Benchmarking Deep Learning Architectures for Unit Commitment in an Isolated Thermal Power System

Victor Ramos-Osuna[1][0009−0009−6139−967X] ✉, José Milla[2][0009−0004−6361−6723], Juan Ignacio Pérez-Díaz[2][0000−0003−0657−8527], Daniel Fernández-Muñoz[3][0000−0003−1789−3151], and Fernando Ortega-Requena[1][0000−0003−4765−1479]

[1] Universidad Politécnica de Madrid, Department of Computer Systems, C. de Alan Turing, s/n, 28031, Madrid, Spain
{victor.ramos.osuna, fernando.ortega}@upm.es

[2] Universidad Politécnica de Madrid, Department of Hydraulic, Energy and Environmental Engineering, c/ Profesor Aranguren s/n, 28040, Madrid, Spain
j.millal@alumnos.upm.es, ji.perez@upm.es

[3] Universidad Politécnica de Madrid, Department of Physics Electronics, Electrical Engineering and Applied Physics, Avda. Complutense 30, 28040, Madrid, Spain
daniel.fernandezm@upm.es

Abstract. The Unit Commitment problem is a critical NP-hard task fundamental to power system economics. To address computational bottlenecks in traditional Mixed-Integer Programming (MIP) solvers, this paper evaluates Deep Learning architectures (Recurrent Neural Networks, one-dimensional Convolutional Neural Networks, and Transformers) to predict binary unit statuses, significantly reducing the MIP search space. We investigate the trade-off between model complexity and performance by comparing simple demand inputs against sophisticated feature sets integrating generator and system-level data. Model hyperparameters were optimized to ensure reliable and robust performance.

Using four years of operational data, split chronologically into training (Dec 2019–Dec 2021) and testing (Dec 2021–Dec 2023) periods, we employed a knee-point Pareto selection strategy to safely maximize solver coverage. Our central and somewhat counter-intuitive finding is that simple, demand-only models outperform their feature-augmented counterparts: a standard Long Short-Term Memory (LSTM) network achieves a 4.05× speed-up over the CPLEX baseline while maintaining 100% solution feasibility and a mean operational cost deviation below 0.01% from the unconstrained optimum.

Keywords: Deep Learning · Feature Engineering · Isolated Power Systems · Mixed-Integer Programming · Unit Commitment

1 Introduction

Unit Commitment (UC) is a cornerstone of power system operations, typically formulated as a large-scale Mixed-Integer Programming (MIP) problem to minimize total costs while satisfying technical constraints [9]. The combinatorial hardness of this problem, driven by minimum up/down times, ramping limits, and reserve coupling across a fleet of dispatchable units, renders the UC problem NP-hard and computationally demanding within the narrow windows required for market-clearing operations [6].

This computational bottleneck has motivated a shift toward Machine Learning (ML) surrogates [7,10]. While recent literature explores Reinforcement Learning [20] and Graph Neural Networks [17] to solve the UC, we believe that the application of sequence modeling still requires further research. This paper adopts a "*predict-then-optimize*" strategy [19], utilizing Deep Learning (DL) to predict binary commitment variables. By using these predictions to prune the MIP search space, we significantly accelerate the solution process while ensuring that the final dispatch satisfies all hard physical constraints [11, 14]. The paper makes two relevant contributions to the state-of-the art, namely:

- We systematically compare Deep Learning architectures to predict binary commitment variables.
- We create sophisticated feature sets integrating system-level data (net demand and spinning reserve requirement) and generator-level technical and economic attributes.

The remainder of this paper is organized as follows. Section 2 presents the UC problem formulation. Section 3 describes the DL architectures, feature engineering, and optimization protocol. Section 4 reports experimental results, and Section 5 concludes with directions for future work.

2 Unit Commitment Problem Formulation

The UC problem is modeled as a deterministic MIP problem using real operational data from the Great Canary island power system [4], which comprises 16 dispatchable thermal units. The generator technical characteristics, cost parameters, demand profiles, and reserve requirements correspond to this system.

2.1 Notation

Indices t, i, and q denote time steps (in hours, h), thermal units, and linear segments of the piecewise linear approximations of the hourly production cost functions, respectively. Parameters: C_i^f is the fuel cost of unit i at minimum power; C_{wc}^{pen} the penalty per MWh of wind energy curtailed; C_i^{su} the start-up cost of unit i; $C_{i,q}^v$ the slope of segment q of unit i's piecewise linear hourly production cost function; D_t the net power demand at time step t; M_i^f and M_i^v the maintenance costs of unit i per start-up and per MWh, respectively;

$P_i^{\min}$ and $P_i^{\max}$ its minimum and maximum power outputs; $P_{i,q}^{inc}$ the maximum length of segment q of unit i's production cost function; P_i^{vc} the interhourly power variation cost of unit i; $sr_d(t)$ the expected increase in power demand between t and t+1; and $sr_{wp}(t)$ the most likely wind power loss at time step t. Decision variables: $p_{i,q,t}$ is the power output of unit i in segment q at time t; $pv_{i,t}$ the power variation of unit i between $t-1$ and t; $u_{i,t} \in \{0,1\}$ equals 1 if unit i is online at time step t and 0 otherwise, and wc_t the wind power curtailment at t; and $\delta_{i,t} \in \{0,1\}$ equals 1 if unit i starts up at time step t.

2.2 Algebraic Formulation

The objective function is aims to minimize the total operation and maintenance cost (1), comprising fuel costs, maintenance costs, start-up costs, and a proxy for the cost due to the wear resulting from interhourly power output variations. The last term in the objective function is an economic penalty aimed to avoid wind curtailment whenever possible.

$$\begin{aligned} \min z = \sum_t \Bigg[\sum_i \Big(u_{i,t} C_i^f + \sum_q p_{i,q,t} C_{i,q}^v + \delta_{i,t}(M_i^f + C_i^{su}) \\ + \big(u_{i,t} P_i^{\min} + \textstyle\sum_q p_{i,q,t}\big) M_i^v + pv_{i,t} P_i^{vc} \Big) + C_{wc}^{pen} wc_t \Bigg] \end{aligned} \tag{1}$$

The objective function is subject to constraints (2)–(7). According to (2) the power demand is covered by the thermal generation and non-curtailed wind power. At each time step, (3) identifies whether unit i starts up. The initial status (on/off) of each unit u_i^0 is part of the input data set; it acts as the boundary condition at $t = 0$ that anchors the start-up logic in (3). The hourly production cost curve of each unit is modelled as a piecewise linear function. The length of each segment in the piecewise linear function is limited by (4). Interhourly power output variations are computed through (5). Two distinct spinning reserve requirements are imposed to ensure system reliability. First, (6) guarantees that the generation schedule is robust against demand and wind forecasting errors. Second, (7) enforces an $N-1$ security criterion: should any single online unit i fail abruptly, the remaining online units must retain enough unused upward capacity (spinning reserve) to instantly cover the lost output. Because this requirement must hold for every unit simultaneously, it couples all binary variables, significantly increasing computational complexity.

$$D_t + wc_t = \sum_i \left(u_{i,t} \cdot P_i^{\min} + \sum_q p_{i,q,t} \right) \quad \forall t \tag{2}$$

$$\delta_{i,t} \geq u_{i,t} - u_{i,t-1} \quad \forall i, t > 1 \tag{3}$$

$$p_{i,q,t} \leq u_{i,t} \cdot P_{i,q}^{inc} \quad \forall i, q, t \tag{4}$$

$$pv_{i,t} \geq \left| (u_{i,t} \cdot P_i^{\min} + \sum_q p_{i,q,t}) - (u_{i,t-1} \cdot P_i^{\min} + \sum_q p_{i,q,t-1}) \right| \quad \forall\ i,\ t \tag{5}$$

$$\sum_i \left(u_{i,t} \cdot \sum_q P_{i,q}^{inc} - \sum_q p_{i,q,t} \right) \geq \max(sr_d(t), sr_{wp}(t)) \quad \forall t \tag{6}$$

$$\sum_{j \neq i} \left(u_{j,t}(P_j^{\max} - P_j^{\min}) - \sum_q p_{j,q,t} \right) \geq u_{i,t} P_i^{\min} + \sum_q p_{i,q,t} \quad \forall i, t \tag{7}$$

3 Methodology

The proposed framework employs a DL model to approximate the optimal unit commitment decisions $u_{i,t}^* \in \{0, 1\}$ for all generation units i and time periods t.

3.1 Feature Engineering and Fusion

A key contribution of this study is the definition of scenario-specific feature sets that integrate both techno-economic unit characteristics and system-level demand information.

Sequence Features (X_{seq}). The baseline input for all models is the net demand profile, defined as system load minus wind generation. For augmented models, this input sequence $X_{seq} \in \mathbb{R}^{24\times3}$ is expanded to include fixed spinning reserve requirement profiles shared across all units. Notably, the predictor receives no information about the initial commitment status u_i^0; rather, it infers the daily commitment trajectory purely from the demand (and, for augmented models, reserve) sequence, delegating the enforcement of inter-temporal constraints anchored on u_i^0 (minimum up/down times and start-up logic) entirely to the downstream MIP. All dynamic input features were standardized using the empirical mean and standard deviation computed *exclusively* on the training partition (Section 3.2), which are then applied unchanged to the test partition at inference time to prevent any leakage from the evaluation horizon.

Static Attribute Features (X_{stat}). To capture unit-level physical and economic constraints, in the case of augmented models, we define a static feature vector for each generator. The static feature vector includes normalized unit-level technical and economic attributes, such as minimum and maximum generation limits, ramping-related costs, maintenance and startup costs and no-load costs. A key derived attribute is the **Aggregated Variable Cost** (C_{var}^{agg}), which compresses the stepwise production cost curve into a single scalar reflecting economic efficiency:

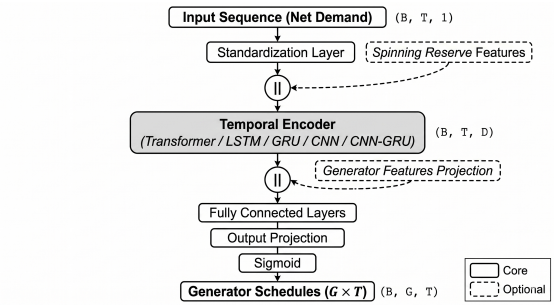


Fig. 1. Proposed Feature Fusion Architecture. The sequence input block (X_{seq}) carries the net demand profile, concatenated with the standardized reserve features for augmented models, and feeds the temporal encoder. The encoded representations are then fused with the static projection block (X_{stat}), which holds the per-generator technical and economic attributes, to predict hourly unit commitments. Labels X_{seq} and X_{stat} in the diagram identify, respectively, the temporal-encoder input and the static-attribute projection branches described in Section 3. Dimensions: B (batch size), T (time in hours), G (generator), D (projection dimension).

$$C^{agg}_{var,i} = \frac{\sum_{q \in Q} C^{v}_{i,q} \cdot P^{inc}_{i,q}}{P^{max}_{i}}. \tag{8}$$

All static features were standardized across units prior to training to ensure numerical stability and comparability.

3.2 Data Partitioning

The study uses four years of operational data from the Great Canary island power system. To rigorously assess generalization to unseen demand and seasonal conditions, the dataset was split chronologically: **Training** from December 19, 2019 to December 19, 2021 (20:00), used for model fitting and hyperparameter optimization; **Testing** from December 19, 2021 to December 19, 2023 (20:00), reserved exclusively for out-of-sample evaluation.

3.3 Deep Learning Architectures

This study benchmarks multiple sequence-to-sequence DL architectures for predicting 24-hour binary unit commitment trajectories from daily net demand inputs. The evaluated models serve as predictors used to reduce the MIP search space following the structure seen in Fig. 1.

Recurrent Models. We evaluate LSTM [16], GRU [2], and BiLSTM variants for their ability to capture the long-term temporal dependencies inherent in thermal UC decisions [12,13]: 32–256 hidden units, 1–4 stacked layers, dropout up to 0.5, and a dense projection head mapping latent states to hourly commitments.

Convolutional Models. Causal 1D temporal convolutions extract local patterns in demand trajectories [18]: 2–4 layers, 16–128 filters, kernel sizes 3–9, followed by a dense projection head.

Hybrid CNN–GRU. The hybrid CNN–GRU couples convolutional feature extraction with recurrent modeling to jointly capture short- and long-term dependencies [5], reusing the standalone CNN and GRU hyperparameter ranges for comparable complexity.

Transformer. Transformer encoders [15] use multi-head self-attention to model global temporal dependencies: model dimension 16–128, 1–4 encoder layers, 2–8 attention heads, with sinusoidal or learnable positional encodings.

Static Feature Embedding. For augmented models, static unit attributes are projected into a low-dimensional embedding (8–16 units) and fused with the temporal representation before prediction, with the embedding size capped to limit overfitting.

3.4 Optimization and Selection Protocol

We employ a multi-objective optimization framework using Optuna [1] to maximize both predictive accuracy and decision coverage. Hyperparameters are optimized via the NSGA-II algorithm [3], an evolutionary procedure that identifies a set of optimal trade-offs (the Pareto frontier) between conflicting objectives, over 100 trials, using a chronological 80/20% split (Dec 2019–Dec 2021) to preserve temporal causality. The 100-trial budget was chosen because the non-dominated front stabilized well before this limit, with later trials yielding only marginal, non-improving candidates. The search covered structural parameters (hidden/model dimension, layers, attention heads), dropout $\sim \mathcal{U}(0, 0.5)$, batch size, and a log-uniform initial learning rate; the retained configurations used a batch size of 32 and per-architecture learning rates spanning 5×10^{-4}–6×10^{-3}. Models train for up to 250 epochs using Adam with cosine-annealing scheduling and early stopping (patience 20) on the validation loss.

Objective Functions and Pareto Selection. To ensure all instances contribute equally regardless of the number of variables fixed per day, the optimization targets are computed per sample and then averaged. Let $s \in \{1, \ldots, N_s\}$ denote a sample (a daily scheduling instance). For a candidate model m and sample s, the high-confidence subset of variables is:

$$\mathcal{F}_s^{(m)} = \left\{ (i,t) \mid p_{i,t,s}^{(m)} \leq 0.1 \vee p_{i,t,s}^{(m)} \geq 0.9 \right\}, \tag{9}$$

where $p_{i,t,s}^{(m)}$ is the predicted probability that unit i is online at time t for sample s.

The selective coverage for an individual sample s is the proportion of variables meeting these thresholds: $c_s^{(m)} = |\mathcal{F}_s^{(m)}|/(N_i \times N_t)$, where $N_i \times N_t$ represents the total number of binary variables in a single daily instance.

The selective accuracy for sample s is evaluated solely over the fixed variables:

$$a_s^{(m)} = \frac{1}{|\mathcal{F}_s^{(m)}|} \sum_{(i,t)\in\mathcal{F}_s^{(m)}} \mathbf{1}\left(\hat{u}_{i,t,s}^{(m)} = u_{i,t,s}^*\right), \tag{10}$$

where $\hat{u}_{i,t,s}^{(m)} = \mathbf{1}(p_{i,t,s}^{(m)} \geq 0.5)$ and $u_{i,t,s}^*$ is the MIP optimum. If $|\mathcal{F}_s^{(m)}| = 0$, then $a_s^{(m)} = 0$.

The final objective functions evaluated by the multi-objective optimizer are the global means across all N_s samples:

$$\mathcal{M}_{cov}(m) = \frac{1}{N_s}\sum_{s=1}^{N_s} c_s^{(m)}, \quad \mathcal{M}_{acc}(m) = \frac{1}{N_s}\sum_{s=1}^{N_s} a_s^{(m)}. \tag{11}$$

The optimization yields a Pareto frontier $\mathcal{P}$ representing the risk-coverage trade-off. To prioritize system security, we filter $\mathcal{P}$ to obtain a viable candidate subset Ω satisfying strict operational minimums: $\tau_{cov} = 85.0\%$ and $\tau_{acc} = 99.0\%$. These values follow a security-first rationale. The accuracy floor is deliberately high (99%): since fixed variables leave the solver's search, a single wrong fixing can render the daily MIP infeasible and trigger the recovery loop (Section 3), so the bound sits just below perfect reliability to keep such events rare while still admitting candidates. The coverage floor (85%) is the minimum search-space reduction at which the speed-up reliably exceeds inference overhead. Models failing these boundaries are discarded to avoid infeasible commitment schedules.

From Ω, the optimal model is selected by locally-normalized knee-point detection. Each candidate's coverage and accuracy are re-normalized to $[0, 1]$ over the local bounds of Ω:

$$\hat{x}_k = \frac{x_k - \min_{j\in\Omega}(x_j)}{\max_{j\in\Omega}(x_j) - \min_{j\in\Omega}(x_j)}, \tag{12}$$

where x_k is the raw metric (accuracy or coverage) for candidate k and j indexes Ω. The chosen model maximizes the perpendicular distance to the trade-off line connecting $(0, 1)$ and $(1, 0)$:

$$d(m) = \frac{|\hat{\mathcal{M}}_{cov}(m) + \hat{\mathcal{M}}_{acc}(m) - 1|}{\sqrt{2}}, \tag{13}$$

identifying the point of diminishing returns among the high-reliability candidates.

Final Production Training. Each selected configuration was retrained on the consolidated training set (Dec 2019–Dec 2021) for the epoch count that gave its best validation performance during the search, after which weights were frozen for unbiased evaluation on the unseen two-year test horizon.

3.5 Adaptive Variable Fixing Strategy

To mitigate the risk of infeasibility in the testing period, we employ an **Iterative Feasibility Recovery** algorithm grounded in selective classification principles. Binary decision variables are fixed when predicted probabilities fall outside a confidence band $(\tau_{low}, \tau_{high})$. If the resulting MIP is infeasible, the thresholds are iteratively tightened (e.g., $(0.1, 0.9) \rightarrow (0.01, 0.99)$), progressively unfixing uncertain variables until feasibility is restored. If feasibility is still not achieved when $(\tau_{low}, \tau_{high}) = (0.0001, 0.9999)$, the fixing strategy is abandoned and the CPLEX solver is applied to the original formulation.

4 Experimental Results

This section evaluates the computational efficiency and solution quality of the proposed architectures over the 730 days of the test horizon. All simulations were performed on an Ubuntu 20.04.6 LTS workstation equipped with an Intel Xeon Gold 6230 CPU (2.10 GHz) and 256 GB of RAM. No GPU acceleration was utilized, ensuring the benchmark reflects pure CPU-based optimization performance.

4.1 Computational Performance

Table 1 summarizes the performance of all benchmarked architectures. The baseline MIP solver requires an average of 0.785 seconds per day. CPLEX was executed via GAMSPy; default settings were used except for a target relative optimality gap of 0.1% (`epgap=0.001`), a search strategy emphasizing optimality (`mipemphasis=2`), and a deterministic parallel search utilizing up to 24 threads, all constrained by a 120 s per-instance time limit. The DL-assisted runs reuse this exact configuration and differ only in fixing high-confidence binaries beforehand, ensuring a fair comparison. While robust, this baseline is outpaced by all DL-based methods, which achieve **2.40x to 4.05x** acceleration.

Two points on solution quality follow. Although Table 1 omits selective accuracy (every selected model meets the $\tau_{acc} = 99\%$ floor by construction), validation-phase selective accuracies were tightly clustered in 99.8–99.99%, so no model merely grazes the threshold. Moreover, fixing high-confidence variables does not degrade the economic optimum: the optimality gap after neural fixing stays at the baseline's order of magnitude ($\approx 5{\times}10^{-4}$), occasionally tighter, confirming the pruning removes redundant branches without excluding the global optimum. The resulting cost deviation is the **Cost** Δ column of Table 1, below 0.05% for every model.

Complexity and Robustness. The **100% feasibility** across all architectures shows the hybrid framework guaranteed a solution for every test instance. This is a joint property of the predictor and the adaptive recovery loop rather than predictive accuracy alone: the loop relaxes fixings whenever infeasibility arises,

so feasibility is ensured by design, while predictive quality governs how often it must intervene. Inference cost scaled with complexity: negligible for standard models ($\approx 2\,\text{ms/day}$) but up to 64.59 ms for the Extra-feature GRU; even so, total solution time stayed well below the CPLEX baseline for every model.

Table 1. Benchmark over the 730-day test period. **Inf.**: mean per-day inference time (s); **Total**: per-day inference+MIP time (s); **Speed-up**: CPLEX-to-model total-time ratio; **Fixed**: mean fraction of binaries fixed (selective coverage); **Cost Δ**: mean relative cost deviation from the unconstrained CPLEX optimum; **Feas.**: fraction of feasible test days. Values are mean±std; best speed-up and coverage in bold.

Architecture	Type	Inf. (s)	Total (s)	Speed-up	Fixed	Cost Δ	Feas.
CPLEX	*Baseline*	–	0.785±0.370	1.00x	0.0%	–	100%
Recurrent Neural Networks (RNNs)							
LSTM	Basic	0.002±0.001	0.198±0.058	**4.05x**	92.6±2.5%	0.00%	100%
LSTM	Extra Feats.	0.009±0.005	0.232±0.075	3.50x	92.3±2.4%	0.02%	100%
BiLSTM	Basic	0.008±0.003	0.303±0.111	2.72x	92.5±2.7%	0.00%	100%
BiLSTM	Extra Feats.	0.022±0.007	0.307±0.113	2.70x	92.5±2.4%	0.01%	100%
GRU	Basic	0.003±0.001	0.230±0.073	3.52x	**93.1±2.4%**	0.00%	100%
GRU	Extra Feats.	0.065±0.011	0.337±0.105	2.40x	92.6±2.4%	0.00%	100%
Convolutional Architectures							
CNN	Basic	0.012±0.005	0.280±0.143	3.11x	87.9±5.2%	0.04%	100%
CNN	Extra Feats.	0.017±0.005	0.285±0.100	2.88x	88.9±2.7%	0.01%	100%
CNN-GRU	Basic	0.005±0.001	0.241±0.085	3.36x	92.0±3.3%	0.01%	100%
CNN-GRU	Extra Feats.	0.045±0.009	0.297±0.101	2.76x	92.1±2.6%	0.01%	100%
Attention Mechanisms							
Transformer	Basic	0.002±0.002	0.285±0.098	2.91x	92.7±2.4%	0.00%	100%
Transformer	Extra Feats.	0.017±0.006	0.315±0.116	2.63x	91.7±2.5%	0.00%	100%

Statistical Significance Analysis. Paired Wilcoxon signed-rank tests across the 730-day horizon show that feature augmentation significantly alters computational performance: LSTM, GRU, CNN-GRU, and Transformer models differ significantly in total solution time ($p < 10^{-4}$), with Basic variants consistently faster, and coverage differences are significant for LSTM ($p < 10^{-10}$), GRU ($p < 10^{-40}$), and Transformer ($p < 10^{-90}$). A two-sample Kolmogorov–Smirnov test further confirms that Basic and Extra configurations produce statistically distinct probability distributions across all architectures ($p < 10^{-100}$). Thus, static inputs reshape the predictive landscape and confidence calibration, but these shifts do not systematically translate into improved prescriptive performance.

Threshold Adaptation Stability. The adaptive recovery mechanism was highly stable over the 730-day horizon: the initial band ($\tau_{low} = 0.1, \tau_{high} = 0.9$) preserved feasibility in the vast majority of instances, with the LSTM Base requiring tightening on only 17 days (2.3%) and the Transformer Base on 2 (0.3%). The CNN Base, consistent with its lower coverage, needed the most adjustment (tightening to 0.99 on 146 days, 0.999 on 23, and 0.9999 on 4). Crucially, iterative tightening restored feasibility in every problematic instance, so the ultimate

fallback—solving the unreduced formulation with CPLEX—was never invoked, confirming the learned distributions were well-calibrated and the fixing strategy robust.

4.2 Efficiency vs. Reliability Trade-off

To isolate search-space pruning from architectural latency, we compute Spearman's rank correlation between coverage and solve time, more robust than linear metrics for the non-linear dynamics of MIP solvers. The consistent negative relationship (r_s from -0.32 to -0.73) confirms that binary fixing reduces branch-and-bound effort. The gains, however, show diminishing returns beyond 92% coverage: the GRU Base attains the highest coverage (93.1%) yet no proportional time reduction over the LSTM Base (92.5%), suggesting a "combinatorial core" of highly constrained variables that dictates the residual complexity once the fundamental UC constraints are met.

4.3 Role of Feature Augmentation

Although feature augmentation significantly modified the underlying probability distributions (KS tests), these shifts did not improve combinatorial reduction or solve time; instead, the inference overhead of processing static inputs dominated any marginal gains in confidence, most starkly for the GRU, whose inference time grew by an order of magnitude with no coverage gain. We therefore conclude that, for this isolated thermal system and our current feature-fusion implementation, static augmentation adds overhead that outweighs any predictive benefit, with temporal demand patterns alone providing sufficient structure for effective variable fixing. We frame this as a property of the present setting rather than a general verdict; richer systems or alternative fusion mechanisms may yet benefit, as discussed in Section 5.

5 Conclusion and Future Work

We benchmarked DL architectures to accelerate the Unit Commitment (UC) problem in isolated power systems. Results demonstrate that **computational simplicity outperforms feature augmentation**. The **standard LSTM** provided the optimal trade-off, achieving a **4.05x speed-up** over the CPLEX baseline with minimal inference latency ($\approx$2 ms) and 92.6% fixing coverage. Statistical tests ($p < 10^{-4}$) confirmed that static feature augmentation altered probabilistic outputs without improving prescriptive performance. Furthermore, the framework is system-independent by design—nothing in the predictor or the fixing-and-recovery loop is specialized to this particular unit set—and it maintained 100% solution feasibility across the testing horizon by utilizing a highly stable adaptive variable-fixing strategy. Two limitations bound these conclusions. First, the study targets a single isolated system of 16 thermal units; scalability to substantially larger UC instances remains to be demonstrated. Second, our

evaluation compares the proposed architectures against one another and a pure CPLEX baseline rather than against published ML-assisted UC methods, since these are reported on different test systems and metrics; a standardized cross-system benchmark would be needed for a fair head-to-head. Future work will extend this methodology to the upcoming Great Canary island hydro-thermal system with Pumped-Hydro Energy Storage [8], where complex state-dependent constraints may finally justify augmented architectures.

Acknowledgments. This research has been partially funded by the Research Council of Norway under the Knowledge-building Project for Industry "Deep integration between machine learning approaches and renewable energy optimization (InterOpt)" (Project No. 352879).

AI-assisted technologies disclosure The authors used a Large Language Model (LLM) for translation and language editing to improve readability. All AI-assisted modifications were thoroughly reviewed, and the authors take full responsibility for the final manuscript's content.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2623–2631. KDD '19, Association for Computing Machinery, New York, NY, USA (Jul 2019). https://doi.org/10.1145/3292500.3330701
2. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (Dec 2014). https://doi.org/10.48550/arXiv.1412.3555
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (Apr 2002). https://doi.org/10.1109/4235.996017
4. Fernández-Muñoz, D., Pérez-Díaz, J.I.: Contribution of non-conventional pumped-storage hydropower plant configurations in an isolated power system with an increasing share of renewable energy. IET Renewable Power Generation **14**(4), 658–670 (2020). https://doi.org/10.1049/iet-rpg.2019.0874
5. Keren, G., Schuller, B.: Convolutional RNN: An enhanced model for extracting features from sequential data. In: 2016 International Joint Conference on Neural Networks (IJCNN). pp. 3412–3419 (Jul 2016). https://doi.org/10.1109/IJCNN.2016.7727636
6. Kumar, V., Naresh, R., Singh, A.: Investigation of solution techniques of unit commitment problems: A review. Wind Engineering **45**(6), 1689–1713 (Dec 2021). https://doi.org/10.1177/0309524X21992446
7. Milla, J., Pérez-Díaz, J.I.: The Use of Artificial Intelligence for the Unit Commitment Problem: State of the Art. In: Belenguer, E., Beltran, H. (eds.) ELECTRIMACS 2024. pp. 713–725. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-73921-7_54

8. Ministerio para la Transición Ecológica y el Reto Demográfico: Sara Aagesen: "El proyecto de Chira-Soria es una excelente muestra de que España sigue apostando por la agenda verde" (Mar 2025), https://www.miteco.gob.es/es/prensa/ultimas-noticias/2025/marzo/sara-aagesen—el-proyecto-de-chira-soria-es-una-excelente-muest.html, accessed: 2026-05-07
9. Montero, L., Bello, A., Reneses, J.: A Review on the Unit Commitment Problem: Approaches, Techniques, and Resolution Methods. Energies **15**(4), 1296 (Jan 2022). https://doi.org/10.3390/en15041296
10. Mosavi, A., Salimi, M., Faizollahzadeh Ardabili, S., Rabczuk, T., Shamshirband, S., Varkonyi-Koczy, A.R.: State of the Art of Machine Learning Models in Energy Systems, a Systematic Review. Energies **12**(7), 1301 (Jan 2019). https://doi.org/10.3390/en12071301
11. Ramesh, A.V., Li, X.: Feasibility Layer Aided Machine Learning Approach for Day-Ahead Operations. IEEE Transactions on Power Systems **39**(1), 1582–1593 (Jan 2024). https://doi.org/10.1109/TPWRS.2023.3266192
12. Ramesh, A.V., Li, X.: Spatio-Temporal Deep Learning-Assisted Reduced Security-Constrained Unit Commitment. IEEE Transactions on Power Systems **39**(2), 4735–4746 (Mar 2024). https://doi.org/10.1109/TPWRS.2023.3313430
13. Salman, D., Kusaf, M.: Short-Term Unit Commitment by Using Machine Learning to Cover the Uncertainty of Wind Power Forecasting. Sustainability **13**(24), 13609 (Jan 2021). https://doi.org/10.3390/su132413609
14. Schmitt, S., Harjunkoski, I., Giuntoli, M., Poland, J., Feng, X.: Fast Solution of Unit Commitment Using Machine Learning Approaches. In: 2022 IEEE 7th International Energy Conference (ENERGYCON). pp. 1–6 (May 2022). https://doi.org/10.1109/ENERGYCON53164.2022.9830191
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6000–6010. NIPS'17 (2017)
16. Vennerød, C.B., Kjærran, A., Bugge, E.S.: Long Short-term Memory RNN (May 2021). https://doi.org/10.48550/arXiv.2105.06756
17. Wei, L., Ai, X., Fang, J., Cui, S., Gao, L., Li, K., Wen, J.: Data-augmentation acceleration framework by graph neural network for near-optimal unit commitment. Applied Energy **377**, 124332 (Jan 2025). https://doi.org/10.1016/j.apenergy.2024.124332
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 Transactions on Sustainable Energy **13**(1), 231–240 (Jan 2022). https://doi.org/10.1109/TSTE.2021.3107848
19. Xavier, Á.S., Qiu, F., Ahmed, S.: Learning to Solve Large-Scale Security-Constrained Unit Commitment Problems. INFORMS Journal on Computing **33**(2), 739–756 (May 2021). https://doi.org/10.1287/ijoc.2020.0976
20. Zhu, Y., Cui, G., Liu, A., Jia, Q.S., Guan, X., Zhai, Q., Guo, Q., Guo, X.: A Reinforcement Learning Embedded Surrogate Lagrangian Relaxation Method for Fast Solving Unit Commitment Problems. IEEE Transactions on Power Systems **40**(5), 3806–3818 (Sep 2025). https://doi.org/10.1109/TPWRS.2025.3529700