Proposal: Topology-Aware Unit Commitment via Spatio-Temporal Graph Neural Networks

sleeping_pizza_

February 10, 2026

1. Abstract

Recent advancements by Venkatesh et al. (2025) successfully introduced a "Feasibility-Guaranteed" approach using a hybrid CNN-BiLSTM model coupled with fuzzy optimization to solve the UC problem, their methodology relies on CNNs for feature extraction. This proposal identifies a critical limitation in that approach: CNNs inherently assume fixed Euclidean data structures, rendering them brittle to topological changes such as $N-1$ contingencies (line outages).

This research proposes a Spatio-Temporal Graph Neural Network (ST-GNN) framework to (hopefully)supersede the CNN-based architecture. By representing the power grid as a dynamic graph, where buses are nodes and transmission lines are edges, the proposed model explicitly preserves spatial fidelity. This allows for zero-shot generalization to $N-1$ contingencies, a capability absent in the baseline CNN approach. This spatial module will be coupled with a sequence modeling layer (benchmarking Long Short-Term Memory networks against Transformer architectures) to capture temporal load dynamics. Crucially, the system retains the proven **Fuzzy Optimization** layer to integrate Machine Learning predictions as non-binding constraints, guaranteeing 100% feasibility while targeting superior optimality gaps and robustness to topology shifts.

2. Proposed Methodology

The proposed architecture consists of three distinct stages:

Stage 1: Topology-Adaptive Spatial Encoding (ST-GNN)

Unlike the CNN approach, which treats the system as an image, we will utilize Graph Convolutional Networks (GCNs) or Graph Attention Networks (GATs).

- *Mechanism:* The GNN aggregates features from neighboring buses. Crucially, the adjacency matrix A will be dynamic, allowing the model to intake "$N-1$" topologies directly during inference.

Stage 2: Temporal Sequence Modeling (Comparative Analysis)

This research will conduct a comparative analysis to determine the optimal temporal architecture for identifying ramping constraints.

- **Candidate A: BiLSTMs** (Baseline architecture).
- **Candidate B: Transformers** (Attention-based, potentially superior for capturing long-range dependencies over the 24-hour horizon).

Stage 3: Feasibility Assurance via Fuzzy MILP

The binary outputs (ON/OFF decisions) from the ST-GNN will be integrated into the MILP formulation using the **Fuzzy Optimization** framework established by Venkatesh et al. This ensures that even if the GNN prediction violates a specific hard constraint, the solver treats the prediction as a "soft" guidance target rather than a rigid command, guaranteeing a feasible solution is always found.

3. Responses to peer criticisms

0.1 1. GNNs are computationally expensive

While GNN training is intensive, online inference is nearly instantaneous, and the model's ability to exploit graph sparsity ensures it scales efficiently for large-scale systems like the Polish 2383-bus.

0.2 2. Generation of (N-1) contingency data

Training data will be generated by augmenting the base dataset through systematically removing transmission lines and re-solving the standard MILP-UC to create a comprehensive topology-variant target set.

0.3 3. Evaluation metrics

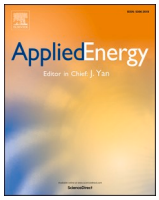
Performance will be measured by the Suboptimality Index (SOI) and computation time reduction, specifically benchmarking "zero-shot" feasibility and schedule similarity on unseen $N-1$ topologies.

Applied Energy 379 (2025) 124923

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy



Feasibility-guaranteed machine learning unit commitment: Fuzzy Optimization approaches

Bala Venkatesh [a,b,*], Mohamed Ibrahim Abdelaziz Shekeew [a,b,d], Jessie Ma [c]

[a] Department of Electrical, Computer, and Biomedical Engineering, Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada
[b] Centre for Urban Energy, Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada
[c] The Department of Systems Design Engineering and the School of Environment, Enterprise, and Development, University of Waterloo, Waterloo, ON, Canada
[d] Electrical Power Engineering Department, Faculty of Engineering, Cairo University, Giza, Egypt

HIGHLIGHTS

- A hybrid CNN and BiLSTM model was created to provide the unit commitment schedules.
- ML predictions modeled as non-binding fuzzy constraints enhancing MILP-UC formulation.
- Fuzzy constraints improved ML decision utilization, ensuring 100 % feasibility.

ARTICLE INFO

Keywords:
Fuzzy
Machine learning
mixed-integer linear programs
Power systems
Unit commitment

ABSTRACT

The unit commitment (UC) problem is solved several times daily in a limited amount of time and is commonly formulated using mixed-integer linear programs (MILP). However, solution time for MILP formulation increases exponentially with the number of binary variables required. To address this, machine learning (ML) models have been attempted with limited success as they cannot be trained for all scenarios, whereby they may contain false predictions leading to infeasibility, hindering their practical applicability. To overcome these issues, we first propose a hybrid deep learning model comprising a convolutional neural network (CNN) and bidirectional long-short-term memory (BiLSTM) to predict the UC decisions. Second, we incorporate these predictions as non-binding fuzzy constraints, enhancing the traditional UC model and creating an ML-fuzzy UC model. Two implementations of non-binding fuzzy constraints are studied. The first develops each ML decision variable as a separate fuzzy set, while the second creates one fuzzy set per hour, considering all decisions within. Introducing ML-UC decisions as non-binding fuzzy constraints ensures the ML-fuzzy UC model has a feasible solution if the basic MILP-UC problem does, while leveraging ML predictions. Moreover, the proposed model benefits from a reduced solution space, leading to substantial reductions in computing time. Results on IEEE 118-bus and Polish 2383-bus systems demonstrate 92 % and 89 % computation time reductions for both systems, respectively and achieve 100 % feasibility for both seen and unseen datasets when the basic MILP-UC problem has a feasible solution.

1. Introduction

Unit commitment (UC) is a fundamental optimization tool for power systems planning and operation. UC is solved by independent system operators (ISOs) to determine the best generation schedule that meets demand bids, the power grid security constraints, power reserve, system reliability requirements, etc. In the day-ahead deregulated energy market, the UC is utilized several times a day before electricity market settlement to ensure the safe operation of the power network and achieve the best economic outcome for market participants [1]. Mathematically, the UC problem is a large-scale mixed-integer (nonconvex) optimization problem involving numerous binary variables and continuous variables, as well as a series of equality and inequality constraints.

The UC problem has been solved using various methods over the years. In the early days of electricity markets, the UC process predominantly relied on the Lagrangian relaxation (LR) method for market

* Corresponding author at: Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada.
E-mail address: bala@torontomu.ca (B. Venkatesh).

https://doi.org/10.1016/j.apenergy.2024.124923
Received 1 March 2024; Received in revised form 23 September 2024; Accepted 11 November 2024
Available online 25 November 2024

Nomenclature

Sets

Ω^{G}	Generation units $(g \in \Omega^{G}, \Omega^{G} = \{1,\ldots,G\})$
Ω^{w}	A generation set for weather-dependent sources i.e., Solar, wind, and biomass $(j \in \Omega^{w}, \Omega^{w} = \{1,\ldots,G^{W}\})$
Ω^{N}	System nodes (busses), (indices, $n,i \in \Omega^{N}$), $\Omega^{N} = \{1,\ldots,N\}$
Ω^{M}	Fuel cost piece-wise segments, $m \in \Omega^{M}, \Omega^{M} = \{1,\ldots,M\}$
Ω^{L}	Power system transmission lines, $l \in \Omega^{L}$
Ω^{T}	Dispatch horizon, t $\in \Omega^{T}, \Omega^{T} = \{1,\ldots,\mathrm{T}\}$
Ω^{T-}	Sub-horizon set $t \in \{2,\ldots,T-1\}$
Λ_n^{G}	Generators connected to node n
Λ_n^{w}	Weather-dependent sources at node n
Ω^{S}	Indices decision set of ML output. $t,g \in \Omega^{S}, \Omega^{S} = \{1,..,TG\}$
$\Upsilon^{u}/\Upsilon^{d}$	Indices set for ML ON/OFF decisions
$\mathscr{O}^{d}$	Vector of ones for ML-OFF decisions with the same size of Υ^{d}. $t,g \in \mathscr{O}^{d}, \mathscr{O}^{d} = [1,1,..]^{1\times YD}$; where YD is the size of Υ^{d}
$\mathscr{S}$	Generated data samples, $s \in \mathscr{S}$
$u_g^{u}(t)$	Individual sets for ML ON decisions
$UG^{u}(t)$	Accumulated sets for ML ON decisions at time t
$u_g^{d}(t)$	Individual sets for ML OFF decisions
$UG^{d}(t)$	Accumulated sets for ML OFF decisions at time t

Parameters

P_g^{U}, P_g^{D}	Upper and lower capacities of generator g
$P_{g,m}^{U}$	Piece-wise generation limit
F_l^{U}	Power flow capacity of transmission line l
c_g^{v}, c_g^{y}	Costs of startup/shutdown for generator g
$c_{g,m}^{p}, c_g^{z}$	Coefficients of piece-wise generation costs
c_g^{r}, c_j^{w}	Costs of reserve and weather-dependent units
X_l	The inductance of transmission line l
γ	Percentile of the online spinning reserve requirement
$d_n(t)$	Per hour load at node n
B	Susceptance matrix for the power system network
R_g^{10}	10-min spinning reserve
R_g^{U}/R_g^{D}	Maximum generation ramp-up/down
T_g^{U}/T_g^{D}	Maximum uptime/downtime limits for generator g
T_g^{int}	The initial unit status at time $t = 1$ (i.e., $T_g^{int} = +4$ meansgenerator g was working for 4 h)
$p_g(int)$	Unit g's active power output at the start of the planning horizon
H_g^{U}	Uptime limit factor to make sure the unit works within its up limit in the planning horizon time. $H_g^{U} = min\left\{T, t+T_g^{U}\right\}$
H_g^{D}	Downtime limit factor. $H_g^{D} = min\left\{T, t+T_g^{D}\right\}$

Variables

$p_{g,m}(t)$	Piece-wise generation at time t
$p_g(t)$	Generation power of unit g at time t
$x_g(t)$	Unit on/off status at time t (binary)
$v_g(t)/y_g(t)$	Startup/shutdown status of unit g at time t
$r_g(t)$	The reserve power of unit g at time t
$f_n(t)$	Hourly power of transmission lines injected to node n
$s(t)$	Hourly requirement of the total system spinning reserve
$\delta_n(t))$	Voltage angle at time t
$\mathrm{p_l^{br}(t))}$	Power flow in transmission line l at time t
$p_j^{w}(t)$	Weather-dependent unit j power forecast at time t
x	MILP-UC variables vector of formulation (1)
$\widetilde{x}$	The reduced variables vector for ML-based UC formulation
$\widetilde{x}(\Delta_S)$	ML decision output vector for a certain data input $(\boldsymbol{\Delta}_S)$
x^{F}	Fuzzy variables vector of formulation (48) or (53)
$\Lambda_g^{u}(t)/\Lambda_g^{d}(t)$	Individual fuzzy set of ON/OFF decisions
$\mu I_g^{u}(t)/\mu I_g^{d}(t)$	Fuzzy membership function of ON/OFF decisions
$\Lambda^{C}(\Delta_S)$	Fuzzy set of MILP-UC operational costs
μC	Fuzzy membership of MILP-UC operational costs
$ZG^{u}(t)/ZG^{d}(t)$	Aggregate fuzzy set of ON/OFF decisions
$\mu G^{u}(t)/\mu G^{d}(t)$	Fuzzy membership of aggregated ON/OFF decisions
λ	Fuzzy objective of fuzzy frameworks

clearing. However, with the significant advancements in commercial mixed-integer linear programming (MILP) solvers, a shift from LR to MILP became practical. This transition was notably spearheaded by the Pennsylvania-New Jersey-Maryland (PJM) system operator in 2004 [2]. However, the effectiveness of MILP solutions for UC is often compromised for two reasons: 1) a high number of binary variables; and 2) a high number of mixed-binary security constraints. Therefore, reducing the number of binary variables and system constraints are the keys to reducing computational complexity and the solving time of large-scale UC problems.

To address these challenges, researchers have developed various strategies to improve the MILP formulation and enhance computational efficiency. In [3], the authors proposed "constraint-and-vertex conversion" and "vertex projection" techniques to tighten the ramping constraints of MILP-UC formulation. Yang et al. [4] introduced tightened three-period unit commitment formulations using additional variables to represent the states of a single unit over three periods. Additionally, Qiu et al. [5] proposed a multistage unit commitment framework that considered the non-anticipative realization of uncertainty in a power system and developed a decomposition framework to enhance the computational complexity. While tightened methods can enhance the MILP-UC problem, they also introduce challenges related to complexity and scalability, which require considerable time to reach the best-known UC solution. These limitations need to be carefully considered when applying such methods to practical power system problems [6]. Consequently, machine learning (ML) has become a crucial tool to address the UC challenges in large-scale power systems.

1.1. Machine learning application for UC

Recently, UC formulations with ML based techniques have evolved into a potent tool to map immense non-linear input-output data and predict the UC outputs with efficiency and precision, thereby addressing the MILP limitations mentioned above. This literature survey classifies ML-based UC methods according to their approaches to the feasibility process.

The feasibility process in UC methods ensures that the solutions generated by ML models adhere to all operational constraints and requirements of the power system. This involves validating and adjusting the ML-generated outputs to meet constraints such as generation limits, ramp rates, startup and shutdown constraints, and reserve requirements. The process may include heuristic adjustments, integration of feasibility layers, or incorporation of external algorithms to ensure that the solutions are both practical and implementable in real-world scenarios.

1.1.1. ML-UC methods without feasibility process

This subsection reviews the literature on the application of ML-UC and highlights the challenges in ensuring feasible integration of ML

within the UC framework. Using historical data, the authors of [7] applied k-nearest neighbors (KNN) to eliminate unnecessary transmission constraints in the UC formulation. However, this method has not provided a feasible approach for removing such constraints from the UC formulation. Further in [8], the neural network (NN) model has been used to classify transmission constraints into two groups, difficult and simple, to minimize UC problem size. To tackle the UC problem, data-driven decisions were used as an expert system from the trained ML system [9]. The authors of [10] provided various ML strategies to improve the UC formulation. The related ML model to our work predicts the binary variables of generators for load input and then applies a hyperplane method to the ML output to construct a predictor that fixes a portion of generation units' decisions. Since this method relies on a heuristic tuning parameter to determine the optimal value for binary variables, there is still a possibility for infeasibility and suboptimality.

In [11], ML has been used to anticipate the uncertainty of line disruptions caused by hurricanes and as an input to solve the stochastic UC challenge. Jiménez et al. [12] proposed a data-driven optimization for UC, employing Q-learning technique to continuously refine UC strategies in real-time operational scenarios. This approach iteratively updates decision policies based on real-time feedback. In [13], Navin NK and Sharma R reframed the UC problem as a multi-agent fuzzy reinforcement learning framework, wherein individual generators assume the role of players collaborating to collectively minimize the total operational cost. De Mars and O'Sullivan [14,15] explored the application of reinforcement learning coupled with tree search algorithms to solve the UC problem. These methods have been applied on a small number of generating units, up to 30 units, while the transmissions constraints have not been considered. Overall, these approaches require a feasibility stage for the ML output for safe operation of the UC process.

In [16], ML output was utilized as a warm-up to remove the redundant transmission constraints in the UC problem. Importantly, for large-scale UC problems, using the ML binary solutions as a warm start does not alter the solution space or the solution method. It simply provides a potential starting point for the optimizer. However, modern MILP solvers (such as Gurobi and CPLEX) [17] are equipped with advanced pre-solve techniques to find the best feasible starting point and precondition the problem, which can render ML-based warm start methods less impactful.

1.1.2. ML-UC methods with external feasibility process

This subsection illustrates the ML models that require an external feasibility stage to ensure feasible operation within the UC framework. In [18], the convolution neural network (CNN) has been trained to predict generation binary schedules. The complete ML output (100 %) has been utilized directly to fix the binary variables of the MILP-UC formulation. A feasibility stage has been added to ensure the ML output is feasible to the minimum up and down constraints; otherwise, the ML output must be changed to sufficiently cover these constraints. In [19], a graph neural network (GNN) is used to enhance reliability assessment in grid operations using a confidence assignment to the ML output. For safe operation, the same feasibility algorithm of [18] has been applied for the ML output. Further, Tang et al. [20] introduced a methodology leveraging graph convolutional networks (GCN) to enhance MILP-UC formulation, integrating power grid topology into the learning process. Their approach involves encoding the power grid's structural information into graph representations, allowing the model to capture spatial dependencies and security constraints.

Hybrid ML approaches have also shown promise to solve the UC challenge and enhance the power systems applications. One such approach leverages historical data for long-short-term memory (LSTM) and light-gradient boosting model (LGBM) to predict renewable energy source outputs, which are then incorporated into a stochastic UC optimization framework [21]. In [22], an online learning framework was developed to create stable integer variables sets to reduce the UC problem dimensionality. The variable sets have been created via comparison between the primary binary solution of the solver with the relaxed solution. However, the incorrect binary fixed variable may lead to infeasibility. In [23], deep reinforcement learning with soft actor-critic model has been used to enhance the UC formulation by training agents to learn optimal policies, which are then integrated with convex optimization techniques. This model provided a feasibility layer for trained data. Nonetheless, the UC may be infeasible for unseen data. Further in [24], a hybrid GNN with LSTM model has been created to capture the generation schedule, aiding in solving reduced security-constrained UC problems.

For the seen data, an external feasibility layer in NN has been added to ensure the generated UC schedule meets the binary constraints in [25]. However, the ML output might be infeasible for untrained data. Logistic regression (LR) and support vector machines (SVM) model has been used to incorporate frequency nadir constraints into the UC problem to ensures system reliability during contingencies [26]. In [27] the authors provide a comprehensive review highlighting current trends in ML applications to the UC problem, discussing challenges like scalability and interpretability. Conversely, fixing the incorrect binaries in UC might lead to an infeasible solution.

1.1.3. Other ML applications

Other ML approaches enhanced power systems applications. For instance, in [28] an improved anti-noise adaptive LSTM model has been developed for the robust remaining useful life (RUL) prediction of lithium-ion batteries. This approach effectively mitigates the impact of noise and variability in battery data, providing more accurate and reliable RUL predictions. Additionally, a hybrid model combining singular filtering, Gaussian process regression, and LSTM has been proposed in [29] for whole-life-cycle remaining capacity estimation of lithium-ion batteries. This model is adapted for fast aging and varying current conditions, ensuring accurate and comprehensive capacity estimations throughout the battery's lifecycle. These advancements in battery life prediction techniques exemplify how ML approaches can address complex challenges in power systems, offering robust solutions for battery management.

Collectively, ML decision output cannot be relied upon in enhancing the UC formulation because this output may contain incorrect decisions and be feasible for a set of constraints at the same time, leading to infeasibility and suboptimality for the whole UC formulation. Therefore, while ML-enhanced UC formulation methods have the potential to solve the UC faster, two main challenges remain:

1) **Suboptimality -** ML models for large-scale power systems cannot have a 100 % accurate prediction, especially when the load pattern has not been seen before. Using false predictions to solve the UC formulation leads to an undesirable suboptimal solution.
2) **Infeasibility -** This issue could happen if the false binary predictions are used to fix the binary variables in the UC formulation.

In our recent works [30,31], the concept of trusted generators' set has been proposed which is defined in the offline training process to use the ML output of those generators in the UC process. To overcome the infeasibility issue, an external feasibility process has been proposed to eliminate the infeasible ML predicted states and converted them into unknown sets. However, considerable ML data is lost due to that infeasibility process. In contrast, the incorporation of fuzzy optimization into ML-based UC methodologies offers a more flexible and adaptive approach to address these challenges.

1.2. Review on fuzzy optimization for UC

Fuzzy optimization (FO), inspired by human reasoning processes, provides a practical approach to managing imprecise or uncertain information. By relaxing constraints within optimization problems such as UC, rigid constraints are replaced with satisfaction functions. These

functions quantify the degree to which constraints are met without strict enforcement, allowing the optimization process to approach constraints flexibly. Incorporating FO effectively addresses uncertainties related to demand variations and renewable energy integration, enhancing the robustness and feasibility of optimization processes. Consequently, FO is a valuable tool for tackling complex real-world problems in power systems.

Researchers have employed FO as viable alternatives for resolving the UC formulations. Saneifard et al. proposed a FO method that handles uncertainties associated with the daily load curve [32]. Saadawi et al. used FO to manage uncertainties in load forecasts, fuel prices, and generator availability [33]. Saber et al. proposed a fuzzy unit commitment scheduling approach using absolutely stochastic simulated annealing, considering load uncertainties to improve the solution quality [34]. Venkatesh et al. proposed a fuzzy MILP-UC model that incorporates wind generators to handle the risks of incorporating wind energy and minimizing UC operating costs [35]. Further, FO has been used to handle transmission lines outage under emergency contingency in [36].

Notably, while fuzzy models have been extensively applied in conventional UC formulations, there is a noticeable gap in the literature regarding the utilization of fuzzy optimization techniques for ML-based UC methodologies. This presents an opportunity for the integration of fuzzy optimization into ML-based UC formulations to address challenges related to infeasibility associated with ML predictions.

1.3. Knowledge gap in surveyed methods

As a continuation of our work, we propose two fuzzy methods to incorporate the full ML output and enhance the MILP-UC formulation using the Fuzzy MILP framework without the need for an external infeasibility process. This study presents a hybrid CNN and Bidirectional Long-Short-Term Memory (BiLSTM) model as an effective and reliable technique for training and predicting large amounts of data. From the literature, hybrid deep learning models show great potential in achieving accurate predictions. BiLSTM introduced by Schuster M and Paliwal KK, is a type of recurrent neural network (RNN) commonly used in natural language processing. Unlike traditional LSTM, BiLSTM processes input in both forward and reverse directions, incorporating information from both sides. LSTM layers consist of interconnected memory modules with gates that manage memory cell states, allowing the network to handle temporal data [37].

Table 1 provides the main differences between the proposed methods (PMs) and the recent ML-based UC models of [8,10,16,18,19,30]. However, to the best of our knowledge, the application of fuzzy optimization on ML-UC methods does not exist.

In this paper, the proposed fuzzy models incorporate ML-predicted decisions into the MILP-UC method as non-binding fuzzy constraints. It enhances the basic MILP UC solution process reducing search space and solution time. The use of fuzzy constraints to introduce ML decision into MILP UC formulation – as non-binding constraints – ensures that the enhanced MILP UC formulation will provide a feasible solution where the basic MILP UC formulation has one.

Table 1
Overview with most related recent literature.

	ML Type	Guaranteed feasibility (if possible)	Feasible process	ML Data	System Scale
[8]	NN	No	X	Losses	118 bus
[10]	KNN	No	X	Losses	>2000 bus
[16]	NN	No	X	Losses	500 bus
[18]	CNN	No	External	Losses	100 unit
[19]	GNN	No	External	Losses	>2000 bus
[30]	NN	No	External	Losses	>2000 bus
PMs	CNN-BiLSTM	Yes	NA	No losses	>2000 bus

To the best of our knowledge, this is the first work to use the hybrid ML-fuzzy MILP-UC formulation to provide a robust solution while leveraging ML prediction to reduce solution time while completely avoiding possible infeasibility and/or suboptimality arising from ML predictions when confronted with unseen scenarios.

A two-stage (offline and online) fuzzy MILP framework for ML-based UC formulations is proposed. In the offline stage, a hybrid CNN-BiLSTM model is trained using the best-known solutions of UC solutions.

The main contributions of the paper are summarized in Fig. 1 and listed as follows.

- Individual Fuzzy Optimization Model (PM1): In the online stage, the first fuzzy model (PM1) is invoked to include individual ML decisions as non-binding fuzzy constraints to augment the MILP UC formulation. Unlike the literature [18,30], this model does not require an external feasibility stage for integers since all UC and fuzzy constraints are optimized once, while fully leveraging ML decisions.
- Aggregate Fuzzy Optimization Model (PM2): Alternatively, in the second approach (PM2), the ML decision outputs are accumulated for all generating units in each hour to propose a smaller set of non-binding fuzzy constraints to enhance MILP UC formulation. This model saves the data lost in the external feasibility process and considers the full UC constraints while leveraging ML decisions. Additionally, a tuning artificial parameter (K-factor) is proposed for the fuzzy models to guarantee the best performance quality in terms of UC solution optimality and computation timesaving.
- The use of non-binding fuzzy constraints allows the proposed method to exploit all ML decisions while avoiding suboptimality and/or infeasibly which plagues other ML methods.

2. MILP-UC problem formulation

The UC process is solved in practice with distinct spatial and temporal data every day. Because of the repeating nature of the UC process, we use a formulation for the UC model from [6]. For power systems, the hourly load forecasts, weather-dependent generation, and generation costs are predetermined in day-ahead UC scenarios. The specific MILP-UC model is expressed as follows:

$$minimize: \mathscr{C}(x) \tag{1}$$

$$\text{subject to}: G(x) \le H \tag{1a}$$

Where $\mathscr{C}(x)$ is the MILP-UC objective function as detailed in (2) and (3). $G(x)$ describes the MILP-UC constraints of (4)–(26) and H is the upper bound for the constraints' matrix G. The vector x contains the optimal UC binary and continuous variables as described in (27). The detailed UC model is illustrated as follows.

2.1. Total UC operational costs

The total UC operational cost $\mathscr{C}(x)$ of (2) comprises the operational costs of all generating units encompassing: the fuels, shutdown, startup,

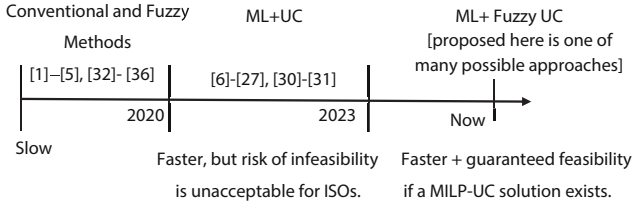


Fig. 1. Evolution of Unit Commitment Methods.

weather-dependent sources, and reserve costs of coal-fired units, and combined cycle gas turbines. The fuel cost function $f_g^p(t)$ as in (3) for unit g is a convex piecewise linear model of the actual quadratic cost.

$$\mathscr{C}(x) = \sum_{t\in\Omega^{\mathcal{T}}}\sum_{g\in\Omega^{\mathcal{G}}}\left(f_g^p(t) + c_g^{st} \bullet y_g(t) + c_g^{r} \bullet r_g(t) + \sum_{j\in\Omega^{re}} c_j^{re}\left(r_j^{re}(t)\right) + c_g^{r}\left(r_g(t)\right)\right) \quad (2)$$

$$f_g^p(t) = \sum_{m\in\Omega^{\mathcal{M}}} c_{g,m}^{p}\left(r_{g,m}(t)\right) + c_g^{z} \bullet x_g(t) \quad (3)$$

Subject to:

2.2. Security constraints on units' decision variables

The logical startup and shutdown decisions at each hour are constrained in (4). The consecutive unit decisions are related to the startup and shutdown decisions as in (5), where the unit could not be turned on and off at the same time.

$$0 \le r_g(t) + y_g(t) \le 1; \forall g \in \Omega^{\mathcal{G}}, t \in \Omega^{\mathcal{T}} \quad (4)$$

$$x_g(t+1) - x_g(t) = r_g(t+1) - y_g(t+1); \forall g \in \Omega^{\mathcal{G}}, t = 1, \ldots, T-1 \quad (5)$$

The previous day's commitment status of each generating unit is a crucial input. This information helps ensure continuity and feasibility in the scheduling process, considering the operational constraints such as minimum uptime and downtime limits as in (6) and (7), respectively.

$$\sum_{t=1}^{T_g^U - T_g^{int}} x_g(t) \ge T_g^U - T_g^{int}; \forall g \in \Omega^{\mathcal{G}}, t \in \left[T_g^{int} > 0, T_g^U > T_g^{int}\right] \quad (6)$$

$$\sum_{t=1}^{T_g^D - T_g^{int}} x_g(t) \le 0; \forall g \in \Omega^{\mathcal{G}}, t \in \left[T_g^{int} < 0, T_g^D > -T_g^{int}\right] \quad (7)$$

Generators' uptime and downtime limits are operational constraints that are critical in the unit commitment process. Uptime limits (8) and (9) refer to the minimum amount of time that a generator must remain online once it has been started. These constraints ensure that once a generator is turned on, it cannot be shut down immediately and must run for a certain minimum duration. Downtime constraint (10) refers to the minimum amount of time that a generator must remain offline once it has been shut down. This constraint is necessary to allow the generator to cool down, undergo maintenance checks, and prepare for the next operational cycle.

$$\sum_{s=t+2}^{H_g^U} x_g(s) \ge r_g(t+1).T_g^U; \forall g \in \Omega^{\mathcal{G}}, t \in \Omega^{\mathcal{T}^-} \quad (8)$$

$$T_g^U r_g(t+1) - \sum_{s=t+2}^{H_g^U} x_g(s) \le T_g^U - T + t; \forall g \in \mathcal{G}, t \in \Omega^{\mathcal{T}^-} \quad (9)$$

$$\left(1 - y_g(t+1)\right) \bullet T \ge \sum_{s=t+1}^{H_g^D} x_g(s); \forall g \in \mathcal{G}, t \in \Omega^{\mathcal{T}^-} \quad (10)$$

2.3. Constraints on power generation

At each period, constraints (11)–(17) ensure the generating power of each unit within the operational capacity, ramp-up, and ramp-down limits. In this subsection, the generation ramping-up and down constraints are considered whereas the generating unit is restricted by its commitment to the previous period. Constraint (11) defines the generation power at each period, which is a summation of generating powers in each piece-wise segment.

$$p_g(t) = \sum_{m\in\mathcal{M}} r_{g,m}(t); \forall g \in \Omega^{\mathcal{G}}, t \in \Omega^{\mathcal{T}} \quad (11)$$

Constraint (12) is used if the unit has a start-up limit, i.e., the commitment ramping up from 0 MW to P_g^D. Otherwise, the unit will be limited to the hourly ramping-up rate R_g^U. Similarly, constraint (13) is used for ramping down and shutdown conditions. Constraints (14) and (15) have been used similar to (12) and (13) but only for the beginning of the dispatch (at $t = 1$).

$$p_g(t) - p_g(t-1) \le R_g^U\left(1 - r_g(t)\right) + P_g^D r_g(t); \forall g \in \Omega^{\mathcal{G}}, t = 2, .., T \quad (12)$$

$$p_g(t-1) - p_g(t) \le P_g^D y_g(t) + R_g^D\left(1 - y_g(t)\right); \forall g \in \Omega^{\mathcal{G}}, t = 2, .., T \quad (13)$$

$$p_g(1) - p_g(int) \le R_g^u\left(1 - r_g(1)\right) + P_g^D r_g(1); \forall g \in \Omega^{\mathcal{G}} \quad (14)$$

$$p_g(int) - p_g(1) \le P_g^D y_g(1) + R_g^d\left(1 - y_g(1)\right); \forall g \in \Omega^{\mathcal{G}} \quad (15)$$

The generation of each piece-wise segment is limited to its maximum, as shown in (16). Therefore, the hourly generation power of each unit, which is the total power in all segments, is constrained by the thermal limit of the generating unit as in (17).

$$0 \le r_{g,m}(t) \le P_{g,m}^U; \forall g \in \Omega^{\mathcal{G}}, m \in \Omega^{\mathcal{M}}, t \in \Omega^{\mathcal{T}} \quad (16)$$

$$x_g(t) \bullet P_g^D \le P_g(t) \le x_g(t) \bullet P_g^U; \forall g \in \Omega^{\mathcal{G}}, t \in \Omega^{\mathcal{T}} \quad (17)$$

2.4. Constraints on power flow

A linearized power balance equation at each network node is modeled in (18). The injected power flow at each bus is obtained as (19). The power of each transmission line is derived from DC power flow as (20). Limits of the power in transmission lines and bus voltage angles are considered in (21) and (22), respectively.

$$\sum_{g\in\Lambda_n^{\mathcal{G}}} p_g(t) + \sum_{j\in\Lambda_n^{re}} r_j^{re}(t) - d_n(t) = f_n(t); \forall n \in \Omega^{\mathcal{N}}, t \in \Omega^{\mathcal{T}} \quad (18)$$

$$f_n(t) = B'.\delta_n(t); \forall n \in \Omega^{\mathcal{N}}, t \in \Omega^{\mathcal{T}} \quad (19)$$

$$p_l^{br}(t) = \frac{\left[\delta_i(t) - \delta_j(t)\right]}{X_l}; \forall (i,j) \in \Omega^{\mathcal{L}}, t \in \Omega^{\mathcal{T}} \quad (20)$$

$$-F_l^U \le p_l^{br}(t) \le F_l^U; \forall l \in \Omega^{\mathcal{L}}, t \in \Omega^{\mathcal{T}} \quad (21)$$

$$\delta^D \le \delta_n(t) \le \delta^U; \forall n \in \Omega^{\mathcal{N}}, t \in \Omega^{\mathcal{T}} \quad (22)$$

2.5. Constraints on the power reserve

The spinning reserve requirement and the power reserve of operating units are constrained in (23)–(26). For interconnected power networks like in the case of Ontario, Canada, the percentile γ characterizes the required ratio of the power reserve. It should equal 25 %, according to the Northeast Power Coordinating Council [38].

$$s(t) \ge x_g(t) \bullet P_g^U; \forall g \in \Omega^{\mathcal{G}}, t \in \Omega^{\mathcal{T}} \quad (23)$$

$$\sum_{g\in\mathcal{G}} r_g(t) \ge \gamma.s(t); \forall \in \Omega^{\mathcal{T}} \quad (24)$$

$$r_g(t) \le x_g(t) \bullet R_g^{10}; \forall g \in \Omega^{\mathcal{G}}, t \in \Omega^{\mathcal{T}} \quad (25)$$

$$r_g(t) \le \sum_{m\in\mathcal{M}}\left(P_{g,m}^U x_g(t) - r_{g,m}(t)\right); \forall g \in \mathcal{G}, t \in \mathcal{T} \quad (26)$$

The full MILP-UC variables vector x is defined as binary variables vector x_B and continuous variables vector x_C as in (27).

$$x = [x_B, x_C]; x_B = \left[z \in \mathbb{Z}_2^{G\times T}, v \in \mathbb{Z}_2^{G\times T}, w \in \mathbb{Z}_2^{G\times T}\right];$$

$$x_C = \left[p \in \mathbb{R}^{(G\times M)\times T}, r \in \mathbb{R}^{G\times T}, s \in \mathbb{R}^{1\times T}, \delta \in \mathbb{R}^{N\times T}, p^{br} \in \mathbb{R}^{L\times T}\right] \quad (27)$$

The units' startup and shutdown variables v, w are dependent variables that mainly can be obtained if the binary UC solution z exists. Thus, we define vector $z \in \mathbb{Z}_2^{G\times T}$, the generation decision is the optimal binary solution of the MILP-UC formulation (1).

UC formulation (1) of large-scale power systems faces exponentially increasing binaries and constraints, introducing non-convexity and mixed-binary constraints, leading to ISOs terminating solvers with large optimality gaps, and causing economic loss. This paper proposes a Hybrid CNN-BiLSTM plus Fuzzy-MILP-UC Method to ensure the feasibility of ML mitigation in UC formulation, addressing the suboptimal and infeasible solutions in the existing literature.

3. Proposed method – overview

In this section, the Hybrid CNN-BiLSTM plus Fuzzy MILP UC framework and solution algorithm are introduced. The complete framework is shown in Fig. 2. The details of the CNN and BiLSTM architectures and the mathematical formulations are shown in the following sections.

Offline Stage - For a brief overview, in the offline stage, using daily data $\boldsymbol{\Delta} = \{\Delta_1, \ldots, \Delta_s\}$ for hourly bus loads, weather-dependent generation data, and daily fuel generation costs, the MILP-UC (1) has been solved for each data instance Δ_s, forming the training data and hourly generation schedules $z(\boldsymbol{\Delta})$ to train the CNN-BiLSTM model.

Online Stage - In the online stage, we propose two fuzzy optimization approaches listed as follows:

a. An individual fuzzy optimization model for ML output (PM1).
b. Aggregate fuzzy optimization model for ML output (PM2).

In the PM1 framework, each ML decision (for each generator at each hour) is modeled as a fuzzy constraint with a satisfaction parameter and appended to the main MILP-UC constraints (1a). The objective function (2) is also transformed into a fuzzy constraint with a satisfaction parameter. Finally, the minimum of all these satisfaction parameters is maximized to get the UC solution.

In the PM2 framework, the ML decision outputs are aggregated for all generators at each hour and then used to model a fuzzy constraint and appended to the main MILP-UC constraints (1a). The objective function (2) is also transformed into a fuzzy constraint with a satisfaction parameter. Lastly, the minimum of all these satisfaction parameters is maximized to obtain the UC solution. Both PM1 and PM2 methods dramatically reduce the search space while retaining the best-known optimal solution. These formulations remain MILP type and hence can be solved using commercial solvers, as completed in this work.

4. Proposed method – offline process

4.1. ML structure-training data environment

In this paper, the CNN-BiLSTM model utilizes the data set $\Delta_s = \{$the

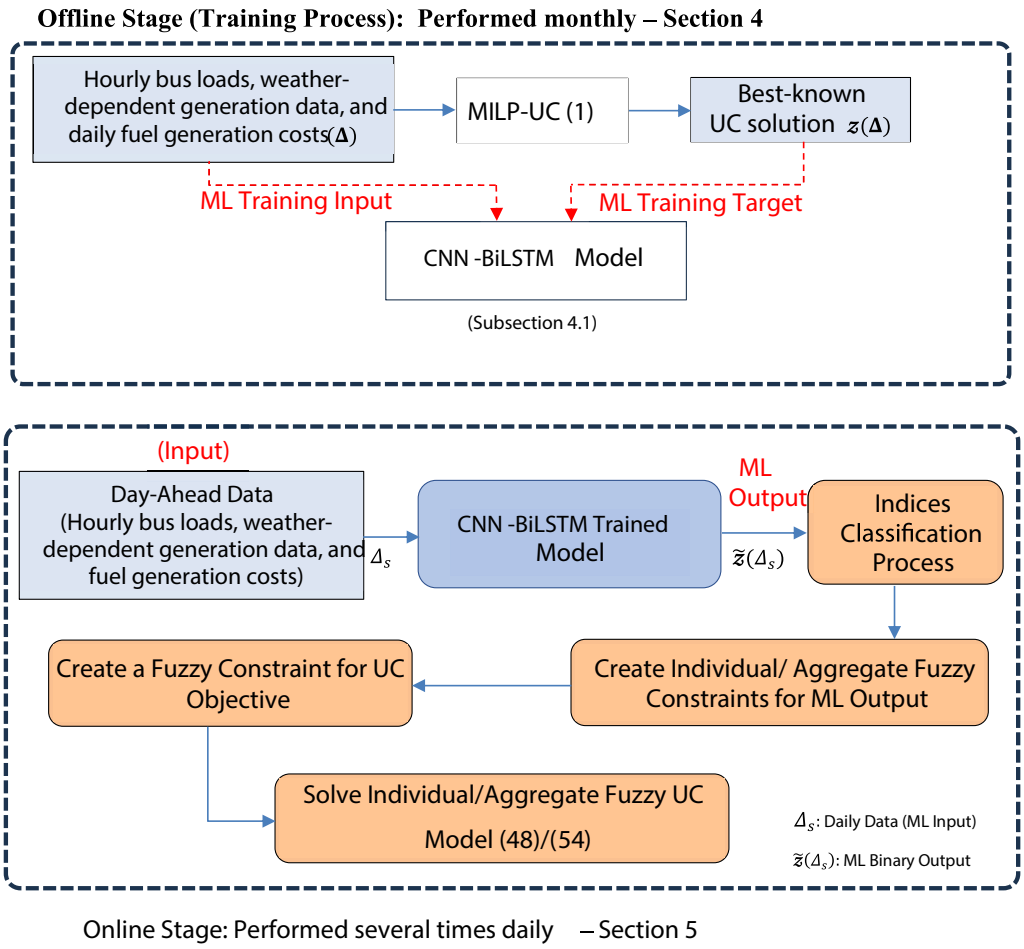


Fig. 2. Framework of fuzzy optimization methods for the ML-based MILP-UC algorithm.

net hourly load data at each bus $Pd_n(t)$, and the generation cost curve data $\mathscr{C}^p_{g,m}\}$ as inputs for the training process to predict the corresponding MILP-UC generation decisions $\tilde{z}_g(t)$. The net hourly load $Pd_n(t)$ is the difference between the forecasted loads $\mathscr{L}_n(t)$ and the accumulated weather-dependent sources $\mathscr{P}^w_j(t)$, i.e., wind/PV at each bus n, as in (28).

$$Pd_n(t) = \mathscr{L}_n(t) - \sum_{j\in\Lambda^w_n} \mathscr{P}^w_j(t); \forall n \in \Omega^{\mathscr{N}}, t \in \Omega^{\mathscr{T}} \tag{28}$$

For each day, the generation cost is considered changeable, and the cost data is assumed to follow Gaussian distribution along the tested samples. The general form for the generated cost data is denoted as a probability density function in (29).

$$F\left(\mathscr{C}^p_{g,m}\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\mathscr{C}^p_{g,m}-\mu}{\sigma}\right)^2}; \forall g \in \Omega^{\mathscr{G}}, m \in \Omega^{\mathscr{M}} \tag{29}$$

where parameters μ and σ are the cost expectation and standard deviation, respectively. Now the full training parameters $\Delta = \{\Delta_1, \ldots, \Delta_s\}$ and $\mathscr{z}(\boldsymbol{\Delta})$ are found using (1).

4.1.1. CNN-BiLSTM model

This study presents a hybrid CNN-BiLSTM model as an effective and reliable technique for training and predicting large amounts of data. From the literature, hybrid deep learning models show great potential in achieving accurate predictions. The structure of the proposed model is shown in Fig. 3.

4.1.2. Convolution neural network

CNN is a type of deep learning model that has been successfully applied to a variety of power systems applications [39,40]. CNN is typically viewed as a hierarchical feature extractor capable of automatically learning high-level features from original sequences. The fundamental structure of CNN consists of layers such as a convolutional layer, pooling layer, and fully connected layer as shown in Fig. 3. Each layer's responsibilities are summarized as follows:

a) CNN's convolution layer generates new feature maps with several convolution kernels. Local feature extraction, where all input maps share kernel weights, works well with convolution. The output of this layer is formulated as a one-dimensional (1-D) layer in (30).

$$Q_j = \Delta_s{}^*\boldsymbol{K}_j + b_j \tag{30}$$

Where $\Delta_s \in \Delta$ the data input for sth sample, $\boldsymbol{K}_j$ is the jth convolution kernel for a 1-D filter, and the corresponding scalar bias is b_j. The output feature at a position t is given in (31).

$$Q_j(t) = \left(\Delta_s{}^*\boldsymbol{K}_j\right)(t) = \sum_{m}^{lm} \Delta_s(t-k) \times \boldsymbol{K}_j(t) \tag{31}$$

where lm is the length of the CNN filter.

b) The pooling layer reduces input map in-plane dimensionality, reducing learnable parameters and preventing overfitting.

c)The fully connected layer maps the features from the convolution layers and pooling layers to the output layer for high-level inference. The hyperbolic tangent function (tanh) is used as an activation function to limit the final CNN output $\boldsymbol{Q}(\Delta_s)$ between $\{-1,1\}$ and the output can be denoted as (32):

$$\boldsymbol{Q}(\Delta_s) = tanh(\Delta_s{}^*W_P + b_P) \tag{32}$$

where W_P and b_P denote the weights and biases of the pooling layer, respectively.

4.1.3. Bidirectional long short-term memory layer

In the BiLSTM model, an additional LSTM layer is introduced to reverse the direction of information flow, and the outputs from both layers are combined using methods like arithmetic mean, sum, multiplication, or concatenation. This bidirectional flow ensures that current information depends on both past and future information. The forward and reverse sequences are denoted by blue and red arrows in Fig. 3, respectively. The proposed equations illustrate the operations of the BiLSTM model.

$$\overrightarrow{h}_t = tanh\left(W_c Q_t + W_{\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \tag{33}$$

$$\overleftarrow{h}_t = tanh\left(W_c Q_t + W_{\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right) \tag{34}$$

$$\widetilde{Y}_t = tanh\left(V_{\overrightarrow{h}} \overrightarrow{h}_t + V_{\overleftarrow{h}} \overleftarrow{h}_t + b_Y\right) \tag{35}$$

where W and V are waiting matrices and $\widetilde{Y}_t$ is the output of BiLSTM at time t. The ML model output binary vector is given by $\tilde{z}(\Delta_s)$. Specifically, for each $\tilde{z}_g(t) \in \tilde{z}(\Delta_s); \forall g \in \Omega^{\mathscr{G}}, t \in \Omega^{\mathscr{T}}$ is set to 1 if $\widetilde{Y}_g(t) \geq 0$ and 0 otherwise.

4.1.4. CNN-BiLSTM Loss Function and Training Algorithm

The loss function measures the difference between each mini-actual batch's output $\widetilde{\boldsymbol{Y}}(\Delta)$ and ground-truth output $\boldsymbol{Y}(\Delta)$, which is normalized to the range of $[-1,1]$. In MATLAB, the loss function used during the

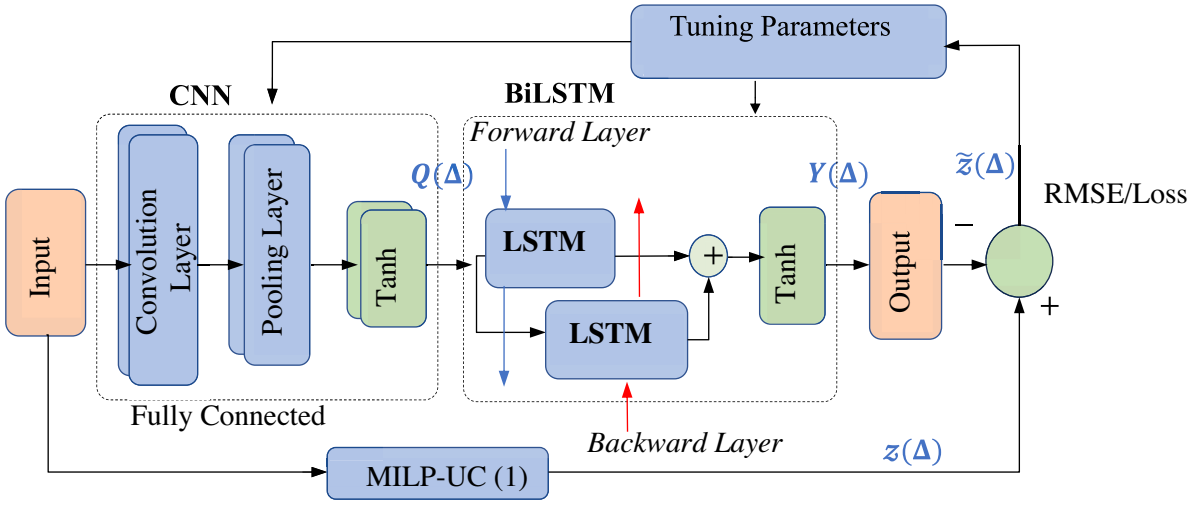


Fig. 3. CNN-BiLSTM Main Architecture.

training process for a regression problem with a CNN-BiLSTM architecture (or any other architecture involving sequence-to-sequence or sequence-to-label prediction) is typically the mean squared error (MSE) as in (36). This loss function measures the average of the squares of the errors between the predicted values and the actual values.

$$L(\Delta) = \frac{1}{T \bullet G \bullet S} \sum_{s \in \mathscr{S}} \sum_{t \in \Omega^{\mathscr{T}}} \sum_{g \in \Omega^{\mathscr{G}}} \left(Y_{t,g}(\Delta_s) - \widetilde{Y}_{t,g}(\Delta_s) \right)^2 \tag{36}$$

where $Y(\Delta) = [Y(\Delta_1), \ldots, Y(\Delta_S)]^{(TG \times S)}$.

In this paper, adaptive moment estimation "*Adam*" is used as a training algorithm for the CNN-BiLSTM model to optimize (36). *Adam* builds upon stochastic gradient descent to solve non-convex problems more quickly and with fewer parameters than many other optimization programs [41]. The tuning parameters of the ML model must be optimized to reduce the Root Mean Squared Error (RMSE) between the estimated and actual system output. These parameters consist of the number of CNN layers, the quantity of learning data, the optimizer, the batch size, the number of LSTM cells, the activation function, and the number of epochs. These parameters are largely determined empirically until they attain optimal performance.

5. Proposed method – Online process

5.1. Conventional ML-based UC feasibility challenge

After MILP-UC has been created as in (1), training data are gathered by solving MILP-UC formulation with a MILP solver such as Gurobi. Using the data vectors $\{\mathbf{\Delta}, \mathscr{x}(\mathbf{\Delta})\}$, the hybrid ML model is well-trained. Online, for one sample of data Δ_S, the ML output is $\widetilde{\mathscr{x}}(\Delta_S)$. Thus, the conventional ML-based UC formulation is stated as follows.

$$minimize : C(\widetilde{\mathscr{x}}) \tag{37}$$

$$\text{subject to} : G(\widetilde{\mathscr{x}}) \leq H \tag{37a}$$

$$\widetilde{\mathscr{x}}_B(\widetilde{\mathscr{x}}) = \widetilde{\mathscr{x}}(\Delta_s) \tag{37b}$$

Where $\widetilde{\mathscr{x}}$ is the reduced variables vector; $\widetilde{\mathscr{x}} = \left\{ \widetilde{\mathscr{x}}_B, \widetilde{\mathscr{x}}_C \right\}, \widetilde{\mathscr{x}}_C \subset \widetilde{\mathscr{x}}_B$ is the generation decision variables vector obtained by the ML. However, prediction errors in $\widetilde{\mathscr{x}}(\Delta_S)$ might lead to infeasibility and/or sub-optimality of (37).

Recent literature, such as [18,19,30], ensures that the ML output is feasible only for the minimum up and down constraints (4)–(10), which are known as binary constraints. While this approach works for the trained dataset, it may not be reliable for unseen data. For example, the ML output might satisfy the binary constraints but could be infeasible for mixed binary constraints, such as power limits (17) and reserve power constraints (26). In eq. (17), if an incorrect unit is fixed to be ON ($\mathscr{x}_g(t) = 1$), it should operate within its upper (P_g^U) and lower (P_g^d) limits as updated in (37c). However, the required power for a feasible solution might fall outside these limits.

$$P_g^D \leq P_g(t) \leq P_g^U; \forall g \in \Omega^{\mathscr{G}}, t \in \Omega^{\mathscr{T}} \tag{37c}$$

To overcome these challenges, we present the following ML-fuzzy MILP UC method to introduce ML-generated decisions into the basic MILP-UC formulation as non-binding fuzzy constraints such that it leverages and exploits ML predictions while avoiding possible infeasibility it might introduce.

5.2. Individual fuzzy optimization model for ML output (PM1)

A fuzzy set is a set of elements with each element qualified by a membership function. A membership function can be defined based on any attribute of choice. In this paper, the Hybrid CNN-BiLSTM model (trained offline), in the online phase considers input data and intelligently determines the UC solution. Fuzzy sets are used in this proposed work to quantify how close the MILP-UC solution is to that intelligently determined by the hybrid CNN-BiLSTM model and maximize this closeness. Accordingly, closeness is defined as a satisfaction function and is maximized in this work. The problem is then reposed with the original constraints (4)–(26) in addition to fuzzy sets constraints to maximize the minimum of these satisfaction parameters. This is detailed below.

5.2.1. Individual fuzzy ML constraints

In this subsection, we define the fuzzy membership functions of the ML output for each generating unit in each hour (individual) to be involved in the decision-making process (MILP-UC).

First, the indices of ML output for the day-ahead data Δ_S is denoted as follows.

$$t, g \in \Omega^{\mathscr{T}}, \Omega^{\mathscr{G}} = \{1, .., T \bullet G\} \tag{38}$$

where the ML output $\widetilde{\mathscr{x}}(\Delta_S)$ is split into two sets of indices. The ON indices are in the set Υ^u and the OFF indices are in the set Υ^d as follows.

$$\Omega^{\mathscr{G}} = \begin{bmatrix} \left\{ \Upsilon^u; t, g \in \Upsilon^u \middle| \widetilde{\mathscr{x}}_g(t) = 1 \right\}, \\ \left\{ \Upsilon^d; t, g \in \Upsilon^d \middle| \widetilde{\mathscr{x}}_g(t) = 0 \right\} \end{bmatrix} \tag{39}$$

Second, a fuzzy set $\Lambda_g^u(t)$ is created for each state in the set of ON indices of ML output Υ^u as in (40). The fuzzy set $\Lambda_g^u(t)$ in a universe of discourse $u_g^u(t)$ is characterized by a membership function $\mu I_g^u(t) : u_g^u(t) \rightarrow \{0, 1\}$. For each element in $u_g^u(t)$, the membership function $\mu I_g^u(t)$ indicates the degree of membership of that element in the fuzzy set $\Lambda_g^u(t)$.

$$\Lambda_g^u(t) = \left\{ \left[u_g^u(t), \mu I_g^u(t) \right]; u_g^u(t) \in \{0, 1\}, t, g \in \Upsilon^u \right\} \tag{40}$$

As shown in Fig. 4 (a), the membership function $\mu I_g^u(t)$ assigns a value between 0 and 1 to each element in $u_g^u(t)$, representing its degree of membership in the fuzzy set $\Lambda_g^u(t)$. A value of 0 means no membership, while a value of 1 means full membership. In the context of ML-based UC, the output of ML in ON indices set Υ^u has the highest probability to be ON. Thus, the ON fuzzy membership will have the highest value of 1 when the fuzzy UC status $\mathscr{x}_{g,t}^F(\Delta_S)$ is identical to the ML prediction (ON state) as defined in (41), while it has a value of 0 when the ML prediction and fuzzy UC solution are different.

$$\mu I_g^u(t) = \mathscr{x}_g^F(t); \forall t, g \in \Upsilon^u \tag{41}$$

where $\mathscr{x}_{g,t}^F(\Delta_S)$ is the fuzzy MILP-UC binary solution.

Similarly, the OFF-fuzzy set and its membership function are defined for the set of OFF indices of ML output Υ^d in (42) and (43) respectively. As shown in Fig. 4 (b) and eq. (43), the OFF-membership function has a highest value when the fuzzy solution $\mathscr{x}_g^F(t)$ is 0 which means it is identical to ML prediction for the OFF units and it has no membership when the ML prediction is incorrect.

$$\Lambda_{g,t}^d(t) = \left\{ \left[u_g^d(t), \mu I_g^d(t) \right]; u_g^d(t) \in \{0, 1\} \right\} \tag{42}$$

$$\mu I_g^d(t) = 1 - \mathscr{x}_g^F(t); \forall t, g \in \Upsilon^d \tag{43}$$

Overall, these membership functions $\mu I_g^u(t)$ and $\mu I_g^d(t)$ quantify the closeness of the ML prediction $\widetilde{\mathscr{x}}_g(t)$ and fuzzy MILP-UC solution $\mathscr{x}_g^F(t)$, for ON and OFF states respectively.

5.2.2. Fuzzy Membership of MILP-UC Objective Function

To establish a fuzzy framework for the UC formulation, the UC

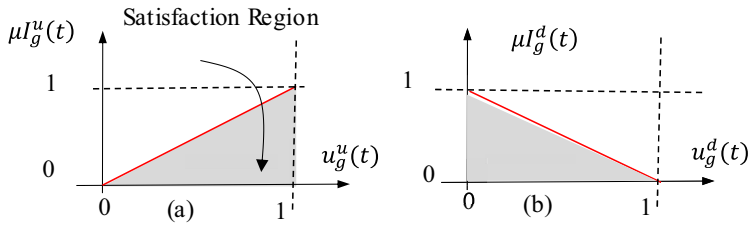


Fig. 4. Individual fuzzy memberships of (a) ON units and (b) OFF units.

objective can be transformed into a fuzzy constraint with an associated membership function. The fuzzy set for UC cost function (1) is defined by establishing a set of values for $\mathscr{C}(x^F)$ such that $\underline{C} \le \mathscr{C}(x^F) \le \overline{C}$. Below is a definition of this fuzzy set and the membership function $\mu C(\mathscr{C}(x^F))$.

$$\Lambda^C(\Delta_S) = \{[\mathscr{C}(x^F), \mu C(\mathscr{C}(x^F))]; \underline{C} \le \mathscr{C}(x^F) \le \overline{C}\} \tag{44}$$

$$\mu C(\mathscr{C}(x^F)) = \frac{\overline{C} - \mathscr{C}(x^F)}{\overline{C} - \underline{C}} \tag{45}$$

Fig. 5 demonstrates the optimal region for the MILP-UC solution. This optimal solution is greater than the lower bound for the operating cost $\underline{C}$ that can be obtained by relaxing the binaries of (1) and solved as LP formulation as in (46).

$$\underline{C} = minimize\ \mathscr{C}(x) \tag{46}$$

$$\text{subject to}: G(x) \le H; x \in \mathbb{R} \tag{46a}$$

Further, the optimal MILP-UC solution is lower than the upper bound $\overline{C}$ of the operating cost that considers all generating units are dispatched and obtained as (46).

5.2.3. Master fuzzy optimization model I

Using the fuzzy set theory, the following general formulation describes the optimal intersection of all fuzzy membership functions. Where the main fuzzy objective λ is used to maximize the minimum of all fuzzy membership functions.

$$Maximize: \lambda \tag{47}$$

$$\text{where}: \lambda \le \begin{Bmatrix} \mu C(x^F) \\ \mu I_g^u(t); \forall t, g \in Y^u \\ \mu I_g^d(t); \forall t, g \in Y^d \end{Bmatrix} \tag{47a}$$

Using the fuzzy framework of (47): 1) the individual fuzzy ML membership functions (41) and (43) and the fuzzy membership of the UC Objective function (45) are replaced by variable λ, and 2) the main MILP-UC constraints of (1a) are added to the complete fuzzy model as in (48).

$$Maximize: \lambda \tag{48}$$

Subject to:

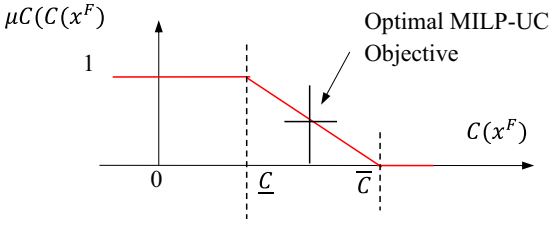


Fig. 5. Fuzzy membership function of the MILP-UC objective.

$$\lambda \le \mathscr{x}_g^F(t); \forall g, t \in Y^u \tag{48a}$$

$$\lambda + \mathscr{x}_g^F(t) \le 1; \forall g, t \in Y^d \tag{48b}$$

$$(\overline{C} - \underline{C})\lambda + \mathscr{C}(x^F) \le \overline{C} \tag{48c}$$

$$G(x^F) \le H \tag{48d}$$

$$0 \le \lambda \le 1 \in \mathbb{R}; x^F = [x_B^F, x_C^F, \lambda] \tag{48e}$$

Where x^F is the UC variables vector obtained by solving the fuzzy UC formulation (48); $x^F = [x_B^F, x_C^F, \lambda]$. $\mathscr{x}_g^F(t) \subset x_B^F$ is the generation decision variables vector.

5.2.3.1. Feasibility proof of PM1 (48). When the ML prediction is entirely inaccurate, resulting in a 100 % prediction error for either the ON or OFF state, the corresponding fuzzy membership value is 0. Consequently, the primary fuzzy objective λ (as defined in eq. (47a)) will also be 0. Substituting $\lambda = 0$ as the worst-case scenario in eq. (48), the non-binding fuzzy constraints of PM1 (48a, 48b, and 48c) are presented as follows.

$$0 \le \mathscr{x}_g^F(t); \forall g, t \in Y^u\ (ON\ units) \tag{49a}$$

$$\mathscr{x}_g^F(t) \le 1; \forall g, t \in Y^d\ (OFF\ units) \tag{49b}$$

$$\mathscr{C}(x^F) \le \overline{C} \tag{49c}$$

$$G(x^F) \le H \tag{49d}$$

In this case, the constraints are feasible because the unit states in eqs. (49a) and (49b) are restricted to their boundaries of either 0 or 1. Additionally, eq. (49c) remains feasible when $\lambda = 0$, as it adheres to the upper bound of the MILP-UC objective. The original MILP-UC constraints (48d) remain unchanged since they are independent of λ. Therefore, the formulation in eq. (48) is always feasible whenever the original MILP-UC is feasible.

5.3. Aggregate fuzzy optimization model for ML output (PM2)

In this subsection, the aggregate fuzzy optimization model (PM2) is presented. In PM2, the ML output is accumulated for all units to create fuzzy constraint sets for each period. The aggregate ON-fuzzy constraints denote that the total ON states of all units in the optimal MILP-UC solution will be close to the total number of ON states predicted by the ML at the same time and vice versa for the aggregate OFF-fuzzy constraints. The advantage of this model is that this model provides only $2 \times T$ fuzzy constraints rather than the $G \times T$ fuzzy constraints of the individual fuzzy model.

5.3.1. Aggregate fuzzy ML constraints

At each period, the aggregate ON and OFF-fuzzy ML sets $\{ZG^u(t), ZG^d(t)\}$ and membership functions $\mu G^d(t), \mu G_t^d(\Delta_S)$ are illus-

trated as follows.

Unlike to the individual fuzzy sets, For the ON-fuzzy set $ZG^u(t)$, the ON-predicted decisions by the ML model are accumulated for each period as in (50). The fuzzy set $ZG^u(t)$ in a universe of discourse $UG^u(t)$ is characterized by a membership function $\mu G^u(t): UG^u(t) \rightarrow \{0,1\}$. For each hour, the membership function $\mu G^u(t)$ indicates the degree of membership of the aggregated ML decisions in the fuzzy set $ZG^u(t)$. Similarly, the OFF-fuzzy set $ZG^d(t)$ has been created as shown in (51).

$$ZG^u(t) = \left\{ (UG^u(t), \mu G^u(t)) \mid 0 \le UG^u(t) \le \sum_{g \in \Upsilon^u} \tilde{z}_g(t) \right\} \tag{50}$$

$$ZG^d(t) = \left\{ (UG^d(t), \mu G^d(t)) \mid 0 \le UG^d(t) \le \sum_{g \in \Upsilon^d} \left(\mathcal{O}_g^d(t) - \tilde{z}_g(t) \right) \right\} \tag{51}$$

where $\mathcal{O}^d$ is a ones' vector for ML-OFF decisions with the same size of Υ^d.

As shown in Fig. 6, the aggregated ON/OFF fuzzy membership functions (52) and (53) are defined to provide a value of 1 when the UC decisions obtained by the fuzzy UC model $z^F(\Delta_s)$ align with the ML prediction and the lowest value of zero if the ML prediction is totally different.

$$\mu G^u(t) = \frac{\sum_{g \in \Upsilon^u} z_g^F(t)}{\sum_{g \in \Upsilon^u} \tilde{z}_g(t)}; \forall t \tag{52}$$

$$\mu G^d(t) = 1 - \left(\frac{\sum_{g \in \Upsilon^d} z_g^F(t)}{\sum_{g \in \Upsilon^d} \left(\mathcal{O}_g^d(t) - \tilde{z}_g(t) \right)} \right); \forall t \tag{53}$$

5.3.2. Master Fuzzy Optimization Model II

Using the fuzzy theory detailed in Subsection 5.2.3, the fuzzy optimization formulation will be created using (47). The aggregated ON/OFF fuzzy membership functions of (52) and (53) and the fuzzy UC objective constraint (45) are transformed as a function of λ using (47b). Further, the main UC constraints are imposed in the following master fuzzy formulation (54).

$$Maximize: \lambda \tag{54}$$

Subject to:

$$\lambda \sum_{g \in \Upsilon^u} \tilde{z}_g(t) - \sum_{g \in \Upsilon^u} z_g^F(t) \le 0; \forall t \tag{54a}$$

$$\lambda \sum_{g \in \Upsilon^d} \left(\mathcal{O}_g^d(t) - \tilde{z}_g(t) \right) + \sum_{g \in \Upsilon^d} z_g^F(t) \le \sum_{g \in \Upsilon^d} \left(\mathcal{O}_g^d(t) - \tilde{z}_g(t) \right); \forall \tag{54b}$$

$$(\overline{C} - \underline{C})\lambda + C(x^F) \le \overline{C} \tag{54c}$$

$$G(x_B^F, x_C^F) \le H \tag{54d}$$

$$0 \le \lambda \le 1 \in \mathbb{R}; x^F = [x_B^F, x_C^F, \lambda] \tag{54e}$$

5.3.2.1. Feasibility Proof of PM2 (54). Similar to the feasibility proof of PM1 by substituting $\lambda = 0$ as the worst-case scenario in eq. (54), the non-binding fuzzy constraints of PM2 (54a, 54b, and 54c) are presented as follows.

$$0 \le \sum_{g \in \Upsilon^u} z_g^F(t); \forall t \in \Upsilon^u \ (ON\ states) \tag{55a}$$

$$\sum_{g \in \Upsilon^d} z_g^F(t) \le \sum_{g \in \Upsilon^d} \mathcal{O}_g^d(t); \forall t \in \Upsilon^d \ (OFF\ states) \tag{55b}$$

$$\mathcal{C}(x^F) \le \overline{C} \tag{55c}$$

$$G(x^F) \le H \tag{55d}$$

For constraint (54a), the aggregated-ON states at each period are restricted by a lower bound value of 0, while in constraint (54b), the aggregated-OFF states at each period are restricted by an upper bound equal to the total number of OFF states. Furthermore, eq. (55c) remains feasible, as it conforms to the upper bound of the MILP-UC objective. The original MILP-UC constraints (55d) remain unchanged because they are independent of λ. Thus, the formulation in eq. (55) is always feasible if the original MILP-UC is feasible. Hence, the proposed formulation leverages ML decisions to exploit them. However, introducing them as non-binding fuzzy constraints ensures that they never cause infeasibility.

The two proposed fuzzy methods (48) and (54) endeavor to maximize λ, the satisfaction variable, from 0 to 1. This formulation variable equals $size(x) + 1$, the extra variable is only for. The additional number of non-binding fuzzy constraints equals $(T \bullet G + 1)$ in PM1. However, these additional fuzzy constraints are limited to $(2T + 1)$ in PM2. Since the ON-fuzzy and OFF-fuzzy membership functions are contradictory, maximizing λ would determine the best compromise solution. The PMs can be solved by any MILP solver to obtain the best-known UC solution vector (x^F) and the fuzzy objective λ to revert feasibly the ML output errors to their optimal decisions.

6. Numerical analysis

Tests are conducted for the IEEE 118-bus (54 generators and 179 transmission lines) and Polish 2383-bus (327 generators and 2896 transmission lines) systems with MATLAB© R2022b, and Gurobi© V 10.0 on a laptop with 32 GB RAM and an Intel Core™ i7–9750 CPU @ 2.6 GHz. Both systems' fuel cost functions are piecewise linear parts with 3 and 1 segments for both systems, respectively. Detailed test case parameters are available in [42]. Initially, the redundant constraints for

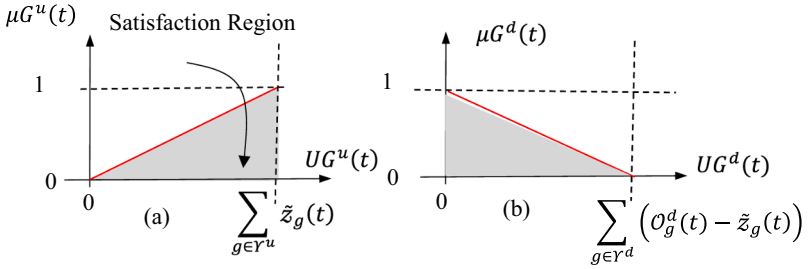


Fig. 6. Aggregated fuzzy memberships of (a) ON units and (b) OFF units.

all formulations are eliminated using the presolve process of Gurobi [43]. Eliminating superfluous constraints is extremely beneficial for decreasing RAM utilization and enhancing computing efficiency as shown in Table 2. For a fair comparison, the presolve process is used for MILP-UC and fuzzy MILP models (PM1 and PM2).

6.1. CNN-BiLSTM training

In this process, we use the MILP-UC solutions of the data set to train the CNN-BiLSTM model as described in Subsection 4.1, and data inputs are generated as detailed in [30] using daily data of the Independent Electricity System Operator (IESO), Ontario, Canada. For the 118-bus, 150 data scenarios are used to train the CNN-BiLSTM model, thus the number of training instances is 424,800 for the input data, and for the output is 194,400. For the 2383-bus system, 200 data scenarios are used for the CNN-BILSTM training, forming ~11.5 M input features and 1.6 M output parameters. The MILP-UC solutions of (1) have been obtained using Gurobi with optimality gaps of 0 and 0.0001 respectively for both systems.

The training tuning parameters and training performance for both tested cases are illustrated in Table 3. Fig. 7 shows the loss function and RMSE performance for the 188-bus system. For both systems, the entire data is partitioned as 70 % for training, 15 % for testing, and 15 % for performance validation. The ML model has been well trained with RMSE less than 0.0014 and Mean Absolute Error (MAE) less than 2E-5 for both systems, corresponding to over 99 % accuracy.

The performance comparison of different machine learning models—ANN, CNN, CNN-LSTM, and CNN-BiLSTM—is shown in Table 4.

Table 4 shows that the CNN-BiLSTM model consistently achieves the lowest RMSE values across training (0.0036), validation (0.0371), and testing (0.0695) phases, indicating superior accuracy and robustness. In contrast, the ANN model has higher RMSE values in all phases, suggesting less effectiveness in capturing complex patterns. The CNN model, while having a lower training RMSE (0.0459), exhibits significantly higher RMSE in validation (0.287) and testing (0.2827), implying overfitting. The CNN-LSTM model performs better with lower RMSE in training (0.0108) and validation (0.0412), though its testing RMSE (0.081) suggests room for improvement. Overall, the CNN-BiLSTM model demonstrates the best fit and generalization capabilities, making it the most suitable for the unit commitment task.

6.2. Computation Quality of Proposed Methods (Online)

In this test, twenty random 24-h bus-wise net loads (IESO load profiles) and generation costs are used. These data correspond to 59,880 and ~ 1.14 M input features for CNN-BiLSTM models of IEEE 118-bus and Polish 2383-bus, respectively. To demonstrate the computational quality of the two ML-fuzzy algorithms, the average solution time and UC objective for both systems are compared with the MILP-UC [6] solution by Gurobi for the same tested data. For the proposed ML-fuzzy algorithms, first, the fuzzy constraint for the MILP-UC objective function is created as in (45).

In PM1, the individual fuzzy constraints have been created using the corresponding ML generation decisions as in (48a) and (48b). Then, the complete model of PM1 is created as in (48) and solved using Gurobi for all tested data of 118-bus and 2383-bus, respectively. Following the same procedure of PM1, the aggregated ON/OFF fuzzy constraints have been modeled as in (54a) and (54b), respectively. Further, the master formulation of PM2 (54) has been solved for the tested cases. The results are reported and compared in Table 5.

Table 2
Main Data of the Tested Systems.

System	118-bus	Polish 2383-bus
# Piece-wise segments	3	1
# MILP-UC binaries	3888	23,544
# MILP-UC cont. Variables	8040	72,912
Constraints Reduction due Presolve	72 %	41 %

Table 3
CNN-BiLSTM Training Parameters and Prediction Errors.

System	118-bus	Polish 2383-bus
# of CNN Layers/Filters/ Kernel size	1/32/3	1/32/3
# of BiLSTM layers/ cells	1/100	1/100
# epochs to train	20,500	20,500
Loss Performance goal/Max time(s)	0/inf	0/inf
Mini batch size /Learning rate	128/0.001	128/0.001
#of input data instances	424,800	~ 11.5e6
# of output data instances	194,400	~ 1.6e6
Prediction Errors (RMSE/MAE)	**0.004/0.00002**	**0.0014/1.9E-6**
Training time (hr.)	0.5	2.0

6.2.1. Solution optimality and computation time

In terms of solution optimality, the suboptimality index (SOI) is defined as the difference between the UC objective obtained by the ML-fuzzy models $\mathscr{C}(x^F)$ and the main UC objective of Gurobi, divided by the main UC objective:

$$SOI = \frac{\mathscr{C}(x^F) - \mathscr{C}(x)}{\mathscr{C}(x)} \tag{56}$$

where $\mathscr{C}(x)$ is the cost of the main MILP-UC formulation of [6] and is considered a benchmark for comparison and obtained by Gurobi at an optimality gap of 0 and 0.0001 for 118-bus and 2383-bus, respectively.

6.2.1.1. IEEE 118-bus. From the results in Table 5, both PMs converge with a very minor SOI of 1.1271E-6 and 0.7654E-6 with average time reductions of 85.9 % and 92.37 %, respectively. As per Table 5, PM2 provides the best performance because the full data generated by the ML model and the added aggregate fuzzy constraints are less than the individual fuzzy constraints of PM1.

6.2.1.2. Polish system of 2383-bus. For the Polish system, SOIs are still marginal between 16.671E-6 to 83.347E-6. In addition, the average computation time reductions for both fuzzy cases are between 88.04 % (90.87 s) to 89.03 % (83.4 s). Since the ML model has been well-trained (MAE equals 1.9E-6), the individual fuzzy constraints provide lower SOI than the aggregate fuzzy model. However, if the ML model has a high degree of errors, PM2 will be more convenient as will be discussed in Subsection 6.2.4.

6.2.2. Optimized Membership (Satisfaction) Functions

The optimized membership functions are the primary factor of the ML-fuzzy model solution performance. Wherein the ML-fuzzy objective function λ provides the lowest membership function of the additional fuzzy constraints (individual or aggregate) and MILP-UC objective function. For instance, the individual membership function of the ON states of the ML provides the similarity between the optimized ON stated by the fuzzy model and the ON prediction state of the ML model.

Table 6 shows these optimized membership functions of the PMs for the two test systems. For the 118-bus system, the PMs provide a high fuzzy satisfaction factor λ over 0.982, minimizing the UC operating cost to the best-known value. It is seen from Table 6 that the minimum satisfaction function of the individual (PM1) and aggregate (PM2) decisions are equal to ones, which means that the ML generation decisions are successfully optimized to the optimal schedules obtained by MILP-UC solutions of (1). In addition, the fuzzy objective λ verifies the lowest of satisfactions which is $\mu C(x^F)$.

PMs give a high fuzzy satisfaction factor (over 0.99) for the Polish 2383-bus system, minimizing UC operational cost to the best-known value as shown in Table 6. However, in PM2 not all satisfaction

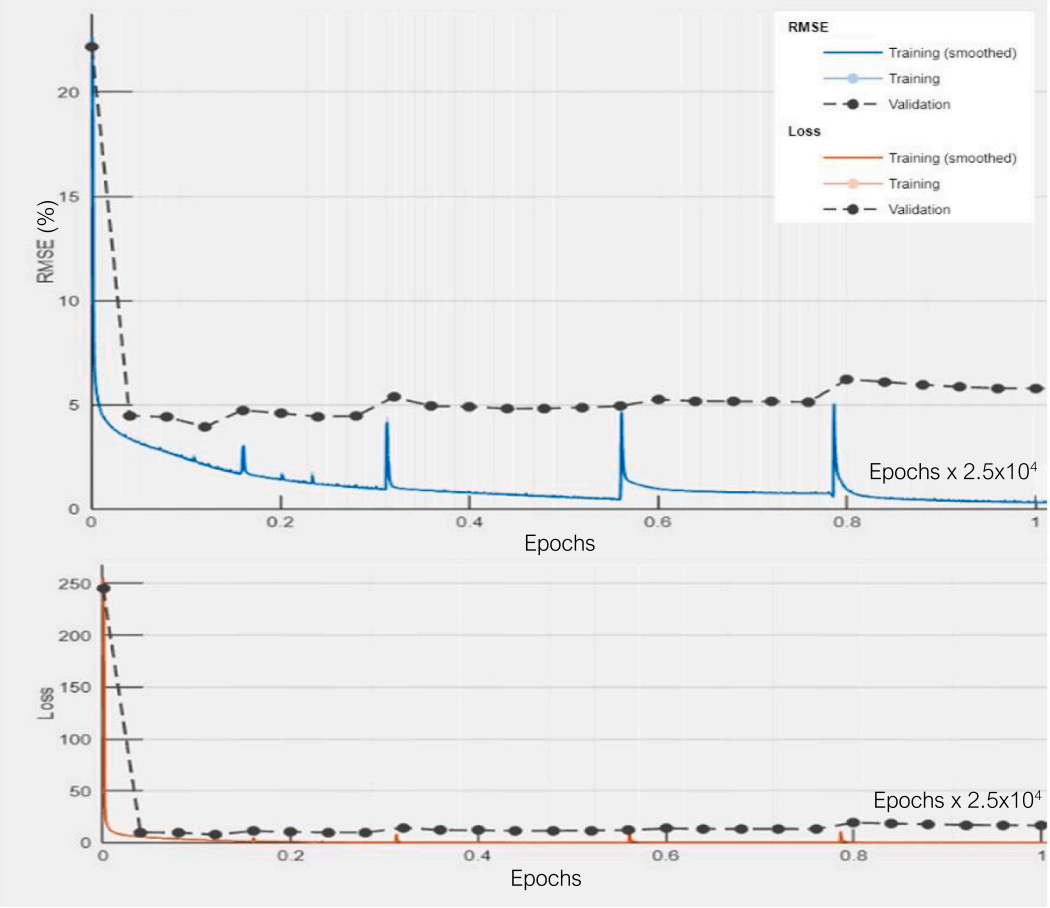


Fig. 7. Training Loss and RMSE Performance for IEEE 118-bus.

Table 4
Performance Comparison of Different Machine Learning Models.

ML Model	RMSE		
	Training	Validation	Testing
ANN	0.0498	0.073	0.0802
CNN	0.0459	0.287	0.2827
CNN-LSTM	0.0108	0.0412	0.081
CNN-BILSTM	0.0036	0.0371	0.0695

Table 5
Solutions Comparison of PMs with MILP-UC methodology.

Solution Factors	IEEE 118-bus			Polish 2383-bus		
	MILP-UC [6]	PM1	PM2	MILP-UC [6]	PM1	PM2
UC-Cost ($M)	1.1913	1.1913	1.1913	23.9924	23.9928	23.9944
SOI (xE-6)	–	1.1271	0.7654	–	16.671	83.347
Solution time(s)	16.7	2.34	1.27	759.8	83.38	90.87
CPU Time Change (%)		−85.9	−92.37		−89.03	−88.04
MILP Gap	0.0			0.0001		

All results are averaged for all tested data.

parameters are optimized to 1 as for the minimum off decision satisfaction $min_t \mu G^d(t)$, which equals 0.997, meaning there is a 0.003 mismatch between the off decision generated by PM2 and the optimal

Table 6
Optimized Membership functions.

Optimized Membership functions	IEEE 118-bus		Polish 2383-bus	
	PM1	PM2	PM1	PM2
λ	0.9822	0.9822	0.9927	0.9924
$\mu C(x^F)$	0.9822	0.9822	0.9927	0.9924
$min_{g,t}\mu I_g^u(t)/min_{g,t}\mu I_g^d(t)$	1/1	–	1/1	–
$min_t \mu G^u(t)/min_t \mu G^d(t)$	–	1/1	–	1/0.997

decisions of MILP-UC (1). The reason behind that is that PM2 uses accumulated fuzzy constraints (timewise) – thus some of the unit decisions might not be optimally scheduled. Therefore, the SOI of PM2 is larger than the SOI of PM1.

6.2.3. Performance of parameterized fuzzy methods (K-Factor)

To study the solution convergence quality for the ML-fuzzy methods, a tuning parameter (K-factor) that matches the satisfaction region is used to limit the fuzzy objective λ. The fuzzy satisfaction constraint for the MILP-UC objective function (45) is updated as follows.

$$\lambda \le \mu C\big(\mathscr{C}(x^F)\big) = K\frac{\overline{C} - \mathscr{C}(x^F)}{\overline{C} - \underline{C}} \tag{57}$$

Accordingly, the constraints (48c) and (54c) in PM1 and PM2 formulations are replaced by (57) and solved for the same tested data of Subsection 6.2. The tuning factor K is set between 0.25 and 1.25 with a step of 0.25; this range covers the satisfaction region, with one step out of the satisfaction limit.

Figs. 8 and 9 show the solution convergence of PMs with the optimal

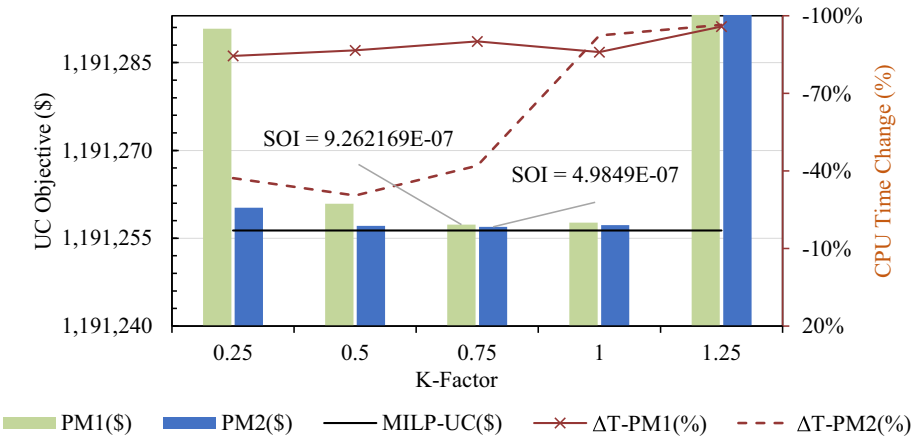


Fig. 8. Optimality Convergence of Fuzzy Methods for IEEE 118-bus.

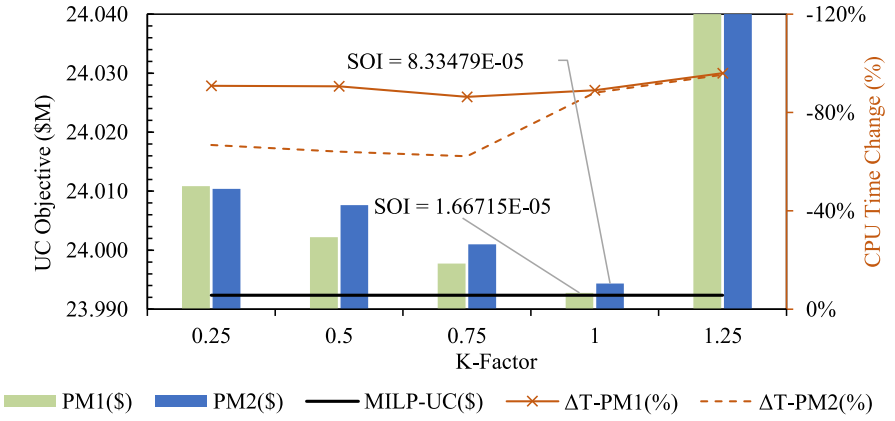


Fig. 9. Optimality Convergence of Fuzzy Methods for Polish 2383-bus.

solution obtained by MILP-UC (1) colored as a black line. As shown, all ML-fuzzy models converge with increasing K from 0.25 to 1, while at K = 1.25 all fuzzy methods diverge because it forces the fuzzy objective to be out of satisfaction region at $\lambda > 1$. It is seen that the SOI of PM2 is marginally better for 118-bus (at K = 0.75). However, PM1 is marginally better for the 2383-bus system (at K = 1). PMs provide up to 92 % CPU time reduction. From the presented results, System Operators can set K at a value closer to 1, where the PMs provide the best performance.

6.2.4. *Feasibility of the PMs with a high rate of weather-dependent sources*

In this subsection, the feasibility test explores two net load profiles, Case 1 and Case 2. Case 1 depicts IESO net load profiles with 50 data

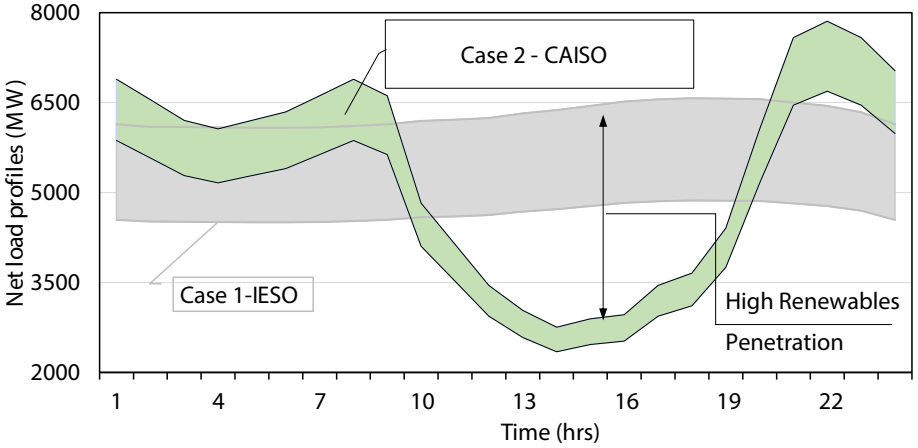


Fig. 10. Netload profiles of Case 1 and Case 2 for IEEE 118-bus.

samples (50 days) selected from Subsection 6.1 within ML-trained data. Case 2, comprising 50 netload samples, illustrates a substantial change in netload due to the high penetration of weather-dependent sources, reaching 75 % of IESO load profiles. Specifically, Case 2 showcases the challenging duck curve load profile of the California Independent System Operator (CAISO) which has never been seen before in ML-trained data.

Both netload cases are rescaled for the IEEE 118-bus system and presented in Fig. 10. The ML model exhibits a low 0.0006 % prediction error for Case 1, contrasting with a higher 24 % error for Case 2, attributed to significant intermittent renewables.

In reference [10], 90 % of ML output is used to fix the binary decisions in the MILP-UC formulation. In contrast, reference [18] utilizes 100 % of the ML predictions to fix the binary variables, effectively transforming the MILP-UC formulation into an LP-UC formulation. Reference [30] employs a different approach by using ML predictions only for the trusted generator set, defined in an offline process, to fix a portion of the UC binary variables.

The results of the proposed method are compared with the recent conventional ML-UC methods and reported in Table 7.

For both ML-fuzzy models, the solutions are always feasible because of using the individual or aggregate fuzzy satisfaction functions for the ML output. These satisfaction functions yield the system to be feasible. As shown in Table 8, for PM1 due to the ML error, the fuzzy objective λ equals zero without any further improvement that yields a higher value of SOI.

In contrast, in PM2 the minimum of the aggregate ON constraints is greater than one, which means some of the ON predicted states of ML were not optimal and the fuzzy model retains some states to be optimal. Thus, the fuzzy objective λ is greater than zero with a much lower value of SOI. For both cases data, the PMs reduce the computational time from 77 to 97.62 %. However, recent ML-UC works [10,18,30] provide an infeasibility rate from 14 % to 100 % for Case 2 data.

Table 9 compares the UC schedules and costs for one load scenario of Case 2 (unseen data) of the IEEE 118-bus system against the main MILP-UC method [6]. The main MILP-UC has a cost of 1.364 million dollars and 714 total ON/OFF states. The individual model (PM1) shows a 5.6 % cost increase and significant deviation with 58.4 % schedule similarity. In contrast, the aggregate model (PM2) has a minimal 0. 22 % cost increase and 95.0 % similarity, closely replicating the main MILP-UC schedule. Overall, from the above results, PM2 provides superior computation quality if the ML provides a high degree of ML errors. That makes this method robust to be used in practice.

6.3. Practical Considerations

6.3.1. Generalization capabilities of the CNN-BiLSTM model

Ensuring robust performance across diverse and unseen data scenarios is crucial for the proposed method. The CNN-BiLSTM model's generalization capabilities are enhanced by its hybrid architecture, which combines CNN to extract local features and BiLSTM to capture long-term dependencies. This hybrid approach improves the model's ability to generalize across different data instances. Training on a diverse and extensive dataset that includes various load profiles, generation costs, and renewable energy outputs helps the model generalize better and reduces the risk of overfitting. Regularization techniques like dropout are used to prevent overfitting by randomly dropping units during training, maintaining robust performance on unseen data. Cross-validation methods evaluate the model's performance on different data subsets, ensuring consistent and reliable results across various data splits, further enhancing robustness to unseen data.

Table 7
Feasibility Analysis and Comparison of PMs with ML-Based Recent Works for Case 1 and Case 2 Netload Data for IEEE 118-bus System.

Cases		PM1	PM2	[30]	[10]	[18]
Case 1 (Seen)	Feasibility	100 %	100 %	100 %	98 %	36 %
	SOI*	0.0	0.0	0.0	0.0	0.0
	CPU-Time*	−95 %	−92 %	−73 %	61 %	47 %
Case 2 (Unseen)	Feasibility	100 %	100 %	86 %	0 %	0 %
	SOI*	0.06	0.00187	0.2	–	–
	CPU-Time*	−97.62 %	−77.15 %	45.38 %	–	–

(*) Results are averaged for all samples and infeasible case's time is considered as 1500s.

6.3.2. Comparative analysis

In evaluating the proposed methods, PM1 and PM2, several key aspects are considered:

- ✓ **Accuracy**: PM1 treats each generator's state individually, providing finer control over each decision, whereas PM2 aggregates states, simplifying the problem and reducing the number of constraints.
- ✓ **Computational Efficiency**: PM1 has a higher computational load due to the larger number of constraints, while PM2 is more efficient due to fewer aggregated constraints.
- ✓ **Flexibility and Robustness**: Both methods ensure feasibility if the basic UC problem has a feasible solution. PM1 is advantageous for scenarios where individual generator decisions are critical, whereas PM2 is more robust to variations in aggregate predictions and better suited for large-scale systems with uncertain and extreme load variations, as illustrated in Subsection 6.2.4.

6.3.3. N-1 contingency constraints

The current approach does not incorporate N-1 constraints directly within the UC optimization model. Instead, after generating the optimal schedule, a separate N-1 contingency analysis is performed by solving N power flow scenarios, each examining the failure of one component at a time. This method is widely used in practice, where the initial schedule is optimized for cost and operational requirements through UC, while reliability is ensured through a post-schedule contingency analysis [44].

7. Conclusion

This paper introduces a novel two-stage Hybrid CNN-BiLSTM with Fuzzy MILP-UC method. In the offline stage, a hybrid CNN-BiLSTM model is trained using net hourly load data and generation costs as inputs, with the corresponding MILP-UC generation decisions as targets. In the online stage, the predicted ML generation decisions are incorporated as non-binding fuzzy constraints in two fuzzy optimization models, PM1 and PM2. The first model treats each ML decision variable as an individual fuzzy set, while the second creates a single fuzzy set per hour, encompassing all decisions for that hour. These fuzzy approaches directly integrate ML predictions into the UC formulations, eliminating the need for external feasibility checks. Of the two models, PM2 demonstrates superior computational performance, as it uses fewer fuzzy constraints than PM1. Additionally, the introduction of a tuning parameter (K-factor) further enhances both solution quality and computational efficiency. Experimental validation on the IEEE 118-bus and Polish 2383-bus systems confirms the method's effectiveness, achieving 100 % feasibility by mitigating infeasible ML predictions and reducing computation time by up to 92 % and 89 %, respectively.

Key Advantages:

- **High Accuracy and Robustness:** The hybrid CNN-BiLSTM model, combined with fuzzy optimization, ensures accurate and reliable predictions across a range of complex scenarios.
- **Flexibility and Adaptability:** Non-binding fuzzy constraints enable the model to adapt to varying operating conditions and uncertainties, maintaining robustness and feasibility.

Table 8
Optimized Membership functions for Case 2 of IEEE 118-bus.

		λ	$\mu C(x^F)$	$min_{g,t}\mu I^u_{g,t}$	$min_{g,t}\mu I^d_{g,t}$	$min_t\mu G^u_t$	$min_t\mu G^d_t$
Case 2	PM1	0.00	0.075	0.00	0.00	–	–
	PM2	0.345	0.347	–	–	1.357	0.382

Table 9
Deviation in Unit Commitment Schedule for One Load Scenario of Case 2 (unseen) of the IEEE 118-Bus.

		UC Cost (M$)	Total (#)		Identical States		Similarity in Full UC Schedule (%)
			ON States	OFF States	ON Schedule	OFF Schedule	
Main MILP-UC [6]		1.364	714	582	–	–	–
Case 2	PM1	1.440 (+5.6 %)	885	411	530	227	**58.4 %**
	PM2	1.367 (+0.22 %)	693	603	671	560	95.0 %

- **Computational Efficiency:** The method's substantial reduction in computation time makes it suitable for real-time and large-scale power system applications.

Potential Applications:

- **Real-Time Operations:** Rapid and accurate unit commitment decisions enhance the responsiveness and stability of real-time power system operations.
- **Renewable Energy Integration:** The model's robustness and adaptability make it particularly effective for systems with high renewable energy penetration, addressing variability and uncertainty.
- **Market Operations:** Improved computational efficiency and reliability can streamline market clearing and settlement processes, leading to better economic and operational outcomes.

Future Research Directions:

- **Nonlinear Constraints:** Extending the method to handle nonlinear constraints would broaden its applicability to more complex power system challenges.
- **Hybrid Optimization:** Combining this method with advanced optimization techniques like stochastic or robust optimization could enhance performance under uncertainty.
- **Scalability and Generalization:** Further exploration of the model's scalability to larger power systems and its generalization to different grid structures can provide deeper insights into its practical viability.

In summary, the proposed Hybrid CNN-BiLSTM with Fuzzy MILP-UC method offers a robust, flexible, and efficient solution to unit commitment problems. It holds significant potential for real-world power system applications, while also opening avenues for future research and optimization advancements.

CRediT authorship contribution statement

Bala Venkatesh: Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition, Formal analysis, Conceptualization. **Mohamed Ibrahim Abdelaziz Shekeew:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Jessie Ma:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

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.

Data availability

Data will be made available on request.

References

[1] Huang Y, Pardalos PM, Zheng QP. Electrical Power unit commitment. Boston, MA: Springer US; 2017. https://doi.org/10.1007/978-1-4939-6768-1.
[2] Streiffert D, Philbrick R, Ott A. A mixed integer programming solution for market clearing and reliability analysis. IEEE Power Eng. Soc. Gen. Meet. 2005, San Francisco, CA, USA: IEEE. 2005. p. 195–202. https://doi.org/10.1109/PES.2005.1489108.
[3] Yan B, Luh PB, Zheng T, Schiro DA, Bragin MA, Zhao F, et al. A systematic formulation tightening approach for unit commitment problems. IEEE Trans Power Syst 2020;35:782–94. https://doi.org/10.1109/TPWRS.2019.2935003.
[4] Yang L, Li W, Xu Y, Zhang C, Chen S. Two novel locally ideal three-period unit commitment formulations in power systems. Appl Energy 2021;284:116081. https://doi.org/10.1016/j.apenergy.2020.116081.
[5] Qiu H, Sun Q, Lu X, Beng Gooi H, Zhang S. Optimality-feasibility-aware multistage unit commitment considering nonanticipative realization of uncertainty. Appl Energy 2022;327:120062. https://doi.org/10.1016/j.apenergy.2022.120062.
[6] Chen Y, Pan F, Qiu F, Xavier AS, Zheng T, Marwali M, et al. Security-constrained unit commitment for electricity market: modeling, solution methods, and future challenges. IEEE Trans Power Syst 2022:1–14. https://doi.org/10.1109/TPWRS.2022.3213001.
[7] Pineda S, Morales JM, Jimenez-Cordero A. Data-driven screening of network constraints for unit commitment. IEEE Trans Power Syst 2020;35:3695–705. https://doi.org/10.1109/TPWRS.2020.2980212.
[8] Yang Y, Lu X, Wu L. Integrated data-driven framework for fast SCUC calculation. IET Gener Transm Distrib 2020;14:5728–38. https://doi.org/10.1049/iet-gtd.2020.0823.
[9] Yang N, Yang C, Wu L, Shen X, Jia J, Li Z, et al. Intelligent Data-Driven Decision-Making Method for Dynamic Multisequence: An E-Seq2Seq-Based SCUC Expert System. IEEE Transactions on Industrial Informatics 2022;18(5):3126–37. https://doi.org/10.1109/TII.2021.3107406.
[10] Xavier ÁS, Qiu F, Ahmed S. Learning to Solve Large-Scale Security-Constrained Unit Commitment Problems. INFORMS J Comput 2020. https://doi.org/10.1287/ijoc.2020.0976.
[11] Mohammadi F, Sahraei-Ardakani M, Trakas D, Hatziargyriou N. Machine learning assisted stochastic unit commitment during hurricanes with predictable line outages. IEEE Trans Power Syst 2021;36:5131–42. https://doi.org/10.1109/TPWRS.2021.3069443.

[12] Jiménez D, Angulo A, Street A, Mancilla-David F. A closed-loop data-driven optimization framework for the unit commitment problem: a Q -learning approach under real-time operation. Appl Energy 2023;330:120348. https://doi.org/10.1016/j.apenergy.2022.120348.
[13] Navin NK, Sharma R. A fuzzy reinforcement learning approach to thermal unit commitment problem. Neural Comput & Applic 2019;31:737–50. https://doi.org/10.1007/s00521-017-3106-5.
[14] De Mars P, O'Sullivan A. Applying reinforcement learning and tree search to the unit commitment problem. Appl Energy 2021;302:117519. https://doi.org/10.1016/j.apenergy.2021.117519.
[15] De Mars P, O'Sullivan A. Reinforcement learning and a* search for the unit commitment problem. Energy AI 2022;9:100179. https://doi.org/10.1016/j.egyai.2022.100179.
[16] Mohammadi F, Sahraei-Ardakani M, Trakas DN, Hatziargyriou ND. Machine Learning Assisted Stochastic Unit Commitment: A Feasibility Study. 2020 52nd north am. Power Symp. NAPS, Tempe, AZ, USA: IEEE. 2021. p. 1–6. https://doi.org/10.1109/NAPS50074.2021.9449789.
[17] Gurobi Optimization LLC. Gurobi optimizer-Presolve tool. https://www.gurobi.com/documentation/current/refman/presolve.html; 2023.
[18] Wu T, Angela Zhang Y-J, Wang S. Deep learning to optimize: security-constrained unit commitment with uncertain wind Power generation and BESSs. IEEE Trans Sustain Energy 2022;13:231–40. https://doi.org/10.1109/TSTE.2021.3107848.
[19] Park S, Chen W, Han D, Tanneau M, Van Hentenryck P. Confidence-aware graph neural networks for learning reliability assessment commitments. IEEE Trans Power Syst 2023:1–12. https://doi.org/10.1109/TPWRS.2023.3298735.
[20] Tang X, Bai X, Weng Z, Wang R. Graph convolutional network-based security-constrained unit commitment leveraging power grid topology in learning. Energy Rep 2023;9:3544–52. https://doi.org/10.1016/j.egyr.2023.02.042.
[21] Bhavsar S, Pitchumani R, Ortega-Vazquez MA, Costilla-Enriquez N. A hybrid data-driven and model-based approach for computationally efficient stochastic unit commitment and economic dispatch under wind and solar uncertainty. Int J Electr Power Energy Syst 2023;151:109144. https://doi.org/10.1016/j.ijepes.2023.109144.
[22] Gao Q, Yang Z, Li W, Yu J, Lu Y. Online learning of stable integer variables in unit commitment using internal information. IEEE Trans Power Syst 2023;38:2947–50. https://doi.org/10.1109/TPWRS.2023.3258699.
[23] Sayed AR, Zhang X, Wang G, Wang Y, Shaaban M, Shahidehpour M. Deep reinforcement learning-assisted convex programming for AC unit commitment and its variants. IEEE Trans Power Syst 2024:1–14. https://doi.org/10.1109/TPWRS.2023.3340674.
[24] Ramesh AV, Li X. Spatio-temporal deep learning-assisted reduced security-constrained unit commitment. IEEE Trans Power Syst 2024;39:4735–46. https://doi.org/10.1109/TPWRS.2023.3313430.
[25] Ramesh AV, Li X. Feasibility layer aided machine learning approach for day-ahead operations. IEEE Trans Power Syst 2024;39:1582–93. https://doi.org/10.1109/TPWRS.2023.3266192.
[26] Rajabdorri M, Kazemtabrizi B, Troffaes M, Sigrist L, Lobato E. Inclusion of frequency nadir constraint in the unit commitment problem of small power systems using machine learning. Sustain Energy Grids Netw 2023;36:101161. https://doi.org/10.1016/j.segan.2023.101161.
[27] Yang Y, Wu L. Machine learning approaches to the unit commitment problem: current trends, emerging challenges, and new strategies. Electr J 2021;34:106889. https://doi.org/10.1016/j.tej.2020.106889.
[28] Wang S, Fan Y, Jin S, Takyi-Aninakwa P, Fernandez C. Improved anti-noise adaptive long short-term memory neural network modeling for the robust remaining useful life prediction of lithium-ion batteries. Reliab Eng Syst Saf 2023;230:108920. https://doi.org/10.1016/j.ress.2022.108920.
[29] Wang S, Wu F, Takyi-Aninakwa P, Fernandez C, Stroe D-I, Huang Q. Improved singular filtering-Gaussian process regression-long short-term memory model for whole-life-cycle remaining capacity estimation of lithium-ion batteries adaptive to fast aging and multi-current variations. Energy 2023;284:128677. https://doi.org/10.1016/j.energy.2023.128677.
[30] Shekeew MIA, Venkatesh B. Learning-assisted variables reduction method for large-scale MILP unit commitment. IEEE Open Access J Power Energy 2023;10:245–58. https://doi.org/10.1109/OAJPE.2023.3247989.
[31] Shekeew MIA, Venkatesh B. Machine learning-additional decision constraints for improved MILP day-ahead unit commitment method. IEEE Access 2023;11:111976–90. https://doi.org/10.1109/ACCESS.2023.3323594.
[32] Saneifard S, Prasad NR, Smolleck HA. A fuzzy logic approach to unit commitment. IEEE Trans Power Syst 1997;12:988–95. https://doi.org/10.1109/59.589804.
[33] El-Saadawi MM, Tantawi MA, Tawfik E. A fuzzy optimization-based approach to large scale thermal unit commitment. Electr Power Syst Res 2004;72:245–52. https://doi.org/10.1016/j.epsr.2004.04.009.
[34] Saber AY, Senjyu T, Miyagi T, Urasaki N, Funabashi T. Fuzzy unit commitment scheduling using absolutely stochastic simulated annealing. IEEE Trans Power Syst 2006;21:955–64. https://doi.org/10.1109/TPWRS.2006.873017.
[35] Venkatesh B, Yu P, Gooi HB, Choling D. Fuzzy MILP unit commitment incorporating wind generators. IEEE Trans Power Syst 2008;23:1738–46. https://doi.org/10.1109/TPWRS.2008.2004724.
[36] Yu P, Venkatesh B. Fuzzy security constraints for unit commitment with outages. IET Gener Transm Distrib 2013;7:1516–26. https://doi.org/10.1049/iet-gtd.2012.0573.
[37] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process 1997;45:2673–81. https://doi.org/10.1109/78.650093.
[38] NPCC Reserve Task Force on Coordination of Operations Revision Review Record. NPCC Northeast Power Coordinating Council (NPCC) Regional Reliability Reference Directory # 5 Reserve. 2024.
[39] Li G, Xie S, Wang B, Xin J, Li Y, Du S. Photovoltaic Power forecasting with a hybrid deep learning approach. IEEE Access 2020;8:175871–80. https://doi.org/10.1109/ACCESS.2020.3025860.
[40] Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, Massachusetts: The MIT Press; 2016.
[41] Kingma DP. Ba J. A Method for Stochastic Optimization: Adam; 2017.
[42] Matpower Repository. IEEE 118-bus and Polish 2383-bus systems Dataset. https://matpower.org/docs/ref/matpower5.0/menu5.0.html.
[43] Gurobi Optimization. LLC. Gurobi Optimizer Version 10.0 2022. https://www.gurobi.com; 2022.
[44] Wood AJ, Wollenberg BF, Sheblé GB. Power generation, operation, and control. 3rd ed. Hoboken: John Wiley & Sons; 2014.