IEEE Open Access Journal of
Power and Energy

Received 20 August 2022; revised 4 November 2022, 12 January 2023, and 9 February 2023; accepted 19 February 2023.
Date of publication 27 February 2023; date of current version 1 March 2023.

Digital Object Identifier 10.1109/OAJPE.2023.3247989

Learning-Assisted Variables Reduction Method for Large-Scale MILP Unit Commitment

MOHAMED IBRAHIM ABDELAZIZ SHEKEEW (Student Member, IEEE)
AND BALA VENKATESH (Senior Member, IEEE)

Department of Electrical, Computer, and Biomedical Engineering, Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada
CORRESPONDING AUTHOR: B. VENKATESH (bala@ryerson.ca)

This work was supported in part by Mitacs Fund; in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Fund; and in part by the Centre for Urban Energy, Toronto Metropolitan University, Toronto, Canada.

ABSTRACT The security-constrained unit commitment (SCUC) challenge is solved repeatedly several times every day, for operations in a limited time. Typical mixed-integer linear programming (MILP) formulations are intertemporal in nature and have complex and discrete solution spaces that exponentially increase with system size. Improvements in the SCUC formulation and/or solution method that yield a faster solution hold immense economic value, as less time can be spent finding the best-known solution. Most machine learning (ML) methods in the literature either provide a warm start or convert the MILP-SCUC formulation to a continuous formulation, possibly leading to sub-optimality and/or infeasibility. In this paper, we propose a novel ML-based variables reduction method that accurately determines the optimal schedule for a subset of trusted generators, shrinking the MILP-SCUC formulation and dramatically reducing the search space. ML indicators sets are created to shrink the MILP-SCUC model, leading to improvement in the solution quality. Test results on IEEE systems with 14, 118, and 300 busses, the Ontario system, and Polish systems with 2383 and 3012 busses report significant reductions in solution times in the range of 48% to 98%. This is a promising tool for system operators to solve the MILP-SCUC with a lower optimality gap in a limited-time operation, leading to economic benefits.

INDEX TERMS Machine learning, mixed-integer linear programming, UC variables reduction, unit commitment.

NOMENCLATURE

A. ACRONYMS

SCUC	Security-Constrained Unit Commitment.
MLVR	Machine learning-based variables reduction.
MILP	Mixed-integer linear programming.
MILP-SCUC	MILP-SCUC formulation (1) – (23).
ML indicators sets	Machine learning indicators sets (28) – (34).
MLVR-MILP-SCUC	MILP-SCUC with machine learning-based variables reduction method (57).
TGS	Trusted-generators set $\boldsymbol{\alpha}$.
IGF	Intelligence generators factor.

B. SETS

SG	Set of generators (for unit index $g \in SG$, $SG = \{1, \ldots, NG\}$).
SB	Set of bus numbers (for bus indices, $i, j \in SB$, $SB = \{1, \ldots, NB\}$).
SM	Set of the number of segments, $m \in SM$, $SM = \{1, \ldots, NM\}$.
SK	Set of the transmission lines, $k \in SK$, $SK = \{1, \ldots, NL\}$.
ST	Set of the number of hours, $t \in ST$, $ST = \{1, \ldots, NT\}$.
$ST1$	Set of the number of hours except for the last hour, $t \in ST1$, $ST1 = \{1, \ldots, NT - 1\}$.
SD	Set of the number of load scenarios, $s \in SD$, $SD = \{1, \ldots, NS\}$.

$\boldsymbol{\alpha}$	Set of trusted generators; $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_g] \in \mathbb{Z}_2^{(1\times NG)}$, α_g =1 (trusted), α_g =0 (untrusted).
Λ^{on}	Set of ML on indicators (28).
Λ^{off}	Set of ML off indicators (29).
Λ^{un}	Set of unknown generation decisions (30).
Ψ	Set of ML startup indicators (31).
Π	Set of ML shutdown indicators (32).
Ψ^{un}	Set of unknown startup decisions (33).
Π^{un}	Set of unknown shutdown decisions (34).

C. PARAMETERS

$\overline{P}_g/\underline{P_g}$	Maximum/minimum capacity of generator g.
$\overline{P}_{gm}/\underline{P_{gm}}$	Maximum/minimum capacity of generator g in segment m.
$\overline{PL}_k$	The capacity of transmission line k.
Cs_g/Cd_g	Costs of startup/shutdown for unit g.
Ca_g, Cb_{gm}	Fuel cost coefficients.
Cr_g	Cost of the reserve requirement.
x_h	Transmission line inductive impedance.
λ	Factor for the online spinning reserve requirement for each hour.
D_{ti}	Net hourly demand which is the difference between the load forecast $d^f_{t,i}$ and the renewable generation at bus i, $w^f_{t,i}$.
B	Power network susceptance matrix.
RG_g^{10}	Spinning reserve capacity for unit g in 10 minutes.
RG_g^{60}	Generating unit ramp up/down limit in 60-minutes.
UT_g	Minimum uptime for generator g.
DT_g	Minimum downtime for generator g.
IT_g	The initial condition for generator g at starting dispatch time $t = 1$ (i.e., $IT_g = +6$ means unit g was working for 6 hours).
$w_{t,i}$	Forecasted renewable energy output at bus i at time t.
β_t	Time-wise power distribution factor.
β_i	Bus-wise load power distribution factor.

D. VARIABLES AND VECTORS

$p_{g,t}$	Hourly generation power of the unit g.
PI	Matrix of the active hourly bus-wise power injection.
$ug_{g,t}$	Unit g decisions (On/Off) at period t, $ug_{gt} \in \{0, 1\}$.
$uu_{g,t}$	Unit g startup status at period t, $uu_{gt} \in \{0, 1\}$.
$ud_{g,t}$	Unit g shutdown status at period t, $ud_{gt} \in \{0, 1\}$.
$\mathrm{sr_t}$	The system spinning reserve requirement for each hour.
$\mathrm{p_{t,g,m}}$	Power dispatch of unit g at period t for segment m.
$\mathrm{r_{t,g}}$	Spinning reserve of unit g in each period t.
$PL_{t,k}$	Power flow in transmission line k at period t.
$\delta_{\mathrm{t,i}}$	Hourly voltage angle at each bus i.
$\boldsymbol{D^s}$	The hourly bus net load data vector for scenario s, $\boldsymbol{D^s} = \left[D^s_{1,1} \cdots D^s_{NT,NB}\right]$.
$\boldsymbol{C^s}$	The daily generation cost curve data $\boldsymbol{C^s} = \left[Cb^s_{1,1} \cdots Cb^s_{NG,NM}\right]$.
$\boldsymbol{ug^s}$	The optimal generation decisions vector.
$\boldsymbol{uu^s}$	The optimal startup vector.
$\boldsymbol{ud^s}$	The optimal shutdown vector.
$\boldsymbol{q^s}$	ML output vector for a particular scenario s.
$\boldsymbol{e^s}$	Prediction error vector for scenario s.

I. INTRODUCTION

SECURITY-CONSTRAINED Unit Commitment (SCUC) is an essential tool for power systems generation scheduling from the day-ahead planning to the real-time operation day. In the Day-Ahead deregulated energy market, Independent System Operators (ISOs) solve the SCUC challenge to determine the least-known operating cost of the power system that satisfies the demand bids, operating reserve, system reliability constraints, etc. This process is performed several times a day before clearing the electricity market to ensure the safe operation of the power grid and the best possible economic outcome for market participants as a whole [1].

In this work, we aim to improve the solution quality (Optimality Gap and Computation Time) of the SCUC problem, which is usually formulated as a mixed-integer linear programming (MILP) problem, using the state of the art of machine learning (ML) techniques. To fathom the SCUC problem dimension, Midcontinent Independent System Operator (MISO), one of the largest ISOs in North America, has a power system network capacity of 175GW, with over 45,000 buses and 1.400 generation units. MISO has solved the SCUC using a commercial MILP solver CPLEX for market clearing since 2009. The Day-Ahead (DA) MILP-SCUC formulation for MISO's network is characterized by over 55,000 binary variables and about 400,000 continuous variables. The constraint matrix has 650,000 rows, 800,000 columns, and about 5 million non-zero elements before the pre-solve process [2]. The solver time for this formulation is about 20 minutes for each running scenario, which leads to several hours to settle the market. For practical purposes, a higher optimality gap is used whereby the best possible integer solution is not found. This causes economic loss [3]. Moreover, many other ISOs (e.g., IESO, ISO New England, and PJM) require about 3 to 5 hours to solve the DA-SCUC process and post the market-clearing prices for the next day [4], [5].

To put this DA-SCUC in context, New England's 32 GW system transacts energy worth $10 billion/year. A $100 million annual benefit would be achieved for only a 1% improvement in solving the DA-SCUC formulation. This is a huge economic implication for only one ISO in the US, showing the great potential of opening the door to improving

the SCUC process and finding efficiencies that benefit all stakeholders [6].

A. MATHEMATICAL FORMULATION

SCUC is characterized as a nonconvex and discrete optimization challenge. It includes a large number of binary variables that represent the hourly power station operation status such as ON/OFF status, startup and shutdown, and numerous continuous variables comprising generation, ramp, and reserve power. Many optimization methods have been utilized to solve the SCUC challenge. The most widely applied methods are Lagrangian relaxation (LR) [7], [8], [9] and MILP [10], [11], [12], [13], [14], [15].

Using LR to solve the SCUC challenge, the objective function (the cost function) is the sum of terms, each involving a set of constraints describing a single unit and a set of coupling constraints (the generation and reserve requirements). The LR model is solved for every single unit, with the set of these solutions providing a lower bound on the optimal value. Then, the dual function is assigned this optimal value to find the best convex lower bound. LR is one of the methods used by ISOs to settle the Day-Ahead market, like PJM before transitioning to the MILP method in 2005 [10]. Recently, some efforts have been exerted to overcome the computational complexity of the large-scale LR-based SCUC problem [7], [8]. In [7], the LR method has been improved using Linear Programming (LP) to solve many single-period SCUC problems. The optimal multipliers of these single-period UC solutions have been used to create a criterion to fix some binary variables, helping to speed up the solution time for the full-scale SCUC problem. In [8], the LR technique is improved to solve the SCUC by replacing the dual optimal solutions by adding quadratic penalties for violating constraints. However, the LR technique still provides a slow convergence time and heavy computational burden for the complex mixed-integer SCUC challenge [9].

Due to the development of state-of-the-art MILP solvers such as CPLEX and GUROBI, the ISOs started to utilize the MILP technique to obtain the best-known mixed-integer DA-SCUC solutions. These solvers first relax integer variables to determine an LP solution. Then, using the branch-and-cut technique, determine the best possible MILP solution, such that the optimality gap (OG) with respect to the LP solution is the least. OG is the difference between the MILP solution and LP solution expressed as a percentage of the LP solution.

For MILP, the DA-SCUC objective function is linearized using the piece-wise linearization method. In addition, the constraints are formulated in the mixed-integer form to obtain the best solution without developing specific heuristics, which will accelerate the development of a program and facilitate its applications to large-scale power systems. However, the MILP-SCUC's solution quality challenges are: 1) an enormous number of binary variables that introduce discrete solution spaces; and 2) a large-scale system of constraints (binary and mixed-binary constraints) [2]. For a complex SCUC challenge as in current power systems, commercial solvers take a large computation time to obtain a MILP solution for a low OG and may not converge.

B. LITERATURE ON IMPROVEMENT IN MILP METHODS

In [11], [12], [13], [14], and [15], improvements to MILP-SCUC formulations considering hydropower generation are reported. In [11], Hydro unit commitment (HUC) has been updated in order to consider a multi-unit pump-storage hydropower station in a practical system. The authors have derived a linear model for the hydropower generation and the corresponding water flow from the reservoir. Then, this model has been added to the main MILP-SCUC formulation. In [12], variable separation and piecewise linearization are used in parallel to linearize the hydro unit power generation with the net head to be adopted in the MILP UC problem. Some other challenges in HUC have been solved, such as vibrating zones, where changes to hydro unit power output have to be permanently avoided as in [13], and the matter of forbidden zones [14]. In [15], a preprocessing phase to calculate the total hydro output based on the limits of the plant volume, downstream flow, and operating zones for each unit, ahead of the UC solution, has been proposed. Due to the traditional MILP-SCUC formulation requiring significant computational time to reach the best-known mixed-integer solution for the large-scale SCUC challenge, researchers have tightened the MILP formulation to overcome the computational complexity [16], [17], [18], [19]. In [16], a piecewise method for tightening the objective function is proposed which is a fundamentally quadratic equation. Startup, shutdown, and ramp constraints have been tightened to reduce the computation time for large scale systems as in [17], [18], [19], and [20].

C. LITERATURE ON USING ML IN UC AND POWER SYSTEMS

In the last decades, due to a lack of data and computational resources, ML has not been used extensively in power systems. Recently, ISOs have made considerable amounts of historical data available in the public domain that can be used for the SCUC challenge. In addition, ML has become a powerful tool that can map vast nonlinear input-output data and predict outputs efficiently and accurately [21], [22], [23], [24], [25], [26], [27]. In [21], ML has been utilized in order to discard the redundant constraints in the transmission-constrained UC challenge, using the historical data. However, the method has not been tested in a complex large scale multi-period SCUC challenge. While in [22], ML has been used to classify the transmission constraints into two sets, hard and easy to be reduced in SCUC problem size. However, this method provides inaccuracies as mentioned by the authors. Data-driven decisions have been extracted from the trained ML to act as an expert system to solve the UC problem [23]. In [24], the authors have presented

different ML techniques to enhance the SCUC formulation. The closest approach for our work is called ML affine subspaces. This method uses the ML model to predict a portion of generators' binary variables, then a hyperplane method is applied to the ML output to create the predictor which fixes generators' binary variables to their best values. Since this method depends on a heuristic tuning parameter to obtain the best value for the binary variables, the possibilities of infeasibility and suboptimality still exist. In [25], ML output has been used as a warm start to speed up the solution of SCUC. Currently, commercial solvers (e.g., Gurobi and CPLEX) provide robust feasible pre-solve processes to warm start the MILP solution process [28]. In [26], ML has been used to predict the uncertainty of line outages due to hurricanes and used as input in solving the stochastic UC challenge. In [27], the ML has been trained to predict the generation binary schedules. The ML full output (100%) has been used directly to fix the binary variables of MILP-SCUC formulation if they are feasible to minimize up and down constraints; otherwise, the ML output should be updated to adequately cover these constraints. However, the ML full output cannot be trusted because the ML output may carry error decisions and be feasible at the same time, which may cause infeasibility and sub-optimality.

D. CONTRIBUTIONS OF THIS PAPER

ML output cannot replace the conventional MILP-SCUC solution with 100% accuracy. In the literature, ML is used to predict a solution (binary or continuous) to warm start the SCUC problem. However, commercial solvers use a pre-solve process to remove redundant constraints [2]. These pre-solve processes may discard user-provided starting solutions and hence render warm start approaches using ML techniques useless. Many works in the literature use full ML output as in [27] or heuristic-based tuning methods as in [24] to fix the binary variables of the SCUC problem. However, they may cause infeasibility or suboptimality. Instead, we provide an ML-based binary variables reduction method using a trusted-generators set to reduce the complexity of the MILP-SCUC formulation that may directly be used with commercial solvers. Reduction of the binary variables yields lesser Mixed-Binary (MB) constraints, decreasing the search space, which reduces the computational time and complexity of the large-scale DA-SCUC problem.

The main contributions of this work are summarized below:

1) A novel ML-based variables reduction (MLVR) method is developed based on the creation of a Trusted-Generators Set (TGS), an offline process. The status of TSG generators can be determined using the trained ML module.
2) Using TGS, ML indicators sets are created on the day-ahead to shrink the MILP-SCUC formulation. That leads to a significant reduction in the search space. The MLVR-MILP-SCUC method offers significant improvement in the solution quality in terms of computation time reduction and optimality gap reduction.
3) The proposed method provides the same optimal solution faster for small-scale systems and with a lower optimality gap for the large-scale systems, such as Polish systems of 2383-bus and 3012-bus, bringing a lower best-known solution that provides economic benefits for the stakeholders.

II. MILP-SCUC PROBLEM FORMULATION

In this section, we present the main MILP-SCUC formulation, which uses three binary variables (unit status, startup, and shutdown).

The optimal solution of the SCUC problem achieves the minimum total operation cost subject to a set of the system security constraints. The objective function (1) includes the shutdown, startup, fuel cost, no-load cost, and reserve power cost. The tensive formulations can be found in [1] and [19]. The fuel cost is a fundamentally quadratic equation, and it is linearized using the piece-wise method to be compatible with MILP formulation.

The SCUC's constraints are classified into three types:

A. CONTINUOUS CONSTRAINTS (2) – (10)

These constraints are dependent only on the continuous variables. Constraint (2) defines the generation power at each period, which is a summation of generating powers in each segment. The generation power at each segment is limited by its maximum as in (3). Constraint (4) ensures the generation, injected powers from transmission lines, and demand are balanced at each bus, where $D_{t,i}$ presents the hourly net load at each node and it includes the bus-wise distribution factor (more details are provided in Section III-A). Node-injected power is created using the DC load flow as in (5). Transmission power flow is calculated and limited as in (6) and (7). Constraint (8) represents the online spinning reserve requirement. The parameter λ characterizes the necessary ratio of the reserve when the system is interconnected with other power grids, as is the case in Ontario where the Northeast Power Coordinating Council requires it to equal 0.25 [29]. Generator ramp rates and bus-voltage angles are limited as in (9) and (10) respectively.

B. MIXED-BINARY CONSTRAINTS (11) – (13)

These constraints depend on binary commitment variables and continuous variables. Constraint (11) ensures the dispatched power of each online generator is within the generation limits. The spinning reserve of each unit is limited by its reserve requirements as in (12) and available power as presented in constraint (13). Spinning reserve capacity at each period equals the capacity of the maximum online generator as enforced in constraint (14).

C. BINARY CONSTRAINTS (14) – (23)

These constraints depend only on the generators' decisions (startup, shutdown, and status variables). Constraint (15)

illustrates the shutdown and startup decisions. The startup and shutdown decisions cannot be made at the same time as enforced by constraint (16). The minimum up and down time of each generator is achieved as in constraints (17) and (18) respectively. Further, the status of the generators for the previous day is considered in (19), (20). Finally, the binary variables (unit status, startup, and shutdown) are constrained in (21) – (23) respectively.

The objective function is defined as:

$$Minimize: \sum_{t=1}^{NT}\sum_{g=1}^{NG}\begin{bmatrix} Cd_g \cdot ud_{t,g} + Cs_g \cdot uu_{t,g} + Cr_g \cdot r_{t,g} \\ +Ca_g \cdot ug_{t,g} + \sum_{m=1}^{NM} Cb_{g,m} \cdot p_{t,g,m} \end{bmatrix} \tag{1}$$

Subject to :

1) Continuous constraints (2) – (10)

$$p_{t,g} = \sum_{m}^{NM} p_{t,g,m}; \quad \forall (t,g) \in \{ST, SG\} \tag{2}$$

$$0 \le p_{t,g,m} \le \overline{P}_{g,m}; \quad \forall (t,g,m) \in \{ST, SG, SM\} \tag{3}$$

$$p_{t,g} - D_{t,i} = PI_{t,i}; \quad \forall g \in i, (t,i) \in \{ST, SB\} \tag{4}$$

$$PI_{t,i} = B' \cdot \delta_{t,i}; \quad \forall (t,i) \in \{ST, SB\} \tag{5}$$

$$PL_{tk} = \frac{\delta_{ti} - \delta_{tj}}{x_k}; \quad \forall (i,j) \in k, (t,k) \in \{ST, SK\} \tag{6}$$

$$-\overline{PL}_k \le PL_{t,k} \le \overline{PL}_k; \quad \forall (t,k) \in \{ST, SK\} \tag{7}$$

$$\sum_{g}^{NG} r_{t,g} \ge \lambda . sr_t; \quad \forall t \in ST \tag{8}$$

$$-RG_g^{60} \le p_{t,g} - p_{t-1,g} \le RG_g^{60}; \quad \forall (t,g) \in \{ST, SG\} \tag{9}$$

$$-\frac{\pi}{2} \le \delta_{t,i} \le \frac{\pi}{2}; \quad \forall (t,i) \in \{ST, SB\} \tag{10}$$

2) Mixed-binary constraints (11) – (14)

$$ug_{t,g} \cdot \underline{P}_g \le p_{t,g} \le ug_{t,g} \cdot \overline{P}_g; \quad \forall (t,g) \in \{ST, SG\} \tag{11}$$

$$r_{tg} \le ug_{tg} \cdot RG_g^{10}; \quad \forall (t,g) \in \{ST, SG\} \tag{12}$$

$$r_{tg} \le \sum_{m}^{NM} \left(\overline{P}_{gm} \cdot ug_{tg} - p_{tgm}\right); \quad \forall (t,g) \in \{ST, SG\} \tag{13}$$

$$sr_t = \max_{g \in SG} \left\{ug_{tg} \cdot \overline{P}_g\right\}; \quad \forall t \in ST \tag{14}$$

3) Binary constraints (15) – (23)

$$ug_{t+1,g} - ug_{tg} = uu_{t+1,g} - ud_{t+1,g}; \quad \forall (t,g) \in \{ST1, SG\} \tag{15}$$

$$0 \le uu_{t,g} + ud_{t,g} \le 1; \quad \forall (t,g) \in \{ST1, SG\} \tag{16}$$

$$us_{t+1,g} . UT_g - \sum_{v=t+2}^{\min\{NT, t+UT_g\}} ug_{v,g} \le \max\left\{1, UT_g - NT + t\right\}; \quad \forall (t,g) \in \{ST1, SG\} \tag{17}$$

$$\left(1 - ud_{t+1,g}\right) \cdot ST \ge \sum_{v=t+1}^{\min\{NT, t+DT_g\}} ug_{vg}; \quad \forall (t,g) \in \{ST1, SG\} \tag{18}$$

$$If\ IT_g > 0\ \&\&\ UT_g > IT_g: \quad \sum_{t=1}^{UT_g - IT_g} ug_{t,g} \ge UT_g - IT_g; \quad \forall g \in SG \tag{19}$$

$$If\ IT_g < 0\ \&\&\ DT_g > -IT_g: \quad \sum_{t=1}^{DT_g - IT_g} ug_{t,g} \le 0; \quad \forall g \in SG \tag{20}$$

$$ug_{tg} \in \{0,1\}; \quad \forall (t,g) \in \{ST, SG\} \tag{21}$$

$$uu_{tg} \in \{0,1\}; \quad \forall (t,g) \in \{ST, SG\} \tag{22}$$

$$ud_{tg} \in \{0,1\}; \quad \forall (t,g) \in \{ST, SG\} \tag{23}$$

When the above-mentioned MILP-SCUC formulation (1) – (23) is used for a large-scale power system network, the number of variables and constraints becomes extremely large. Notably, the binary variables introduce non-convexity in addition to numerous mixed-binary constraints. Due to this, the solver's computational time exponentially increases with the problem size. However, ISOs terminate the solver at a large optimality gap to obtain the solution within a limited time. As a result, stakeholders miss economic benefits.

III. SYSTEM MODEL OF MLVR-MILP-SCUC METHOD

This paper aims to create a trusted set of generators using the ML model. This set is used to accurately predict scheduling decisions. These sets are used to reduce the MILP-SCUC problem size.

A. ALGORITHM #1 – ML MODEL TRAINING AND TRUSTED GENERATORS SET CREATION [OFF-LINE]

A unit's startup and shutdown optimal vectors, ***uu*** and ***ud***, are dependent variables that are obtainable from the unit's status vector ***ug***. As a result, the primary binary optimal solution for MILP-SCUC (1) is only ***ug***.

In this paper, the ML model is trained by the net hourly load data at each bus and the generation cost curve data as an input vector, and the status of generators at each period as the target output. The net hourly load is defined as the difference between the forecasted demand d_{it}^f and renewable energy source w_{it}^f at each bus as in (24).

$$D_{t,i} = d_{t,i}^f - w_{t,i}^f; \quad \forall (t,i) \in \{ST, SB\} \tag{24}$$

Thereafter, we propose a trusted-generators set (TGS). In the DA-SCUC process, considering load and generation cost curve data and using the ML model, the decisions of the TGS can be determined. It reduces the variable map as shown in Fig. 1 that illustrates the main concept of using the proposed ML model. Thus, the search space of the SCUC formulation is reduced, and therefore the computation quality is improved. It allows the MILP solver to explore solutions with a lower OG within the available time and hence provides a lower-costing solution, benefiting all. Algorithm #1 [off-line] details the procedures to train the ML model (Stage 1) and create the TGS (Stage 2).

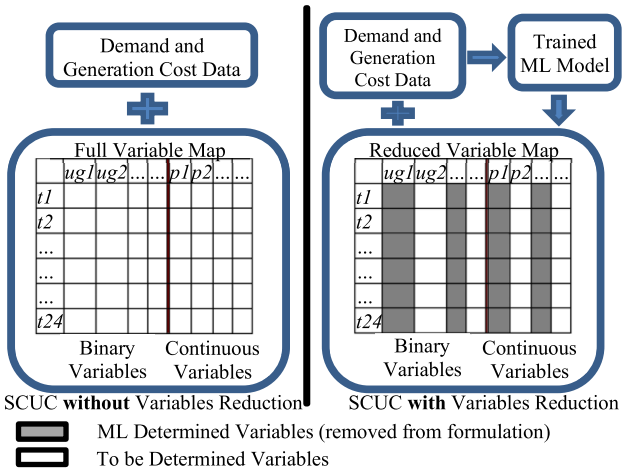

FIGURE 1. **MLVR concept.**

Algorithm 1 MLVR Method [Off-Line]

Stage 1: ML-Training Model (off-line)

1 Use [$\boldsymbol{D}$, $\boldsymbol{C}$] and $\boldsymbol{ug}$ from the historical data if available. If not, **Solve** the MILP-SCUC (1) – (23) for the net forecasted $\boldsymbol{D}$ scenarios **and** create $\boldsymbol{ug}$.

Stage 2: Create Trusted-Generators Set, $\boldsymbol{\alpha}$

2 Use (25) on the ML-output $\boldsymbol{q}$ to obtain the error matrix $\boldsymbol{e}$ for all data sets.
3 Use (26) to determine TGS, $\boldsymbol{\alpha}$.
4 Calculate η using (27)
5 if $\eta \neq 0$, then
Create $\boldsymbol{\alpha}$
else
Retrain the ML model with larger data and different training methods.

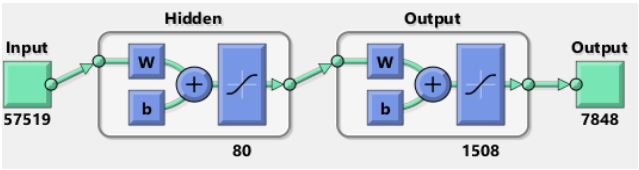

FIGURE 2. **Three layers neural network architecture for the 2383-bus system.**

Stage 1 – The premise is that most ISOs have SCUC data comprised of input (net hourly demand at each bus and the generation cost curve data) and output (generator status and power output) for more than 10 years, for example at MISO [30]. Using this data, the net hourly bus demand, generation cost curve data, and corresponding SCUC schedules of generators can be collated and used to create an ML model.

In this paper, the ML model is trained using a three-layers feedforward backpropagation Artificial Neural Network (ANN), a powerful supervised learning tool that maps the non-linear input-output relationship. Fig. 2 shows the fully connected ANN architecture. Hyperbolic tangent sigmoid is the activation function used in the hidden layers. Furthermore, the Robust backpropagation algorithm (Rprop) is used for the training process to combat system scalability. Rprop is the fastest method for pattern recognition. Unlike Levenberg–Marquardt, it needs less memory to train well [31].

The ANN tuning parameters must be well-tuned to minimize Mean Squared Error (MSE). These parameters include the number of layers, epochs, learning data, optimizer, batch size, neurons, and activation function. These metrics are essentially empirical and tuned until the best performance is achieved.

The ML model is trained using the hourly bus net load data vector $\boldsymbol{D^s} = \left[D^s_{1,1} \cdots D^s_{NT,NB}\right]$ and piecewise linear generation cost curve data $\boldsymbol{C^s} = \left[Cb^s_{1,1} \cdots Cb^s_{NG,NM}\right]$ as input and the generating unit decisions $\boldsymbol{ug^s} = \left[ug^s_{1,1} \cdots ug^s_{NT,NG}\right]$ as output for hundreds of available scenarios $s \in SD$. Although the ML model would map input and output, there would be errors in the output. In [27], it is similarly pointed out that this mapping might not be accurate. However, a subset of generators – the TGS – would be predicted perfectly by the well-trained ML model. Hence, in this paper, we introduce the concept of a TGS.

Stage 2 – We analytically determine a set of generators, TGS, that predicts the status of generators with 100% accuracy (without prediction errors) in all tested data. This process is performed in the off-line mode once the ML model is well trained.

Accordingly, the trained ML model uses the net demand data and generation cost curve data and accurately predicts the status of a TGS.

In stage 2, the ML output error for each trained load scenario 's' is defined as $\boldsymbol{e^s} \in \mathbb{Z}_+^{(ST \cdot SG)}$ and it can be computed as the absolute difference between the actual SCUC schedules $\boldsymbol{ug^s}$ and the ML output $\boldsymbol{q^s} = \left[q^s_{1,1} \cdots q^s_{NT,NG}\right] \in \mathbb{Z}_2^{(NT \times NG)}$ as in (25).

$$\boldsymbol{e^s} = \left|\boldsymbol{ug^s} - \boldsymbol{q^s}\right|, \in \mathbb{Z}_+^{(NT \times NG)} \tag{25}$$

This error is analyzed for each generator in the system to create the TGS that discards any generator that provides false predictions in the error matrix. TGS is defined as $\boldsymbol{\alpha} = \left[\alpha_1 \cdots \alpha_g\right] \in \mathbb{Z}_2^{(1 \times NG)}$ where α_g is a binary that defines the trusted or untrusted generator as in (26).

$$\alpha_g = \begin{cases} 1, \ (Trusted) & if \sum_s \sum_t \boldsymbol{e^s_{tg}} = 0 \\ 0, \ (Untrusted) & otherwise \end{cases} \tag{26}$$

This set defines the intelligence generators factor (IGF) of the power system, as below in (27).

$$\eta = \left(1/NG\right) \cdot \sum_g \alpha_g \tag{27}$$

Further, IGF η is used as a performance factor for the ML training model. Hence, a lower IGF indicates the training data

is not enough to create the TGS. Consequently, using the ML output of TGS, the MLVR method is created by introducing ML variables indicators, as in the following subsection, that shrink the MILP-SCUC variable map and mixed binary constraints. Hence, both the problem size and the search space are decreased.

In summary, as shown in Algorithm #1 [off-line] the MLVR method trains the ML model in off-line mode using this vast amount of data (net hourly bus-wise load and generation cost curve data as input and corresponding SCUC schedules of generators as output), checks the output for errors, and determines a TGS as outlined in this section.

B. ML-VARIABLES REDUCTION METHOD – DEVELOPMENT OF THE ML INDICATORS SETS [DAY-AHEAD]

In the Day-Ahead Unit Commitment Process, the ML model considers day-ahead net demand and the generation cost curve data $(\boldsymbol{D^s}, \boldsymbol{C^s})$. It generates an output schedule for generators $\boldsymbol{q^s}$. Only statuses corresponding to the TGS $\boldsymbol{\alpha}$ are to be retained and used in the MILP-based SCUC formulation (1) – (23). To filter the ML output $\boldsymbol{q^s}$, we define $\Lambda = [\Lambda^{on} \cup \Lambda^{off}]$ as a set of decisions (on and off sets) corresponding to TSG as declared in (28) and (29) respectively. This set is part of the optimal binary solution ($\Lambda \subset \boldsymbol{ug^s}$) and the unknown binary part can be defined as Λ^{un} in (30).

$$\Lambda^{on} := \left\{ \left\{ \Lambda_1, \ldots, \Lambda_{t,g} \right\},\ \Lambda_{t,g} = 1 : q_{tg} = 1, \alpha_g = 1 \right\} \tag{28}$$

$$\Lambda^{off} := \left\{ \left\{ \Lambda_1, \ldots, \Lambda_{t,g} \right\},\ \Lambda_{t,g} = 0 : q_{tg} = 0, \alpha_g = 1 \right\} \tag{29}$$

$$\Lambda^{un} := \left\{ \left\{ \Lambda_1^{un}, \ldots, \Lambda_{t,g}^{un} \right\},\ \forall \alpha_g = 0, t \in ST \right\} \tag{30}$$

Since the startup/shutdown variables are dependent on the unit status, two more indicator sets $\{\Psi, \Pi\}$ are produced. These sets are part of the optimal startup and shutdown vectors ($\boldsymbol{uu^s}$, $\boldsymbol{ud^s}$). Each element in $\{\Psi, \Pi\}$ is obtained as in (31) and (32) respectively. Further, the sets of unknown startup and shutdown decisions are created in (33) and (34) respectively.

$$\Psi_{t,g} = \begin{cases} 1, & if\ q_{t+1,g} - q_{tg} = 1 \\ 0, & Otherwise, \end{cases} \quad \forall \alpha_g = 1, \quad t \in ST1 \tag{31}$$

$$\Pi_{t,g} = \begin{cases} 1, & if\ q_{t,g} - q_{t+1,g} = 1 \\ 0, & Otherwise, \end{cases} \quad \forall \alpha_g = 1,\ t \in ST1 \tag{32}$$

$$\Psi^{un} := \left\{ \left\{ \Psi_{1,1}^{un}, \ldots, \Psi_{t,g}^{un} \right\}, \forall \alpha_g = 0, t \in ST1 \right\} \tag{33}$$

$$\Pi^{un} := \left\{ \left\{ \Pi_{1,1}^{un}, \ldots, \Pi_{t,g}^{un} \right\}, \forall \alpha_g = 0, t \in ST1 \right\} \tag{34}$$

These four sets (Λ^{on}, Λ^{off}, Ψ, Π) have to undergo a feasibility check (35) to make sure each trusted generator in TGS is feasible for the binary constraints (15) – (23), *i.e.,* the minimum up and down constraints. If a generator in TGS violates, its decisions should be discarded, and all sets have to be updated accordingly. Finally, these four ML indicators sets form the ML binary variables reduction method that shrinks the MILP-SCUC problem (1) – (23).

---

for each unit $g = 1 : NG$, $g \in \boldsymbol{\alpha}$ **do**
  if $\alpha_g = 1$, **then**

$$Minimize : \sum_t \begin{bmatrix} Cd_g \cdot ud_{tg} + Cs_g \cdot \\ uu_{tg} + Ca_g \cdot ug_{tg} \end{bmatrix}, \tag{35a}$$

Subject to: (15) – (23), (35b)

  if (35) **is infeasible**
    Remove all decisions corresponding to g in all predicted set (Λ^{on}, Λ^{off}, Ψ, Π).
  end if

Export the updated decisions sets (Λ^{on}, Λ^{off}, Ψ, Π) and update the unknown decision sets (Λ^{un}, Ψ^{un}, Π^{un}) accordingly.

IV. PROPOSED MLVR-MILP-SCUC METHOD – COMPLETE REAL-TIME MODEL

A. EFFECT OF ML INDICATOR SETS ON THE SCUC PROBLEM SIZE

In this subsection, we present the theory of the effect of using ML sets on the MILP-SCUC problem size. These sets as discussed predict only binary variables for TGS, thus the total binary variables in the main formulation will be decreased. In addition, if the predicted status is off, some of the continuous variables (i.e., p_{tg}, r_{tg}) will be reduced. As a result, the size of the constraints (mixed-binary, and binary constraints) will be decreased, while the continuous constraints will be changed based on the total number of the off/on statuses as shown in the following subsections.

1) UPDATED CONTINUOUS CONSTRAINTS

The constraints in (2), (3), and (9) are replaced by (36) – (39) respectively as below:

$$p_{t,g} = \sum_m^{NM} p_{t,g,m};\ \forall (t,g) \in \left\{ \Lambda^{on}, \Lambda^{un} \right\} \tag{36}$$

$$p_{t,g} = 0; \quad \forall (t,g) \in \left\{ \Lambda^{off} \right\} \tag{37}$$

$$0 \le p_{t,g,m} \le \overline{P}_{g,m};\ \forall (t,g,m) \in \left\{ \Lambda^{on} \cup \Lambda^{un}, SM \right\} \tag{38}$$

$$-RG_g^{60} \le p_{t,g} - p_{t-1,g} \le RG_g^{60};\ \forall (t,g) \in \left\{ \Lambda^{on} \cup \Lambda^{un} \right\} \tag{39}$$

2) UPDATED MIXED-BINARY CONSTRAINTS (11) – (14)

Part of the MB constraint (11) will be transformed into a continuous constraint as in (40) and the remaining part will be a reduced MB constraint as in (41). Consequently, constraints (12) – (14) are updated as in (42) – (47) respectively.

As shown, the constraints in (42) – (44) are continuous constraints.

$$\underline{P_g} \le p_{t,g} \le \overline{P}_g;\ \forall (t,g) \in \left\{\Lambda^{on}\right\} \tag{40}$$

$$ug_{t,g} \cdot \underline{P_g} \le p_{t,g} \le ug_{t,g} \cdot \overline{P}_g;\ \forall (t,g) \in \left\{\Lambda^{un}\right\} \tag{41}$$

$$r_{tg} = 0;\ \ \forall (t,g) \in \left\{\Lambda^{off}\right\} \tag{42}$$

$$r_{tg} \le RG_g^{10};\ \ \forall (t,g) \in \left\{\Lambda^{on}\right\} \tag{43}$$

$$r_{tg} \le \sum_{m}^{NM} \left(\overline{P}_{gm} - p_{tgm}\right);\ \forall (t,g) \in \left\{\Lambda^{on}\right\} \tag{44}$$

$$r_{tg} \le ug_{tg} \cdot RG_g^{10};\ \ \forall (t,g) \in \left\{\Lambda^{un}\right\} \tag{45}$$

$$r_{tg} \le \sum_{m}^{NM} \left(\overline{P}_{gm} \cdot ug_{tg} - p_{tgm}\right);\ \ \forall (t,g) \in \left\{\Lambda^{un}\right\} \tag{46}$$

$$sr_t = \max_{\{v \in \Lambda^{on}, g \in \Lambda^{un}\}} \left\{\overline{P}_v, u_{tg} \cdot \overline{P}_g\right\},\ \ \forall t \in ST \tag{47}$$

3) UPDATED BINARY CONSTRAINTS (15) – (23)

The constraints in (15) – (23) are replaced by (48) – (56) respectively as below:

$$ug_{t+1,g} - ug_{tg} = uu_{t+1,g} - ud_{t+1,g};\ \forall (t,g) \in \left\{\Psi^{un}\right\} \tag{48}$$

$$0 \le uu_{t,g} + ud_{t,g} \le 1;\ \forall (t,g) \in \left\{\Pi^{un}\right\} \tag{49}$$

$$us_{t+1,g}.UT_g - \sum_{v=t+2}^{\min\{NT, t+UT_g\}} ug_{v,g} \le \max\left\{1, UT_g - NT + t\right\};\ \forall (t,g) \in \left\{\Psi^{un}\right\} \tag{50}$$

$$\left(1 - ud_{t+1,g}\right) \cdot ST \ge \sum_{v=t+1}^{\min\{NT, t+DT_g\}} ug_{vg};\ \forall (t,g) \in \left\{\Pi^{un}\right\} \tag{51}$$

If $IT_g > 0$ && $UT_g > IT_g$:

$$\sum_{t=1}^{UT_g - IT_g} ug_{t,g} \ge UT_g - IT_g;\ \forall (t,g) \in \Lambda^{un} \tag{52}$$

If $IT_g < 0$ && $DT_g > -IT_g$:

$$\sum_{t=1}^{DT_g - IT_g} ug_{t,g} \le 0;\ \ \forall (t,g) \in \Lambda^{un} \tag{53}$$

$$ug_{tg} \in \{0,1\};\ \ \forall (t,g) \in \left\{\Lambda^{un}\right\} \tag{54}$$

$$uu_{tg} \in \{0,1\};\ \ \forall (t,g) \in \left\{\Psi^{un}\right\} \tag{55}$$

$$ud_{tg} \in \{0,1\};\ \ \forall (t,g) \in \left\{\Pi^{un}\right\} \tag{56}$$

Now the complete MILP-SCUC method with ML variables reduction is formulated below in (57). The new formulation is to be solved for undetermined continuous variables and unknown binary variables that equal the original binary variables minus ML binary indicators.

$$\text{Minimize} \quad (1) \tag{57a}$$

Subject to:

1. Updated continuous constraints (57b)

$$(4)\text{–}(8),\ (10),\ (36)\text{–}(40),\ (42)\text{–}(44) \tag{57b}$$

2. Updated MB constraints (57c)

$$(41),\ (45)\text{–}(47) \tag{57c}$$

3. Updated binary constraints (57d)

$$(48)\text{–}(56) \tag{57d}$$

Solve (57) to get the decisions of Λ^{un}, Ψ^{un}, Π^{un} and the remaining continuous variables.

Note that the original problem (1) – (23) has $3 \times NG \times NT$ binary variables and continuous variables $p_{t,g,m}, r_{tg}, sr_{t,g}, \delta_{t,n}$. The effect of the MLVR method is illustrated in the following knowledge matrices. The main objective of the MLVR reduces the binary variables by $\left[3NG \times NT - size(\Lambda^{\text{on}}, \Lambda^{\text{off}}, \Psi, \Pi)\right]$ as shown in the knowledge matrix (A). When the trusted generator predicts off status, continuous variables as shown in knowledge matrix (B) will be reduced. As a result, the size of the MILP-SCUC formulation is shrunk, the solution space is reduced, and the solution speed is increased without a degradation in quality.

A. Knowledge Matrix of 24 hours $\forall(t,g) \in \{ST, SG\}$

Condition	α_g	1	0
Outcome	u_{tg}	Known	Unknown
Outcome	us_{tg}	Known	Unknown
Outcome	ud_{tg}	Known	Unknown

B. Knowledge Matrix of each *hour*

Condition	α_g	1	1
Condition	u_{tg}	1	0
Outcome	p_{tg}	Exists (Unknown)	Doesn't exist
Outcome	p_{tgm}	Exists (Unknown)	Doesn't exist
Outcome	r_{tg}	Exists (Unknown)	Doesn't exist

B. ALGORITHM #2 - MILP-SCUC WITH MLVR METHOD [DAY-AHEAD]

The full MILP-SCUC with MLVR method is shown in Algorithm #2 and Fig. 3. As in Algorithm #1, the off-line ML training model aims to train massive data comprising the hourly bus-wise loads and corresponding generation schedules. By analyzing the ML model in the off-line stage, a TGS $\boldsymbol{\alpha}$ is created. The trained ML model and TGS $\boldsymbol{\alpha}$ are used in the day-ahead Algorithm #2. In the day ahead, with a day-ahead of hourly bus-wise net demand, output from the trained ML model is obtained. ML indicators are created as shown in (28) – (34) to form the shrunk MLVR-MILP-SCUC formulation (57).

Fig. 3 illustrates the complete framework given in this paper for the MLVR-MILP-SCUC challenge. It is imperative to note that the ML model provides feasible generation schedules to the standard MILP-SCUC formulation. The MILP-SCUC with the MLVR method (57) can be efficiently solved using any commercial MILP solver.

V. CASE STUDIES RESULTS

Cases for the IEEE 14-bus (5 generators), 118-bus (54 generators), 300-bus (69 generators), Polish 2383-bus

Algorithm 2 MILP-SCUC Formulation With MLVR Method [Day-Ahead]

Stage 1: Development of the ML indicators sets

1 Collate the day-ahead hourly bus-wise net loading data and the generation cost curve data ($\boldsymbol{D}^s$, $\boldsymbol{C}^s$).
2 Use the trained ML model to determine the predicted generation schedule $\boldsymbol{q}^s$.
3 Use TGS (α) to create the sets of indicators as in (28) – (34),
  3a **Check the feasibility** (35) for each generator $g \in \boldsymbol{\alpha}$ using ML indicators (Λ^{on}, Λ^{off}, Ψ, Π).
  3b **If infeasible,** then delete predicted decisions of g from ML indicator sets.
4 Export the updated (Λ^{on}, Λ^{off}, Ψ, Π).

Stage 2: Enhance MILP-DAUC Formulation with MLVR method

1 **Create** the MLVR-MILP-SCUC formulation as shown in (57a) – (57d).
2 **Solve** the complete MLVR-MILP-SCUC formulation (57) with a standard MILP solver and issue commitment schedules plus dispatch results.

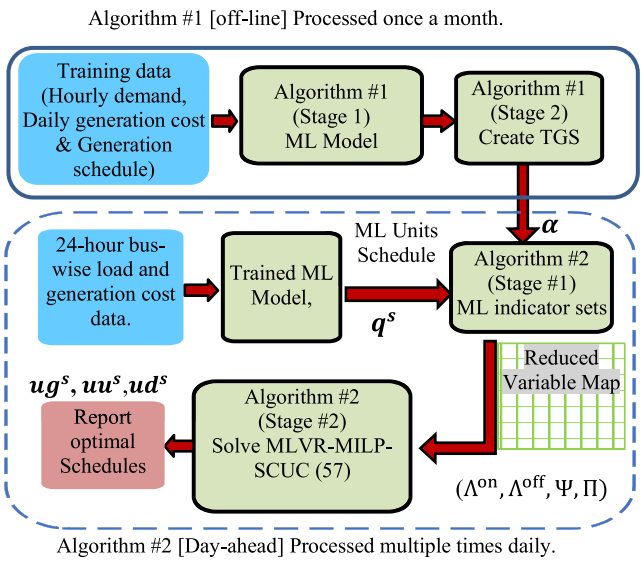

FIGURE 3. **Complete framework for the MILP-SCUC with MLVR method.**

(327 generators), Polish 3012-bus (502 generators), and the 10-zone Ontario (131 generators −15 node) are used to test the proposed MLVR-MILP-SCUC method. The tests are implemented in MATLAB V. 9.7.012 (R2021A) with the commercial solver Gurobi V 10.0 on a PC with an i7, 3.60 GHz Intel™ processor, and 8 GB of memory.

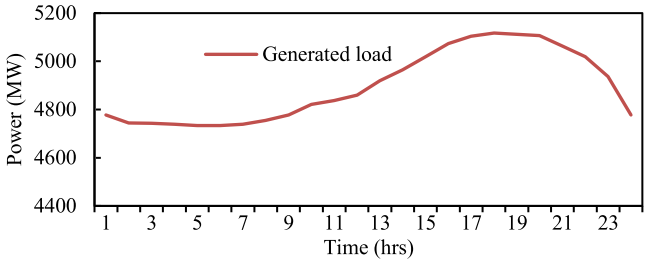

FIGURE 4. **Generated load scenario (118-bus) rescaled for each tested system.**

A. DATA AVAILABILITY AND PREPARATION

The proposed method is applied to a wide range of system sizes using massive data for training. The data for the benchmark systems are collected from the Matpower Repository [32] and we have created the Ontario Zonal system from the available online available data of the IESO, the Ontario system operator [33], and the Ontario Zonal Network is shown in [34]. The load profiles are created to fulfill all possible load variations in the systems using two realistic factors: the time-wise power distribution factor of total peak load, β_t and bus-wise load power distribution factor, β_i. We examined hourly demand data from IESO to create a realistic distribution for these factors [33]. β_t presents the load change along the 24-hour. By investigation, we found β_t is changing from 0.8 to 0.93 of the day peak, and the peak system load in the original cases is always below 71% of the total generation capacity. Thus, $\beta_i = \hat{d}_i/\hat{L}_i$, where $\hat{d}_i$ and $\hat{L}_i$ present the average load and the peak load at bus i respectively.

The load profile for each scenario is defined as in (58). Fig. 4 presents one of the generated load scenarios for 118-bus systems. Loads of other systems are rescaled and generated by the same concept.

$$d_{ti}^{s} = \left(\sum_{n}^{NG} \underline{PG_n} + \left(\sum_{n}^{NG} \overline{PG_n} - \underline{PG_n}\right) \cdot \frac{s \cdot 0.71}{NS}\right) \cdot \beta_t \cdot \beta_i \tag{58}$$

where s is the scenario number and NS is the total number of load scenarios.

The cost of generation is considered to meet the variations in fuel prices. To illustrate how the price is created, first, the generating units are classified into three categories: a) base generation (i.e., hydro and nuclear); b) middle-peak units; and c) peak units. We assumed the base generation capacity is 60% of the largest generation units, then 20% for each middle-peak and peak units. Second, we considered the prices mentioned in [32] and [33] as the average for each tested case. Then, we assumed the base generation cost is the average, and the prices of the middle-peak and peak units are changing. The total generation cost change for the load scenarios of the 118-bus is plotted as a probability distribution function as shown in Fig. 5 and rescaled for the other systems.

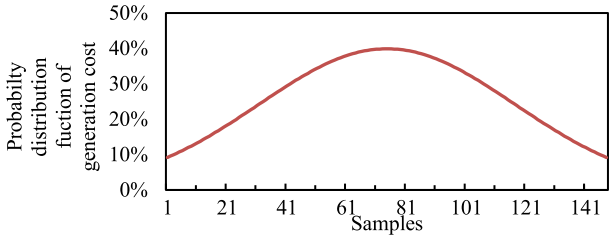

FIGURE 5. Probability distribution function for the generation cost of 118-bus rescaled for each tested system.

TABLE 1. Tested systems data, ML training parameters, and accuracy.

System	IEEE 14-bus	IEEE 118-bus	IEEE 300-bus	Ontario System	Polish 2383-bus	Polish 3012-bus
# Generators	5	54	69	131	327	502
# Transmission Lines	20	186	411	19	2,896	3,572
# Of Segments NM	3	3	3	1	1	1
# NN layers / # neurons	3/60	3/80	3/80	3/80	3/80	3/80
ML input vector $[D, C]$ size	351	2,994	7,407	491	57,519	72,790
ML output vector $\boldsymbol{ug}$ size	120	1,296	1,656	3,144	7,848	12,048
Number of Samples	500	148	299	208	198	185
# epochs (x1000)/ Perf. goal	10.5/0	10.5/0	10.5/0	10.5/0	10.5/0	10.5/0
Min performance gradient	1e-60	1e-60	1e-60	1e-60	1e-60	1e-60
Max validation failures	3000	3000	3000	3000	3000	3000
Learning rate	0.01	0.01	0.01	0.01	0.01	0.01
Weight increment	1.2	1.2	1.2	1.2	1.2	1.2
ML Training Accuracy (%)	99.9	99.8	95	98.7	97.2	97.4
ML Testing Accuracy (%)	99	97.8	93	98	96.4	96.2
ML training time (s)	350	1,300	1,100	850	3,500	4,000

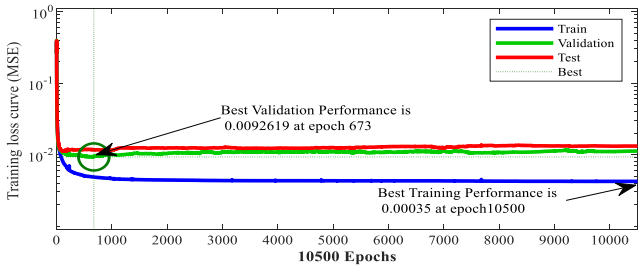

FIGURE 6. ANN Training performance loss curve for Ontario system.

B. ML TRAINING MODEL AND DEVELOPMENT OF ML TGS (α) - OFF-LINE ALGORITHM #1

To illustrate the training process, for the Polish 2383-bus hourly example, the ML model is trained using 198 bus-wise demand data, forming the hourly bus vector $\boldsymbol{D} = [57, 192 \times 198]$, and the generation prices $\boldsymbol{C} = [327 \times 198]$. Both are used as the ML input vector, with the generation commitment schedules $\boldsymbol{ug} = [7, 848 \times 198]$ used as a target vector. The target vector $\boldsymbol{ug}$ is created using the solutions of MILP-SCUC formulation in (1) – (23) for load scenarios $\boldsymbol{D}$ and the generation prices $\boldsymbol{C}$. For all systems, the stop time is set as 1800 s, with different optimality gap settings relative to the systems' scale.

The ML training process is performed similarly for all other systems. ML model tuning parameters and training accuracy results are reported in Table 1. Fig. 6 shows the ANN training performance loss curve for the Ontario system as a training sample result. The training data set is divided into 70% for training and 15% each for testing and validation.

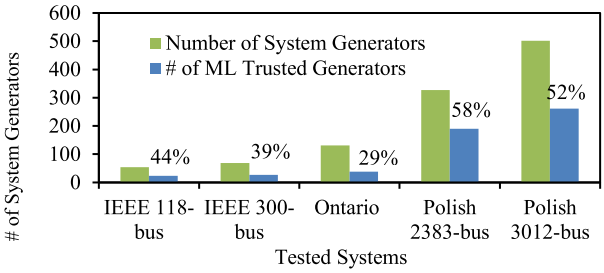

FIGURE 7. Size of ML trusted generators set and the intelligence factor for the tested systems.

The input-output ratio can affect the training time of the ML model. A larger input-output ratio can lead to longer training times because the network has more weights and biases to adjust during training, and each weight and bias must be updated for the network to learn. Additionally, a larger input-output ratio can also increase the complexity of the network, which can also contribute to longer training times as shown in Table 1.

The development of the power network, the load increase, price changes of fuels, and the units all will be captured in this training stage. In addition, we recommend continuous offline training once a month or every couple of weeks if further changes are happening in the power network.

Inspired by the well-trained ML model for each tested system in Stage #1 of Algorithm #1, the TGS α is created as per Stage #2 of Algorithm #1. TGS provides a set of generators that is perfectly determined by a trained ML model without errors. For the IEEE-14 bus system, the ML model perfectly predicts the behavior of three generators out of the total five generators (60%). This percentage is termed as the Intelligence Generators

Factor (IGF) (η). As illustrated in Fig. 7, IGF is higher for systems with a larger number of generators. For Polish 2383-bus, and Polish 3012-bus systems, IGF (η) is consistently larger than 52%.

C. TESTING THE MILP-SCUC FORMULATION WITH MLVR METHOD – DAY-AHEAD ALGORITHM 2

The ML reduced variable map is created using ML indicator sets, which are presented in Stage #1 of Algorithm #2. All indicators sets have passed the feasibility condition (35) and have then been used to shrink the MILP-SCUC formulation as in (57). In the following Table 2, randomly selected data sets are used to study the effectiveness of the proposed method. In addition, the effect of the MLVR method on the problem size is tabulated. Consequently, these ML indicators sets decrease the number of binary variables and marginally decrease the continuous variables of the turned-off units for each tested system. As a result, the mixed-binary and binary constraints are decreased by significant percentages

TABLE 2. Effect of MLVR method on the MILP-SCUC problem size.

	IEEE 14-bus	IEEE 118-bus	IEEE 300-bus	Ontario system	Polish 2383-bus	Polish 3012-bus
Testing data size	500	148	299	208	198	185
TGS (%)	3 (60%)	24 (44%)	27 (39%)	38 (29%)	190 (58%)	261 (52%)
# ML ON (OFF) reduced variables	72 (0)	216 (360)	192 (456)	888 (24)	504 (4,056)	1.248 (5,016)
# ML Startup (Shutdown) variables	72 (72)	576 (576)	648 (648)	912 (912)	4,560 (4,560)	6,264 (6,264)
# Binary Variables	144 (-60%)	2,160 (-44%)	3,024 (-39.1%)	6,696 (-29%)	9,864 (-58.1%)	17,352 (-52%)
# Cont. Variables	840 (0%)	6,744 (-16.1%)	12,480 (-9.9%)	6,624 (-0.36%)	68,856 (-5.56%)	91,392 (-5.2%)
# Continuous Constraints	2,016 (21.7%)	14,976 (-2.3%)	30,048 (-2.8%)	10,104 (34.9%)	193,008 (-3.3%)	248,136 (-2.5%)
# Mixed-binary Constraints	216 (-40%)	2,376 (-38.9%)	3,216 (-35.3%)	7,584 (-19.6%)	10,368 (-56%)	18,600 (-48.5%)
# Binary Constraints	263 (-68.9%)	4,470 (-51%)	6,423 (-44.9%)	14,767 (-33.3%)	18,403 (-60%)	34,204 (-59.7%)

for all systems, while the continuous constraints are increased in some systems, depending on the ratios between reduced on/off statuses as in Table 2.

For the 14-bus system using the MLVR method, the MILP-SCUC binary constraints are reduced by 68.9%, and the binary variables are reduced by 60%. For the IEEE 118-bus system, the ML indicators' sets decrease the binary variables by 44%. As a result, a significant shrinking of the MILP-SCUC problem size is achieved and the problem complexity is improved by reducing the mixed-binary constraints and binary constraints by 38.9% and 51% respectively.

For the IEEE 300-bus system, the ML model perfectly predicts 39% of the output; thus, the mixed-binary and binary constraints decrease by −35.3% and −44.9% respectively.

For the Ontario and Polish systems, the mixed-binary constraints decrease by 19.6%, 56%, and 48.5% respectively. Clearly, when the ML model has a high intelligence factor, MLVR-MILP-SCUC is extremely efficient. Therefore, the MLVR-MILP-SCUC complexity is significantly reduced.

As illustrated in Table 3, the effectiveness of the proposed method is demonstrated by comparing its average solution quality (average of cost, computation time, and solver optimality gap) for all random tested data with that of the conventional MILP-SCUC method (1) – (23) mentioned in [1], [19], and [27]. In Stage #2 of Algorithm #2, the MLVR-MILP-SCUC formulation (57) is shrunk by the ML variables reduction method and solved by Gurobi, a MILP commercial solver.

IEEE systems results of the MLVR-MILP-SCUC method present significant benefits in the solution quality without degradation in the average best-known mixed binary solution obtained by the MILP-SCUC method and equal 60 %, 89%, and 60% reductions in the solution time, respectively. From the average cost perspective, the MLVR-MILP-SCUC method introduces the same best-known solution of the MILP-SCUC method with the same average 0% converged OG for IEEE 14-bus and 118-bus and 0.0001% converged OG for IEEE 300-bus.

TABLE 3. Performance of the proposed MLVR-MILP-SCUC method with comparisons with the MILP-SCUC Method [27]*.

	IEEE 14-bus	IEEE 118-bus	IEEE 300-bus	Ontario system	Polish 2383-bus	Polish 3012-bus
Search Space	2^{120}	2^{1296}	2^{1656}	2^{3144}	2^{7848}	2^{12048}
Search Space Change	2^{48}	2^{720}	2^{1008}	2^{2232}	2^{3288}	2^{5784}
Avg. Solver Time (s)	0.13	5.4	100	2.94	300	575
Time reduction	-60%	-89%	-60%	-98%	-48%	-49%
OG Setting (%)	0.0	0.0	0.0001	0.0	0.01	0.03
Avg. converged OG MILP-SCUC (%)	0.0	0.0	0.0001	0.0	0.01	0.024
Avg. converged OG MLVR-SCUC (%)	0.0	0.0	0.0001	0.0	0.007	0.022
Avg. Cost of MILP-SCUC (M$)	1.47	1.201	9.898948	28.5714	27.168953	67.460026
Avg. Cost of MLVR-SCUC (M$)	1.47	1.201	9.898948	28.5714	27.167587	67.459020
Avg. Reduction in Cost (%)	0.00%	0.00%	0.00%	0.00%	0.005%	0.0014%
Avg. benefits ($)	0	0	0	0	1366	1006

Note (*): Objective function benefits are derived from numerical efficiency of the MILP solver, due to reduced size of the proposed MLVR-SCUC method.

Table 3 illustrates the benefits of using the MLVR-MILP-SCUC method (57) over the MILP-SCUC method (1) – (23). Results demonstrate the effectiveness of the proposed method in terms of the computation complexity and economic benefits. In terms of the computation time for Ontario, the average computation time is decreased by 98% with the same average optimal cost. For the Polish 2383-bus and 3012-bus systems, the ML model predicts 58% and 52% of the generation schedules respectively, leading to a significant reduction in binary variables, binary constraints, and MB constraints, and therefore a reduction in the search space. As a result, the average computation time is improved to 1.92 and 1.96 times faster respectively. Regarding the effect of the MLVR-MILP-SCUC method on system optimality, as shown in Table 3, the average optimality gap is improved to 0.007% and 0.022% respectively, with a very minor average operating cost change of −0.005% and −0.0014% respectively.

The demonstrated results effectively illustrate the superiority of using the MLVR method to reformulate the MILP-SCUC, as the MLVR-MILP-SCUC method offers a much-improved solution quality for all tested systems, from small to large scale systems. As shown from the series of the tested systems, from small scale to large scale, the benefits of using the MLVR-MILP-SCUC method are not limited to the improvements in computation time. There are also improvements in the optimality gap which are very essential for the ISOs to gain economic advantage for all stakeholders.

VI. SOLUTION QUALITY FACTORS AND COMPARISON

A. INTELLIGENCE GENERATORS AND SOLUTION TIME FACTORS

It is important to inspect the impact of ML indicator sets. They update the MILP-SCUC formulation by reducing the

binary variables and system constraints. Therefore, the solution space is decreased, and solution complexity improves without creating infeasibility or deterioration of the best-known mixed integer solution. Energized by the results presented in Table 3, we create Fig. 8 to show the correlation between the IGF of the MLVR-MILP-SCUC method and the computation time for small-scale systems and large-scale systems. As shown, the IGF leads to more than 60% time reduction for small-scale systems and more than 48% for large systems. Thus, the increase of IGF leads to a greater computation time reduction, establishing a positive correlation.

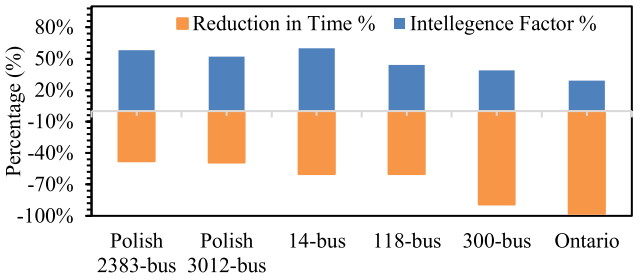

FIGURE 8. Correlation between IGF of MLVR-MILP-SCUC method and computation time reduction.

B. OPTIMALITY GAP, SOLUTION TIME, AND OPTIMAL SOLUTION

Practically, for a large-scale system, it is hard to obtain a MILP-SCUC solution with a zero-optimality gap for all days, and it requires much computational burden. Solvers' OG might reach 0%, because the MILP solvers can update the lower bound based on LP relaxation at branching nodes, resulting in a higher lower bound for minimization problems. However, the actual OG is calculated by using the lower bound of the original MILP formulation. Therefore, it is higher than the solver OG.

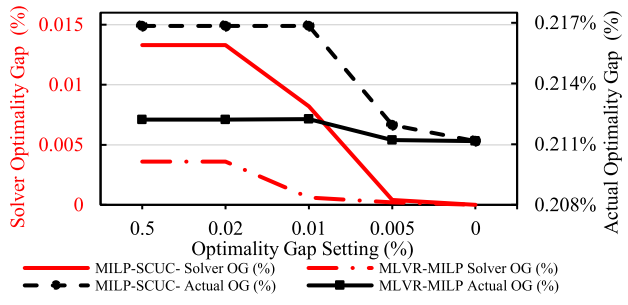

FIGURE 9. Optimality gap convergence evolution for Polish 2383-bus system using MLVR-MILP-SCUC and MILP-SCUC methods.

Fig. 9 shows the convergence effect of the solver and actual OG in the case of using the MLVR-MILP-SCUC and MILP-SCUC methods for a particular scenario for the Polish 2383-bus system, considering a large enough maximum running time (8000s). As shown, at a 0.5% OG setting, the MILP-SCUC method converges at a larger solver-OG (0.014%) and actual OG (0.217%). However, for the proposed model, the system converges early at a lower solver-OG (0.0036%) and actual-OG (0.212%). Due to that, there is a negative marginal change in the operating cost of (0.00045%), as shown in Fig. 10, with a time reduction of 69%. In the case of a 0% OG setting, both methods converge with no cost difference and a time reduction of 80% for the proposed method.

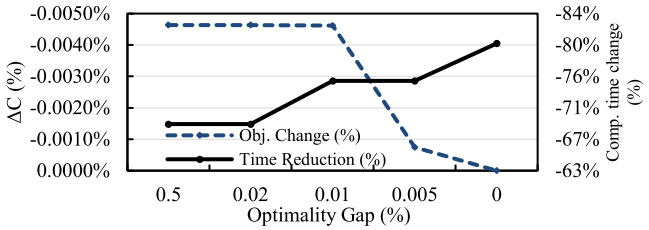

FIGURE 10. Solution quality evolution of the proposed MLVR-MILP-SCUC and conventional SCUC methods for Polish 2383-bus system.

Clearly, from Fig. 9 and Fig. 10, the proposed method provides only a very marginal cost reduction for large-scale tested systems that are solved at greater than 0% OG, and it provides time reductions for all OG cases as a result of the numerical efficiency of the MILP solver.

Fig. 11 illustrates the change in the operating cost for the large Polish 2383-bus for all tested scenarios at a practical OG value of 0.01%. It shows that the proposed method provides a very marginally lower solution for 25% of the tested data and the same solution as the MILP-SCUC method for 72% of the tested data, while 3% of the tested data provides a very marginal increase of 0.00025%. The reason for a very marginally lower or higher solution is attributed to the computational aspects of the solver in such large problems.

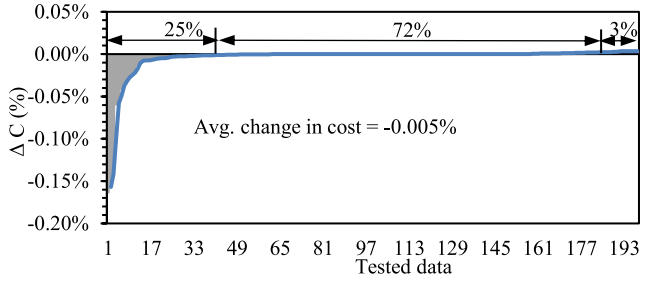

FIGURE 11. Operating cost change of the proposed MLVR-MILP-SCUC for all tested data of Polish 2383-bus System at OG = 0.01%.

C. COMPARISON WITH RECENT ML-SCUC WORKS

To show the effectiveness of the proposed method in comparison with recent ML-SCUC methods, we compare our methodology with the most related works in [24] and [27]. For a fair comparison, the hourly net demand at each bus has been used as input for the ML model. Table 4 provides these results. In this comparison, we used 50 random samples for

testing the IEEE 118-bus and Polish 3012-bus systems. It is noted that [24] and [27] provide results that are infeasible for a few samples. Considering the execution time for infeasible samples as 1500 s, the average time is computed and compared. Obviously, it is seen that the proposed method provides superior performance in terms of execution time and consistently provides a feasible and optimal solution, in contrast to [24] and [27] which remain infeasible sometimes.

To sum up, the proposed MLVR method provides a reduction in the variables map, and therefore the problem size is reduced. This effect is reflected in the improvement of the system search space and solution quality compared with the MILP-SCUC method. The benefits of the proposed method for all tested data are tabulated in Table 3. The tested systems' results demonstrate that the MLVR method can play a vital role in enhancing the MILP-SCUC formulation to achieve significant improvements in solution quality in terms of the computation time reduction and lower optimality gap that results in lower optimal operation costs and hence, economic benefits.

TABLE 4. Comparison with recent ML-SCUC methodologies.

ML-SCUC Methods	Avg. Solution Time	Reduction in Cost	Prediction Error	Variables Reduction	Infeasibility Ratio	Feasible Samples
118-bus						
[27]	-47%*	0.00%	3.47%	100%	64%	18
[24]	-61%*	0.00%	0.045%	74%	0.02%	49
Proposed Method	-73%	0.00%	0.00%	44%	0.00%	50
3012-system						
[27]	-	-	3.25%	100%	100%	0
[24]	- 28%*	0.00 %	0.66%	86%	0.04 %	48
Proposed Method	- 49%	- 0.01%	0.00%	52%	0.00%	50

Note (*): Solution time for infeasible samples is considered as 1500 s.

D. PRACTICAL IMPLICATIONS AND CONTINUOUS OFFLINE TRAINING

The ML training process could be performed continually (i.e., monthly or weekly) to cover the changes in the system topology, prices, and generation outages. If the fuel price of a generator changes and has been seen before by the ANN model, the ML should be able to predict that change. Finally, the proposed method is independent of MILP-UC formulation. It can work with any MILP-UC formulation.

VII. CONCLUSION

In this paper, a novel machine learning variables reduction method is created to enhance the MILP-SCUC formulation. The proposed method is implemented in two off-line stages in Algorithm #1. In the first stage, the ML model is trained to map the 24-hourly generation decisions with 24-hourly bus-wise net demands and the piecewise linear generation cost curve data. Using Algorithm #1 stage 2, the well-trained ML model is utilized to accurately predict outputs for a set of intelligence generators. This set is defined as TGS (α), and the percentage of intelligence generators (IGF) is used as a performance factor for the effectiveness of the proposed method. For the tested cases IEEE-14, IEEE-118, 300-bus, Ontario, Polish-2383, and Polish 3012-bus systems, the IGF equals 60%, 44%, 39%, 29%, 58%, and 52% respectively. In real-time, for a day-ahead bus hourly load and the generation prices, the corresponding ML output is obtained using the ML model created in Algorithm #1. Using Algorithm #2, The ML indicators are feasibly created for the TGS (28) – (34). These indicators sets are used to reduce the number of binary variables of the MILP-SCUC problem, also reducing the mixed-binary constraints by significant values. As a result, the MILP-SCUC formulation is shrunk and updated to form the MLVR-MILP-SCUC formulation (57) that leads to a reduction in the search space and an improvement in computation quality. The proposed method provides a faster computation time without degrading the optimal solution. As shown in the results section, the reductions in solution times are 60%, 89%, 60%, 98%, 48%, and 49% for the IEEE tested systems, Ontario, Polish-2383, and Polish-3012 bus systems, respectively. The results conclusively validate the benefits of the MLVR-MILP-SCUC method. In addition, it provides a trusted tool for ISOs to solve the MILP-SCUC faster with a lower optimality gap within the limited time in real-time use, leading to lower operation costs.

REFERENCES

[1] Y. Huang, P. M. Pardalos, and Q. P. Zheng, *Electrical Power Unit Commitment: Deterministic and Two-Stage Stochastic Programming Models and Algorithms*. Boston, MA, USA: Springer, 2017, doi: 10.1007/978-1-4939-6768-1.

[2] Y. Chen, F. Wang, Y. Ma, and Y. Yao, "A distributed framework for solving and benchmarking security constrained unit commitment with warm start," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 711–720, Jan. 2020, doi: 10.1109/TPWRS.2019.2930706.

[3] 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, doi: 10.1109/TPWRS.2016.2530811.

[4] *Part 9.0: Day-Ahead Commitment Process Overview*. Accessed: Jan. 19, 2022. [Online]. Available: https://www.ieso.ca/-/media/Files/IESO/cument-Library/Market-Rules-and-Manuals-Library/market-manuals/y-ahead-commitment/MM9-dacp-manual.pdf

[5] (Oct. 1, 2020). *ISO New England Manual For Market Operations M-11. Revision 60*. Accessed: Jan. 19, 2022. [Online]. Available: https://www.iso-ne.com/static-assets/documents/2020/10/m_11_revision_60_effective_10_1_2020.pdf

[6] (Apr. 2022). *Energy Market Introduction*. Accessed: Jan. 19, 2022. [Online]. Available: https://www.iso-ne.com/static-assets/documents/22/05/20220425-01-wem101-energy-market-introduction-PRINT.pdf

[7] X. Li, Q. Zhai, J. Zhou, and X. Guan, "A variable reduction method for large-scale unit commitment," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 261–272, Jan. 2020, doi: 10.1109/TPWRS.2019.2930571.

[8] X. Sun, P. B. Luh, M. A. Bragin, Y. Chen, J. Wan, and F. Wang, "A novel decomposition and coordination approach for large day-ahead unit commitment with combined cycle units," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5297–5308, Sep. 2018, doi: 10.1109/TPWRS.2018.2808272.

[9] B. Saravanan, S. Das, S. Sikri, and D. P. Kothari, "A solution to the unit commitment problem—A review," *Frontiers Energy*, vol. 7, no. 2, pp. 223–236, Jun. 2013, doi: 10.1007/s11708-013-0240-3.

[10] B. Knueven, J. Ostrowski, and J.-P. Watson, "On mixed-integer programming formulations for the unit commitment problem," *INFORMS J. Comput.*, vol. 32, pp. 855–1186, Jun. 2020, doi: 10.1287/ijoc.2019.0944.

[11] A. Borghetti, C. D'Ambrosio, A. Lodi, and S. Martello, "An MILP approach for short-term hydro scheduling and unit commitment with head-dependent reservoir," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1115–1124, Aug. 2008, doi: 10.1109/TPWRS.2008.926704.

[12] Y. Chen, F. Liu, B. Liu, W. Wei, and S. Mei, "An efficient MILP approximation for the hydro-thermal unit commitment," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3318–3319, Jul. 2016, doi: 10.1109/TPWRS.2015.2479397.
[13] C. Cheng, J. Wang, and X. Wu, "Hydro unit commitment with a head-sensitive reservoir and multiple vibration zones using MILP," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4842–4852, Nov. 2016, doi: 10.1109/TPWRS.2016.2522469.
[14] Z. Zhao, C. Cheng, S. Liao, Y. Li, and Q. Lu, "A MILP based framework for the hydro unit commitment considering irregular forbidden zone related constraints," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 1819–1832, May 2021, doi: 10.1109/TPWRS.2020.3028480.
[15] L. S. M. Guedes, P. de Mendonça Maia, A. C. Lisboa, D. A. G. Vieira, and R. R. Saldanha, "A unit commitment algorithm and a compact MILP model for short-term hydro-power generation scheduling," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3381–3390, Sep. 2017, doi: 10.1109/TPWRS.2016.2641390.
[16] L. Wu, "A tighter piecewise linear approximation of quadratic cost curves for unit commitment problems," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2581–2583, Nov. 2011, doi: 10.1109/TPWRS.2011.2148370.
[17] G. Morales-Espana, J. M. Latorre, and A. Ramos, "Tight and compact MILP formulation of start-up and shut-down ramping in unit commitment," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1288–1296, May 2013, doi: 10.1109/TPWRS.2012.2222938.
[18] G. Morales-Espana, J. M. Latorre, and A. Ramos, "Tight and compact MILP formulation for the thermal unit commitment problem," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4897–4908, Nov. 2013, doi: 10.1109/TPWRS.2013.2251373.
[19] B. Yan et al., "A systematic formulation tightening approach for unit commitment problems," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 782–794, Jan. 2020, doi: 10.1109/TPWRS.2019.2935003.
[20] T. Zhao, H. Zhang, X. Liu, S. Yao, and P. Wang, "Resilient unit commitment for day-ahead market considering probabilistic impacts of hurricanes," 2019, *arXiv:1911.07599*.
[21] S. Pineda, J. M. Morales, and A. Jimenez-Cordero, "Data-driven screening of network constraints for unit commitment," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 3695–3705, Sep. 2020, doi: 10.1109/TPWRS.2020.2980212.
[22] Y. Yang, X. Lu, and L. Wu, "Integrated data-driven framework for fast SCUC calculation," *IET Gener., Transmiss. Distribution*, vol. 14, no. 24, pp. 5728–5738, Dec. 2020, doi: 10.1049/iet-gtd.2020.0823.
[23] N. Yang et al., "Intelligent data-driven decision-making method for dynamic multi-sequence: An E-Seq2Seq based SCUC expert system," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3126–3137, May 2021, doi: 10.1109/TII.2021.3107406.
[24] Á. 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, Oct. 2020, doi: 10.1287/ijoc.2020.0976.
[25] F. Mohammadi, M. Sahraei-Ardakani, D. N. Trakas, and N. D. Hatziargyriou, "Machine learning assisted stochastic unit commitment: A feasibility study," in *Proc. 52nd North Amer. Power Symp. (NAPS)*, Apr. 2021, pp. 1–6, doi: 10.1109/NAPS50074.2021.9449789.
[26] F. Mohammadi, M. Sahraei-Ardakani, D. Trakas, and N. Hatziargyriou, "Machine learning assisted stochastic unit commitment during hurricanes with predictable line outages," *IEEE Trans. Power Syst.*, vol. 36, no. 6, pp. 5131–5142, Nov. 2021, doi: 10.1109/TPWRS.2021.3069443.
[27] T. Wu, Y.-J. Angela Zhang, and S. Wang, "Deep learning to optimize: Security-constrained unit commitment with uncertain wind power generation and BESSs," *IEEE Trans. Sustain. Energy*, vol. 13, no. 1, pp. 231–240, Jan. 2022, doi: 10.1109/TSTE.2021.3107848.
[28] (2022). *Gurobi Optimizer Version 10.0*. https://www.gurobi.com/
[29] *NPCC Reserve Task Force on Coordination of Operations Revision Review Record, NPCC Northeast Power Coordinating Council (NPCC) Regional Reliability Reference Directory # 5 Reserve*. New York, NY, USA. Accessed: Feb. 13, 2022. [Online]. Available: https://www.npcc.org/content/docs/public/program-areas/standards-and-criteria/regional-criteria/directories/directory-5-reserve-20200426.pdf
[30] *MISO Market Report Archives*. Accessed: Feb. 13, 2022. [Online]. Available: https://www.misoenergy.org/markets-and-operations/real-time–market-data/market-reports/market-reportarchives/#t=10&p=0&s=MarketReportPublished&sd=desc
[31] M. A. Riedmiller and H. A. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1, Apr. 1993, pp. 586–591, doi: 10.1109/ICNN.1993.298623.
[32] *Matpower Repository: IEEE 14-Bus, 118-Bus, 300-Bus and Polish 2383-Bus and 3012-Bus Systems Datasets*. Accessed: Feb. 13, 2022. [Online]. Available: https://matpower.org/docs/ref/matpower5.0/nu5.0.html
[33] I*ESO Public Reports: Zonal System Data*. IESO. Toronto, ON, Canada. Accessed: Feb. 13, 2022. [Online]. Available: http://reports.ieso.ca/public/
[34] J. Ma and B. Venkatesh, "Integrating net benefits test for demand response into optimal power flow formulation," *IEEE Trans. Power Syst.*, vol. 36, no. 2, pp. 1362–1372, Mar. 2021, doi: 10.1109/TPWRS.2020.3020856.

MOHAMED IBRAHIM ABDELAZIZ SHEKEEW (Student Member, IEEE) was born in Egypt. He received the B.Sc. and M.Sc. degrees (Hons.) in electrical engineering from Cairo University, Cairo, Egypt, in 2012 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Centre for Urban Energy, Toronto Metropolitan University, Toronto, ON, Canada. His research interests include power system optimization, energy storage, and renewable energy resources integration.

BALA VENKATESH (Senior Member, IEEE) received the Ph.D. degree from Anna University, India, in 2000. He is currently a Professor and the Academic Director of the Centre for Urban Energy, Toronto Metropolitan University, Toronto, Canada. His research interests include power system analysis and optimization. He is a registered Professional Engineer in the province of Ontario, Canada.