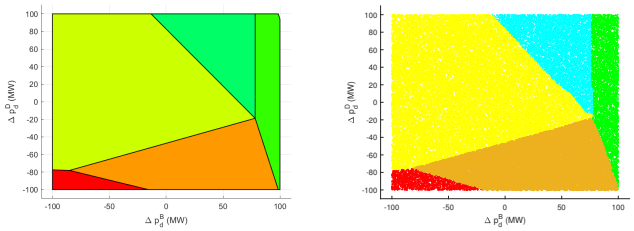


Fig. 7: Topology and parameters of PJM 5-bus system

A test case of LMP is constructed based on a modified PJM 5-bus system. The original system can be found in [35]. The topology and parameters of the system are displayed in Fig. 7. The varying parameters in the problem are the load variations at bus B and D with the range of $[-100, 100]$MW. 10000 samples are generated in the proposed method, including 8000 training samples and 2000 test samples. The critical regions acquired by the theoretical analysis and the proposed method with ANN are shown in Fig. 8a and Fig.8b respectively.

The LMP in each critical region is a constant, which can be accurately inferred by (19). Therefore, the error of the proposed method only comes from the misclassification of patterns, i.e., the error on the division of critical regions. Results show that when top-2 predictions are provided, all the test samples are given a correct pattern, and all the predicted LMPs are accurate.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2021.3088951, IEEE Transactions on Sustainable Energy



(a) Critical regions by (21) (b) Critical regions by learning

Fig. 8: Comparison of the critical regions

C. Incorporating Contingency Constraints

Sometimes, contingency constraint is also included in the UC problem, ensuring the security of power flow redistribution after the tripping of a transmission line. The $N-1$ contingency constraint can be cast as [36]

$$-F_l \leq \sum_i \pi_{il}^k p_i^g - \sum_j \pi_{jl}^k p_j^d \leq F_l, \ \forall l, \forall k \tag{22}$$

where p_i^g represents the generator output at bus i; p_j^d is the load demand at bus j; F_l is the power flow capacity of line l; π_{il}^k/π_{jl}^k represents the power transfer distribution factor from bus i/j to line l under contingency scenario k. In UC problem, constraint (22) holds in every period.

As the contingency constraint (22) is linear, the UC problem remains an MILP/MIQP. However, the contingency constraint set introduces a large number of inequalities in the parametric programming problem, and the majority of them are redundant, or in other words, removing any of the redundant constraints does not change the feasible set of the original UC problem. Hence, to improve computational efficiency, it is helpful to remove redundant constraints in advance. To this end, several redundancy filtering methods have been proposed, such as those in [36]–[38]. Here we apply a data-driven method to reduce the size of the contingency constraint set. First, the UC problem without contingency constraints is solved under each set of parameters in the training set, and line power flow under each contingency can be calculated based on the optimal generation schedule. With N_{train} training samples, we can get N_{train} groups of line flow data. By traversing such data, the overloaded lines and the corresponding contingency scenarios are obtained. Assuming that the number of training samples is sufficient, we can pick up necessary line flow constraints that could impact the feasible region of the SCUC problem and circumvent redundancy.

V. Case Study

In this section, the proposed method is tested via the network constrained unit commitment problem (NCUC). Solar outputs are regarded as the varying parameters in the problem. The detailed model of NCUC is given in the Appendix, which has the canonical form of (1). Testing systems include the IEEE 57-bus system and a real-world 1881-bus system in China. The case studies aim to exhibit the performance of the proposed learning method for solving a single instance of MIQP (1) with a fixed θ, from the perspectives of computation efficiency, infeasibility rate, and optimality gap. In practice, MIQP (1) can be solved in a rolling horizon fashion with the latest forecast of θ, so that unit on-off status can be updated sub-hourly, for example, 15 minutes, supporting the operation of fast-response gas-fired units. As the renewable forecast is beyond the scope of this paper, the rolling horizon implementation is omitted from the case study.

A. IEEE 57-bus system

The System data can be found in Matpower toolbox [39]. Two solar stations with 100MW rated capacity connect to the system at bus 8 and bus 9. Parameter θ includes solar outputs of two stations in the next 24 periods, so the dimension of θ is 48. In order to fully sample the parameter space, adequate solar output curves should be obtained. In this study, the solar data is acquired based on the method in [40].

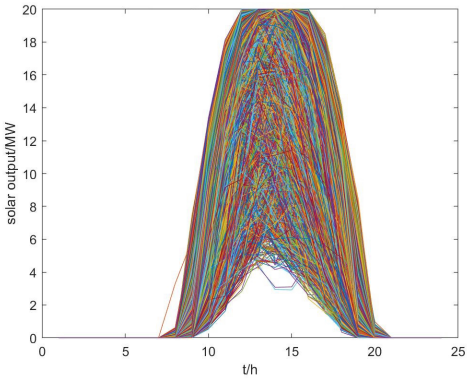


Fig. 9: Solar curves in the sample set

ANN is applied to learn the optimal patterns. Each of the two hidden layers consists of 30 nodes with ReLU activation, and the output layer is activated by the softmax function in order to predict the probabilities of each pattern.

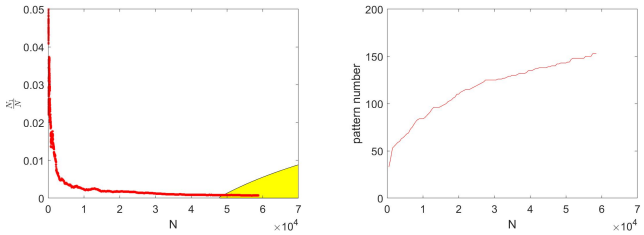
The experiment is implemented on a laptop with an Intel i7-1065G7 CPU and 16 GB memory. Optimization problems are solved by CPLEX 12.9 with YALMIP interface [41]. ANN is implemented by TensorFlow [42].

For online prediction, 5 candidate patterns are selected and the final predicted solution is the optimal one among the 5 solutions. The ANN is trained with 60000 sampled curves shown in Fig. 9. Another 20000 samples are used to examine the performance of the ML method.

First, the effect of the Good-Turing estimator is examined and exhibited in Fig. 10. In Fig. 10(a), it is observed that the ratio N_1/N decreases during sampling, indicating that there are fewer patterns that appeared only once with the increase of the number of samples. The shaded area in Fig. 10(a) contains the points where the stopping criterion (7) is met. When the sampling number N is greater than 50000, the curve of $N \sim N_1/N$ enters the shaded area, and hence the sampling procedure is terminated. Fig. 10b displays the change of pattern number during sampling. It can be seen that when $N = 50000$, the number of optimal patterns is around

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2021.3088951, IEEE Transactions on Sustainable Energy

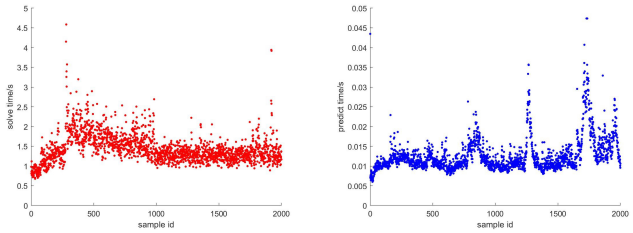
150, and the further growth of N does not bring to a notable increase of new patterns, validating the effectiveness of the stopping criterion.



(a) Change of N_1/N (b) Change of pattern number

Fig. 10: Validation of the Good-Turing estimator

The computation time is checked. For each validation sample, we call CPLEX to solve the corresponding NCUC; the time is recorded and shown in Fig. 11a. The validation samples are sent to the ANN, and the online prediction is executed; the time is exhibited in Fig. 11b. It is observed that compared to directly solving NCUC with CPLEX, the acceleration brought by the ML method is about two orders of magnitudes on average, because only a set of linear equations is solved during the online prediction stage. The acceleration could be more significant if the system is larger and possesses more generators.



(a) Solution time of CPLEX (b) Online prediction time of the proposed method

Fig. 11: Comparison of solution time.

Table I shows the results of top-5 prediction with different numbers of training samples. The size of the training set is within 60000, while another 20000 samples serve as the validating set. The size of the pattern set denotes the number of optimal patterns that have appeared in the training set. The infeasibility rate means the proportion of test samples that cannot be associated with a feasible solution with the predicted patterns. The sub-optimality rate means the proportion of test samples that can only acquire a suboptimal solution with the predicted patterns. The maximum gap of sub-optimality for the test set is defined as

$$\max_i \frac{obj_i^{pred} - obj_i^{actu}}{obj_i^{actu}} \tag{23}$$

where obj_i^{pred} and obj_i^{actu} represent the predicted objective value and the actual one (reported by CPLEX with the default optimality gap) of the i-th test sample respectively.

From Table I, we can observe that the infeasibility rate continues to drop as more patterns are discovered. When the size of the training set increases to 60000, only one sample among the 20000 test samples is predicted to be infeasible. Additionally, the sub-optimality rate is also small enough, and the gap of sub-optimality is no more than 0.21%, indicating that even if a suboptimal value is predicted, it is still very close to the true optimal one. The above results validate the feasibility and efficiency of the proposed method.

TABLE I: Test results by top-5 prediction

Training set size	10000	20000	40000	60000
Size of pattern set	84	110	134	153
Infeasibility rate	0.1%	0.05%	0.025%	0.005%
Suboptimality rate	0.13%	0.085%	0.05%	0.055%
Max suboptimality gap	0.21%	0.21%	0.21%	0.13%

It is found that the value of k also influences the quality of predictions, as an increase of k implies a higher possibility of including the true optimal pattern. Table II and Table III provide additional results with $k = 30$ and $k = N_p$ respectively, where N_p is the size of the pattern set, implying the latter one traverses all the optimal patterns found with the training data. Results in Table I-Table III show that a larger k leads to a reduced feasibility rate and optimality gap. Table III suggests that when the predictions are acquired by traversing the optimal pattern set, all the predicted solutions are feasible, and the maximum optimality gap is no more than 0.0012%, demonstrating the effectiveness of the proposed method. Although a larger value of k leads to an increase in the prediction time, the online time is still kept within 0.16s even if all the optimal patterns are traversed, which is fast enough in real applications.

TABLE II: Test results by top-30 prediction

Training set size	10000	20000	40000	60000
Size of pattern set	84	110	134	153
Infeasibility rate	0	0	0	0
Suboptimality rate	0.06%	0.035%	0.025%	0.02%
Max suboptimality gap	0.21%	0.21%	0.13%	0.08%

TABLE III: Test results by traversing the pattern set

Training set size	10000	20000	40000	60000
Size of pattern set	84	110	134	153
Infeasibility rate	0	0	0	0
Suboptimality rate	0.055%	0.035%	0.01%	0.01%
Max suboptimality gap	0.21%	0.21%	0.0012%	0.0012%

Table IV compares the performance of the proposed method with a model-free direct prediction method recently proposed in [43]. It predicts each binary variable as a binary classification task. For the continuous variable, k-means clustering is implemented to approximate the division of critical regions, followed by linear regressions between each continuous variable and parameters in each cluster. Such a method directly predicts the optimal solution without exploiting the solution

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSTE.2021.3088951, IEEE Transactions on Sustainable Energy

structure of the optimization model discussed in Section III. The method is applied to the UC problem of the above 57-bus system. The same 60000 samples are used to train the model.

TABLE IV: Comparison of two ML methods

Method	Proposed method	Direct prediction
Infeasibility rate	0	21.65%
Suboptimality rate	0.01%	57.16%
Max suboptimality gap	0.0012%	6.94%
Average online computation time	0.01s	0.01s

Table IV suggests that although both methods have similar computation time, the direct prediction method exhibits much higher infeasibility rates and optimality gaps, because it neglects the constraints as well as the intrinsic solution structure of the UC model, and predicts the binary variables and continuous variables separately. In contrast, the proposed method takes full advantage of the optimization model, so it is able to provide high-quality solutions.

Further tests considering $N-1$ line contingency are also carried out. Before the training phase, the constraint reduction method in Section IV-C is applied to accelerate sample generation. Compared with conventional UC problems, the inclusion of contingency constraint leads to the growth of optimal patterns, therefore, a larger dataset with 120000 training samples and 20000 test samples are used. The results of the top-50 prediction are shown in Table V. It can be seen that the size of the pattern set approximately converges to 700 as the training set size increases, and the proposed method exhibits a satisfactory performance. When the training set size reaches 120000, only 0.01% of test samples are predicted to be infeasible, 0.04% of predicted solutions are suboptimal, and the maximum sub-optimality gap is no more than 0.1%, demonstrating the effectiveness of the proposed method on contingency-constrained UC problems.

TABLE V: Test results considering contingency

Training set size	20000	50000	80000	120000
Size of pattern set	403	569	664	716
Infeasibility rate	0.07%	0.05%	0.03%	0.01%
Suboptimality rate	0.09%	0.04%	0.03%	0.04%
Max suboptimality gap	0.14%	0.084%	0.084%	0.082%

B. A real-world 1881-bus system

A real-world power system in China is used to test the performance of the proposed method on large-scale systems. It consists of 1881 buses, 2760 transmission lines, 180 thermal generators, and 3 wind farms. The total installed capacity of thermal units is 48.7GW, the peak load is 41.8GW, and the capacity of each wind farm is 1.5GW. The classification model is trained on 2 servers, each with 4×22-core Intel Xeon CPU (E5-2699 v4 @ 2.20GHz) and 1024 GB memory. Optimization problems are solved by CPLEX 12.9. The sample set consists of 40000 training samples and 10000 test samples.

To maintain a fair comparison, the offline solution time of CPLEX and the online prediction time of the proposed method are tested on the same laptop used previously. Using the default setting with an optimality gap of 0.01%, CPLEX solver fails to find an optimal solution in 10 hours. So the criterion for termination is manually modified to a 0.1% optimality gap. We solve 100 instances with samples randomly chosen from the 10000 testing samples and record the statistical performance of CPLEX. Results are displayed in Table VI, from which we can see that it takes CPLEX around 1 hour to solve the UC problem. For the proposed ML method, $k=50$ is used for the ANN-based classification method. The online prediction time of all the 10000 samples is also recorded and exhibited in Fig. 12. The average online processing time is only 0.42s, achieving an acceleration of four orders of magnitudes.

TABLE VI: Comparison results on the 1881-bus system

Method	CPLEX solver	Proposed method
Average solution time	3448s	0.42s
Maximum solution time	4026s	1.41s
Minimum solution time	2574s	0.35s
Standard Deviation of solution time	386s	0.07s

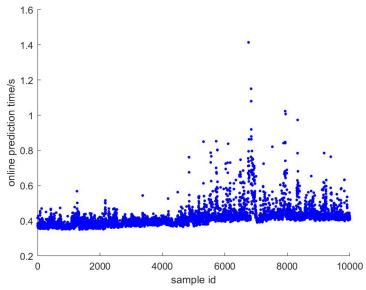


Fig. 12: Online prediction time of the proposed method

TABLE VII: Test results on the 1881-bus system

Training set size	10000	20000	30000	40000
Size of pattern set	192	243	270	288
Infeasibility rate	0.22%	0.16%	0.11%	0.10%
Max suboptimality gap	0.0020%	0.0018%	0.0018%	0.0018%

Table VII presents the quality of the solution offered by the proposed method. In this case, as CPLEX fails to find the true optimal solution with default optimality gap, the approximate optimal solution with an optimality gap of not more than 0.1% is used as the benchmark; obj_i^{CPLEX} denotes the objective value of sample i at this solution. The maximum sub-optimality gap is defined as

$$\max_i \frac{obj_i^{pred} - obj_i^{CPLEX}}{obj_i^{CPLEX}} \tag{24}$$

where obj_i^{pred} is the objective value of sample i at the predicted solution. Table VII suggests that when the number of samples is sufficiently large, only 0.1% of solutions for test samples are infeasible, and the objective value predicted by the proposed method is very close to the one offered by CPLEX. In summary, the proposed method has the potential to be used in real-world large-scale power systems.

VI. Conclusions

This paper proposes a classification based method to predict the optimal solution of power system dispatch problems modeled by MIQPs/MILPs. Instead of directly learning the optimal strategy, the proposed method learns the map from the parameter to the optimal patterns which are discrete and accounts for mandatory constraints. The tree-based classification gives interpretable policy and the ANN-based one has better efficiency in high-dimensional problems. Results of the DCOPF problem is compared with the analytical method, and validates the accuracy. Case studies on the UC problem demonstrate that the online processing time is less than a second, because the online prediction entails solving only a set of linear equations. The strategy offered by the ML method exhibits a small optimality gap with a magnitude of 0.1%, which is generally acceptable in practice. LMP can be obtained conveniently without extra effort.

Future work is to increase the efficiency of sampling: using more dedicated samples that are properly chosen rather than randomly chosen, so as to reduce the required number of samples while maintaining a sufficiently low infeasibility rate and optimality gap. Extending the method in a dynamic environment and tackling multi-stage decision-making problems are also promising directions.

Appendix A
Formulation of NCUC

An NCUC problem can be modeled as the following MIQP:

$$\begin{aligned} \min \quad & \sum_{t=1}^{T} \sum_{i} \left(F_{it}^{U}(u_{it}, v_{it}, z_{it}) + F_{it}^{C}(p_{it}) \right) && \text{(25a)} \\ \text{s.t.} \quad & \textbf{Cons-Binary}, \ \textbf{Cons-Continuous} \end{aligned}$$

where $z_{it} = 1/0$ means the unit is on/off in period t; $u_{it} = 1$ means the unit is startup in period t; otherwise $u_{it} = 0$; $v_{it} = 1$ means the unit is shutdown in period t; otherwise $v_{it} = 0$; p_{it} is the output of unit i in period t; function

$$F_{it}^{U}(u_{it}, v_{it}, z_{it}) = SU_i u_{it} + SD_i v_{it} + SZ_i z_{it} \tag{25b}$$

consists of startup cost, shutdown cost and running cost, where SZ_i is the fixed running cost coefficient, and SU_i/SD_i is the startup/shutdown cost coefficient. Function

$$F_{it}^{C}(p_{it}) = a_i p_{it}^2 + b_i p_{it} \tag{25c}$$

is the variable operation cost, depending on the output level. **Cons-Binary** encapsulates all constraints involving binary variables:

$$\begin{aligned} -z_{i(t-1)} + z_{it} - z_{ik} \le 0, \forall i, \forall t, \\ \forall k \in \{t, \dots, T_i^{on} + t - 1\} \end{aligned} \tag{25d}$$

$$\begin{aligned} z_{i(t-1)} - z_{it} + z_{ik} \le 1, \forall i, \forall t, \\ \forall k \in \left\{t, \dots, T_i^{off} + t - 1\right\} \end{aligned} \tag{25e}$$

$$-z_{i(t-1)} + z_{it} - u_{it} \le 0, \forall i, \forall t \tag{25f}$$

$$z_{i(t-1)} - z_{it} - v_{it} \le 0, \forall i, \forall t \tag{25g}$$

$$z_{it}, u_{it}, v_{it} \in \{0, 1\}, \forall i, \forall t \tag{25h}$$

where T_i^{on} and T_i^{off} are the minimum-up and minimum-down time of unit i; constraints (25d)-(25e) denote the minimum-up and minimum-down time of each generator, respectively; inequality (25f) forces $u_{it} = 1$ when $z_{i(t-1)} = 0$ and $z_{it} = 1$ to represent the startup action, otherwise $u_{it} = 0$ because of the minimization of objective function. Inequality (25g) influences v_{it} in a similar way to represent the shutdown action. **Cons-Continuous** includes all constraints involving continuous variables:

$$z_{it} p_i^l \le p_{it} \le z_{it} p_i^m, \forall i, \forall t \tag{25i}$$

$$\begin{aligned} p_{it} - p_{i(t-1)} \le \left(2 - z_{i(t-1)} - z_{it}\right) p_i^l \\ + \left(1 + z_{i(t-1)} - z_{it}\right) RU_i, \forall i, \forall t \end{aligned} \tag{25j}$$

$$\begin{aligned} p_{i(t-1)} - p_{it} \le \left(2 - z_{i(t-1)} - z_{it}\right) p_i^l \\ + \left(1 - z_{i(t-1)} + z_{it}\right) RD_i, \forall i, \forall t \end{aligned} \tag{25k}$$

$$\sum_{\mathrm{i}} p_{it} + \sum_{k} p_{kt}^s = \sum_{j} p_{jt}^d, \forall t \tag{25l}$$

$$\begin{aligned} -F_l \le \sum_i \pi_{il} p_{it} + \sum_k \pi_{kl} p_{kt}^s \\ - \sum_j \pi_{jl} p_{jt}^d \le F_l, \forall l, \forall t \end{aligned} \tag{25m}$$

where p_i^l and p_i^m are the minimum and maximum output of generator i, RU_i/RD_i are the ramp-up/ramp-down limits of generator i; p_{jt}^d represents power demand of bus j at time t; p_{kt}^s is the renewable output at bus k and time t; F_l is the line flow limit of line l; π_{il} represents the power transfer distribution factor from bus i to line l; (25i) indicates generation capacity, (25j) and (25k) prescribe upward and downward ramping limits, (25l) imposes system-wide power balancing, and (25m) stipulates the flow capacity limit of each transmission line.

All the constraints (25d)-(25m) are linear, and the problem is an MIQP. Let $x = \{p_{i,t} | \forall i, \forall t\}$ and $y = \{z_{it}, u_{it}, v_{it} | \forall i, \forall t\}$ represent continuous and discrete variables respectively. Parameter is renewable output $\theta = \{p_{kt}^s | \forall k, \forall t\}$. Then NCUC comes down to the canonical form in (1).

For more comprehensive discussions on the concrete optimization models of UC, ED and OPF, please refer to [44].

References

[1] Y. Chen, A. Casto, F. Wang, Q. Wang, X. Wang, and J. Wan, "Improving large scale day-ahead security constrained unit commitment performance," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4732–4743, Nov. 2016.

[2] Y. Tang, K. Dvijotham, and L. S. H., "Real-time optimal power flow," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2963–2973, Nov. 2017.

[3] Y. Zhang, E. Dall'Anese, and M. Hong, "Dynamic ADMM for real-time optimal power flow," in *2017 IEEE Global Conference on Signal and Information Processing*, Montreal, QC, Canada, Nov. 2017, pp. 1085–1089.

[4] F. Guo, C. Wen, J. Mao, J. Chen, and Y.-D. Song, "Hierarchical decentralized optimization architecture for economic dispatch: A new approach for large-scale power system," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 523–534, Feb. 2017.

[5] C.-Y. Zhang, C. P. Chen, M. Gan, and L. Chen, "Predictive deep Boltzmann machine for multiperiod wind speed forecasting," *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1416–1425, Oct. 2015.

[6] C. Li, G. Tang, X. Xue, A. Saeed, and X. Hu, "Short-term wind speed interval prediction based on ensemble GRU model," *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1370–1380, Jul. 2020.

[7] M. Khodayar and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 670–681, Apr. 2019.

[8] Q. Zhu, J. Chen, D. Shi, L. Zhu, X. Bai, X. Duan, and Y. Liu, "Learning temporal and spatial correlations jointly: A unified framework for wind speed prediction," *IEEE Trans. Sustain. Energy*, vol. 11, no. 1, pp. 509–523, Jan. 2020.

[9] O. Abedinia, M. Lotfi, M. Bagheri, B. Sobhani, M. Shafie-khah, and J. P. S. Catalao, "Improved EMD-based complex prediction model for wind power forecasting," *IEEE Trans. Sustain. Energy*, vol. 11, no. 4, pp. 2790–2802, Oct. 2020.

[10] C. Feng, M. Cui, B. Hodge, S. Lu, H. F. Hamann, and J. Zhang, "Unsupervised clustering-based short-term solar forecasting," *IEEE Trans. Sustain. Energy*, vol. 10, no. 4, pp. 2174–2185, Oct. 2019.

[11] L. Gigoni, A. Betti, E. Crisostomi, A. Franco, M. Tucci, F. Bizzarri, and D. Mucci, "Day-ahead hourly forecasting of power generation from photovoltaic plants," *IEEE Trans. Sustain. Energy*, vol. 9, no. 2, pp. 831–842, Apr. 2018.

[12] S. Wen, C. Zhang, H. Lan, Y. Xu, Y. Tang, and Y. Huang, "A hybrid ensemble model for interval prediction of solar power output in ship onboard power systems," *IEEE Trans. Sustain. Energy*, vol. 12, no. 1, pp. 14–24, Jan. 2021.

[13] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang, and G. Liu, "Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting," *IEEE Trans. Sustain. Energy*, vol. 11, no. 2, pp. 571–583, Apr. 2020.

[14] W. Zhang, Y. Luo, Y. Zhang, and D. Srinivasan, "SolarGAN: Multivariate solar data imputation using Generative Adversarial Network," *IEEE Trans. Sustain. Energy*, vol. 12, no. 1, pp. 743–746, Jan. 2021.

[15] X. Lei, Z. Yang, J. Yu, J. Zhao, Q. Gao, and H. Yu, "Data-driven optimal power flow: A physics-informed machine learning approach," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 346–354, Jan. 2021.

[16] R. Nayak and J. Sharma, "A hybrid neural network and simulated annealing approach to the unit commitment problem," *Comput. Electr. Eng.*, vol. 26, no. 6, pp. 461–477, Aug. 2000.

[17] D. N. Vo and W. Ongsakul, "Economic dispatch with multiple fuel types by enhanced augmented Lagrange Hopfield network," *Appl. Energy*, vol. 91, no. 1, pp. 281–289, Mar. 2012.

[18] Z. Hu, Y. Xu, M. Korkali, X. Chen, L. Mili, and J. Valinejad, "A Bayesian approach for estimating uncertainty in stochastic economic dispatch considering wind power penetration," *IEEE Trans. Sustain. Energy*, vol. 12, no. 1, pp. 671–681, Jan. 2021.

[19] T. Yu, J. Liu, K. Chan, and J. Wang, "Distributed multi-step Q(λ) learning for optimal power flow of large-scale power grids," *Int. J. Electr. Power Energy Syst.*, vol. 42, no. 1, pp. 614–620, Nov. 2012.

[20] Z. Yan and Y. Xu, "Real-time optimal power flow: A Lagrangian based deep reinforcement learning approach," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 3270–3273, Jul. 2020.

[21] W. Liu, P. Zhuang, H. Liang, J. Peng, and Z. Huang, "Distributed economic dispatch in microgrids based on cooperative reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2192–2203, Jun. 2018.

[22] E. Jasmin, T. I. Ahamed, and T. Remani, "A function approximation approach to reinforcement learning for solving unit commitment problem with photo voltaic sources," in *2016 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*. Trivandrum, India: IEEE, Dec. 2016, pp. 1–6.

[23] M. Zhou, B. Wang, T. Li, and J. Watada, "A data-driven approach for multi-objective unit commitment under hybrid uncertainties," *Energy*, vol. 164, pp. 722–733, Dec. 2018.

[24] G. Longoria, A. Davy, and L. Shi, "Subsidy-free renewable energy trading: A meta agent approach," *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1707–1716, Jul. 2020.

[25] L. Xi, L. Yu, Y. Xu, S. Wang, and X. Chen, "A novel multi-agent DDQN-AD method-based distributed strategy for automatic generation control of integrated energy systems," *IEEE Trans. Sustain. Energy*, vol. 11, no. 4, pp. 2417–2426, Oct. 2020.

[26] S. Wang, S. Bi, and Y. J. A. Zhang, "Reinforcement learning for real-time pricing and scheduling control in EV charging stations," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 849–859, Feb. 2021.

[27] D. Bertsimas and B. Stellato, "The voice of optimization," *Machine Learning*, vol. 110, no. 2, pp. 249–277, 2021.

[28] M. Costley, M. J. Feizollahi, S. Ahmed, and S. Grijalva, "A rolling-horizon unit commitment framework with flexible periodicity," *Int. J. Electr. Power Energy Syst*, vol. 90, pp. 280–291, Sep. 2017.

[29] M. Zhou, B. Wang, and J. Watada, "Deep learning-based rolling horizon unit commitment under hybrid uncertainties," *Energy*, vol. 186, p. 115843, Nov. 2019.

[30] E. A. Bakirtzis, P. N. Biskas, D. P. Labridis, and A. G. Bakirtzis, "Multiple time resolution unit commitment for short-term operations scheduling under high renewable penetration," *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 149–159, Jan. 2014.

[31] D. A. McAllester and R. E. Schapire, "On the convergence rate of Good-Turing estimators." in *COLT*, 2000, pp. 1–6.

[32] S. K. Murthy, S. Kasif, and S. Salzberg, "A system for induction of oblique decision trees," *J. Artif. Intell. Res.*, vol. 2, pp. 1–32, 1994.

[33] E. N. Pistikopoulos, *Multi-parametric programming*. Wiley-vch, 2007.

[34] Y. Ji, R. J. Thomas, and L. Tong, "Probabilistic forecasting of real-time LMP and network congestion," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 831–841, Mar. 2016.

[35] F. Li and R. Bo, "Small test systems for power system economic studies," in *IEEE PES general meeting*. Minneapolis, MN, USA: IEEE, Jul. 2010, pp. 1–4.

[36] A. J. Ardakani and F. Bouffard, "Acceleration of umbrella constraint discovery in generation scheduling problems," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 2100–2109, Jul. 2015.

[37] D. A. Tejada-Arango, P. Sánchez-Martın, and A. Ramos, "Security constrained unit commitment using line outage distribution factors," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 329–337, Jan. 2018.

[38] A. S. Xavier, F. Qiu, F. Wang, and P. R. Thimmapuram, "Transmission constraint filtering in large-scale security-constrained unit commitment," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 2457–2460, May 2019.

[39] 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 Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[40] S. Pfenninger and I. Staffell, "Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, no. 1, pp. 1251–1265, Nov. 2016.

[41] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Proc. 2004 IEEE Int. Conf. Robot. Autom*, Taipei, Taiwan, Sep. 2004, pp. 284–289.

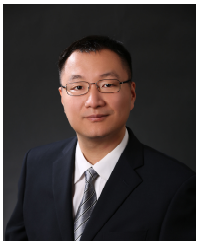
[42] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265–283.

[43] A. Shokry, S. Medina-González, and A. Espuña, "Mixed-integer multi-parametric approach based on machine learning techniques," in *Computer Aided Chemical Engineering*. Elsevier, 2017, vol. 40, pp. 451–456.

[44] W. Wei and J. Wang, *Modeling and Optimization of Interdependent Energy Infrastructures*. Switzerland: Springer, 2020.

Mingxuan Li received the B.S. degree from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree.

His research interests include machine learning, renewable power generation, and power system operation.

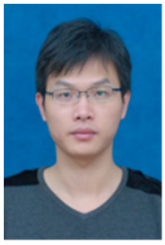


Wei Wei (M'15-SM'18) received the Bachelor and Ph.D. degrees, both in electrical engineering, from Tsinghua University, Beijing, China, in 2008 and 2013, respectively.

He was a Postdoctoral Researcher with the Tsinghua University from 2013 to 2015. He was a Visiting Scholar with Cornell University, Ithaca NY, USA, in 2014, and with the Harvard University, Cambridge MA, USA, in 2015. He is currently an Associate Professor with Tsinghua University. His research interests include computational optimization and energy system economics.

Yue Chen (S'16-M'21) received B.E. and Ph.D. degrees in Electrical Engineering from Tsinghua University, Beijing, China, in 2015 and 2020, respectively. She holds the B.S. degree in Economics from Peking University, Beijing, China, in 2017.

From 2018 to 2019, she was a visiting student at California Institute of Technology, Pasadena, CA, USA. Currently, she is an Assistant Professor with the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong. Her research interests include optimization, game theory, mathematical economics, and their applications in smart grid and integrated energy systems.



Ming-Feng Ge (M'17) received the B.Eng. degree in automation and the Ph.D. degree in control theory and control engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2016, respectively.

He is currently a Professor with the School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan. His current research interests include networked robotic systems, smart grids and human-in-the-loop systems.

João P. S. Catalão (Senior Member, IEEE) received the M.Sc. degree from the Instituto Superior Tecnico (IST), Lisbon, Portugal, in 2003, and the Ph.D. degree and Habilitation for Full Professor from the University of Beira Interior (UBI), Covilha, Portugal, in 2007 and 2013, respectively.

Currently, he is a Professor at the Faculty of Engineering of the University of Porto (FEUP), Porto, Portugal, and Research Coordinator at INESC TEC. He was also appointed as Visiting Professor by North China Electric Power University, Beijing, China. He was the Primary Coordinator of the EU-funded FP7 project SiNGULAR (Smart and Sustainable Insular Electricity Grids Under Large-Scale Renewable Integration), a 5.2-million-euro project involving 11 industry partners. He was the recipient of the 2011 Scientific Merit Award UBIFE/Santander Universities, the 2012 Scientific Award UTL/Santander Totta, the 2016-2017-2018 FEUP Diplomas of Scientific Recognition, the 2017 Best INESC-ID Researcher Award, and the 2018 Scientific Award ULisboa/Santander Universities, in addition to an Honorable Mention in the 2017 Scientific Award ULisboa/Santander Universities. He is a Top Scientist in the Guide2Research Ranking, which lists only scientists having h-index equal or greater than 40. He has won four Best Paper Awards at IEEE four Conferences and one Best Ph.D. Poster Award.