Confidence-Aware Graph Neural Networks for Learning Reliability Assessment Commitments

Seonho Park, Wenbo Chen, Dahye Han, Mathieu Tanneau, and Pascal Van Hentenryck

Abstract—Reliability Assessment Commitment (RAC) Optimization is increasingly important in grid operations due to larger shares of renewable generations in the generation mix and increased prediction errors. Independent System Operators (ISOs) also aim at using finer time granularities, longer time horizons, and possibly stochastic formulations for additional economic and reliability benefits. The goal of this article is to address the computational challenges arising in extending the scope of RAC formulations. It presents RACLEARN that 1) uses a Graph Neural Network (GNN) based architecture to predict generator commitments and active line constraints, 2) associates a confidence value to each commitment prediction, 3) selects a subset of the high-confidence predictions, which are 4) repaired for feasibility, and 5) seeds a state-of-the-art optimization algorithm with feasible predictions and active constraints. Experimental results on exact RAC formulations used by the Midcontinent Independent System Operator (MISO) and an actual transmission network (8965 transmission lines, 6708 buses, 1890 generators, and 6262 load units) show that the RACLEARN framework can speed up RAC optimization by factors ranging from 2 to 4 with negligible loss in solution quality.

Index Terms—Graph neural network, optimization, reliability assessment commitments, security constrained unit commitment, uncertainty quantification.

I. INTRODUCTION

A. Background and Motivation

The Security-Constrained Unit Commitment (SCUC) problem is a critical tool to ensure reliable and economical scheduling of energy production. Its reliability is however being challenged by the rapid transition from fossil-fueled generators to renewable energy sources and the policies supporting/accelerating this migration. California, for instance, aims at having 100% of the state’s electricity procured and served using renewable energy by 2045 [1]. Integrating more renewable energy sources into the grid creates significant volatility, both in front and behind the meter, and increased forecasting errors in net load (load minus renewable generation). For these reasons, Independent System Operators (ISOs) use various Reliability Assessment Commitment (RAC) models to secure sufficient resources in the day-ahead (DA) and real-time markets. For instance, MISO, the Midcontinent Independent System Operator, uses both Day-Ahead Forward Reliability Assessment Commitment (DA-FRAC) and Look-Ahead Commitment (LAC) for reliability assessment. The DA-FRAC is executed after the DA market, replaces demand bids by load forecasts, and decides additional commitments of the generators abiding by the DA market decisions. The LAC is executed every 15 minutes on a rolling basis and provides an additional chance to turn on generators for a 3-hour look-ahead window. Both DA-FRAC and LAC exploit updated forecasts for load and renewable generation.

ISOs have been striving to curtail the execution times of the SCUC, FRAC, and LAC models in order to obtain additional safety and economic gains. This includes the opportunity to have finer time granularity and longer horizons in the model formulations and to explore scenario-based approaches. For example, MISO has decreased the DA market-clearing time from 4 hours to 3 hours in 2016 [6] and is committed to decreasing it further.

The RAC formulations are Mixed Integer Linear Programming (MILP) that are computationally challenging. They are operated by MISO to deal with significant changes in load and renewable energy forecasts. A RAC optimization helps operators determine which additional generators must be committed to taking into account the uncertainty in forecasting. Even though the number of variables is smaller than in the SCUC (because must-run commitments have already been determined by the day-ahead or prior RAC commitments), the RAC formulations must give operators reliable solutions to different scenarios as fast as possible, which makes a machine-learning (ML) approach particularly appealing. Fortunately, the FRAC and RAC are often run on similar instances. Indeed, the pool of generators and the topology of the network evolve slowly, while the demand and renewable generation lead to instances that exhibit strong correlations and similar patterns. This presents a significant opportunity for ML techniques that can then be used to speed up the solution process of RAC formulations.

This is precisely the goal of this article, which proposes the RACLEARN framework for accelerating the RAC optimizations. RACLEARN consists of three key components: 1) a Graph Neural Network (GNN) based architecture that predicts both the generator commitments and the active transmission constraints;
2) the use of an epistemic uncertainty measurement to associate a confidence value with each commitment prediction; and 3) a feasibility restoration that transforms possibly infeasible instances with fixed commitment variables into feasible ones in polynomial time. Note that the proposed RACLEARN framework consolidates all those components using end-to-end learning for accelerating the RAC optimization.

The contributions of the article can be summarized as follows:

1) The RACLEARN framework: The article presents an end-to-end optimization learning approach to obtain near-optimal solutions to industrial RAC formulations and reduce solving times by factors from 2 to 4 on an actual grid.

2) The GNN-based architecture: RACLEARN includes a novel GNN-based architecture to predict generator commitments and active transmission constraints simultaneously. The GNN-based architecture includes novel encoder-decoder layers leading to a compact design with a small number of trainable parameters, which is critical to scale to realistic networks. Experimental results show that the proposed architecture provides significant improvements compared to deep and shallow baselines.

3) The confidence measurement: RACLEARN uses confidence measurements to select a subset of high-confidence commitment predictions. Experimental results show that the confidence values are strongly indicative of being accurate commitment estimates.

4) The polynomial-time feasibility restoration step: RACLEARN includes a polynomial-time feasibility restoration optimization that transforms possibly infeasible instances into feasible instances that will lead to near-optimal solutions.

The rest of the article is organized as follows. Section II reviews prior related works. Section III revisits RAC formulations and their solution techniques. Section IV gives an overview of the RACLEARN framework. Section V presents the proposed GNN-based architecture. Section VI describes how to compute the confidence measurements with the GNN-based architecture. Section VII introduces the PREDICT-REPAIR-OPTIMIZE heuristic for accelerating RAC. Section VIII reports the experimental settings and results. Section IX contains the concluding remarks.

II. RELATED WORK

Solving SCUC and RAC formulation is computationally challenging and alternative MIP formulations of SCUCs have been developed to improve efficiency; they include tightening the constraint polytope (e.g., [7], [8], [9]), constraints elimination (e.g., [6], [10]) and decomposition method (e.g., [11], [12]).

The RAC formulations, which include the DA-FRAC and LAC optimization, have been developed and operated by MISO [2], [3], [4], [5]. However, to the best of the authors’ knowledge, ML schemes to accelerate the RAC optimization process have not yet been studied. Instead, in recent years, ML techniques have been proposed to approximate market-clearing algorithms and/or to accelerate their underlying optimization processes in general [13], [14], [15], [16].

For speeding up the SCUC, Xavier et al. [13] used Support Vector Machine (SVM) and k-nearest neighbor by approximating active transmission constraints and fixing some reliable variables. Recently, Ramesh and Li [14] proposed an ML-aided reduced MILP for SCUC problems; they measured the Softmax-based scores of the ML outputs to fix the commitment variables and restore the feasibility of the commitment outputs of the ML model in the post-processing of the ML inference. This article tackles a similar problem for RACs, but proposes a unified framework that is trained in an end-to-end manner to jointly predict generator commitments and active transmission constraints, and estimate the confidence of the commitment predictions.

In the context of DC-Optimal Power Flow (DC-OPF), deep learning has been shown to approximate optimal solutions successfully, given a variety of input configurations (e.g., [17], [18]). Optimal solutions of the non-convex AC-OPF have also been successfully approximated by deep learning on simpler test cases (e.g., [19], [20], [21], [22], [23]). The combination of deep learning and Lagrangian duality reducing constraint violations, together with a feasibility restoration process using load flows, was shown to find high-quality solutions to real-world AC-OPF test cases [24], [25], [26]. A DNN architecture exploiting the patterns appearing in real-world market-clearing solutions was shown to approximate the solution of large-scale Security-Constrained Economic Dispatch (SCED) problems effectively [27]. Feasibility restoration, which is imperative in this context, can be obtained by warm-starting optimization solvers with deep-learning approximations, (e.g., [28], [29], [30]) or finding active constraints in SCUC (e.g., [13]), DC-OPF (e.g., [31]), and AC-OPF (e.g., [32], [33]).

Deep learning has also been exploited to find better heuristics. Those approaches include approximating Newton’s method for solving AC-OPFs (e.g., [34]) and estimating parameters in the ADMM algorithm via a reinforcement learning scheme (e.g., [35]). These attempts usually have been framed in more general optimization settings beyond the power-system domain. Those include reinforcement learning for general integer programming (e.g., [36]), combinatorial optimization (e.g., [37]), and MIP (e.g., [38]). GNN, a DNN architecture for graph-structured data, has been utilized especially in AC-OPF [21], [30], [39].

After Softmax-based entropy to measure confidence was first introduced in [40], confidence estimation using epistemic uncertainty then became one of the active topics in the deep learning research community. It can be achieved by estimating data density probability (e.g., [41], [42], [43]) or uncertainty quantification (e.g., [44], [45], [46], [47]). Unfortunately, epistemic uncertainty-based confidence estimation on the ML-based output has not been thoroughly studied in power system applications.

III. RAC FORMULATION AND SOLUTION TECHNIQUES

This article focuses on the RAC problem, a variant of SCUC defined and operated by MISO. The detailed formulation is fully stated in [5]. The problem can be modeled as an MILP problem.
of the form

$$\begin{align*}
\text{(RAC)} \quad & \min_{x,y} \quad c^T x + q^T y \\
\text{s.t.} \quad & x \in X, \\
& Ax + By \geq b, \\
& H y \geq g,
\end{align*}$$

(1a)

(1b)

(1c)

(1d)

where $x$ denotes the binary commitment/startup/shutdown variables, and $y$ denotes the continuous variables such as energy and reserve dispatches, branch flows, etc. Constraints (1b) denote generator-level commitment-related constraints, such as minimum run/down times and must-run commitments, and binary requirements on $x$ variables. Constraints (1c) denote generator-level dispatch-related constraints such as minimum/maximum limits and ramping constraints. Finally, constraints (1d) denote system-wide reliability constraints, namely, power balance, minimum reserve requirements, and transmission constraints. Note that, in MISO’s formulation, the reliability constraints (1d) are treated as soft: they can be violated, but doing so incurs a (large) penalty.

Indeed, ML-based methodologies for RAC acceleration can be directly applied to SCUC with minor adjustments. However, our attention remains solely on RAC. While RAC and SCUC share similarities in their formulations, RAC requires greater acceleration due to the need to effectively manage significant variations in forecasts in real-time.

Despite the significant progress in MILP solver technology and in constructing tight MILP formulations for SCUC (see, e.g., [48]), industry-size instances remain challenging [49]. In order to accelerate the process, transmission constraints are typically handled in a lazy fashion. Accordingly, the RACLearn framework uses the iterative optimization procedure suggested in [10] as the baseline optimization algorithm. The algorithm first solves the relaxed problem with no transmission constraint and then identifies the violations of these constraints. The $k$ most violated transmission constraints are added into the relaxed problem, and the process is repeated until there is no constraint violation. Note that identifying the active set of transmission constraints beforehand may allow to solve the problem in a single iteration ideally. The RACLearn framework includes ML techniques to predict this active set and prior work indicates that such prediction can improve run times [13].

IV. THE RACLEARN FRAMEWORK

The RACLearn framework can be summarized as follows: 1) Use the GNN-based architecture to predict the generator commitments and active transmission constraints; 2) Use epistemic uncertainty to obtain a confidence value for each commitment prediction, and fix a subset of high confident commitment estimates; 3) Transform a RAC instance with fixed high-confidence commitment variables into the feasible instance using the feasibility restoration step; and, 4) Fix the feasible commitment predictions and seed the active transmission constraints in RAC optimization.

The next sections describe these steps in detail.

V. THE GNN-BASED ARCHITECTURES

The first component of the RACLearn framework is a novel GNN-based encoder-decoder architecture depicted in Fig. 1. Let $B$ be the set of bus indices, $G$ the set of generator indices, $L$ the set of transmission line indices, $D$ the set of load demand indices, and $T$ the set of time periods (commitment intervals). The ML model approximating the RAC aims at predicting commitment decisions and active transmission constraints from the problem inputs. The problem inputs include the generator and load configurations $g_R$ and $l_T$, detailed in Table I. It is assumed that the network topology is fixed and the difference between the instances can be described by $g_R$. The forecasting of load demand and renewable generation.

The ML model can be viewed as a parametric function $F_\theta$, with trainable parameters $\theta$, that estimates the binary commitment decisions $\tilde{x}_g$ and the active transmission constraints $\tilde{a}_l$ as follows:

$$\begin{align*}
\{\tilde{x}_g\}_{g \in G}, \{\tilde{a}_l\}_{l \in L} = F_\theta (\{g_R\}_{g \in G}, \{l_T\}_{l \in L}).
\end{align*}$$

To account for the high-dimensional and graph-structured nature of the input data $\{g_R\}_{g \in G}, \{l_T\}_{l \in L}$, this article proposes a GNN-based encoder-decoder architecture. The proposed architecture consists of three parts:

1) The encoders encode the input configurations of generators and loads into the latent vectors, which are then aggregated at the bus level.

2) The core GNN layers evolve the embedded vectors at the buses, and eventually, generate bus representations.
3) The decoders outputs the optimal commitment estimates and active transmission constraint estimates.

The rest of this section discusses this architecture in detail.

A. Encoders

a) Generator Encoder: For each generator, the input feature $g$ changes from instance to instance. This input feature $g$ includes the maximum capacity, pricing curve parameters, and so on. This input feature $g$ is flattened to be a one-dimensional vector and embedded into a higher-dimensional latent vector $\tilde{g}$ through the generator encoder $Enc_g$, i.e., $\tilde{g} = Enc_g(g)$.

b) Load Encoder: Similarly, an input feature for each load unit $l$ (i.e., the load forecast) is embedded into a latent vector $\tilde{l}$ using the load encoder $Enc_l$, i.e., $\tilde{l} = Enc_l(l)$.

The generator and load encoders are Multilayer Perceptron (MLP)-based structures, comprised of fully-connected layers followed by ReLU activations. The generator and load encoder structures are formulated as

\[
\begin{align*}
    h^{(l)} &= x, \\
    \tilde{h}^{(l)} &= h^{(l-1)}W^{(l)} + b^{(l)}, \\
    h^{(l)} &= \max(\tilde{h}^{(l)}, 0),
\end{align*}
\]

where $x$ denotes the generator or load input features. Also, at each layer $l$, $W^{(l)}$ is a (trainable) weight matrix, and $b^{(l)}$ is a (trainable) bias vector. Note that the trainable parameters involved in the generator and load encoders are shared across all the generators and load units, yielding an efficient structure.

c) Generating Bus Feature: Vectors $g$ and $l$ are then $l_2$-normalized, and the bus feature $f_b \in \mathbb{R}^d$ at bus $b$ is defined by concatenating the sums of the generator and load features into

\[
f_b = \left[ \sum_{g \in G_b} \tilde{g}_g \mid \sum_{d \in D_b} \tilde{l}_d \right],
\]  

(3)

where $[\cdot|$ denotes vector concatenation, and $G_b$ and $D_b$ are the sets of the generator and load indices attached to bus $b$, respectively.

B. Graph Neural Networks

Once the bus features for every bus are prepared, the whole bus features are denoted as a matrix form of $X \in \mathbb{R}^{B \times d}$ where each row corresponds to the bus feature $f$ and $B$ represents the number of buses in the system.

In this article, to embed the output bus features $Y$ from the input bus feature $X$, two types of GNN layer are explored: GCN [50] and SIGN [51].

a) Graph Convolutional Network (GCN): The power system is viewed as an unweighted undirected graph $\mathcal{N} = (B, \mathcal{L})$. The symmetric adjacency matrix of this graph is denoted by $A \in \mathbb{R}^{B \times B}$, i.e., $A_{ij} = A_{ji} = 1$ if there is a transmission line between bus $i$ and bus $j$. The self-loop added adjacency matrix $\tilde{A}$ is defined as $\tilde{A} = A + I$.

A GCN layer typically utilizes the normalized adjacency matrix $\tilde{A}$ defined as: $\tilde{A} = D^{-1/2}AD^{-1/2}$, where $D$ is the degree matrix, i.e., $D = \text{diag}(\sum_{j=1}^{B} \tilde{A}_{1j}, \ldots, \sum_{j=1}^{B} \tilde{A}_{Bj})$. The GCN propagates the information to the neighbor nodes along the edge using graph diffusion operations. Given the matrix form of the input bus features $X \in \mathbb{R}^{B \times d}$, the $d'$-dimensional output bus features $Y \in \mathbb{R}^{B \times d'}$ is obtained as

\[
Y = \text{GCN} \left( X | \tilde{A} \right) = \sigma \left( \tilde{A}X\Theta \right),
\]

where $\sigma(\cdot)$ represents the ReLU activation, and $\Theta \in \mathbb{R}^{d \times d'}$ is a matrix of trainable parameters.

The GCN layer only propagates messages to neighboring nodes, i.e., 1-hop neighbors. To propagate the message throughout the network, a multiple GCN layers stacked structure can be utilized. Using bus embedding $H^{(l-1)} \in \mathbb{R}^{B \times d^{(l-1)}}$ as the input of $l$th GCN layer, the output is given by

\[
H^{(l)} = \text{GCN}_l \left( H^{(l-1)} | \tilde{A} \right) = \sigma \left( \tilde{A}H^{(l-1)}\Theta^{(l)} \right),
\]
where \( \Theta^{(l)} \in \mathbb{R}^{d^{(l-1)} \times d^{(l)}} \) is trainable parameters of the layer \( l \) and \( \mathbf{H}^{(0)} = \mathbf{X} \). Then, the output of the final GCN layer \( \mathbf{H}^{(L)} \) of the \( L \)-stacked GCN structure is regarded as the output bus feature \( \mathbf{Y} \). It is still an open question whether a multiple-stacked GNN structure gives an enhanced performance in general [51]. Thus, recent research has utilized different types of diffusion operators to replace the normalized adjacency matrix and allow messages to propagate beyond 1-hop neighbors [52], [53]. For instance, S-GCN [52] employs the \( \rho \)th power of the normalized adjacency matrix to encompass \( p \)-hop neighbors in its propagation, i.e., S-GCN is defined as

\[
\mathbf{Y} = \sigma \left( \mathbf{A}^\rho \mathbf{X} \Theta \right).
\]

b) Scalable Inception Graph Neural Network (SIGN): Another type of GNN layer, SIGN [51], uses Inception-like architecture [54] on graph-structured data, and generalizes the use of various diffusion operators in the GCN structure. In particular, SIGN concatenates multiple output bus features from various diffusion operators. In addition to the use of the adjacency matrix that only covers the spatial domain, the SIGN architecture considered in this article uses the personalized PageRank (PPR) based spectral diffusion operator \( \mathbf{S} \) [53] and its powers. The SIGN layer can be formalized as

\[
\mathbf{Y} = \sigma \left( \text{MLP} \left( \left\{ \xi \left( \| \mathbf{g} \|_2 \mathbf{X} \mathbf{A}^i \right), \sum_{i=1}^{s} \text{GCN}(\mathbf{X} | \mathbf{S}^i) \right\} \right) \right),
\]

where \( \xi \) and \( \sigma \) are nonlinear activation functions. In practice, PReLU and ReLU have been used for \( \xi \) and \( \sigma \), respectively. Also, MLP denotes fully connected layers followed by ReLU activations, \( p \) and \( s \) are hyperparameters representing the maximum powers used to construct representational features in parallel for the normalized adjacency matrix and the PPR diffusion operator, respectively.

C. Decoders

Denote the output bus feature of bus \( j \) from the GNN as \( \mathbf{f}_j \), which corresponds to the \( j \)th row of \( \mathbf{Y} \). It remains to estimate the optimal commitment decisions and active transmission constraints.

a) Commitment Decoder: Since multiple generators can be located at the same bus, another generator encoder (independent and distinct from the previous generator encoder) is used to generate another embedded vector \( \mathbf{g}_i \). This vector is concatenated to the bus features \( \mathbf{f}_j \) (with \( i \in \mathcal{G}_j \)) to obtain a generator-level representation \( \tilde{\mathbf{g}}_i \), i.e., \( \tilde{\mathbf{g}}_i = [\mathbf{f}_j | | \mathbf{g}_i] \). Then, the MLP-based commitment decoder \( \text{Dec}_g \) uses \( \tilde{\mathbf{g}}_i \) as an input and outputs a \( |\mathcal{T}| \)-dimensional vector \( \tilde{x}_i \) that represents the commitment decision likelihood for the generator \( i \), i.e., \( \tilde{x}_i = \text{Dec}_g (\tilde{\mathbf{g}}_i) \). The \( t \)th element of \( \tilde{x}_i \) represents the probability of committing generator \( i \) at time step \( t \in \mathcal{T} \). This value is the output of a sigmoid function attached to the final layer of the decoder \( \text{Dec}_g \).

b) Transmission Decoder: The vector \( \mathbf{e}_l \) of transmission line \( l \), linking buses \( i \) and \( j \), can be represented by the permutation-invariant element-wise summation of the associated output bus-representations, i.e., \( \mathbf{e}_l = \tilde{\mathbf{f}}_i + \tilde{\mathbf{f}}_j \). The MLP-based transmission decoder \( \text{Dec}_e \) uses vector \( \mathbf{e}_l \) to estimate whether the transmission constraints are tight or not, i.e., \( \tilde{a}_l = \text{Dec}_e (\mathbf{e}_l) \) for all transmissions \( l \in \mathcal{L} \). Note that \( \tilde{a} \) is a \( 2|\mathcal{T}| \) dimensional vector where 2 comes from the lower and upper limits of each transmission constraint. Like the commitment decoders, \( \tilde{a}_l \) represents the sigmoid-based binary probabilities that the transmission constraints of transmission \( l \) are active.

VI. CONFIDENCE MEASUREMENT

The second component of the RACLEARN framework is the confidence measurement of the prediction. Unlike other critical applications such as semantic segmentation for self-driving AI [55] or medical image diagnosis [56], ML approaches in power systems have not been extensively studied through the lens of uncertainty quantification. The RACLEARN framework remedies this limitation and provides an epistemic uncertainty-based confidence measure for each output of the ML model. Specifically, to quantify the confidence in each commitment prediction, RACLEARN exploits MC-dropout [46], one of the popular tools to measure epistemic uncertainty (also called model uncertainty) by approximating a Bayesian neural network [45]. A dropout layer, at training time, chooses nodes at random, and sets the trainable parameters associated with those nodes to zero in order to prevent overfitting during training. The dropout layer is also used at inference to express the stochasticity involved in the input data. That is, instead of providing a single prediction, MC-dropout generates \( T \) of them. The mean \( \mu \) of these \( T \) predictions is then used as the output prediction, while the variance \( \sigma \) serves as a proxy for the epistemic uncertainty, i.e.,

\[
\mu(\tilde{x}) = \frac{1}{T} \sum_{t=1}^{T} \tilde{x}^{(t)},
\]

\[
\sigma(\tilde{x}) = \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} (\tilde{x}^{(t)} - \mu(\tilde{x}))},
\]

When the model is to estimate on unseen or unfamiliar input instances, \( \sigma \) tends to be higher, representing the higher epistemic uncertainty involved in the prediction from the model. Thus, the confidence measure, which has the reciprocal relationship with the epistemic uncertainty, can be defined as

\[
\text{Conf}(\tilde{x}) := \frac{1}{\sigma(\tilde{x})}.
\]

RACLEARN generates confidence values for the commitment of each generator at each time step. The implementation attaches a dropout layer with a dropout probability of 0.5 to the penultimate layer of the commitment decoder.

VII. THE PREDICT-REPAIR-OPTIMIZE HEURISTIC

The last component of the RACLEARN framework is the feasibility restoration. It is motivated by a key challenge faced in solving RAC (or SCUC in general): to find high-quality feasible solutions quickly [6], [49]. RACLEARN addresses this challenge with PREDICT-REPAIR-OPTIMIZE heuristic involving the feasibility restoration step, which is guaranteed to return a feasible solution. The PREDICT-REPAIR-OPTIMIZE heuristic
consists of four components; 1) estimating the active transmission constraints and add at the beginning of the iterative optimization procedure, 2) fixing generator commitments with high confidence, 3) executing the feasibility restoration step to repair the fixed commitments, and 4) solving the reduced RAC problem w.r.t. the remaining continuous variables.

A. Estimating Active Transmission Constraints

The first step in the PREDICT-REPAIR-OPTIMIZE heuristic is to estimate the active transmission constraints. Fig. 2 highlights that few transmission constraints are active in an optimal RAC solution. This justifies the use of the iterative optimization procedure [10] that first solves the relaxed problem where there is no transmission constraints and then adds violated transmission constraints lazily until there is no violation. To accelerate the optimization procedure, akin to the approach suggested in [13], it is desirable to estimate the active likely transmission constraints and add those constraints at the beginning of the iterative procedure in order to decrease the number of iterations. Again, the transmission decoder generates a sigmoid-based binary probability as its output to estimate the activeness of a transmission constraint.

B. Fixing Generator Commitments

The second step of the PREDICT-REPAIR-OPTIMIZE heuristic fixes the binary variables corresponding to confident commitment decisions. The confidence level needed to fix the commitment decisions is a hyper-parameter that is studied in Section VIII.

C. Feasibility Restoration

Fixing confident commitments may result in an infeasible RAC problem instance. Specifically, the prediction $\hat{x}$ may conflict with the commitment-level constraints (1b) in the RAC formulation. To address this issue, the RACLEARN framework repairs the prediction $\hat{x}$ to obtain commitment decisions $\tilde{x}$ that satisfy (1b). Then, the commitment decisions are fixed to $\tilde{x}$, and the RAC is re-optimized, yielding $\tilde{y}$ such that $(\tilde{x}, \tilde{y})$ is feasible for Problem (1). The rest of this section describes the repair step and proves that it runs in polynomial times.

Let $\hat{x}$ be the optimal commitment estimates to a RAC instance. This prediction does not guarantee that $\hat{x}$ will satisfy the combinatorial constraints (1b). To alleviate this issue, the article introduces a repair step which, akin to a projection step, seeks a feasible $x \in \mathcal{X}$ that is close to $\hat{x}$. Formally, the repair step solves

$$\tilde{x} \in \arg \min_x \mathcal{H}(x, \hat{x})$$

subject to $x \in \mathcal{X}$,

where $\mathcal{H}(x, \hat{x})$ is the Hamming distance between $x$ and $\hat{x}$.

Theorem 1: Problem (5) can be solved in polynomial time.

Proof: First, because commitment constraints are formulated at the generator level, Problem (5) can be decomposed into $G$ single-generator problems, where $G$ is the number of generators. Second, because $x, \hat{x}$ are binary vectors, objective (5a) is linear. Thus, Problem (5) reduces to optimizing a linear objective over the commitment constraints for a single generator. This is solvable in polynomial time using the dynamic-programming algorithm of Frangioni and Gentile [57].

Theorem 2: The PREDICT-REPAIR-OPTIMIZE heuristic returns a feasible solution $(\tilde{x}, \tilde{y})$ in polynomial time.

Proof: By Theorem 1, a feasible $\tilde{x}$ can be obtained in polynomial. Then, recall that constraints (1d) are soft. In addition, the structure of constraints (1c) is such that, for any $x$ that satisfies (1b), there exists a vector $y$ that satisfies (1c). For instance, it suffices to set the reserve dispatch of each generator to zero, and its energy dispatch to its minimum output at every time period. Finally, once the binary variables $x$ are fixed, Problem (1) reduces to a linear program, which can be solved in polynomial time to obtain $\tilde{y}$. 

VIII. EXPERIMENTAL RESULTS

A. Experimental Settings

1) The Case Studies: Experiments were conducted on the RTE system in France. The system contains 8965 transmission lines and 6708 buses where 1890 generators and 6262 load units are located. Solar/wind power generation and load demand forecast were obtained through LSTM-based models [58] trained on the publicly-available ground truth obtained from RTE [59]. The stochastic forecast scenarios are generated via MC-dropout [46]. Supply offer bids were synthetically generated by the approach proposed in [60].

Two distinct types of RAC problems are studied: DA-FRAC and LAC. For gathering DA-FRAC instances, the DA market (DA-SCUC) is first executed. The must-run commitments for the DA-FRAC are then extracted from the DA-SCUC solution. For gathering LAC instances, the DA-FRAC is first executed and the must-run commitments are extracted from its solution. A LAC instance is solved every 15 minutes and it uses all the commitment decisions of the DA-FRAC and prior LAC executions as the must-run commitments. Fig. 3 shows that the majority of the commitment decisions are already determined by prior
optimization decisions. Ratios of the must-run commitments are above 80% and 85% for DA-FRAC and LAC, respectively.

2) Data Generation: The formulations of the DA-FRAC, LAC, and DA-SCUC were modeled with JuMP [61] and solved through Gurobi 9.5 [62]. To gather the instances, the optimizations were solved on Linux machines in the PACE Phoenix cluster at Georgia Tech [63]. The data instances are gathered by setting the optimality gap to 0.1% and using 2 threads and 8 GB RAM. The time limit was set to 3600 seconds and the parameter MIPFocus was set to 1 for focusing on finding feasible solutions. The remaining parameters use the default values. Only the instances with the optimal solutions that are obtained within the time limit are included in the dataset for stable training. When generating instances, transmission constraints are added in an iterative manner as suggested in [10]. The solution process starts with no transmission constraints. At each iteration, the 15 most violated transmission constraints are added into the formulation sequentially until no violations are observed.

The number of data instances for each month is around 9000 for the DA-FRAC and 8600 for the LAC. The training and testing procedures are performed on a monthly basis, i.e., the training is performed on data instances of a month, and the testing is conducted on the data instances of the following month. Thus, in 2018, 11 independent models for each month (February to December) are trained and tested. Note that this experimental setting inherently features a distribution shift [64], since the data distributions of the training and testing datasets are different. It mimics the natural way of deploying the models in real-world settings, using historical data to infer the upcoming commitment decisions. For each month of training, 100 instances are extracted and used for validation. 1000 testing instances are also sampled from the next month’s instances for measuring and reporting the performance results.

3) Training Configuration: The proposed GNN-based architecture and other neural network baselines were implemented and trained using PyTorch. The proposed architecture is trained in an end-to-end manner by minimizing binary cross-entropy loss with respect to the commitment estimates and active transmission constraint estimates in a supervised way. The minibatch size was set to 8 for training. The AdamW optimizer [65] with a learning rate of $1e^{-4}$ and a weight decay of $1e^{-6}$ was used for updating the trainable parameters. Early stopping was applied with a the patience of 5, while the learning rate decayed by 0.95 when the validation loss was worse than the best validation loss at every epoch. The maximum epoch was set to 20. The training and testing process was performed using Tesla V100 GPU on machines with Intel CPU cores at 2.7 GHz.

B. Commitment Prediction Accuracy

This section reports the performance of the optimal commitment predictions of the proposed GNN-based models as well as various baselines. The RACLEARN framework was tested with several GNN layers: While the encoder and decoder structures remain the same for every architecture, the GNN layers to propagate bus representations are different: they are SIGN, GCN-1 (a single GCN layer), and GCN-5 (5 stacked GCN layers).

1) Baselines: The RACLEARN architectures are compared to two neural network baselines: MLP has three fully-connected layers of 512 hidden nodes and flattens all the input features; Encoders+MLP uses the encoder structures of RACLEARN, flattens the embedded generator/load features, and applies 3 fully-connected layers, followed by ReLU activations, with 512 hidden nodes instead of using the GNN and decoder structures of RACLEARN.

The RACLEARN framework is also compared to the following shallow models taken from scikit-learn [66]. SVM-poly is an SVM model with a polynomial kernel, SVM-rbf is an SVM model with radial basis function-based kernel, and RF is a random-forest model. All those shallow models use the default parameter settings. Those shallow models are not scalable to the case study and hence a naïve autoencoder (AE) was used to decrease the dimensionality of the input feature space. AE has three MLP layers for each encoder and decoder. The encoder embedding vector has a dimensionality of 0.2% of that of the input features. The number of hidden nodes is set to the mean of dimensions of the input and embedding features. This AE model is also trained using the AdamW optimizer with a learning rate of $5e^{-5}$ up to 20 epochs with a minibatch of 16 instances. When training, must-run commitments are masked out to focus on the non-must-run commitments.

2) Accuracy Results: The prediction results for DA-FRAC and LAC are depicted in Tables II and III. The tables report the mean accuracy together with 95% confidence intervals. The
metrics are calculated using bootstrapping with 1000 samples from the testing set with replacement. The performance measures are the averaged accuracy and the area under the receiver operating characteristic curve (AUROC). In all cases, GNN-based models are superior to the swallow and MLP-based models in terms of accuracy and AUROC. The prediction accuracy is higher for the LAC than for the DA-FRAC, which is explained by the smaller variability in forecasts and the shorter time horizon of the LAC.

Fig. 4 depicts the monthly performance of the models (the shallow models are excluded because of their poor performances). It highlights that the winter season (from October to March) is more challenging to predict and has more outliers. The figure also indicates that the GNN-based architectures outperform their MLP-based counterparts by a significant margin.

C. Commitment Prediction Confidence

Among the GNN-based models, the SIGN model was chosen for the rest of the experiments. Fig. 5 shows the average testing accuracy and its corresponding normalized confidence values from (4) of the commitment predictions at each time step for the DA-FRAC and the LAC. Recall that the numbers of time steps for the DA-FRAC and the LAC are 24 and 12. For the LAC, the accuracy of the predictions decreases as the time step increases. This comes from the fact that the LAC is run over a rolling horizon during operations. Earlier steps have more must-run commitments and more accurate forecasts. For the DA-FRAC, each instance predicts the next day, the forecasts for each hour are more consistent, and fewer must-run commitments are imposed. In this case, the accuracy is consistent within the range between 0.95 and 0.99. Interestingly, in both cases, the confidence value is high and in agreement with the accuracy. For instance, in LAC, the average confidence value at step 12 is significantly lower than other steps, which is indicative of less accurate results. This is confirmed by the accuracy results. Note that the confidence value is obtained without relying on the ground truth.

| Model       | Accuracy     | AUROC        |
|-------------|--------------|--------------|
| SVM-poly    | 0.913(0.666,0.949) | 0.916(0.640,0.958) |
| SVM-zbf     | 0.912(0.692,0.951) | 0.883(0.664,0.961) |
| RF          | 0.916(0.733,0.944) | 0.905(0.683,0.961) |
| MLP         | 0.982(0.966,0.991) | 0.965(0.678,0.986) |
| Encoders+MLP| 0.986(0.973,0.991) | 0.980(0.839,0.993) |
| GCN-1       | 0.994(0.990,0.998) | 0.995(0.975,1.000) |
| GCN-5       | 0.994(0.990,0.998) | 0.995(0.973,0.999) |
| SIGN        | 0.994(0.990,0.998) | 0.995(0.976,0.999) |

The bounds of the 95% confidence intervals are presented between parentheses. The best values are in bold.

GNN-based architectures. This can be attributed to the fact that the power grid can be characterized as a small-world network, where most nodes can be reached from every node with a limited number of hops. As such, the generation commitment can be estimated with the localized features; the loads and the generator’s features from neighboring nodes, rather than from distant nodes.
The RACLEARN framework uses commitment predictions to fix binary variables, reducing the dimensionalities of these optimization problems. Moreover, the confidence values associated with the predictions provide valuable information about which predictions must be used in this process in order to balance the quality of the solutions and the dimensionality reduction. Using lower confidence predictions may deteriorate the quality of the solutions, while only using the predictions with the highest confidence may not provide significant computational benefits. Fig. 6 explores this trade-off. The x-axis captures the percentage of highly confident commitment predictions selected to fix variables in the optimization models, which is called the confident subset. The y-axis captures the average accuracy of the selected predictions. The percentages are ranked in decreasing order in Fig. 6: a percentage of 100% means that all the predictions will be used; a percentage of 30% means that the confident subset consists of the 30% of predictions with the highest confidence values. The figure shows how fast the average accuracy increases monotonically as the size of the confident subset decreases for both FRAC and the LAC cases. For the LAC, the accuracy can increase to around 1.000 as soon as the confident subset contains less than 50% of the predictions. Similarly, for the DA-FRAC, the accuracy increases to around 0.998 if low-confidence commitment estimates are rejected, whereas when all commitment estimates are concerned, the accuracy was 0.981. These predictions based on the MC-dropout-based confidence measure are likely to be accurate, leading to an attractive tradeoff between solution quality and dimensionality reduction in the reduced optimization.

Fig. 6 also includes the binary entropy (denoted as Entropy) for comparison. This measure is directly induced by the softmax-based probability [40], which has been utilized in the previous ML-based approaches in power system applications [13], [14]. The figure shows that the proposed confidence measure (based on MC-dropout) performs better than the binary entropy to classify the confident subset. This coincides with the comparison results in [56] in the medical domain.

D. Prediction of Active Transmission Constraints

Actual unit-commitment problems have few active transmission constraints and previous works have shown that they can be predicted with reasonable accuracy by identifying those transmissions [13], [68]. To evaluate the performance of the RACLEARN framework in predicting active constraints, this article adopts baselines that exploit prior runs of the RAC model, i.e., the FRAC of the previous day or the LAC that was run 15 minutes before. In the baseline, all the constraints are considered active if they are included in a transmission line that had at least one active constraint in the prior run. In RACLEARN, transmission constraint predictions whose probability are above 0.1 are regarded as being active.

Table IV presents experimental results showing the benefits of predicting active constraints compared to the baselines. Those predictions are expected to obtain high recall values while having a small number of active constraint predictions. The recall value, i.e., the true positive rate, is the fraction of correctly predicted active constraints among all constraints that are actually active. The recall value is an appropriate measure because the false negatives (the constraints predicted as not active, but active actually) should be added to the formulation subsequently, which leads to excessive time for solving the problem.

On average, there are about 125 active constraints in the DA-FRAC and 71 in the LAC. On average, the baseline isolates 965.24 constraints (DA-FRAC) and 242.46 constraints (LAC) that can be active for recall values of 0.822 (DA-FRAC) and 0.979 (LAC). In contrast, RACLEARN identifies 409.91 (DA-FRAC) and 141.18 (LAC) possible active constraints on average. The recall values are 0.914 and 0.981 for the DA-FRAC and LAC, respectively. Hence, compared to the baseline, RACLEARN provides better recall values (especially for the DA-FRAC) and adds fewer constraints to the optimization model, demonstrating its practical value.

E. Optimization Speedups

The results for the optimization speedups are reported in Table V. The baseline is the iterative optimization procedure [10] with $k = 15$ (adding 15 transmission constraints at each iteration of the overall procedure) as suggested in [13]. According to [69], at MISO, the DA-FRAC and LAC have to be solved within 40 and 10 minutes. The results in elapsed times of the baseline for DA-FRAC and LAC are 126.06 and 42.71 seconds. These times are higher than those obtained for the MISO system because the RTE system is significantly larger and more congested in general. For RACLEARN, Table V reports the results when the confident subset contains 0%, 30%, 50%, 70%, 90%, and 100% of the predictions. For the LAC, the results...
This article proposed the RACLearn framework to speed up the solving of RAC problems and, in particular, the DA-FRAC and LAC. RACLearn consists of three key components: 1) a dedicated GNN-based architecture to predict generator commitments and active transmission lines; 2) the use of the GNN-based architecture and MC-dropout to associate a confidence value to each commitment prediction; and 3) a feasibility-restoration procedure that transforms the prediction into a high-quality feasible solution in polynomial time. Experimental results on the French transmission system and the DA-FRAC and LAC formulations used by MISO indicate that RACLearn can provide significant speedups in solving for negligible loss in solution quality. The RACLearn framework sheds light on the use of confidence measurement and uncertainty quantification in ML approaches to power systems. Future research will explore how to further improve the confidence calibration and the architectural design of the GNNs. The RACLearn framework can be generalized to the stochastic optimization versions of the FRAC and the LAC. Additionally, this framework can be further enhanced to address network topology changes, thereby extending its applicability to accelerate optimal transmission planning or generalize transmission contingencies.

REFERENCES

[1] CEC, “New data indicates California remains ahead of clean electricity goals,” 2022. [Online]. Available: https://www.energy.ca.gov/news/2022-02/new-data-indicates-california-remains-ahead-clean-electricity-goals
[2] M. Massi, H. Song, M. Hong, J. Wan, Y. Chen, and E. Zak, “The security-constrained commitment and dispatch for midwest ISO day-ahead co-optimized energy and ancillary service market,” in Proc. IEEE Power Energy Soc. Gen. Meeting, 2009, pp. 1–8.
[3] Y. Chen, V. Ganugula, J. Williams, J. Wan, and Y. Xiao, “Resource transition model under MISO MIP based look ahead commitment,” in Proc. IEEE Power Energy Soc. Gen. Meeting, 2012, pp. 1–8.
[4] Y. Chen, Q. Wang, X. Wang, and Y. Guan, “Applying robust optimization to MISO look-ahead commitment,” in Proc. IEEE PES Gen. Meeting Conf. Expo., 2014, pp. 1–5.
[5] MISO, “Reliability assessment commitment software formulations and business logic,” Oct-13-2021, business Practices Manual, Energy and Operating Reserve Markets Attachment C., BPM-002-222, [Online]. Available: https://els.psc.mo.gov/mips/commoncomponents/viewdocument.aspx?DocId=936255868.
[6] 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.
[7] K. Pan and Y. Guan, “A polyhedral study of the integrated minimum-up/down time and ramping polytope,” 2016, arXiv:1604.02184.
[8] P. Danesi-Kurt, S. Kiczukayvuz, D. Rajan, and A. Atamtürk, “A polyhedral study of production ramping,” Math. Program., vol. 158, no. 1, pp. 175–205, 2016.
[9] J. Ostrowski, M. F. Anjos, and A. Vannelli, “ Tight mixed integer linear programming formulations for the unit commitment problem,” IEEE Trans. Power Syst., vol. 27, no. 1, pp. 39–46, Feb. 2012.
[10] A. S. Xavier, F. Qiu, F. Wang, and P. R. Thirumurugan, “Transmission constraint filtering in large-scale security-constrained unit commitment,” IEEE Trans. Power Syst., vol. 34, no. 3, pp. 2457–2460, May 2019.
[11] K. Kim, A. Botterud, and F. Qiu, “Temporal decomposition for improved unit commitment in power system production cost modeling,” IEEE Trans. Power Syst., vol. 33, no. 5, pp. 5276–5287, Sep. 2018.
[12] M. J. Feizollahi, M. Costley, S. Ahmed, and S. Grigjulav, “Large-scale decentralized unit commitment,” Int. J. Elect. Power Energy Syst., vol. 73, pp. 97–106, 2015.
[13] A. S. Xavier, F. Qiu, and S. Ahmed, “Learning to solve large-scale security-constrained unit commitment problems,” INFORMS J. Comput., vol. 33, no. 2, pp. 739–756, 2021.
[14] A. V. Ramesh and X. Li, “Feasibility layer aided machine learning approach for proshade operations,” IEEE Trans. Power Syst., Apr. 2023.

[15] X. Tang, X. Bai, Z. Weng, and R. Wang, “Graph convolutional network-based security-constrained unit commitment leveraging power grid topology,” IEEE Trans. Power Syst., vol. 9, no. 4, pp. 3618–3628, Jul. 2021.

[16] M. I. A. Shekew and B. Venkatesh, “Learning-assisted variables reduction method for large-scale MILP unit commitment,” IEEE Open Access J. Power Energy, vol. 10, pp. 245–258, 2023.

[17] A. Velloso and P. Van Hentenryck, “Combining deep learning and optimization for preventive security-constrained DC optimal power flow,” IEEE Trans. Power Syst., vol. 36, no. 4, pp. 3544–3552, 2023.

[18] X. Pan, T. Zhao, M. Chen, and S. Zhang, “DeepOPF: A deep neural network approach for security-constrained DC optimal power flow,” IEEE Trans. Power Syst., vol. 36, no. 3, pp. 1725–1735, May 2021.

[19] X. Pan, T. Zhao, and M. Chen, “DeepOPF: Deep neural network for DC optimal power flow,” in Proc. IEEE Int. Conf. Control. Comput. Technol. Smart Grids, 2019, pp. 1–6.

[20] A. S. Zamzam and K. Baker, “Learning optimal solutions for extremely fast AC optimal power flow,” in Proc. IEEE Int. Conf. Control. Comput. Technol. Smart Grids, 2020, pp. 790–792.

[21] B. Donon, Z. Liu, W. Liu, I. Guyon, A. Marot, and M. Schoenauer, “Deep statistical solvers,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 710–721.

[22] F. Cengil, H. Nagarajan, R. Bent, S. Eksioglu, and B. Eksioglu, “Learning to accelerate globally optimal solutions to the AC optimal power flow problem,” Electric Power Syst. Res., vol. 212, 2022, Art. no. 108275.

[23] S. Park and P. Van Hentenryck, “Self-supervised primal-dual learning for constrained optimization,” Proc. 57th AAAI Conf. Artif. Intell., vol. 37, no. 4, pp. 4052–4060, Feb. 2023.

[24] F. Fioretto, T. W. Mak, and P. Van Hentenryck, “Predicting AC optimal power flows: Combining deep Learning and lagrangian dual methods,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 630–637.

[25] M. Chatzos, F. Fioretto, T. W. Mak, and P. Van Hentenryck, “High-fidelity machine learning approximations of large-scale optimal power flow,” 2020, arXiv:2006.16356.

[26] M. Chatzos, T. W. Mak, and P. Vanhentenryck, “Spatial network decomposition for fast and scalable AC-OPF learning,” IEEE Trans. Power Syst., vol. 37, no. 4, pp. 2601–2612, Jul. 2022.

[27] W. Chen, S. Park, M. Tanneau, and P. Van Hentenryck, “Learning optimization proxies for large-scale security-constrained economic dispatch,” Electric Power Syst. Res., vol. 213, 2022, Art. no. 108566.

[28] K. Baker, “Learning warm-start points for AC optimal power flow,” in Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process., 2019, pp. 1–6.

[29] L. Chen and J. E. Tate, “Hot-starting the AC power flow with computational neural networks,” 2020, arXiv:2004.03942.

[30] F. Diehl, “Warm-starting AC optimal power flow with graph neural networks,” in Proc. 37th AAAI Conf. Artif. Intell., 2023, pp. 1–6.

[31] D. Kota and S. Misra, “Learning for DC-OPF: Classifying active sets normally reduced formulation of OPF through meta-optimization,” 2019, arXiv:2004.00950.

[32] K. Baker, “A learning-based quasi-newton method for AC optimal power flow,” 2020, arXiv:2007.06074.

[33] S. Zeng, A. Kody, Y. Kim, K. Kim, and D. K. Molzahn, “A reinforcement learning approach to parameter selection for distributed optimal power flow,” IEEE Trans. Power Syst., vol. 32, no. 4, pp. 7910–7921.

[34] Y. Tang, S. Agrawal, and Y. Faenza, “Reinforcement learning for integer problem,” in Proc. IEEE 57th Annu. Allerton Conf. Commun. Control Comput., 2019, pp. 1050–1059.

[35] S. Park, G. Adosoglu, and P. M. Pardalos, “Interpreting rate-distortion of variational autoencoder and using model uncertainty for anomaly detection,” Ann. Math. Artif. Intell., vol. 90, pp. 735–752, 2021.

[36] B. Knueven, J. Ostrowski, and J.-P. Watson, “On mixed-integer programming formulations for the unit commitment problem,” INFORMS J. Comput., vol. 32, no. 4, pp. 857–876, 2020.

[37] Y. Chen et al., “Security-constrained unit commitment for electricity market: Modeling, solution methods, and future challenges,” IEEE Trans. Power Syst., early access, Oct. 10, 2022, doi: 10.1109/TPWRS.2022.3213001.

[38] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. Int. Conf. Learn. Representations, 2017.

[39] A. Rossi, F. Frasca, B. Chamberlain, D. Eynard, M. Bronstein, and F. Moreno, “SIGN: Scalable inception graph neural networks,” ICML 2020 Workshop Graph Representation Learn. Beyond, 2020.

[40] F. Wu, A. Souza, T. Zhang, C. Fitty, T. Yu, and K. Weinberger, “Simplifying graph convolutional networks,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 6861–6871.

[41] J. Klicpera, S. Weißenberger, and S. Gümennenn, “Diffusion improves graph learning,” in Proc. Conf. Neural Inf. Process. Syst., 2019.

[42] C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1–9.

[43] A. Kendall and Y. Gal, “What uncertainties do we need in Bayesian deep learning for computer vision?,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 3580–3590.

[44] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, “Leveraging uncertainty information from deep neural networks for disease detection,” Sci. Rep., vol. 7, no. 1, pp. 1–14, 2017.

[45] A. Frangi and C. Gentile, “Solving nonlinear single-unit commitment problems with ramping constraints,” Operations Res., vol. 54, no. 4, pp. 767–775, 2006.

[46] J. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[47] RTE, “eco2mix - all of France’s electricity data in real time,” 2022. [Online]. Available: https://www.rte-france.com/en/eco2mix

[48] M. Chatzos, M. Tanneau, and P. Van Hentenryck, “Data-driven time series reconstruction for modern power systems research,” Electric Power Syst. Res., vol. 212, 2022, Art. no. 108589.

[49] I. Dunning, J. Huchette, and M. Lubin, “JuMP: A modeling language for mathematical optimization,” SIAM Rev., vol. 59, no. 2, pp. 295–320, 2017.

[50] Gurobi Optimization, LLC, “Gurobi optimizer reference manual,” 2021. [Online]. Available: https://www.gurobi.com

[51] PACE, “Partnership for an advanced computing environment (PACE),” 2017. [Online]. Available: http://www.pace.gatech.edu

[52] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Dataset Shift in Machine Learning, Cambridge, MA, USA: MIT Press, 2008.

[53] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in Proc. Int. Conf. Learn. Representations, 2017.

[54] F. Pedregosa et al., “Decoste: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[55] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” Nature, vol. 393, no. 6684, pp. 440–442, 1998.

[56] L. A. Roald and D. K. Molzahn, “Implied constraint satisfaction in power system optimization: The impacts of load variations,” in Proc. IEEE 57th Annu. Allerton Conf. Commun. Control Comput., 2019, pp. 308–315.

[57] N. Barry et al., “Risk-aware control and optimization for high-renewable power grids,” 2022, arXiv:2204.00950.
Seonho Park received the Ph.D. degree in industrial and systems engineering from the University of Florida, Gainesville, FL, USA. He is currently a Postdoctoral Fellow with the Georgia Institute of Technology, Atlanta, GA, USA. His research interests include optimization and machine learning. His current research focuses on utilizing machine learning techniques to accelerate the optimization process in power system applications.

Wenbo Chen received the B.S. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently working toward the Ph.D. degree in machine learning with the Georgia Institute of Technology, Atlanta, GA, USA. His research focuses on fusing machine learning & optimization into large-scale, intelligent systems with applications in energy systems and supply chains.

Dahye Han received the undergraduate degree in mathematics from Washington University, St. Louis, MO, USA. She is currently working toward the Ph.D. degree in operations research with the H. Milton Stewart School of Industrial and Systems Engineering, Georgia Tech, Atlanta, GA, USA. Her research interests include mixed-integer linear and nonlinear optimization, decision-making under uncertainty, and integration of future energy resources into the power grid. Before joining Georgia Tech, she was a Senior Analyst with National Economic Research Associates, Washington University.

Mathieu Tanneau received the M.Sc. degree from Ecole polytechnique, Palaiseau, France, and the Ph.D. degree from Polytechnique Montréal, Montréal, QC, Canada. He is currently a Research Engineer with the Georgia Institute of Technology, Atlanta, GA, USA. His research focuses on the combination of optimization and machine learning, with applications to power systems operations.

Pascal Van Hentenryck is currently the A. Russell Chandler III Chair and a Professor with the H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, GA, USA, and the Associate Chair for innovation and entrepreneurship. He is also the Director of the NSF AI for advances in optimization. His current research interests include machine learning, optimization, and with applications in energy and supply chains. He is an INFORMS Fellow and a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI). He was the recipient of two honorary doctoral degrees, and teaching excellence awards at Brown University, Providence, RI, USA, and Georgia Tech. His several optimization systems have been in commercial use for more than 20 years.