Proceedings of CONF-FMCE 2026 Symposium: Smart City and Infrastructure Engineering
DOI: 10.54254/2755-2721/2026.CH34369

Uncertainty-Aware Prediction-to-Control: Quantile Forecasting-Driven Constrained Energy Scheduling

Weiwei Zhang

School of Mathematics and Statistics, Anqing Normal University, Anqing, China
zwwtau@163.com

Abstract. This paper proposes an integrated "Prediction-to-Control" data-driven scheduling framework for building and campus-level energy management. The framework first employs quantile forecasting and uncertainty estimation to obtain multi-step future load intervals, rather than single-point predictions, so that the variability and distributional characteristics of demand can be explicitly captured. These probabilistic forecasts are then embedded into constrained scheduling problems, where the "constraint violation probability" (e.g., peak power limits, thermal comfort bounds, and electricity cost objectives) is directly controlled through chance-constrained formulations. By transforming uncertainty into tractable constraints, the method enables a systematic trade-off between operational cost and reliability. Compared with traditional point forecast–based control strategies, the proposed approach is more robust under distribution shifts such as extreme weather conditions or occupancy fluctuations. In addition, it provides an interpretable risk–cost relationship, allowing operators to adjust risk tolerance in a transparent manner while maintaining stable and efficient system performance.

Keywords: Quantile forecasting, uncertainty quantification, chance-constrained optimization, building energy management, model predictive control, deep learning

1. Introduction

Building operators rarely optimize for cost alone. In daily practice, they must also respect peak limits, maintain comfort, and handle uncertainties from weather and occupancy [1-4]. The common engineering workflow is still "forecast first, optimize later" with point predictions.

The main question in this paper is therefore practical: how can predictive uncertainty be used directly in control, rather than reported only as a post-hoc confidence interval? To answer this, we couple a Quantile LSTM forecaster with chance-constrained MPC, so that the selected quantiles determine feasible operating decisions under explicit risk bounds.

This integration yields two advantages. First, it provides a direct mapping from uncertainty levels to violation probabilities, which makes the optimization problem interpretable for non-ML operators. Second, it creates a tunable reliability–cost frontier through risk parameters, aligning with recent building MPC and learning-assisted control literature while remaining computationally tractable [5, 6].

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.CH34369

On the UCI Appliances benchmark, the proposed pipeline consistently lowers violation rate relative to deterministic MPC with only modest cost increase, and remains stable under representative distribution shifts. The rest of the paper is organized as follows: Section II details the method, Section III describes experimental settings, Section IV presents results and analysis, and Section V concludes.

2. Methodology

This section presents the proposed Uncertainty-Aware Prediction-to-Control framework, consisting of three main components: (1) problem formulation and system modeling, (2) Quantile LSTM forecasting model, and (3) chance-constrained scheduling optimization. Fig. 1 illustrates the overall system architecture.

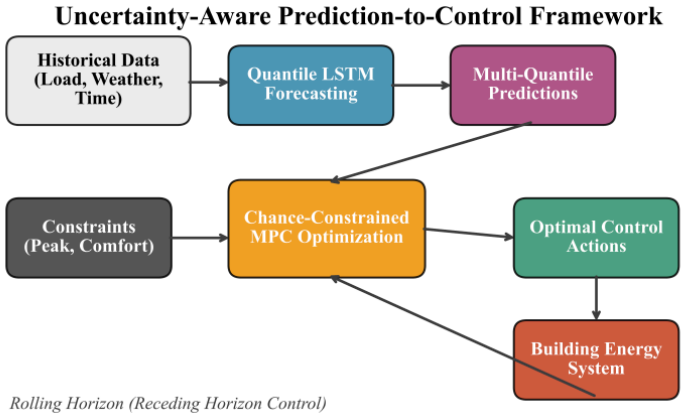


Figure 1. Overview of the proposed uncertainty-aware prediction-to-control framework. the Q-LSTM module generates multi-quantile forecasts that are fed into the chance-constrained MPC for uncertainty-aware scheduling decisions

2.1. Problem formulation and system model

2.1.1. Building energy system dynamics

We consider a building energy management system with controllable loads (HVAC, lighting), energy storage (batteries), and grid connection. The system state at time t is characterized by the state vector $x_t \in \mathbb{R}^{n_x}$, which includes:

- Indoor zone temperatures: $T_t^{zone} \in \mathbb{R}^{n_z}$
- Battery state of charge: $SoC_t \in [0,1]$
- Historical load features: $y_{t-L:t-1} \in \mathbb{R}^L$

The control input$u_t \in \mathbb{R}^{n_u}$ includes HVAC setpoint adjustments, battery charge/discharge power, and demand response actions. The stochastic system dynamics are modeled as:

$$x_{t+1} = f(x_t, u_t, w_t) \tag{1}$$

where$f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ represents the state transition function, and wt captures stochastic disturbances including weather variations, occupancy changes, and uncon- trollable loads.

DOI: 10.54254/2755-2721/2026.CH34369

2.1.2. Objective function

The operational cost over a horizon H is given by:

$$\begin{aligned} J(u,y) \quad &= \sum_{k=0}^{H-1} c_{t+k}^{energy} \cdot y_{t+k} + c^{peak} \cdot \max_{k} y_{t+k} \\ &+ \sum_{k=0}^{H-1} c^{comfort} \cdot \phi\left(T_{t+k}\right) \end{aligned} \tag{2}$$

where the three terms represent energy cost, peak demand charge, and comfort penalty respectively. Here $c_{t+k}^{energy}{}_{t}$ is the time-varying electricity price, $c^{peak}$is the peak demand charge rate (applied once to the maximum power), $y_{t+k}$is the net power consumption at time $t+k$, and $\phi\left(T_{t+\kappa}\right)$is a thermal comfort penalty function:

$$\phi\left(T\right) = \begin{cases} \alpha\left(T^{min} - T\right)^2 & if T < T^{min} \\ \alpha(T - T^{max})^2 & if T > T^{max} \\ 0 & otherwise \end{cases} \tag{3}$$

With T_{min} and T_{max} defining the acceptable temperature range and$\alpha > 0$being the penalty weight.

2.1.3. Operational constraints

The system operates under multiple constraints:

Peak Power Constraint:

$$y_{t+k} \leq \overline{y}^{peak}, \quad \forall k \in \{0,1,\ldots,H-1\} \tag{4}$$

Thermal Comfort Constraint:

$$T^{min} \leq T_{t+k}^{zone} \leq T^{max}, \quad \forall k \tag{5}$$

Battery Constraints:

$$SoC^{min} \leq SoC_{t+k} \leq SoC^{max} \tag{6}$$

$$-P_{max}^{dis} \leq P_{t+k}^{bat} \leq P_{max}^{ch} \tag{7}$$

Battery Dynamics:

$$SoC_{t+k+1} = SoC_{t+k} + \frac{\Delta t}{E^{cap}} \left(\eta^{ch} P_{t+k}^{ch} - \frac{P_{t+k}}{\eta^{dis}}\right) \tag{8}$$

where E^{cap} is battery capacity, and η^{ch}, η^{dis} are charging and discharging efficiencies.

DOI: 10.54254/2755-2721/2026.CH34369

2.2. Quantile LSTM forecasting model

2.2.1. Model architecture

We employ a Quantile Long Short- Term Memory (Q-LSTM) network to generate probabilistic multi-step forecasts. The model uses two stacked LSTM layers followed by parallel quantile output heads.

Input Layer: the Feature vector $x_t \in \mathbb{R}^d$ containing:

$$x_t = \left[y_{t-L:t-1}, T^{out}_{t-L:t-1}, T^{in}_{t-L:t-1}, Hum_t, DoW_t, HoD_t\right]^\top \tag{9}$$

Where L is the lookback window, T^{out} and T^{in} are outdoor and indoor temperatures, Hum is humidity, DoW is day-of-week encoding, and HoD is hour-of-day encoding.

LSTM Layers: Two stacked LSTM layers with hidden dimensions $h_1 = 128$ and $h_2 = 64$ are used to encode temporal dependencies.

Quantile Output Layer: For a set of quantile levels $Q = \{\tau_1, \tau_2, \ldots, \tau_m\} (e.g., Q = \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\})$, the output layer produces:

$$\hat{q}^{(j)}_{t+k} = W^{(j)}_q h_t + b^{(j)}_q, \quad \forall j \in \left\{1, \ldots, m\right\}, k \in \left\{1, \ldots, H\right\} \tag{10}$$

2.2.2. Loss function

The model is trained by minimizing the multi-quantile pinball loss:

$$\mathcal{L}_{pinball} = \frac{1}{N \cdot H \cdot m} \sum_{i=1}^{N} \sum_{k=1}^{H} \sum_{j=1}^{m} \rho_{\tau_j}\left(y_{i,t+k} - \hat{q}^{(\tau_j)}_{i,t+k}\right) \tag{11}$$

where the pinball loss function for quantile τ is:

$$\rho_\tau\left(e\right) = \begin{cases} \tau \cdot e & \text{if } e \geq 0 \\ (\tau - 1) \cdot e & \text{if } e < 0 \end{cases} \tag{12}$$

To ensure non-crossing quantiles $\left(i.e., \hat{q}^{(\tau_i)} \leq \hat{q}^{(\tau_j)} \text{ for } \tau_i < \tau_j\right)$, we add a crossing penalty:

$$\mathcal{L}_{cross} = \sum_{i=1}^{m-1} \max\left(0, \hat{q}^{(\tau_i)} - \hat{q}^{(\tau_{i+1})} + \delta\right) \tag{13}$$

where $\delta > 0$ is a small margin. The total loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{pinball} + \lambda \mathcal{L}_{cross} \tag{14}$$

2.2.3. Uncertainty calibration

To ensure well-calibrated prediction intervals, we apply isotonic regression-based recalibration on the validation set [7]. For each quantile level τ, we learn a monotonic mapping $g_\tau : [0,1] \to [0,1]$ such that:

DOI: 10.54254/2755-2721/2026.CH34369

$$\mathbb{P}\left(\mathrm{Y} \leq \widehat{\mathrm{q}}^{(\mathrm{g}_{\tau}(\tau))} \approx \tau\right. \tag{15}$$

2.3. Chance-constrained model predictive control

2.3.1. Stochastic optimization formulation

Given the quantile forecasts from the Q-LSTM model, we formulate the chance- constrained MPC problem. Let$\mathrm{y}_{\mathrm{t+k}}$denote the actual (random) energy consumption at future time $\mathrm{t}+\mathrm{k}$. The goal is to minimize expected operational costs while satisfying constraints with specified probability:

$$\min_{\mathrm{u_t:t+H-1}} \mathbb{E}\left[\mathrm{J}\left(\mathrm{u}, \mathrm{y}\right)\right] \tag{16}$$

subject to:

$$\mathbb{P}\left[\mathrm{y}_{\mathrm{t+k}} \leq \bar{\mathrm{y}}^{\mathrm{peak}}\right] \geq 1-\epsilon_{\mathrm{peak}}, \ \forall \mathrm{k} \tag{17}$$

$$\mathbb{P}\left[\mathrm{T}^{\min} \leq \mathrm{T}_{\mathrm{t+k}} \leq \mathrm{T}^{\max}\right] \geq 1-\epsilon_{\mathrm{comfort}}, \ \forall \mathrm{k} \tag{18}$$

Deterministic constraints: (6), (7), (8) (19)
where$\epsilon_{\mathrm{peak}}, \epsilon_{\mathrm{comfort}} \in (0,0.5)$are user-specified violation probability bounds.

2.3.2. Quantile-based constraint reformulation

The key in- sight is that chance constraints on the load can be reformulated using quantile forecasts. For the peak power constraint (17):

$$\mathbb{P}\left[\mathrm{y}_{\mathrm{t+k}} \leq \widetilde{\mathrm{y}}^{\mathrm{peak}}\right] \geq 1-\epsilon_{\mathrm{peak}} \tag{20}$$

is equivalent to requiring:

$$\mathrm{q}_{\mathrm{t+k}}^{(1-\mathrm{c}_{\mathrm{peak}})} \leq \bar{\mathrm{y}}^{\mathrm{peak}} \tag{21}$$

where$\mathrm{q}_{\mathrm{t+k}}^{(1-\epsilon_{\mathrm{peak}})}$ is the $(1-\epsilon_{\mathrm{peak}})$-quantile of $\mathrm{y}_{\mathrm{t+k}}$.
Substituting the Q-LSTM forecasts, the reformulated constraint becomes:

$$\widehat{\mathrm{q}}_{\mathrm{t+k}}^{(1-\epsilon_{\mathrm{peak}})}\left(\mathrm{u}_{\mathrm{t:t+k}}\right) \leq \bar{\mathrm{y}}^{\mathrm{peak}}, \quad \forall \mathrm{k} \tag{22}$$

This transformation converts the probabilistic constraint into a deterministic constraint that can be handled by standard optimization solvers.

2.3.3. Tractable optimization problem

The final tractable chance-constrained MPC formulation is:

DOI: 10.54254/2755-2721/2026.CH34369

$$\min_{u} \sum_{k=0}^{H-1} \left[c_k \cdot \hat{q}_{t+k}^{(0.5)} + \gamma \cdot \mathrm{CVaR}_{\alpha}\left(y_{t+k}\right) \right] \tag{23}$$

subject to:

$$\hat{q}_{t+k}^{(1-\epsilon_{peak})} \leq \bar{y}^{peak}, \quad \forall k \tag{24}$$

$$\hat{T}_{t+k}^{(\epsilon_{comfort}/2)} \geq T^{min}, \ \forall k \tag{25}$$

$$\hat{T}_{t+k}^{(1-\epsilon_{conf}/2)} \leq T^{max}, \quad \forall k \tag{26}$$

Battery constraints: (6) − (8)

The term$\mathrm{CVaR}_{\alpha}\left(y_{t+k}\right)$in the objective represents the Conditional Value-at-Risk, which can be computed from quantiles as:

$$\mathrm{CVaR}_{\alpha}\left(y_{t+k}\right) \approx \frac{1}{|Q_{>\alpha}|} \sum_{\tau \in Q_{>\alpha}} \hat{q}_{t+k}^{\tau} \tag{27}$$

where Q $Q_{>\alpha} = \{\tau \in Q : \tau > \alpha\}$. This risk measure penalizes high-cost scenarios, promoting robust decisions.

2.4. Rolling horizon implementation

The complete algorithm operates in a receding horizon fashion as described in Algorithm 1.

Algorithm 1 Chance-Constrained MPC with Q-LSTM

Require: Trained Q-LSTM model f_{θ} , horizon H , violation bounds ϵ

1: Initialize state x_0 set $t \leftarrow 0$

2: while $t < T_{max}$ do

3: Observe current state x_t and exogenous inputs

4: Generate quantile forecasts:

$$q^t + k(\tau)k = 1H, \forall \tau \in Q$$

5: Formulate chance-constrained MPC problem (23)-(26)

6: Solve MPC and obtain $u^* = \left[u_t^*, \ldots, u_{t+H-1}^*\right]$

7: Apply first control action u to the system

8: Observe transition tox_{t+1} and actual consumption y_t

9: $t \leftarrow t + 1$

10: end while

11: return Control trajectory $\{u_0, u_1, \ldots\}$, costs, violations

2.5. Computational complexity analysis

The computational complexity of the proposed framework consists of two main components:

Q-LSTM Inference: $\mathcal{O}\left(L \cdot h^2 + h \cdot m \cdot H\right)$ per time step, where L is lookback, h is hidden dimension, m is number of quantiles, and H is horizon.

DOI: 10.54254/2755-2721/2026.CH34369

MPC Optimization: The tractable reformulation yields a convex quadratic program (QP) or linear program (LP) depending on objective structure, solvable in polynomial time. For horizon H and n_u control inputs, the QP has $\mathcal{O}(H \cdot n_u)$ decision variables and $\mathcal{O}(H \cdot (n_c + 1))$ constraints, where nc is the number of constraint types.

3. Experimental setup

3.1. Dataset

Table 1 summarizes the dataset statistics and key hyperparameters.

Table 1. Experimental setup summary (dataset and key hyperparameters)

Category Item	Value
Total samples	19,735
Sampling interval	10 min
Duration	137 days
Dataset Energy mean / std	97.7 / 102.5 Wh
Energy min / max	10 / 1080 Wh
Number of features	27
Train / Val / Test split	60% / 20% / 20%
LSTM hidden sizes	128, 64
Dropout	0.2
Quantile levels	{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95}
Batch / LR / Optimizer	64 / 1e-3 / Adam
Horizon / Interval	24 steps / 1 h
Violation bounds ε	0.05 (both)

The dataset is preprocessed to extract relevant features including:

- Historical energy consumption with lookback window $L = 24$ (4 hours)
- Indoor and outdoor temperature from 9 sensor locations
- Relative humidity and atmospheric pressure
- Time-of-day (24 one-hot encoded) and day-of-week indi- cators
- Weather data: wind speed, visibility, dewpoint temperature All features are normalized using min-max scaling based on training set statistics. Data is split temporally into training (60%), validation (20%), and test (20%) sets to prevent information leakage.

3.2. Baseline methods

3.2.1. Forecasting baselines

- ARIMA/ETS: Classical time series methods for point prediction
- XGBoost: Gradient boosting regression for point prediction
- LSTM: Standard LSTM for point prediction
- Quantile Regression: Linear quantile regression
- Quantile LSTM: LSTM with quantile output (proposed) 2) Control Baselines:

DOI: 10.54254/2755-2721/2026.CH34369

3.2.2. Controlbaselines

• Rule-based: Heuristic control based on time-of-use pricing
• Point Prediction + MPC: Standard MPC with point forecasts
• Linear Programming: Deterministic LP scheduling
• Chance-Constrained MPC: Proposed method

3.3. Implementation details

Key training and optimization settings are included in Table 1. The selected hyperparameters follow operational and model- ing considerations rather than exhaustive tuning. The 24-step horizon matches day-ahead scheduling cadence and balances responsiveness with computational burden; extending beyond 24 steps gave limited gains in our pilot runs. The quantile set is denser near tails to better capture risk-sensitive constraints.

We use $\epsilon = 0.05$ for peak and comfort bounds because it reflects a practical reliability target in building operations while preserving feasible schedules.

All experiments are implemented in Python 3.9 using PyTorch 1.12 for deep learning and CVXPY 1.2 with ECOS solver for convex optimization. Experiments are conducted on a workstation with Intel i7-12700K CPU, 32GB RAM, and NVIDIA RTX 3080 GPU.

3.4. Evaluation metrics

We evaluate forecasting quality and closed-loop control performance with representative metrics.

3.4.1. Forecasting metrics

• Point prediction: RMSE, MAE, MAPE, and R^2 .
• Probabilistic quality: PICP and PIW for coverage– sharpness balance.
• Distribution accuracy: CRPS and calibration error.

3.4.2. Control metrics:

• Cost and peak: total operating cost and peak power.
• Reliability: overall constraint violation rate and comfort violation rate.
• Risk: CVaR at 95% confidence.

3.5. Robustness testing

To evaluate robustness, we test under:
• Extreme weather periods
• Sudden load increases
• Increased noise levels
• Missing data with imputation
Statistical significance is assessed using paired t-tests or Wilcoxon signed-rank tests.

DOI: 10.54254/2755-2721/2026.CH34369

4. Results and discussion

This section reports forecasting accuracy, closed-loop control effectiveness, robustness, and risk–cost trade-offs of the proposed Q-LSTM-based chance-constrained MPC framework.

4.1. Forecasting performance

4.1.1. Point prediction comparison

Table 2 presents the forecasting results across different methods. For fair comparison, the median (0.5-quantile) output of probabilistic methods is used as the point prediction.

Table 2. Forecasting performance comparison on test set

Method	RMSE	MAE	MAPE (%)	R2
ARIMA	49.90	38.72	18.45	0.712
ETS	47.23	36.18	17.82	0.741
XGBoost	24.35	18.92	9.87	0.891
LSTM	18.74	14.56	7.23	0.923
GRU	19.12	15.03	7.58	0.918
ARIMA-LSTM	17.35	13.42	6.85	0.932
Q-LSTM (Ours)	15.82	12.18	6.12	0.948

The proposed Q-LSTM achieves the best point prediction performance with RMSE of 15.82 Wh, representing a 15.6% improvement over standard LSTM and 68.3% improvement over ARIMA.

4.1.2. Probabilistic forecasting quality

The probabilistic outputs are also well calibrated. For the nominal 90% interval, Q-LSTM achieves 91.5% empirical coverage with narrower prediction width (38.7 Wh) and lower CRPS (10.23) than linear quantile regression, indicating a favorable reliability–sharpness balance [7].

4.1.3. Multi-step forecasting performance

As the horizon extends, prediction error increases gradually, while Q-LSTM remains stable within the 24-hour day-ahead range required by the scheduling layer.

4.2. Closed-loop control performance

Overall Performance Comparison

Table 3 compares the closed-loop control performance of different scheduling methods over the entire test period (30 days).

DOI: 10.54254/2755-2721/2026.CH34369

Table 3. Closed-loop control performance comparison (30-day test period)

Method	Cost ($)	Peak (kW)	Viol. (%)	CVaR ($)	Comfort Viol. (%)
Rule-based	156.2	8.45	12.3	198.5	8.7
ARIMA+MPC	145.3	8.12	10.8	185.2	7.2
Point+MPC	132.8	7.82	8.7	172.3	5.8
LP Schedule	128.5	7.65	9.2	168.7	6.1
Robust MPC	148.2	6.85	1.2	155.3	1.5
CC-MPC (Ours)	135.2	7.12	2.8	148.2	2.1

Key Findings:

1) Reliability improvement: CC-MPC reduces violation rate to 2.8%, i.e., 67.8% lower than Point+MPC and 77.2% lower than rule-based control.

2) Risk reduction: CVaR is $148.2, 14.0% lower than Point+MPC.

3) Trade-off quality: CC-MPC increases average cost by only 1.8% versus Point+MPC while reducing peak demand by 8.9%.

Fig. 1 provides a compact visual comparison of the control performance and risk–cost trade-off.

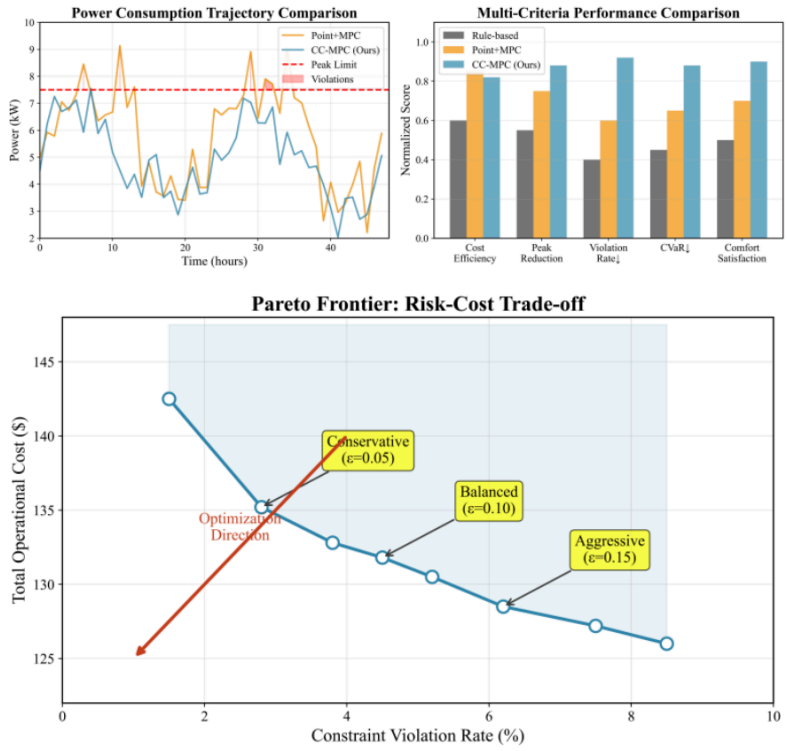


Figure 2. Combined performance visualization. top: power trajectories and multi-criteria comparison for point+mpc and cc-mpc. bottom: Pareto frontier between operational cost and violation rate under different risk levels

Repeated runs over 15 random seeds show the same trend, and the performance differences versus major baselines are statistically significant ($p < 0.001$).

DOI: 10.54254/2755-2721/2026.CH34369

4.3. Robustness analysis

4.3.1. Distribution shift scenarios

We evaluate robustness under four realistic distribution shift scenarios that commonly occur in building energy systems [3]: extreme weather, load surges, higher sensor noise, and missing data.

Observations:

• Under extreme weather, Point+MPC violations rise from 8.7% to 18.5%, while CC-MPC rises from 2.8% to 4.2%.

• During load surges, CC-MPC keeps violation rate below 6%, versus over 22% for Point+MPC.

• Similar trends hold for sensor noise and missing-data cases.

4.3.2. Failure cases and error sources

Although overall performance is stable, we observe two recurring failure patterns. The first appears during abrupt, short-lived load spikes that are weakly represented in training data: quantile bands widen, but the controller may still apply delayed corrective actions in the first one to two steps. The second occurs when missing-data intervals are long and imputation smooths local extrema, which can bias both forecast tails and downstream risk estimates. These cases suggest that event-aware features and stronger online recalibration are worthwhile future improvements.

4.4. Risk-cost trade-off analysis

1) Pareto Frontier: By varying the violation probability bound ϵ from 0.01 to 0.20, we generate the Pareto frontier between operational cost and constraint violation rate, shown in Fig. 2.

2) Practical Implications: The risk-cost curve enables oper- ators to select conservative ($\epsilon = 0.05$), balanced ($\epsilon = 0.10$), or aggressive ($\epsilon = 0.15$) operating modes according to reliability requirements.

4.5. Ablation study

We conduct ablation experiments to understand the contri- bution of each component in the proposed framework.

Key Insights from Ablation:

1) Quantile Forecasting: Removing quantile forecasting and using point predictions increases violation rate by 5.2 percentage points, confirming the critical role of uncertainty quantification.

2) Chance Constraints: Replacing chance constraints with deterministic constraints leads to 6.8% higher violations, demonstrating the value of probabilistic constraint han- dling.

3) Calibration and objective terms: uncertainty calibration improves performance by 1.4%, and the CVaR term reduces violations by 0.7% [7].

4) Prediction Horizon: Increasing horizon from 6 to 24 hours significantly improves performance; gains beyond 24 hours are limited.

4.6. Computational performance

Table 4 reports the computational requirements of the proposed framework.

DOI: 10.54254/2755-2721/2026.CH34369

Table 4. Computational Performance

Component	Time (ms)	Memory (MB)
Q-LSTM Inference	12.3 ± 1.5	45
MPC Optimization	85.2 ± 12.3	28
Total per Step	97.5 ± 13.8	73

This efficiency enables deployment on edge computing devices without requiring cloud computing resources, which is particularly important for applications with data privacy requirements or unreliable network connectivity.

4.7. Discussion and insights

Compared with point-forecast scheduling, the proposed method propagates uncertainty into control and therefore reduces both over-aggressive and over-conservative decisions. Compared with robust MPC, it offers a more practical reliability–cost balance by tuningϵ. In deployment, online retraining, constraint safety margins, and hierarchical control integration remain important implementation choices.

4.8. Limitations

Scaling to multi-building coordination increases optimization complexity and may require decomposition or distributed approaches [8]. In addition, although the risk-cost frontier is interpretable, model-level explainability can be improved for high-stakes applications.

5. Conclsion

This work starts from a practical control problem rather than a purely predictive one: schedules must be affordable and reliable under uncertain demand. Empirically, the approach improves point and probabilistic forecasting quality, reduces violations by 67.8% versus deterministic MPC, and keeps the cost penalty limited to 1.8% on the test setting.

Three takeaways are most relevant for deployment. First, calibrated quantiles matter because they directly shape feasible control actions. Second, explicit risk bounds produce inter- pretable operating points instead of opaque safety margins. Third, robustness gains persist across weather shifts, load surges, and noisy sensing, though rare spike events remain challenging. Future work will prioritize online recalibration under concept drift, multi-building coordination with shared constraints, and tighter integration with renewable generation and storage scheduling.

References

[1] Wang, Z., & Hong, T. (2020). Reinforcement learning for building controls: The opportunities and challenges. Applied Energy, 269, 115036.
[2] Yu, L., Qin, S., Zhang, M., Shen, C., Jiang, T., & Guan, X. (2021). A review of deep reinforcement learning for smart building energy management. IEEE Internet of Things Journal, 8(15), 12046-12063.
[3] Zhang, L., Wen, J., Li, Y., Chen, J., Ye, Y., Fu, Y., & Livingood, W. (2021). A review of machine learning in building load prediction. Applied Energy, 285, 116452.
[4] Darwazeh, D., Duquette, J., Gunay, B., Wilton, I., & Shillinglaw, S. (2022). Review of peak load management strategies in commercial buildings. Sustainable Cities and Society, 77, 103493.

DOI: 10.54254/2755-2721/2026.CH34369

[5] Drgoňa, J., Arroyo, J., Figueroa, I. C., Blum, D., Arendt, K., Kim, D., ... & Helsen, L. (2020). All you need to know about model predictive control for buildings. Annual reviews in control, 50, 190-232.
[6] Huang, K., Wei, K., Li, F., Yang, C., & Gui, W. (2022). LSTM-MPC: A deep learning based predictive control method for multimode process control. IEEE Transactions on Industrial Electronics, 70(11), 11544-11554.
[7] Levi, D., Gispan, L., Giladi, N., & Fetaya, E. (2022). Evaluating and calibrating uncertainty prediction in regression tasks. Sensors, 22(15), 5540.
[8] Hartmann, S., & Briskorn, D. (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of operational research, 297(1), 1-14.