Highlights

Hybrid modeling approach for better identification of building thermal network model and improved prediction

Sang woo Ham, Donghun Kim

- Development of a hybrid modeling approach that enhances the long-term temperature or load predictions of a gray-box model by combining a machine-learning model for unmeasured disturbances.
- Investigation of the limitations of various system identification approaches for a gray-box model under unmeasured disturbances.
- Development of the design and model selection processes for the robust hybrid modeling approach, considering the building thermal process.
- Achieved better long-term (1-day) prediction performance of the hybrid approach compared to the gray-box model in advanced predictive control applications for both simulated and experimental data.

Hybrid modeling approach for better identification of building thermal network model and improved prediction

Sang woo Ham[a], Donghun Kim[a]

[a]*Building Technology & Urban Systems Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA*

Abstract

The gray-box modeling approach, which uses a semi-physical thermal network model, has been widely used in building prediction applications, such as model predictive control (MPC). However, unmeasured disturbances, such as occupants, lighting, and in/exfiltration loads, make it challenging to apply this approach to practical buildings. In this study, we propose a hybrid modeling approach that integrates the gray-box model with a model for unmeasured disturbance. After reviewing several system identification approaches, we systematically designed the unmeasured disturbance model with a model selection process based on statistical tests to make it robust. We generated data based on the building model calibrated by real operational data and then trained the hybrid model for two different weather conditions. The Hybrid model approach demonstrates the reduction of RMSE approximately 0.2-0.9°C and 0.3-2°C on 1-day ahead temperature prediction compared to the Conventional approach for mild (Berkeley, CA) and cold (Chicago, IL) climates, respectively. In addition, this approach was applied for experimental data obtained from the laboratory building to be used for the MPC application, showing superior prediction performances.

Keywords: Gray-box model, Building modeling, Unmeasured disturbances, Hybrid modeling, Neural network, Building control, Machine learning

Nomenclature

CONV: Conventional system identification

HVAC: Heating, ventilation, and air-conditioning

HP-RTU: Heat pump rooftop unit

ID: Input disturbance system identification

MPC: Model predictive control

OD: Output disturbance system identification

RTF: Runtime fraction

RTU: Rooftop unit

RMSE: Root mean squared error

LD: Lumped term for all the unmeasured disturbances

($\mathbf{A}(\cdot)$, $\mathbf{B_u}(\cdot)$, $\mathbf{B_w}(\cdot)$, $\mathbf{B_g}(\cdot)$, $\mathbf{C}(\cdot)$): A state space model structure that maps θ to building dynamics (i.e., G_u, G_w, and G_g)

($\mathbf{A_d}(\cdot)$, $\mathbf{B_{d,u}}(\cdot)$, $\mathbf{B_{d,w}}(\cdot)$, $\mathbf{B_{g,w}}(\cdot)$, $\mathbf{C_d}(\cdot)$): A discretized state space model of ($\mathbf{A}(\cdot)$, $\mathbf{B}_\mathrm{u}(\cdot)$, $\mathbf{B}_\mathrm{w}(\cdot)$, $\mathbf{B}_\mathrm{g}(\cdot)$, $\mathbf{C}(\cdot)$)

$A_{\mathbf{win}}$**:** Effective window area of a zone window [kW/m^2]

b: Bias vector of neural network.

$C_{\mathbf{w},i}$**:** Thermal capacitance of wall mass of ith zone [kWh/K]

$C_{\mathbf{za},i}$**:** Thermal capacitance of zone air of ith zone [kWh/K]

c: Cell state vector in LSTM

dow: Day of week [-]

e**:** Zero mean white noise

($\mathcal{F}(\cdot)$, $\mathcal{G}(\cdot)$): A state space model structure that maps ρ to lumped disturbance dynamics (i.e., H)

f**:** Convective fraction of the incident solar radiation of a zone window [-]

$G_{\mathbf{u}}$**:** A dynamic system that maps $\mathbf{u}$ to y_za

$G_{\mathbf{w}}$**:** A dynamic system that maps $\mathbf{w}$ to y_za

$G_{\mathbf{g}}$**:** A dynamic system that maps $\dot{Q}_g$ to y_za

how: Hour of week [h]

hod: Hour of day [h]

H**:** Dynamics of lumped output disturbances

h: Hidden state vector in RNN/LSTM

$i_{\mathbf{heat}}, i_{\mathbf{cool}}$**:** Binary indicators for heating/cooling stage [-]

n_***:** Number of *.

n_ψ**: Number of input features in disturbance model [-**

$n_{\mathbf{k},\psi}$**:** Input sequence length for disturbance model [-]

$n_{\mathbf{k},\xi}$**:** Output sequence length for disturbance model [-]

$n_{\mathbf{layer}}$**:** Number of hidden layers in neural network [-]

$n_{\mathbf{z}}$**:** Size of hidden layer [-]

$n_{\mathbf{channel}}$**:** Number of convolution channels [-]

$n_{\mathbf{filter}}$**:** Size of convolution filter [-]

$n_{\mathbf{pool}}$**:** Pooling size in CNN [-]

$\dot{Q}_{\mathbf{g}}$**:** Unmeasured heat gains of a zone [kW]

$\dot{q}_{\mathbf{sol,win}}$**:** Incident solar radiation per area on a zone window [kW/m^2]

$\dot{Q}_{\mathbf{hc}}$**:** Rated heating($\dot{Q}_{\mathrm{h}}$)/cooling($\dot{Q}_{\mathrm{c}}$) capacity of a zone HVAC unit [kW]

($R_{\mathbf{zw}}$, $R_{\mathbf{zo}}$**):** Thermal resistances between temperature nodes of a zone [K/kW]

t, k**:** Continuous and discrete time

$T_{\mathbf{za}}$**:** Air temperature of a zone [°C]

$T_{\mathbf{w}}$**:** Wall thermal mass temperature of a zone [°C]

$T_{\mathbf{oa}}$**:** Outdoor air temperature [°C]

$T_{\mathbf{csp}}$**:** Cooling setpoint temperature [°C]

$T_{\mathbf{hsp}}$**:** Heating setpoint temperature [°C]

T_s**:** Sampling time [s]

u: Vector of control inputs (i.e., heating or cooling operation stages, [u_{h}, u_{c}])

$u_{\mathbf{h}}$, $u_{\mathbf{c}}$**:** Heating and cooling stages of a HVAC unit [-].

$u_{\mathbf{hc}}$**:** Combined heating/cooling control signal [-]

W: Weight matrix of neural network.

w: Vector of measured disturbances (i.e., [T_{oa}, $\dot{Q}_{\mathrm{sol,win}}$])

weekday: Binary indicator of weekday/weekend [-]

x: Vector of state variables (i.e., [T_{w}, T_{za}])

$\hat{\mathbf{x}}(k|j)$**:** Vector of estimated (predicted) state variables at time k from the data at j

$\boldsymbol{\nu}$**:** Vector of lumped output disturbances [°C]

$y_{\mathbf{za}}$**:** Measured thermostat temperature of a zone [°C]

β_* : Regression parameters of * variable in model selection process.

ε**:** One step ahead prediction error

$\boldsymbol{\Sigma}_{\mathbf{x_{ID}}}$**:** State noise covariance matrix

$\boldsymbol{\Sigma}_{y_{\mathbf{za}}}$**:** Measurement noise covariance matrix

$\boldsymbol{\zeta}$**:** Vector of internal state of lumped output disturbances

θ**:** Physical parameters consisting of thermal resistances and capacitances, [C_{w}, C_{za}, R_{zw}, R_{zo}, f,A_{win}, $\dot{Q}_{\mathrm{h}}$, $\dot{Q}_{\mathrm{c}}$]

ρ**:** Parameters that constructs dynamics of lumped output disturbances, i.e. H

$(\omega_{\mathbf{l}}, \omega_{\mathbf{u}})$**:** Weights on optimization variables for (Γ_{l}, Γ_{u})

$\mathcal{D}(k)$**:** Set of measured data from time step from timestep from 1 to k.

χ_* : Independent variable (*) of the regression in model selection process.

$\upsilon_*(c)$**:** Temperature prediction error by using unmeasured disturbance model on * dataset (i.e., either train or test).

$\boldsymbol{\psi}$**:** Input vector of unmeasured disturbance model.

φ**:** Activation function in neural network

$\boldsymbol{\xi}$**:** Output vector of unmeasured disturbance model.

1. Introduction

The gray-box modeling approach, often referred to as the semi-physical thermal network model, is extensively utilized in building energy prediction applications. Its applications include model predictive control (MPC) [1, 2] for enhancing energy efficiency [3] and advancing decarbonization efforts [4], due to its flexible structure. This method models a complex building using a simplified R-C (thermal resistance and capacitance) network model, while retaining the physical principles of the building's thermal characteristics.

The conventional system identification method for gray-box models generally involves estimating parameters by minimizing the sum of squared errors between the predicted and actual temperatures n steps ahead [5, 6]. This technique uses readily accessible data, such as disturbances (weather conditions) and control signals (heating and cooling operation signals). However, there are also disturbances that are not easily measurable like internal heat gains from occupants, lighting, appliances, and air infiltration or exfiltration. These sources can contribute significantly to the overall heat gain but are challenging to quantify in actual buildings due to the lack of sensors, which can degrade the quality of system identification outcomes [7, 8]. In literature, the aggregate of heat gains from unmeasured disturbance is often simplified into a single term, presumed to be proportional to the total of all electrical loads [9, 3]. Nonetheless, installing the necessary power sensors for accurate measurement is expensive and typically unfeasible, especially in small and medium commercial buildings.

The failure in system identification can lead to significant issues in predicting building energy performance. For instance, unmeasured disturbances—such as all unaccounted internal heat gains—might be expressed as exaggeratedly high thermal capacitances and solar heat gains in the system identification results. Consequently, this misrepresentation could worsen the system's ability to accurately predict the building's thermal behavior in response to future heating and cooling control signals.

Various system identification techniques have been proposed to accurately estimate the model parameters of a gray-box model under unmeasured disturbances. Kim et al. [7] proposed a system identification algorithm that accounts for the dynamics of unmeasured disturbances. They treated all the internal heat gains as a lumped unmeasured disturbance term (LD) and appended its dynamics to the building gray-box model for system identification. The authors utilized the impact of LD on the measurement (i.e., zone air temperature), referred to as the output disturbance (OD) approach, and solved the system identification problem using the prediction-error method (i.e., minimizing one-step prediction errors with state filtering) [10]. Coffman and Barooah [11] suggested a similar but slightly different approach by directly including the LD term as a heat gain in the gray-box model (i.e., the input disturbance (ID) approach). They achieved system identification by minimizing one-step-ahead prediction errors via the Kalman filter, but the selection

of state variances served as tuning parameters to correctly capture the variations of the IDs. Zeng et al. [12] further developed the ID approach in the frequency domain with physical constraints. Although this approach makes the system identification problem a convex optimization problem, the parameter values of the gray-box model may lose their physical scales. On the other hand, the LD term has also been modeled as a black-box function of time-related features (e.g., time of day, day of the week, and day of the year) via a feed-forward neural network, and the system identification was conducted together with the neural network [13]. Similarly, Kumar et al. [14] modeled the LD term as piecewise constants and included it in the system identification. Having a separate model for the LD term in the system identification can capture the complex and non-linear characteristics of unmeasured disturbances, but an iterative process may be required to prevent the overall building thermal dynamics from being overfitted by the model. Finally, the theoretical analysis of the LD approach [8] claims that the success of system identification under unmeasured disturbances heavily relies on the quality of training data (i.e., data with uncorrelated control inputs, and measured and unmeasured disturbances). This can be achieved by manually assigning cooling or heating signals (exciting the system) for sufficient periods [15]. Nevertheless, in practice, extensive manual excitation is often not feasible for buildings in use, given the slow thermal response of the building and the seasonal characteristics of outdoor conditions.

The predictive accuracy of the gray-box model for future scenarios is often compromised, even with precisely identified system parameters, if it does not account for unmeasured disturbance profiles. In building simulations, such disturbances—which include heat gains from occupants, appliances, and infiltration—are typically represented by scheduled values, either deterministic [16] or stochastic [17]. While this method is appropriate for white-box simulations aimed at estimating average energy usage based on standard disturbance schedules for building design, it may not suffice for precise predictive applications for real buildings.

Therefore, estimating unmeasured disturbances from available data is essential. Although a Kalman filter [18, 11] or a particle filter [19] can derive these disturbances from measured data, they do not offer future predictions. To address this, some research [20, 21] has developed black-box models that forecast non-HVAC energy consumption based on historical data. Additionally, two studies [13, 14] have suggested employing a neural network-based black-box model to detect unmeasured disturbances using temporal features and incorporate them into MPC. One study conducted simultaneous identification of both the gray-box and neural network models, while the other trained the neural network model subsequent to the gray-box model identification. Yet, the concern regarding the potential overfitting of neural network models remains unresolved. Additionally, as highlighted earlier, there is a critical need for high-quality training data accompanied by sufficient system excitation.

In summary, the incorporation of unmeasured disturbance dynamics are important to improve the quality of conventional system identification for real buildings. Additionally, an additional model for unmeasured disturbances is essential for accurately predicting future outcomes. While previous studies have demonstrated promising prediction results with their data, further investigation is needed to comprehend the variances among different system identifications and the effects of gray-box model quality and unmeasured disturbance structures on predictions. This necessity arises from the fact that gray-box model quality cannot be assured in real buildings due to sensor scarcity and low data quality.

In this research, we introduce a hybrid approach that combines a neural network-based machine-learning model for unmeasured disturbances with a gray-box model for predictive applications. Firstly, we conduct a simulation case study to underscore the deficiencies of conventional identification algorithms in handling unmeasured disturbances. We then propose and compare alternative identification algorithms to mitigate these challenges. Subsequently, we examine the limitations of these alternatives in predictive applications and propose a methodology for designing and selecting the unmeasured disturbance model. This methodology aims to prevent overfitting by considering the input-output structure of the gray-box model. Finally, we

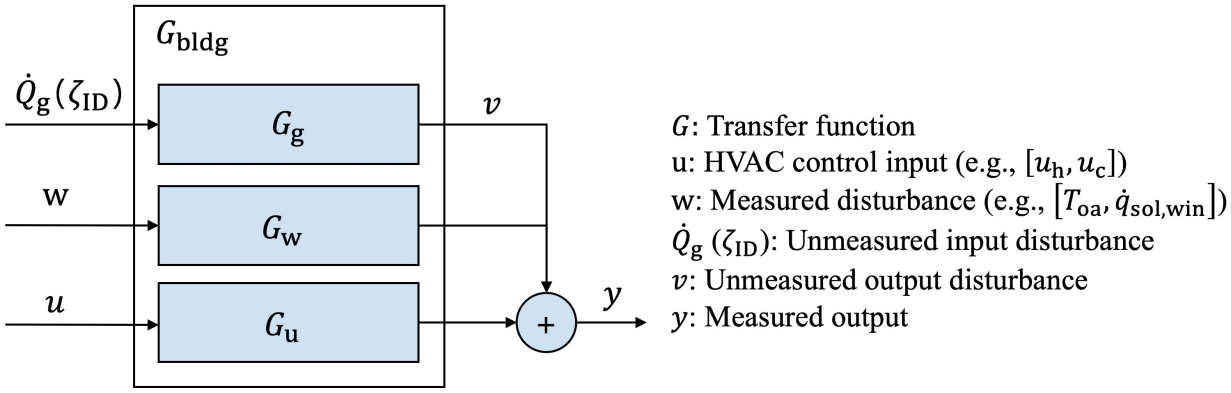


Figure 1: Relationship between inputs and output in a building transfer function.

assess the performance of our proposed model on both generated simulation and experimental data.

2. Comparisons of system identification approaches under unmeasured disturbance and effect on prediction: simulation case study

In this section, a case study is presented to compare the performance of various system identification approaches when significant unmeasured disturbances are present. Fig. 1 shows the relationship between the inputs and output of a building envelope system transfer function, $G_{\text{bldg}} : \left(\mathbf{w}, \mathbf{u}, \dot{Q}_{\text{g}}\right) \rightarrow y$. As described in section 1, $\dot{Q}_{\text{g}}$ is not available in real buildings. Therefore, the system identification methods are classified based on how they treat this unmeasured disturbance. The conventional system identification (CONV) approach treats it as white noise [5, 6]. The input disturbance approach (ID) models the unmeasured disturbance as a lumped input disturbance term (ζ_{ID}) and includes it in the system identification [11]. In contrast, the output disturbance approach (OD) represents the lumped disturbance (LD) term in the system identification through the measurement process (i.e., y) [7, 8]. In the following sections, the mathematical details of each system identification approach are described, and their performances are compared.

2.1. True system description and data generation

Evaluating the effectiveness of different system identification methods for an actual building is challenging because the true system dynamics and parameters are unknown. To address this, we developed a theoretical building envelope model, referred to as the *True model* (TRUE), to generate synthetic building operational data for our analysis. The TRUE model was calibrated using a dataset from a single-zone laboratory building (FLEXLAB, [22]) that represents an office environment. This dataset includes weather data, thermostat setpoints, HVAC operation data, and all unmeasured disturbances such as plug loads, lighting loads, occupancy, ventilation, and infiltration. The TRUE model adopts the 2R–2C network shown in Fig. 2 and follows the state-space form given in Eq. 1 (state transition process) and Eq. 2 (measurement process).

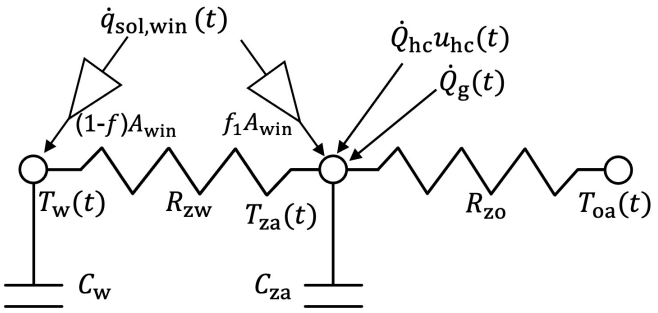


Figure 2: A simple RC network model for a case study building.

$$\underbrace{\begin{bmatrix}\dot{T}_{\text{w}}(t)\\ \dot{T}_{\text{za}}(t)\end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix}\frac{-1}{C_{\text{w}}R_{\text{zw}}} & \frac{1}{C_{\text{w}}R_{\text{zw}}}\\ \frac{-1}{C_{\text{za}}R_{\text{zw}}} & \frac{-1}{C_{\text{za}}R_{\text{zw}}}+\frac{-1}{C_{\text{za}}R_{\text{zo}}}\end{bmatrix}}_{\mathbf{A}}\underbrace{\begin{bmatrix}T_{\text{w}}(t)\\ T_{\text{za}}(t)\end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix}0 & \frac{(1-f)A_{\text{win}}}{C_{\text{w}}}\\ \frac{1}{C_{\text{za}}R_{\text{zo}}} & \frac{fA_{\text{win}}}{C_{\text{za}}}\end{bmatrix}}_{\mathbf{B}_{\text{w}}}\underbrace{\begin{bmatrix}T_{\text{oa}}(t)\\ \dot{q}_{\text{sol,win}}(t)\end{bmatrix}}_{\mathbf{w}} + \underbrace{\begin{bmatrix}0\\ \frac{\dot{Q}_{\text{hc}}}{C_{\text{za}}}\end{bmatrix}}_{\mathbf{B}_{\text{u}}}\underbrace{\begin{bmatrix}u_{\text{hc}}(t)\end{bmatrix}}_{\mathbf{u}} + \underbrace{\begin{bmatrix}0\\ \frac{1}{C_{\text{za}}}\end{bmatrix}}_{\mathbf{B}_{\text{g}}}\underbrace{\begin{bmatrix}\dot{Q}_{\text{g}}(t)\end{bmatrix}}_{\dot{\mathbf{Q}}_{\text{g}}} \tag{1}$$

$$y_{\text{za}} = \underbrace{\begin{bmatrix}0 & 1\end{bmatrix}}_{\mathbf{C}}\mathbf{x} \tag{2}$$

The parameters to be estimated through system identification are $\theta = \left[C_{\text{w}}, C_{\text{za}}, R_{\text{zw}}, R_{\text{zo}}, f, A_{\text{win}}, \dot{Q}_{\text{h}}, \dot{Q}_{\text{c}}\right]$ and were tuned using the complete set of measurements, including $\dot{Q}_{\text{g}}$. The TRUE model parameter values were set as follows: C_{w} = 4.0 kWh/K, C_{za} = 1.0 kWh/K, R_{zw} = 1.2 K/kW, R_{zo} = 9 K/kW, f = 0.3, A_{win} = 3.0 m^2, $\dot{Q}_{\text{h}}$ = 6 kW, and $\dot{Q}_{\text{c}}$ = –6 kW. An ON/OFF controller was implemented to maintain the indoor temperature, with a minimum ON/OFF duration of 5 minutes.

A two-week dataset was generated from the TRUE model using Oakland, CA weather data [23], as shown in Fig. 3. Here, T_{oa} is the outdoor air temperature, T_{za} is the zone air temperature, T_{csp} is the room cooling setpoint, T_{hsp} is the room heating setpoint, $\dot{q}_{\text{sol,win}}$ is the incident solar radiation per unit area on a zone window, u_{h} and u_{c} are the heating and cooling ON/OFF signals (fraction values in Fig.3 indicate 15-minute moving averages), and $\dot{Q}_{\text{gain}}$ is the sum of all unmeasured disturbances (i.e., plug loads, lighting loads, occupancy, ventilation, and infiltration).

During weekdays, the cooling setpoint is set to 23°C–25°C during occupied hours (6:00–19:00) and 28°C–30°C during unoccupied hours. In the data generation process, internal heat gains from plug loads, lighting, occupancy, ventilation, and infiltration are included as a lumped term ($\dot{Q}_{\text{g}}$) with stochastic random variations. However, to test the performance of the CONV, ID, and OD system identification algorithms under a realistic scenario, $\dot{Q}_{\text{g}}$ is assumed to be unknown.

It is further assumed that experiments can be designed to actively control the indoor temperature setpoint (within an acceptable room air temperature range) during weekends (i.e., unoccupied periods) to improve

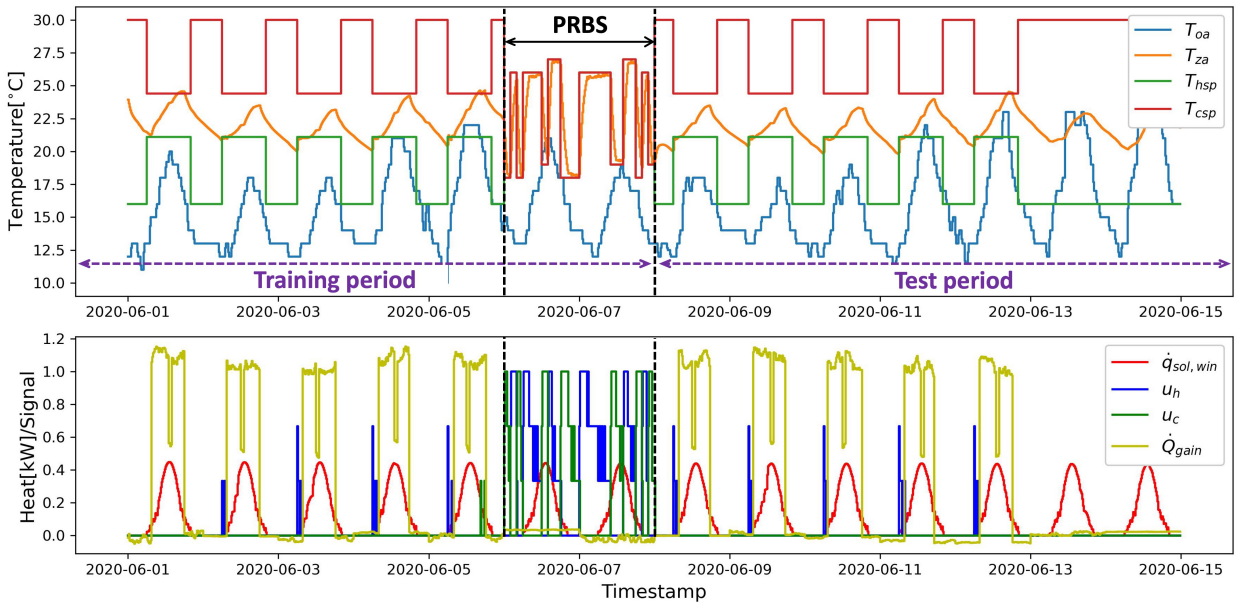


Figure 3: Synthetic data from the TRUE model to evaluate the performance of different identification algorithms (15-minute moving average).

system identification. This is a practical scenario when conducting system identification in real buildings [24]. On the first weekend (two days), the setpoint was perturbed according to a pseudo-random binary sequence (PRBS) with a 2-hour time scale and 4th order [15]. The binary signal was mapped to sampled setpoints between 18°C and 25°C.

2.2. Descriptions of system identification approaches

Eqs. 1–2 were discretized with a 15-minute sampling interval using a zero-order hold method [25]. Three different system identification approaches (CONV, ID, and OD) were then applied to the generated data. While all three approaches share the same discretized model structure for the building envelope dynamics, they differ in the structure of their disturbance models. It is important to note that the unmeasured disturbance ($\dot{Q}_{\mathrm{g}}$) was not provided to the identification algorithms. In other words, the identified model G maps only the measured disturbances (T_{oa}, $\dot{q}_{\mathrm{sol,win}}$) and the control input ($\dot{Q}_{\mathrm{hc}}$) to the indoor air temperature (y_{za}), as shown in Fig. 1.

2.2.1. Conventional simulation error minimization approach

The CONV approach assumes that all the unmeasured disturbance can be expressed as white noise ($e_{\mathrm{conv}} : \varepsilon_{\mathrm{conv}}(k) \sim N(0, \sigma^2_{\mathrm{conv}})$) in the measurement process, and the discretized system can be written as Eq. (3);

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_{\mathrm{d}}\mathbf{x}(k) + \mathbf{B}_{\mathrm{w,d}}\mathbf{w}(k) + \mathbf{B}_{\mathrm{u,d}}\mathbf{u}(k) \\ y(k) &= \mathbf{C}_{\mathrm{d}}\mathbf{x}(k) + e_{\mathrm{conv}}(k) \end{aligned} \tag{3}$$

The set of parameters (θ^*_{conv}) is estimated by minimizing the sum of squared errors between simulation ($\hat{y}(k;\theta) = \mathbf{C}_{\text{d}}\hat{\mathbf{x}}(k;\theta)$) and measurement ($y(k)$) via the nonlinear optimization (Eqs. 4-5);

$$\begin{aligned}\hat{\mathbf{x}}(k+1;\theta) &= \mathbf{A}_{\text{d}}(\theta)\hat{\mathbf{x}}(k;\theta) + \mathbf{B}_{\text{w,d}}(\theta)\mathbf{w}(k) + \mathbf{B}_{\text{u,d}}(\theta)\mathbf{u}(k)\\ y(k) &= \mathbf{C}_{\text{d}}\hat{\mathbf{x}}(k;\theta) + \varepsilon_{\text{conv}}(k;\theta)\end{aligned} \tag{4}$$

$$\theta^*_{\text{conv}} = \arg\min_{\theta} \sum_{k=1}^{N} \left(\varepsilon_{\text{conv}}(k;\theta)\right)^2, \tag{5}$$

where subscript d indicates a discretized system and, k is a discrete time step.

In the 7-day training data, the initial state (i.e., $\mathbf{x}(0)$) is obtained via a Kalman filter [25] by using the first-day data. Then, the following 6 days are predicted via simulation (Eq. 4). The optimization bounds of parameters are set to [0.1, 40] for all R and C parameters (i.e., $C_{\text{w}}, C_{\text{za}}, R_{\text{zw}}, R_{\text{zo}}$), [1e-6, 1] for f, and [0.1, 25] for A_{win}.

2.2.2. Input disturbance identification approach

The ID approach assumes that unmeasured disturbances come from the input channel, i.e., the heat gain term, and treats the input disturbance as an additional dynamic state. This can be written as an augmented state space format (Eq. 6) [11, 26].

$$\begin{aligned}\underbrace{\begin{bmatrix}\dot{\mathbf{x}}\\ \dot{\zeta}_{\text{ID}}\end{bmatrix}}_{\dot{\mathbf{x}}_{\text{ID}}} &= \underbrace{\begin{bmatrix}\mathbf{A} & \mathbf{A}_{\zeta_{\text{ID}}}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}}_{\mathbf{A}_{\text{ID}}}\underbrace{\begin{bmatrix}\mathbf{x}\\ \zeta_{\text{ID}}\end{bmatrix}}_{\mathbf{x}_{\text{ID}}} + \mathbf{B}_{\text{w}}\mathbf{w} + \mathbf{B}_{\text{u}}\mathbf{u} + \mathbf{e}_{\text{x}_{\text{ID}}}, \text{ and } \mathbf{A}_{\zeta_{\text{ID}}} = \begin{bmatrix}\mathbf{0}\\ \frac{1}{C_{\text{za}}}\end{bmatrix}\\ y_{\text{za}} &= \underbrace{\begin{bmatrix}\mathbf{C} & 0\end{bmatrix}}_{\mathbf{C}_{\text{ID}}}\mathbf{x}_{\text{ID}} + e_{\text{ID}}\end{aligned} \tag{6}$$

where $\mathbf{e}_{\mathbf{x}_{\text{ID}}}$ and e_{ID} are state and measurement noises. ζ_{ID} represents the lumped input disturbance term.

The key idea in this approach is to treat ζ_{ID} as Wiener process, so therefore, it behaves as the Brownian motion after a discretization according to its noise level (i.e., $\zeta_{\text{ID}}(k+1) = \zeta_{\text{ID}}(k) + \varepsilon_{\zeta_{\text{ID}}}(k|\theta)$).

For the system identification, the ID approach first calculates one-step-ahead prediction errors via the following steps. The system equation (Eq. 6) is discretized as Eq. 7. From the initial states ($\hat{\mathbf{x}}_{\text{ID}}(1|1)$), the next time states ($\hat{\mathbf{x}}_{\text{ID}}(2|1)$) and zone air temperature ($\hat{y}_{\text{za}}(2|1)$) are predicted through Eq. 6. The prediction error (i.e., innovation, ε_{ID}) is estimated through Eq. 8, and then, the predicted states are updated by using the innovation and optimal Kalman gain ($\mathbf{K}(k|\theta)$) (Eq. 9). The process in Eqs. 7-9 is sequentially repeated for the whole data ($k = 1, 2, ..., N$). The ID approach finds a set of parameters by minimizing the square sum of one-step ahead prediction error (Eq. 10).

$$\begin{aligned}\hat{\mathbf{x}}_{\text{ID}}(k+1|k;\theta) &= \mathbf{A}_{\text{d,ID}}(\theta)\hat{\mathbf{x}}_{\text{ID}}(k|k) + \mathbf{B}_{\text{w,d}}(\theta)\mathbf{w}(k) + \mathbf{B}_{\text{u,d}}(\theta)\mathbf{u}(k)\\ \hat{y}_{\text{ID}}(k+1|k;\theta) &= \mathbf{C}_{\text{d,ID}}\hat{\mathbf{x}}_{\text{ID}}(k+1|k)\end{aligned} \tag{7}$$

At each time step k, the optimal Kalman gain is obtained using the Kalman filter [25]. The gain is estimated based on the state noise covariance ($\mathbf{\Sigma}_{\mathbf{x}_{\text{ID}}}$, i.e., $\boldsymbol{\varepsilon}_{\text{x}_{\text{ID}}}(k) \sim N(0, \mathbf{\Sigma}_{\mathbf{x}_{\text{ID}}})$) and the measurement noise covariance $\mathbf{\Sigma}_{y_{\text{za}}}$ in the discretized system. The measurement noise covariance is set to $0.25^2/T_{\text{s}}$, based on the sensor noise level ($\pm 0.5°\text{C}$) and the discretization sampling time T_{s}. The state noise covariance is modeled

using two additional parameters, i.e., $\mathbf{\Sigma}_{\mathbf{x}_{\mathrm{ID}}} = \mathrm{diag}(\sigma_x^2, \sigma_x^2, \sigma_{\zeta_{\mathrm{ID}}}^2)$, which determine the degree of state update in $\hat{\mathbf{x}}_{\mathrm{ID}} = \left[\hat{\mathbf{x}}, \hat{\zeta}_{\mathrm{ID}}\right]^{\intercal}$ (Eq. 9). The optimization bounds of these two parameters are set to [1e^{-9}, 1].

$$\varepsilon_{\mathrm{ID}}(k+1;\theta) = y(k+1) - \hat{y}_{\mathrm{ID}}(k+1|k;\theta) \tag{8}$$

$$\hat{\mathbf{x}}_{\mathrm{ID}}(k+1|k+1;\theta) = \hat{\mathbf{x}}_{\mathrm{ID}}(k+1|k;\theta) + \mathbf{K}(k+1;\theta)\varepsilon_{\mathrm{ID}}(k+1;\theta) \tag{9}$$

$$\theta_{\mathrm{ID}}^* = \arg\min_{\theta} \sum_{k=1}^{N} \left(\varepsilon_{\mathrm{ID}}(k;\theta)\right)^2 . \tag{10}$$

2.2.3. Output disturbance identification approach

The OD approach [7, 8] does not explicitly model the input disturbances. Instead, it tries to model the effect of unmeasured heat gains on the output (i.e., room air temperature). The aggregated contribution of the unknown heat sources to the output is called the output disturbance, as opposed to the input disturbance. The OD approach models the output disturbance as a filtered process of white noise ($e_{\mathrm{OD}}(k)$), which is called output disturbance ($v_{\mathrm{OD}}(k)$ in Eq. 11) and Fig. 1. The output disturbance dynamics can be modeled with two more parameters, ρ_1 and ρ_2 (Eq. 11).

$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}_{\mathrm{d}}\mathbf{x}(k) + \mathbf{B}_{\mathrm{w,d}}\mathbf{w}(k) + \mathbf{B}_{\mathrm{u,d}}\mathbf{u}(k) \\
y(k) &= \mathbf{C}_{\mathrm{d}}\mathbf{x}(k) + v_{\mathrm{OD}}(k) \\
\zeta_{\mathrm{OD}}(k+1) &= \underbrace{[\rho_1]}_{\mathcal{F}}\zeta_{\mathrm{OD}}(k) + \underbrace{[\rho_2]}_{\mathcal{G}}e_{\mathrm{OD}}(k) \\
v_{\mathrm{OD}}(k) &= \zeta_{\mathrm{OD}}(k) + e_{\mathrm{OD}}(k)
\end{aligned} \tag{11}$$

For each time step, the prediction error (i.e., innovation, $\varepsilon_{\mathrm{OD}}$) is estimated via Eq. 12. Then it is used to calculate next time prediction (Eq. 13-14).

$$\varepsilon_{\mathrm{OD}}(k;\theta) = y(k) - \hat{y}_{\mathrm{OD}}(k;\theta) \tag{12}$$

$$\begin{aligned}
\hat{\zeta}_{\mathrm{OD}}(k+1;\theta) &= \mathcal{F}(\theta)\hat{\zeta}_{\mathrm{OD}}(k;\theta) + \mathcal{G}(\theta)\varepsilon_{\mathrm{OD}}(k) \\
\hat{v}_{\mathrm{OD}}(k;\theta) &= \hat{\zeta}_{\mathrm{OD}}(k;\theta) + \varepsilon_{\mathrm{OD}}(k;\theta) \\
\hat{\mathbf{x}}(k+1;\theta) &= \mathbf{A}_{\mathrm{d}}(\theta)\hat{\mathbf{x}}(k;\theta) + \mathbf{B}_{\mathrm{w,d}}(\theta)\mathbf{w}(k) + \mathbf{B}_{\mathrm{u,d}}(\theta)\mathbf{u}(k) \\
\hat{y}_{\mathrm{OD}}(k+1;\theta) &= \mathbf{C}_{\mathrm{d}}\hat{\mathbf{x}}(k+1;\theta) + \mathcal{F}(\theta)\hat{\zeta}_{\mathrm{OD}}(k+1;\theta)
\end{aligned} \tag{13}$$

The optimal set of parameters is estimated by minimizing the square sum of one-step ahead prediction error (Eq. 14). The optimization bounds of ρ_1 and ρ_2 are set to [-0.999, 0.999] as suggested in [7].

$$\theta_{\mathrm{OD}}^* = \arg\min_{\theta} \sum_{k=1}^{N} \left(\varepsilon_{\mathrm{OD}}(k;\theta)\right)^2 \tag{14}$$

2.3. System identification results and discussion

For each identification algorithm, the optimization problem is non-convex, and therefore, we randomly sampled initial starting points and repeatedly solved the optimization problems 50 times to find a better optimal solution. The estimation results of CONV, ID, OD identification approaches are summarized in Table 2.3.

	C_w [kWh/K]	C_za [kWh/K]	R_zw [K/kW]	R_zo [K/kW]	f [-]	A_win [m^2]	$\dot{Q}_\mathrm{h}$ [kW]	$\dot{Q}_\mathrm{c}$ [kW]	σ_x(ID) ρ_1(OD)	σ_ζ(ID) ρ_2(OD)
TRUE	4.0	1.0	1.2	9.0	0.3	3.0	6.0	-6.0		
CONV	40	0.4	6.2	40	1.0	1.1	1.5	-1.3		
ID	20.6	1.4	0.8	3.4	0.03	20.0	6.7	-8.7	1e-3	2e-2
OD	10.6	1.3	0.8	4.1	0.2	12.4	6.7	-8.6	0.99	0.99

Table 1: Comparison of estimated parameters from system identification approaches

Table 2.3 shows that the CONV method yielded inaccurate parameter estimates. In particular, it significantly overestimated the thermal capacitance (C_w) and the thermal resistance (R_zo). This occurred because the method attempted to explain the relationship between inputs and outputs without considering heat gain information, leading to compensatory changes in the physical parameters. As a result, the large thermal mass (C_w) acted like a heat source during the day to offset the unmeasured heat gains, while R_zo was overestimated to retain heat in the thermal mass overnight. In addition, the estimated heating and cooling rates ($\dot{Q}_\mathrm{h}$ and $\dot{Q}_\mathrm{c}$) were substantially underestimated, producing values that do not match the physical scales of the actual heating and cooling capacities.

In contrast, the ID and OD algorithms include an additional degree of freedom through parameters associated with the disturbance models, allowing them to better explain the input–output relationship under unmeasured disturbances. Overall, incorporating disturbance dynamics improves parameter estimation when such disturbances are present.

The ID method produced improved estimates of R_zo, indicating its ability to capture small unmeasured disturbances. However, it struggled with abrupt changes in these disturbances, such as during transitions between occupied and unoccupied periods. In such cases, the method compensated by overestimating the solar radiation parameters (A_win and f). The OD method, on the other hand, provided more accurate estimates for most parameters, although they were not entirely precise. As noted in [7], input disturbances are often noisier and more volatile than output disturbances, resulting in a higher-frequency power spectrum. In contrast, output disturbances tend to have a higher power spectrum in the lower-frequency range, which can be easier to model in certain applications.

Since all methods deviate from the TRUE model, it is important to evaluate their ability to capture the building's thermal dynamics and to identify potential issues in prediction applications. To this end, Bode magnitude plots of the different approaches are compared with the TRUE model in Fig. 4. Both the ID and OD models outperform the CONV method overall. Specifically, the CONV method failed to capture the behavior of u_h and u_c in the high-frequency range, whereas the ID and OD approaches performed better in this regard. This suggests that short-term heating and cooling behaviors can be captured more effectively with ID and OD, highlighting the importance of selecting an appropriate identification method for specific predictive applications.

The step responses of the measured disturbances (T_oa and $\dot{q}_\mathrm{sol,win}$) and control inputs (u_h and u_c) over a 12-hour period are compared with the TRUE system in Fig. 5. For the heating control input, all methods

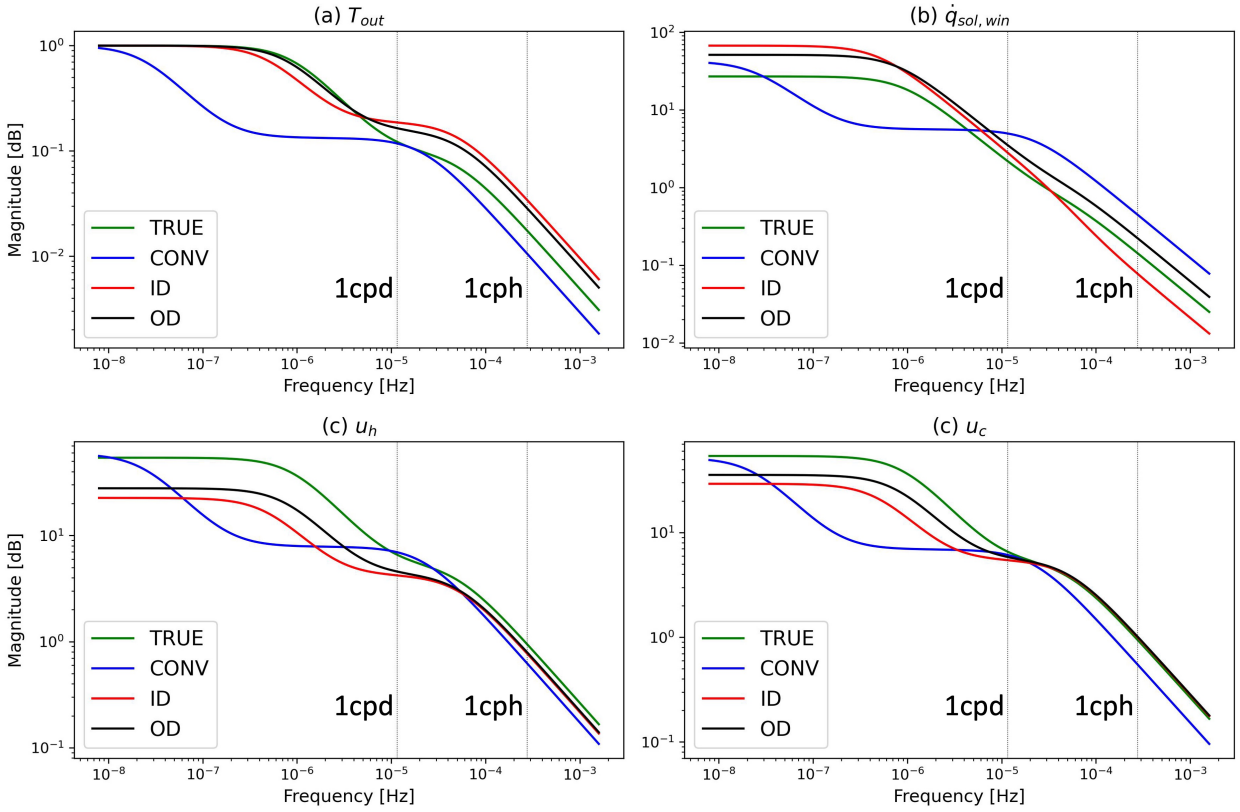


Figure 4: Comparison of Bode magnitude plots of measured disturbances (T_{oa} and $\dot{q}_{\mathrm{sol,win}}$) and control inputs ($u_{\mathrm{h}}$ and u_{c}) for each method. (1 cph = 2.8e-4 Hz, 1 cpd = 1e-5 Hz)

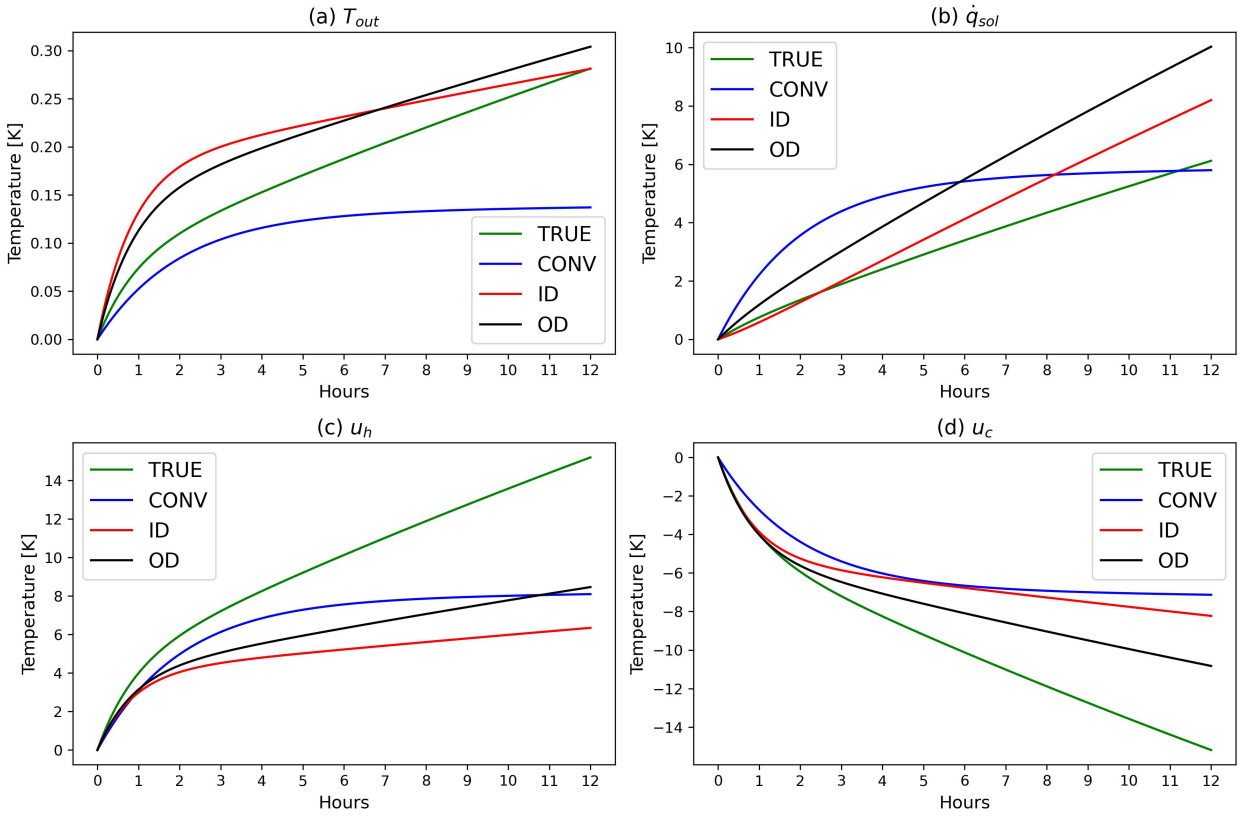


Figure 5: Comparison of step response of measured disturbances (T_{oa} and $\dot{q}_{\mathrm{sol,win}}$) and control input ($u_{\mathrm{h}}$ and u_{c}) for each method.

perform well up to 30 minutes but show a rapid decline thereafter. For the cooling control input, ID and OD maintain good performance during the first 1.5 hours, whereas CONV performs poorly even over a short-term horizon.

To compare the performance of each method, three-day-ahead temperature predictions using unmeasured disturbances as input data are evaluated (Fig. 6). As expected, the ID and OD methods show superior performance compared to the CONV method. Since the CONV method embeds all unmeasured disturbance information in its parameters, its temperature predictions deviate significantly when the magnitude of the unmeasured disturbance is large. Although small deviations are observed, the predictions of the ID and OD methods are generally close to the TRUE predictions. However, their performance declines during the weekend, when large unmeasured disturbances are absent, as their parameters perform poorly in capturing slow and long-term dynamics (i.e., the low-frequency region in Fig. 4).

For evaluating model accuracy, we compared the predicted heating and cooling loads with measurements, in addition to using the typical cross-validation strategy that directly compares output predictions (in our case, thermostat temperatures) with measurements from a validation dataset. This is important for control applications (e.g., MPC), where accurately estimating the required heating or cooling rate is essential—more

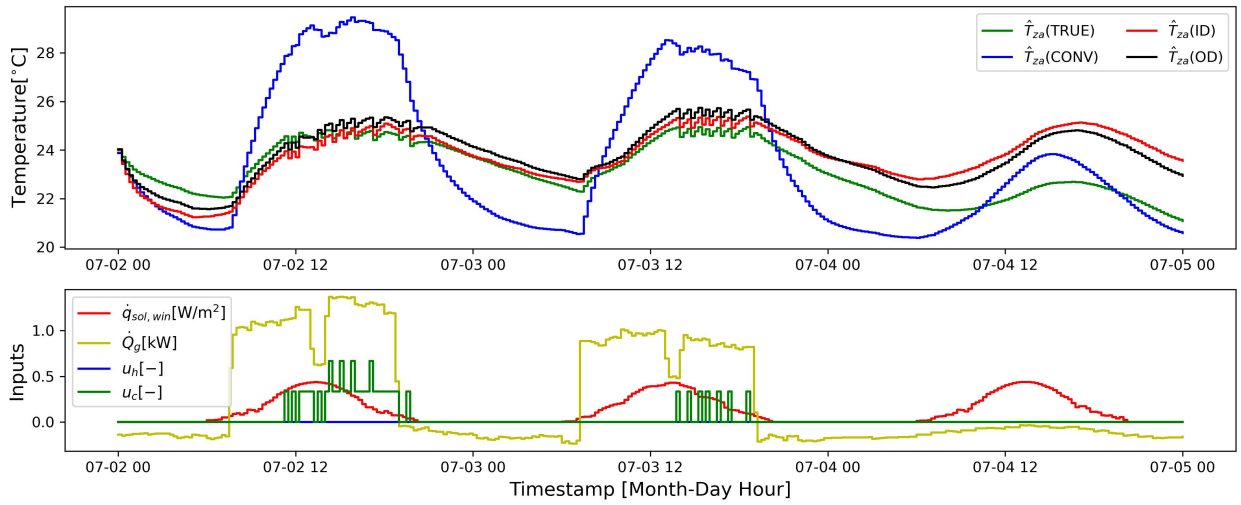


Figure 6: Comparison of indoor temperature predictions with unmeasured disturbance as an input for each method.

specifically, the runtime fraction (RTF) of the heating and cooling stages in our case, i.e., $\bar{u}_{\mathrm{h}}(k)$ and $\bar{u}_{\mathrm{c}}(k)$. To achieve this, the state-space model in Eqs. 1–2 was rewritten as Eqs. 15–16, ignoring the error noise term. The required RTF ($\hat{\mathbf{u}}(k)$) and the next-time states ($\hat{\mathbf{x}}(k+1)$) were estimated using Eq. 16. In this case, $(\mathbf{C}_{\mathrm{d}}\mathbf{B}_{\mathrm{d,u}}(\theta))$ is not invertible, so its pseudo-inverse, $(\mathbf{C}_{\mathrm{d}}\mathbf{B}_{\mathrm{d,u}}(\theta))^{\dagger}$, was used.

$$\begin{aligned}\mathbf{y}(k+1) &= \mathbf{C}_{\mathrm{d}}(\theta)\mathbf{x}(k+1)\\ &= \mathbf{C}_{\mathrm{d}}(\theta)\left(\mathbf{A}_{\mathrm{d}}\mathbf{x}(k) + \mathbf{B}_{\mathrm{d,u}}(\theta)\mathbf{u}(k) + \mathbf{B}_{\mathrm{d,w}}(\theta)\mathbf{w}(k)\right)\end{aligned} \tag{15}$$

$$\begin{aligned}\hat{\mathbf{u}}(k) &= (\mathbf{C}_{\mathrm{d}}\mathbf{B}_{\mathrm{d,u}}(\theta))^{\dagger}\left[\mathbf{y}(k+1) - \mathbf{C}_{\mathrm{d}}(\theta)\left(\mathbf{A}_{\mathrm{d}}\hat{\mathbf{x}}(k) + \mathbf{B}_{\mathrm{d,w}}(\theta)\mathbf{w}(k)\right)\right]\\ \hat{\mathbf{x}}(k+1) &= \mathbf{A}_{\mathrm{d}}\hat{\mathbf{x}}(k) + \mathbf{B}_{\mathrm{d,u}}(\theta)\hat{\mathbf{u}}(k) + \mathbf{B}_{\mathrm{d,w}}(\theta)\mathbf{w}(k)\end{aligned} \tag{16}$$

In Fig. 7, the required cooling RTF at each sampling time to maintain the measured temperature is compared across the methods. Similar to the temperature prediction results, the ID and OD methods perform well during cooling periods. However, all methods show some deviations during unoccupied periods.

To summarize, the OD method provides the best performance, although all approaches fail to capture long-term predictions. The ID method shows performance similar to the OD method, whereas the CONV method performs substantially worse than the other two, as expected. As described in [7], the output disturbance is a low-pass filtered version of the input disturbance, meaning that the smoother signal can be well modeled by the disturbance dynamics (Eq. 11). Similarly, the high-frequency nature of the input disturbance can be influenced by the estimation of the noise parameter scales ($\mathrm{e}_{\mathrm{x_{ID}}}$ in Eq. 6). Therefore, the OD method is adopted in this research due to both its performance and robustness.

2.4. Challenges in prediction application

In predictive applications such as MPC, accurate prediction performance is critically important. However, even with a perfect gray-box model, prediction becomes difficult when unmeasured disturbances are

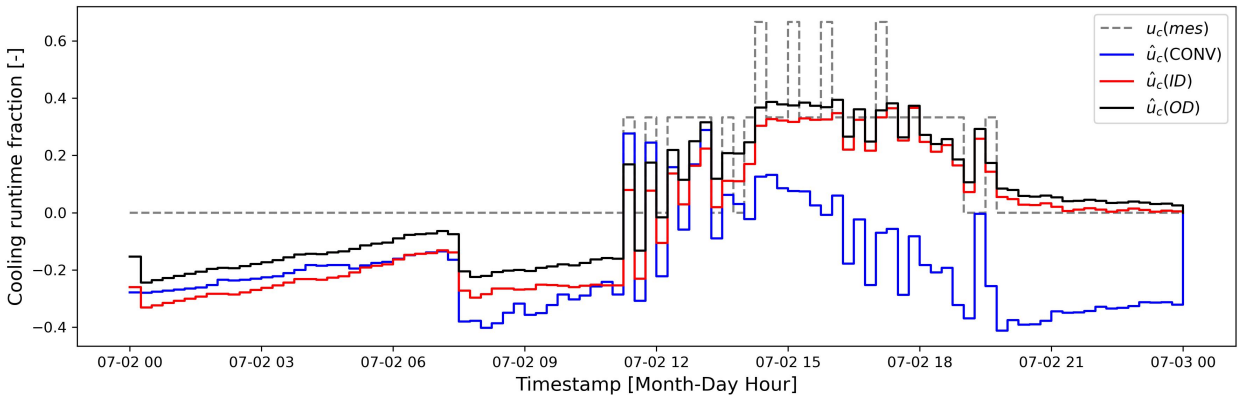


Figure 7: Comparison of required cooling runtime fraction to maintain the measured temperature.

significant. To illustrate this, consider a building subject to large unmeasured disturbances. In such cases, predicting the building dynamics is extremely challenging because the magnitude of the unmeasured disturbance is unknown. This section presents a detailed simulation study to highlight this issue and emphasize the need for an additional modeling approach—namely, the hybrid modeling method described in the following section—to address this limitation.

To evaluate the predictive performance of the models, we compared the one-day-ahead room air temperature predictions of the TRUE, CONV, and OD models with measurements (mes) (Fig. 8). Heat gain information (shown as the yellow-dotted line in the bottom figure) was deliberately excluded to simulate a realistic prediction scenario. Without this information, all models deviated from the measurements. Although OD performed better because it implicitly embeds heat gain effects in its parameters, this does not necessarily indicate superiority, as even the TRUE model produced incorrect predictions. The CONV model was expected to capture more heat gain information through its parameters, yet its predictions were not close to either the TRUE model or the measurements. These results indicate that, in addition to obtaining accurate parameters, it is essential to develop methods for forecasting unmeasured disturbances and compensating for parameter inaccuracies in predictive applications such as MPC.

3. Hybrid modeling approach for predictive application

3.1. Overview of hybrid modeling approach

In this section, we present a hybrid modeling approach that combines a gray-box building model with a machine-learning model to forecast unmeasured disturbances for predictive applications (Hybrid approach). Figure 9 illustrates the overall structure of the Hybrid approach in comparison with a conventional prediction approach (Conventional approach).

In the Conventional approach, the building model (i.e., a transfer function between the inputs (control inputs, measured disturbances, and unmeasured disturbances) and the output (temperature) as shown in Fig. 1) predicts future temperature using forecasts of future measured disturbances (i.e., T_{oa} and $\dot{q}_{\mathrm{sol,win}}$) and future control inputs (u_{h} and u_{c}). However, as discussed in Section 2.4 (see Fig. 8), this approach does not provide reliable predictions when the impact of unmeasured disturbances is significant, regardless of the quality of the gray-box building model.

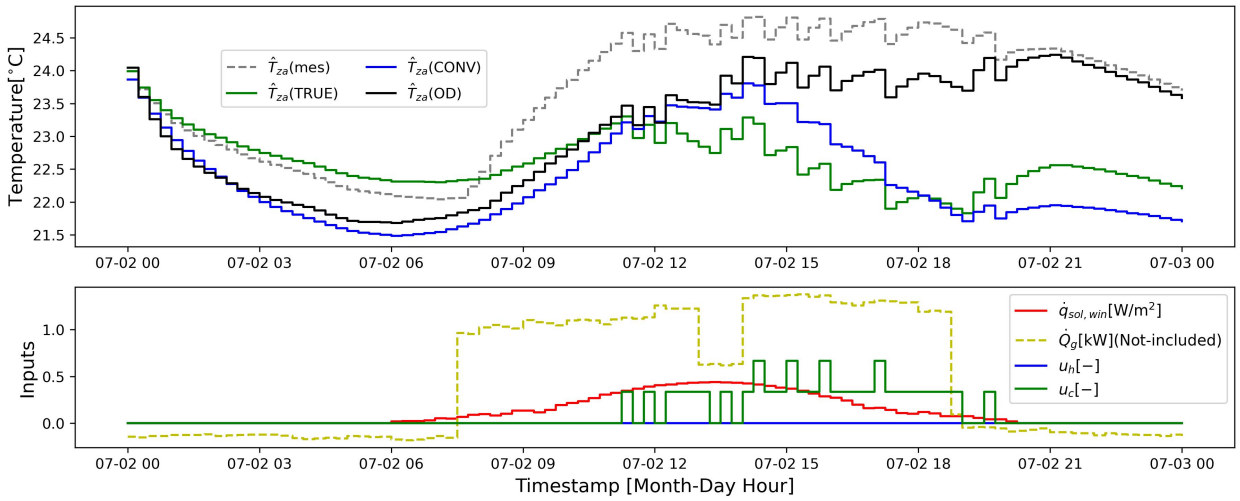


Figure 8: 1-day ahead prediction of the TRUE, CONV and OD models without heat gain information.

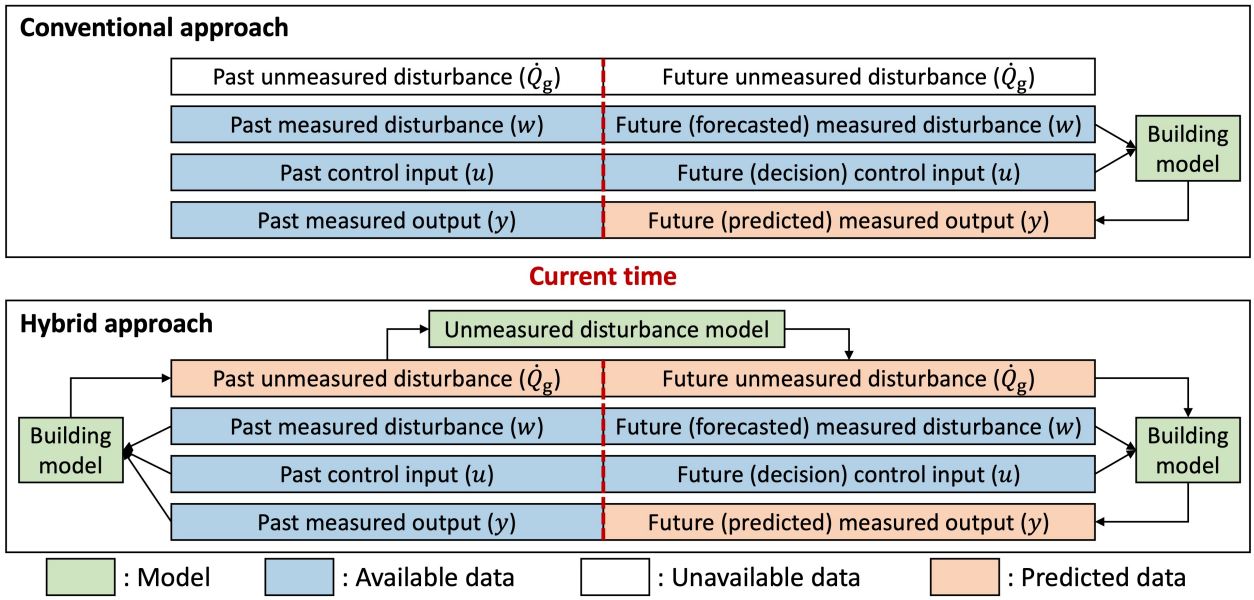


Figure 9: Comparison between Conventional and Hybrid approach.

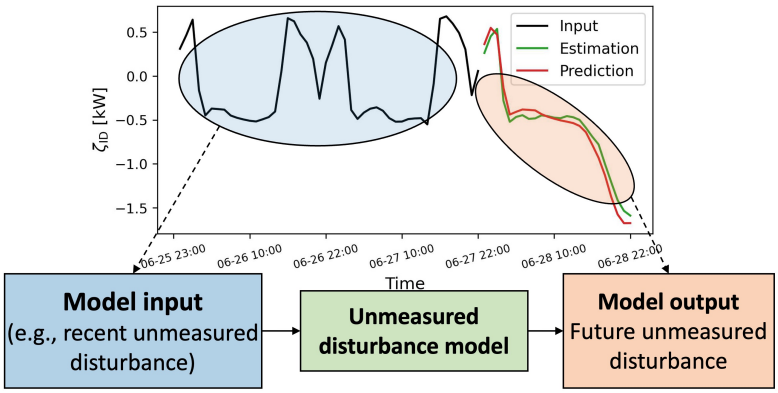


Figure 10: A graphical example of the disturbance model (ID case).

In contrast, the Hybrid approach incorporates predicted future unmeasured disturbances by using a disturbance model. The key steps are as follows:

1. Model identification — A gray-box building model is obtained through OD system identification.
2. Disturbance estimation — Using the building model, unmeasured disturbances (IDs, $\hat{\zeta}_{\mathrm{ID}}$, or ODs, $\hat{\nu}_{\mathrm{OD}}$) are estimated via Eq. 9 (ID) or Eq. 13 (OD).
3. Disturbance model development — The estimated IDs or ODs are separated into input (past) and output (future) components for developing a disturbance model. For example, a graphical representation of the disturbance model for the ID case is shown in Fig. 10.
4. Future disturbance prediction — The disturbance model predicts future unmeasured disturbances, which are then included in the gray-box building model for future predictions. Specifically, the predicted future IDs or ODs are directly used for temperature forecasting, as shown in Eqs. 17 and 18.

$$\begin{aligned} \hat{\mathbf{x}}_{\mathrm{ID}}(k+1;\theta) &= \mathbf{A}_{\mathrm{d,ID}}(\theta)\hat{\mathbf{x}}_{\mathrm{ID}}(k;\theta) + \mathbf{B}_{\mathrm{w,d}}(\theta)\mathbf{w}(k) + \mathbf{B}_{\mathrm{u,d}}(\theta)\mathbf{u}(k) \\ \hat{y}_{\mathrm{ID}}(k;\theta) &= \mathbf{C}_{\mathrm{d,ID}}\hat{\mathbf{x}}_{\mathrm{ID}}(k) \end{aligned} \tag{17}$$

where $\hat{\mathbf{x}}_{\mathrm{ID}}(k;\theta) = \left[\hat{\mathbf{x}}(k;\theta), \hat{\zeta}_{\mathrm{ID}}(k)\right]$, and $\hat{\zeta}_{\mathrm{ID}}(k)$ is predicted ID at time k.

$$\begin{aligned} \hat{\mathbf{x}}(k+1;\theta) &= \mathbf{A}_{\mathrm{d}}(\theta)\hat{\mathbf{x}}(k;\theta) + \mathbf{B}_{\mathrm{w,d}}(\theta)\mathbf{w}(k) + \mathbf{B}_{\mathrm{u,d}}(\theta)\mathbf{u}(k) \\ \hat{y}_{\mathrm{OD}}(k) &= \mathbf{C}_{\mathrm{d}}\hat{\mathbf{x}}(k;\theta) + \hat{v}_{\mathrm{OD}}(k) \end{aligned} \tag{18}$$

where $\hat{v}_{\mathrm{OD}}(k)$ is predicted OD at time k.

3.2. Design of Hybrid model structure

The first step in designing the Hybrid model is to understand the relationship between the quality of the gray-box model and the estimated unmeasured disturbances obtained from it (via Eq. 17 or 18). True unmeasured disturbances include various factors such as internal heat gains from occupants, appliances, plug

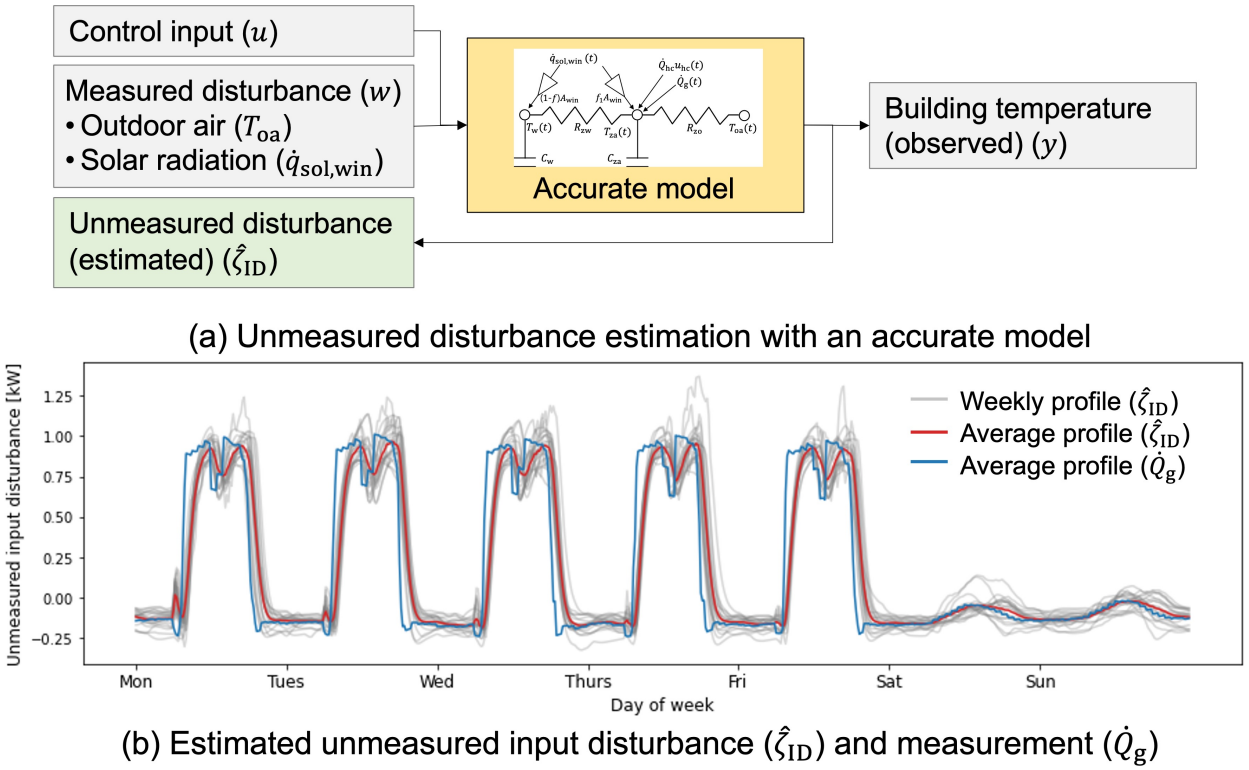
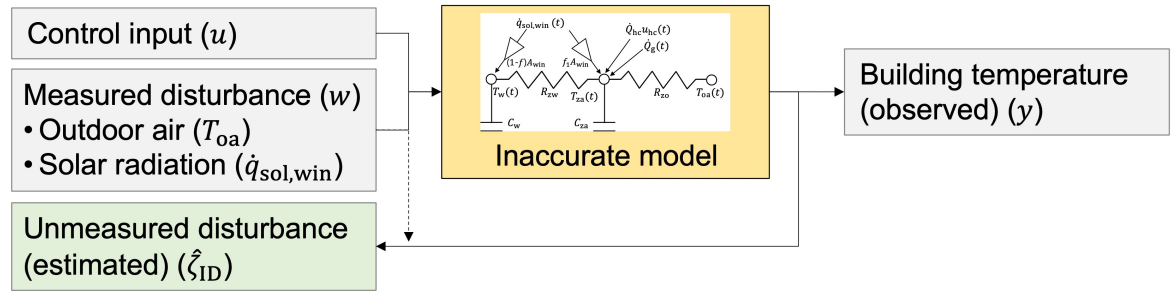


Figure 11: Unmeasured disturbance estimation with the accurate model.

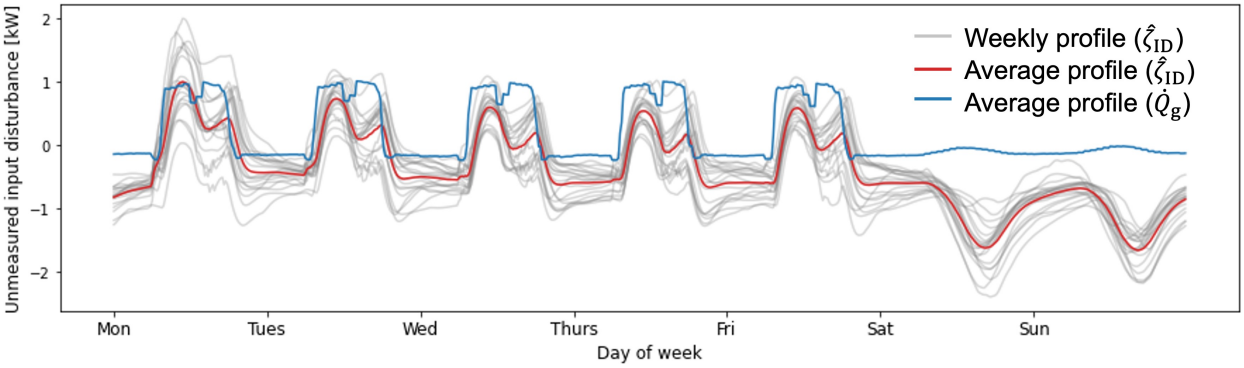
loads, infiltration, and ventilation due to window openings. Because of their stochastic nature, it is common to represent the average profiles of unmeasured disturbances using time-related features (e.g., daily or weekly occupancy schedules) [13, 27]. However, when significant unmeasured disturbances are present, the accuracy of the gray-box model decreases (Section 2.3). As a result, the estimated unmeasured disturbances may deviate from the true values. Therefore, it is important to understand how the quality of the gray-box model affects both the estimation of unmeasured disturbances and the overall performance of the Hybrid model.

Figure 11(a) presents a schematic diagram of unmeasured disturbance estimation ($\hat{\zeta}_{ID}$) using an accurate gray-box (TRUE) model. Based on the data shown in Fig. 3, the average estimated weekly profiles are compared with the average measured values ($\dot{Q}_g$) in Fig. 11(b). As shown in Fig. 3, each day's internal heat gains were generated according to human behavior patterns with added random noise. Consequently, the estimated weekly profiles also exhibit a recurring weekly pattern with some variability, and their average profile ($\hat{\zeta}_{ID}$) closely matches the average measured profile ($\dot{Q}_g$). This demonstrates that when the gray-box model is accurate, it is possible to extract meaningful information about unmeasured disturbance profiles from the data, which can then be modeled using time-related features.

However, when the estimated gray-box model is inaccurate due to significant unmeasured disturbances, the estimated unmeasured disturbances do not reliably represent the true disturbance profiles. Figure 12(a) shows a schematic diagram illustrating the estimation of unmeasured disturbances using an inaccurate model,



(a) Unmeasured disturbance estimation with an inaccurate model



(b) Estimated unmeasured input disturbance ($\hat{\zeta}_{ID}$) and measurement ($\dot{Q}_g$)

Figure 12: Unmeasured disturbance estimation with the inaccurate model.

while Fig. 12(b) compares the average estimated weekly profiles ($\hat{\zeta}_{ID}$) with the measured values ($\dot{Q}_g$). Although the estimated disturbances still exhibit weekly patterns, they deviate from the measured values. Furthermore, compared to the accurate model case shown in Fig. 11(a), the week-to-week estimations display greater variability, suggesting the presence of additional dynamics beyond internal heat gains. Notably, as illustrated in Fig. 12(a), information from the control inputs and measured disturbances must be incorporated into the estimation process when the system model lacks accuracy.

In this research, a deep learning model was selected to represent unmeasured disturbances among various machine-learning approaches. Deep learning has emerged as one of the most successful machine learning techniques in recent years, owing to its flexibility and ability to capture the nonlinear and complex characteristics of data, supported by rapid advancements in computing power [28]. Consequently, deep learning models have been widely applied in time-series forecasting [29, 30, 31]. Moreover, deep learning models can inherently account for characteristics of time-series data—such as stationarity, seasonality, and dynamic structure—without the need to explicitly define them in the forecasting model [32].

Due to the nature of deep-learning model, it involves an infinite number of possible hyperparameter combinations (e.g., network size, activation functions, optimization parameters). The complexity of a deep learning model is generally governed by four factors: model framework, model size, optimization process, and data complexity [33]. Model framework refers to the type of architecture used (e.g., feedforward neural

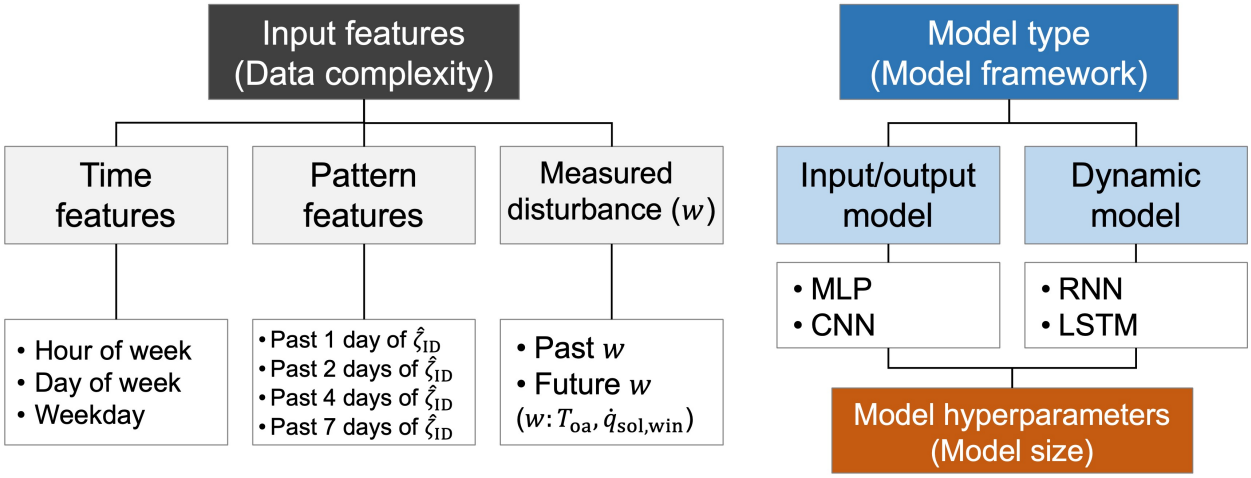


Figure 13: Model design matrix for the unmeasured disturbance model.

network, dynamic neural network) and the choice of activation functions. Model size relates to the number and width of hidden layers. The optimization process refers to the settings of the optimizer, such as the optimization algorithm and learning rate. Finally, data complexity concerns the dimensionality, distribution, and volume of the data.

Unmeasured disturbance data typically exhibits site-specific stochastic profiles, but often with periodic patterns (i.e., different buildings have different profiles). Therefore, it is important to identify model types and hyperparameters that can provide robust performance across different situations (i.e., buildings) by effectively capturing periodic patterns in the data. To explore various combinations, a model design matrix for the deep learning model (Fig. 13) was created based on the factors influencing model complexity, with the goal of selecting the most appropriate architecture. The optimization process factor was omitted because, for this relatively low-order problem (i.e., weekly patterns), it is not difficult to reach a global minimum without modifying optimization hyperparameters (e.g., learning rate).

The input features, which determine the data dimensionality, consist of three components: time, pattern, and measured disturbance. Time features (e.g., hour of the week) and pattern features (past n days of $\hat{\zeta}_{\mathrm{ID}}$) are commonly used in time-series modeling, as they capture time-specific and autoregressive characteristics. In addition, measured disturbances (w), as shown in Fig. 12, are also included as input features.

Four types of models are investigated in this study: Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM). The MLP and CNN have an input–output structure that maps an input time-series vector ($\boldsymbol{\psi} \in \mathbb{R}^{(n_\psi n_{\mathrm{k},\psi})}$) to an output time-series vector ($\boldsymbol{\xi} \in \mathbb{R}^{n_{\mathrm{k},\xi}}$), referred to as a feed-forward network. The input time-series vector is constructed by concatenating all input features over the input time period ($n_{\mathrm{k},\psi}$) into a one-dimensional vector. The output time-series vector is a one-dimensional vector representing the estimated unmeasured disturbance for the prediction time horizon ($n_{\mathrm{k},\xi}$).

In contrast, the RNN and LSTM are dynamical models that map the current input features ($\boldsymbol{\psi}(k) \in \mathbb{R}^{n_\psi}$) and the previous hidden states ($\mathbf{h}(k-1)$) to the updated hidden states ($\mathbf{h}(k)$) and output ($\boldsymbol{\xi}(k) \in \mathbb{R}^1$) at each time step k—a structure referred to as a feedback network.

Specifically, the MLP is a fully connected feed-forward neural network [28]. In each layer, the input time-series vector ($\boldsymbol{\psi}$) passes through a linear transformation (defined by weights $\mathbf{W}_i$ and biases $\mathbf{b}_i$), followed by an activation function (φ) and a dropout layer. This process is repeated n_{layer} times. Finally, an additional linear transformation (with parameters $\mathbf{W}_0$ and $\mathbf{b}_0$) is applied to produce the output ($\boldsymbol{\xi}$), as expressed in Eq. 19.

$$
\begin{aligned}
&\mathbf{z}_1 = \text{dropout}(\varphi(\mathbf{W}_1\boldsymbol{\psi} + \mathbf{b}_1)) \\
&\text{for } \ i \text{ in } 2 : n_{\text{layer}} : \\
&\quad \mathbf{z}_i = \text{dropout}(\varphi(\mathbf{W}_i\mathbf{z}_{i-1} + \mathbf{b}_i)) \\
&\boldsymbol{\xi} = \mathbf{W}_\xi \mathbf{z}_{n_{\text{layer}}} + \mathbf{b}_\xi
\end{aligned}
\tag{19}
$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ are weights and bias parameters that maps hidden layers from $\mathbf{z}_{i-1}$ to $\mathbf{z}_i$, $\mathbf{z}_i \in \mathbb{R}^{n_z}$ is hidden variable of i layer. φ is activation function, dropout is a dropout layer, which is commonly used for preventing over-fitting [28], and the following model size combinations were investigated ($n_{\text{layer}} \in [1, 2, 4]$, $n_z \in [10, 20, 50, 100]$, and $\varphi \in [\text{ReLU}, \text{SELU}, \text{GELU}]$ [28]).

CNN is widely used for image classification problems because it automatically extracts meaningful features from raw data through the convolution kernel [28]. It has a similar structure to MLP, but the convolution filter and max-pooling layer are used instead of the linear layer, and one more linear layer is for the last convolution layer (Eq. 20).

$$
\begin{aligned}
&\mathbf{z}_0 = \varphi(\text{maxpool}(\text{conv}(\boldsymbol{\psi}))) \\
&\text{for } \ i \text{ in } 1 : n_{\text{layer}} : \\
&\quad \mathbf{z}_i = \varphi(\text{maxpool}(\text{conv}(\mathbf{z}_{i-1}) \\
&\boldsymbol{\xi} = \mathbf{W}_\xi(\text{dropout}(\varphi(\mathbf{W}_{n_{\text{layer}}+1}\mathbf{z}_{n_{\text{layer}}} + \mathbf{b}_{n_{\text{layer}}+1}))) + \mathbf{b}_\xi
\end{aligned}
\tag{20}
$$

where conv is a convolution filter and maxpool is a max-pooling layer, and other notations are the same as MLP. The following model size combinations were investigated ($n_{\text{layer}} \in [1, 2]$, $n_z \in [10, 50, 100]$, $n_{\text{channel}} \in [10, 50, 100]$, $n_{\text{filter}} \in [6, 12]$, $n_{\text{pool}} \in [0, 4]$, and $\varphi \in [\text{ReLU}, \text{SELU}, \text{GELU}]$ [28]).

RNN is a neural network that uses an RNN cell [28, 34] as a recurrent layer. The cell maps input ($\boldsymbol{\psi}(k) \in \mathbb{R}^{n_\psi}$) and previous hidden states ($\mathbf{h}(k-1)$) to the current hidden states ($\mathbf{h}(k)$). The output ($\boldsymbol{\xi}(k) \in \mathbb{R}^1$) is calculated by applying a linear ($\mathbf{W}_h$ and $\mathbf{b}_h$), an activation, a dropout, and a linear ($\mathbf{W}_\xi$ and $\mathbf{b}_\xi$) layers to the hidden states ($\mathbf{h}(k)$) for each time step (k) (Eq. 21).

$$
\begin{aligned}
&\text{for } \ k \text{ in } 1 : (n_{\text{k},\psi} + n_{\text{k},\xi}) : \\
&\quad \mathbf{h}(k) = \text{RNN}\left(\mathbf{h}(k-1), \boldsymbol{\psi}(k-1)\right) \\
&\text{for } \ k \text{ in } 1 : (n_{k,\xi}) : \\
&\quad \boldsymbol{\xi}(k) = \mathbf{W}_\xi(\text{dropout}(\varphi(\mathbf{W}_h\mathbf{h}(k) + \mathbf{b}_h))) + \mathbf{b}_\xi
\end{aligned}
\tag{21}
$$

where $\mathbf{h}$ is a hidden state vector that holds memory. $\boldsymbol{\psi}(k)$ is subset of $\left(\text{time}(k), w(k), \hat{\zeta}_{\text{ID}}(k)\right)$, and $\hat{\xi}_{(n_{\text{k},\psi}+1):(n_{\text{k},\psi}+n_{\text{k},\xi})}$ are used to for $\hat{\zeta}_{(n_{\text{k},\psi}+1):(n_{\text{k},\psi}+n_{\text{k},\xi})}$. The following model size combinations were investigated ($n_{\text{layer}} \in [1, 2, 4]$, $n_z \in [10, 20, 40, 60]$, and $\varphi \in [\text{ReLU}, \text{SELU}, \text{GELU}]$ [28]).

LSTM is one type of RNN using a LSTM cell [28, 35]. It maps input ($\boldsymbol{\psi}(k) \in \mathbb{R}^{n_\psi}$), previous hidden states ($\mathbf{h}(k-1)$), and previous cell states ($\mathbf{c}(k-1)$) to the current hidden ($\mathbf{h}(k)$) and cell stats ($\mathbf{c}(k)$). The

output ($\boldsymbol{\xi}(k) \in \mathbb{R}^1$) is calculated by applying a linear ($\mathbf{W}_h$ and $\mathbf{b}_h$), an activation, a dropout, and a linear ($\mathbf{W}_\xi$ and $\mathbf{b}_\xi$) layers to the hidden states ($\mathbf{h}(k)$) for each time step (k) (Eq. 22).

$$\begin{aligned} &\text{for } \; k \text{ in } 1:(n_{k,\psi}+n_{k,\xi}): \\ &\quad \mathbf{h}(k), \mathbf{c}(k), = \text{LSTM}\left(\mathbf{h}(k-1), \mathbf{c}(k-1), \boldsymbol{\psi}(k-1)\right) \\ &\text{for } \; k \text{ in } 1:(n_{k,\xi}): \\ &\quad \boldsymbol{\xi}(k) = \mathbf{W}_\xi(\text{dropout}(\varphi(\mathbf{W}_h\mathbf{h}(k)+\mathbf{b}_h)))+\mathbf{b}_\xi \end{aligned} \tag{22}$$

where $\mathbf{h}$ and $\mathbf{c}$ are hidden and cell state vector that holds short- and long-term memory, respectively. $\mathbf{h}$ is a hidden state vector that holds memory. $\boldsymbol{\psi}(k)$ is subset of $\left(\text{time}(k), w(k), \hat{\zeta}_{\text{ID}}(k)\right)$, and $\hat{\xi}_{(n_{\text{k},\psi}+1):(n_{\text{k},\psi}+n_{\text{k},\xi})}$ are used to for $\hat{\zeta}_{(n_{\text{k},\psi}+1):(n_{\text{k},\psi}+n_{\text{k},\xi})}$. The following model size combinations were investigated ($n_{\text{layer}} \in [1,2,4]$, $n_{\text{z}} \in [10,20,40,60]$, and $\varphi \in [\text{ReLU}, \text{SELU}, \text{GELU}]$ [28]).

3.3. Model selection

In this section, we present the model selection process for the unmeasured disturbance model. When selecting a deep learning model, the typical approach is to identify the optimal set of parameters that yield the best evaluation metrics while avoiding overfitting, often through techniques such as cross-validation [36], given a set of hyperparameters. Hyperparameters themselves can be optimized by repeating the parameter optimization process, for example, via Bayesian optimization [28]. However, since our model design matrix is relatively small (Fig. 13), we can determine the best set of hyperparameters and parameters using a simple grid search.

Unmeasured disturbance data is inherently site-specific (i.e., different profiles for different occupants and buildings). Therefore, it is important to identify a model that can capture the underlying pattern of the input–output relationship, as described in Section 3.2. By accurately capturing this pattern, the model can achieve robust performance in extrapolation tasks without overfitting the training data. This requires first determining whether the model structure is suitable for representing the input–output relationship pattern. The expressible functional space of the model is determined primarily by the model framework (model type) and input features (data complexity). In addition, model size, defined by the depth and width of the network [33], influences the model's complexity. When the model type and input features are appropriate for capturing the input–output pattern, the model can perform well if it has at least the minimum required size (i.e., minimum model order).

If the model is trained correctly to avoid overfitting (e.g., via cross-validation), unnecessary parameters will decay when the model size exceeds what is needed. In such cases, the model's performance will remain consistent across a range of model sizes, indicating robustness. Conversely, if the model type and input features are poorly suited to the input–output relationship, the model will likely overfit without truly learning the underlying pattern, leading to greater sensitivity to hyperparameters related to model size.

Our model development process follows three key steps:

1. Model training — The dataset is split into training and testing sets. All cases in the model design matrix are included in the training phase. An optimizer identifies the parameters that minimize the training error. To prevent overfitting, we employ early stopping [28], which halts optimization when the testing error begins to increase. Each model iteration involves tuning various hyperparameters, including the number and size of hidden layers, activation functions, convolution filter dimensions, pooling layer sizes, and more. We iteratively search for the hyperparameter combination that minimizes the root-mean-squared error (RMSE) on the test set. PyTorch [37] is used for model implementation and training. To assess generalizability across seasons, we train the model on one month of data and test it on the subsequent six months. We also include

data from two contrasting climates: Berkeley, CA (mild climate) and Chicago, IL (four distinct seasons). All data is resampled to hourly intervals, assuming hourly occupancy patterns for unmeasured disturbances [13, 27], and standardized using z-score normalization.

2. Pattern expressiveness assessment — We evaluate the model's ability to capture the input–output relationship pattern using regression-based statistical tests. Regression variables are derived from the input features and model types to assess their influence on the test error. If including certain input features or using a specific model type results in statistically significant performance improvements, we infer that the model has greater potential to capture the underlying pattern. To avoid performance degradation from undersized models, only the top three models from the regression test are considered for further evaluation. The regression model is formulated in Eq. 23.

$$\begin{aligned} \upsilon_{\text{test}}(c) =& \beta_0 + \beta_{\text{CNN}}\chi_{\text{CNN}}(c) + \beta_{\text{RNN}}\chi_{\text{RNN}} + \beta_{\text{LSTM}}\chi_{\text{LSTM}}(c) + \beta_{\text{time}}\chi_{\text{time}}(c) + \\ & \beta_{\text{pattern}}\chi_{\text{pattern}}(c) + \beta_{\text{past-w}}\chi_{\text{past-w}}(c) + \beta_{\text{future-w}}\chi_{\text{future-w}}(c) + \beta_{\text{ID}}\chi_{\text{ID}}(c) \end{aligned} \tag{23}$$

where c is each case identifier, υ_{test} is a root-mean squared error of temperature prediction (Eqs. 17-18) on test data, χ_{CNN}, χ_{RNN}, and χ_{LSTM} are a binary indicator of whether to use each model type (i.e., 0 is MLP model), χ_{time} is a binary indicator of inclusion of any time feature, χ_{pattern} is a numeric number of the length of pattern feature, $\chi_{\text{past-w}}$ is a binary indicator of inclusion of past w feature, $\chi_{\text{future-w}}$ is a binary indicator of inclusion of future w feature, χ_{ID} is a binary indicator of either ID model or OD model (ID is 1).

The regression model is trained using a Bayesian approach [38], which allows direct interpretation of the marginal effect of each variable (β_*). For instance, if the posterior distribution of $\beta_{\text{future-w}}$ shows that only 1% of samples are greater than zero, this implies that including $\beta_{\text{future-w}}$ in the model would reduce the prediction error with 99% probability compared to the case without its inclusion.

3. Once several combinations of input features and model types are identified in the second step, the robustness of these combinations is evaluated by analyzing the variation in prediction error across different model sizes. For this comparison, each pair of combinations is tested statistically to determine whether their error distributions differ significantly. Each error distribution is modeled as a log-normal distribution, and the resulting pairs are compared using the methodology described in Eqs. 24–25.

$$\begin{aligned} &\text{for } c \text{ in all cases} \\ &\quad \mathrm{P}(\mu_\upsilon(c)) = \text{Normal}(0, 10) \\ &\quad \mathrm{P}(\sigma_\upsilon(c)) = \text{HalfNormal}(0, 1) \\ &\quad \mathrm{P}(\upsilon_{\text{test}}(c)|\mu_\upsilon(c), \sigma_\upsilon(c)) = \text{logNormal}\left(\upsilon_{\text{test}}(c)|\mu_\upsilon(c), \sigma_\upsilon(c)\right) \\ &\quad \mathrm{P}(\mu_\upsilon(c), \sigma_\upsilon(c)|\upsilon_{\text{test}}(c)) \propto \mathrm{P}(\upsilon_{\text{test}}(c)|\mu_\upsilon(c), \sigma_\upsilon(c))\mathrm{P}(\mu_\upsilon(c))\mathrm{P}(\sigma_\upsilon(c)) \end{aligned} \tag{24}$$

where $\mu_\upsilon(c)$ and $\sigma_\upsilon(c)$ are mean and standard deviation of Log-Normal distribution. Log-normal distribution is used because RMSE is a non-negative value.

$$\text{Probability of } c_1 \text{ is better than } c_2\text{: } \mathrm{P}\left((\tilde{\upsilon}_{\text{test}}(c_1)|\upsilon_{\text{test}}(c_1) - \tilde{\upsilon}_{\text{test}}(c_2)|\upsilon_{\text{test}}(c_2)) < 0\right) \tag{25}$$

where $\tilde{\upsilon}_{\text{test}}(c)|\upsilon_{\text{test}}(c)$ is posterior predictive samples [38].

Finally, once suitable model cases that are robust with respect to model size have been identified, the final model can be selected by also considering practical factors, such as the amount of required data. For example, if the model is to be used in MPC, predictions are generated at every MPC sampling time (i.e., control time step). If the model requires n days of recent data for Past w and ζ_{ID}, then n days of data

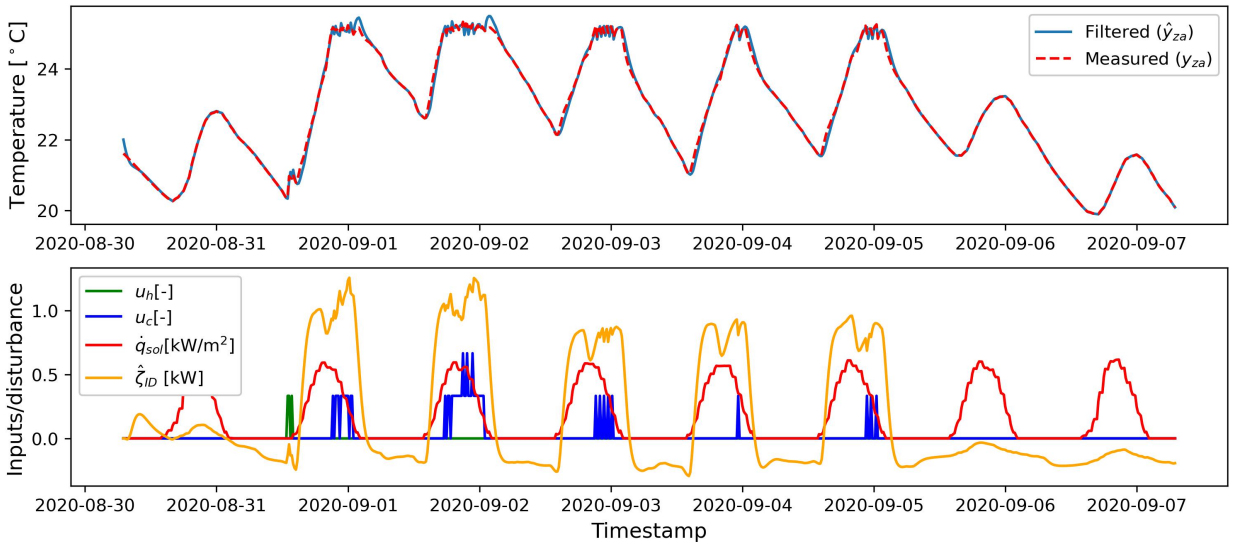


Figure 14: Estimated input disturbance profile in a week.

must be retrieved from the database for all zones at each prediction step. This can create computational and database burdens in the MPC framework. Therefore, it is advisable to minimize the amount of recent data required.

In addition, the ID model may be preferred over the OD model in certain cases. Figures 14 and 15 show the estimated ID and OD profiles for a one-week period. While the ID profile directly represents unmeasured heat gains, the OD profile reflects the cumulative impact of ID on the zone air temperature. Consequently, if the performances of the two models are similar, the ID model's outputs are more straightforward to interpret.

4. Result

The unmeasured disturbance model was trained and evaluated using the data generated in Section 2.1. However, since the weather conditions in the Berkeley, CA area are mild and lack distinct seasonal variations, additional datasets were generated to represent four seasons. These datasets were created using weather conditions from Chicago, IL [23], while maintaining similar building conditions.

4.1. Unmeasured disturbance model

Figs. 16 and 17 present randomly sampled prediction results of a well-performing unmeasured disturbance (ID) model on the training and test datasets for Chicago weather, respectively. The selection of this model was carried out in the subsequent section through the model selection process. In the top panels, the predicted ID profiles are compared with the measured profiles (i.e., those estimated from data using Eq. 9). As shown in Fig. 3, the magnitude of unmeasured disturbances is generally higher on weekdays than on weekends. The selected model demonstrates strong predictive performance, with close alignment between predicted and measured profiles for both the training and test datasets. Consequently, the Hybrid approach

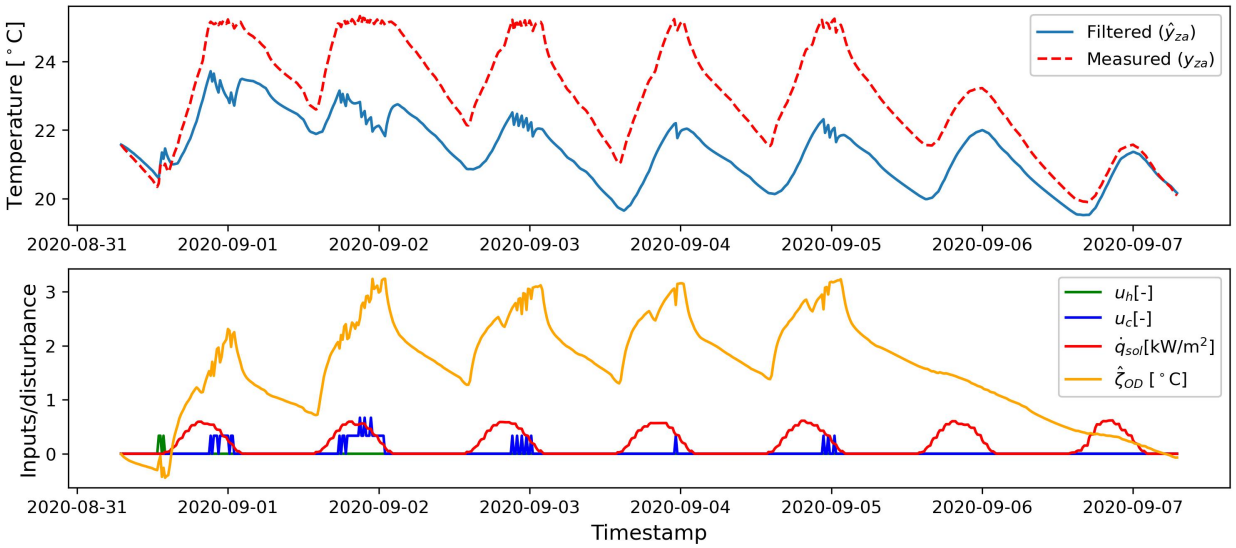


Figure 15: Estimated output disturbance profile in a week.

(h) reduces the RMSE for one-day-ahead temperature prediction by approximately 0.3–2.0°C compared to the Conventional approach (c).

Figures 18 and 19 show the prediction results of a poorly performing unmeasured disturbance (ID) model on both the training and test datasets. These results underscore the critical impact of model quality on prediction performance. Although the model was trained with measures to prevent overfitting, its inadequate structure fails to capture the essential features of the data. As a result, the model exhibits overfitting and poor extrapolation capability, as evidenced by the substantial difference between its predictions on the training data (Fig. 18) and the test data (Fig. 19).

4.2. Model selection

One of the primary contributions of this study is the demonstration of a systematic approach for determining the structure of an unmeasured disturbance model through a model selection process. As described in Section 3.3, the model's ability to capture the input–output relationship pattern is assessed using a regression model (Eq. 23), and the posterior distributions of the regression parameters are visualized in Fig. 20.

Compared to the MLP model type, most CNN and LSTM samples show a reduction in prediction error, whereas the RNN does not. However, some CNN and LSTM samples exhibit errors greater than zero (opposite for the RNN). This is primarily because the regression analysis only reflects the marginal effect of each variable without accounting for potential interaction effects. Nonetheless, the inclusion of the time feature and future w consistently reduces errors. In contrast, the pattern features and past w tend to increase errors. Considering the scales and distributions of the regression parameters, the positive effects of the time feature and future w are clear, while the effects of the other variables remain inconclusive.

In addition, the use of the ID model (β_{output}) leads to lower errors. This can be attributed to the structural difference between the ID and OD models. While the OD model's predictions are directly added to the temperature prediction, the ID model predicts unmeasured heat gains, whose influence on temperature

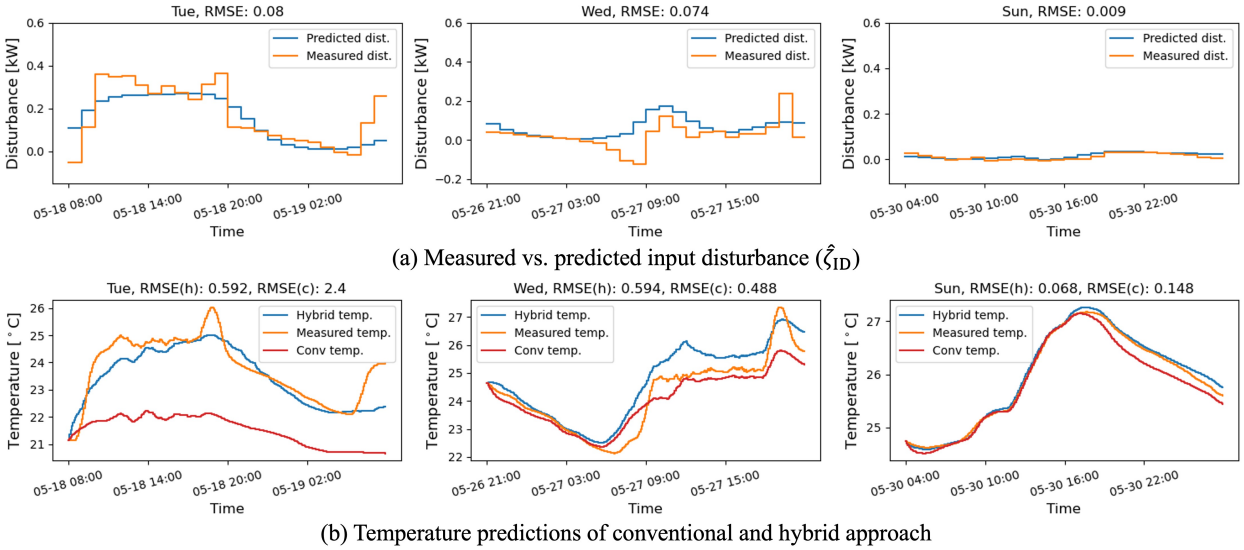


Figure 16: Prediction results of a good model on train data; Day information and RMSE are shown on the top

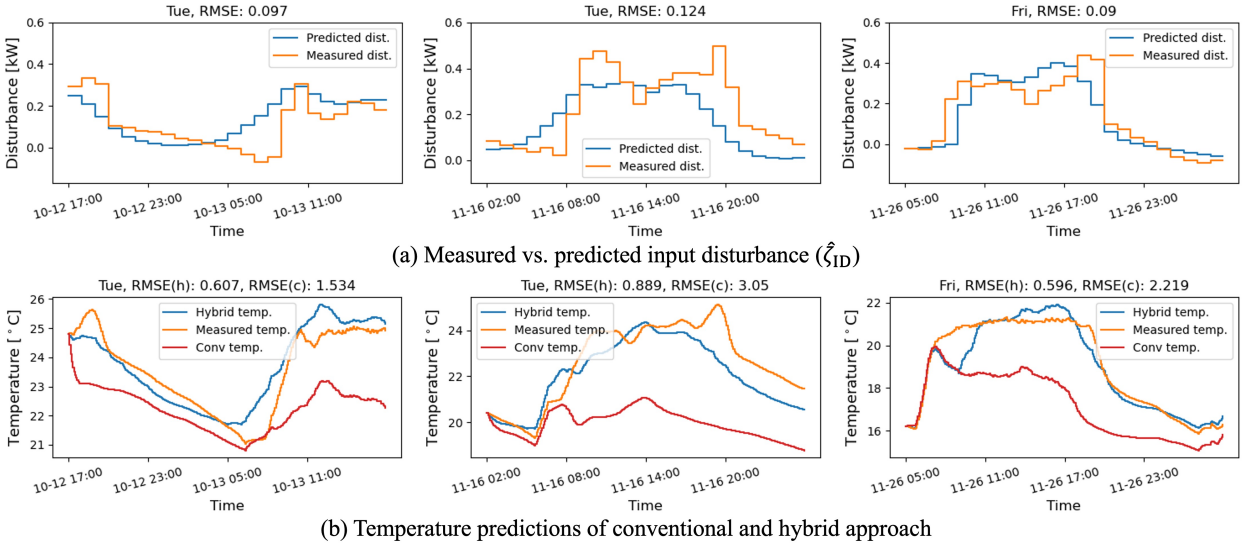
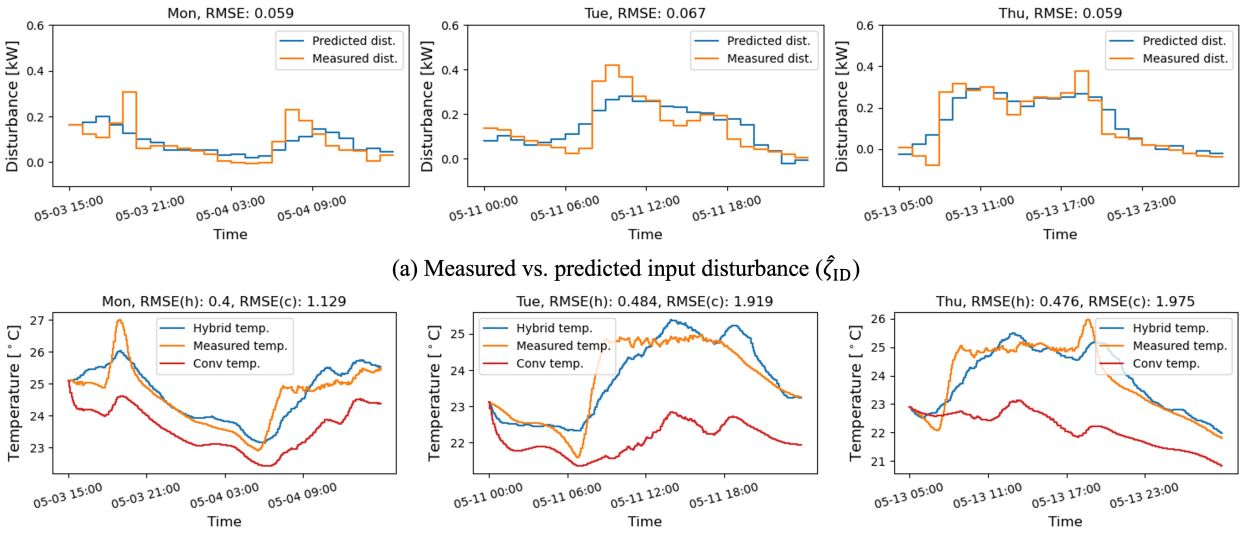


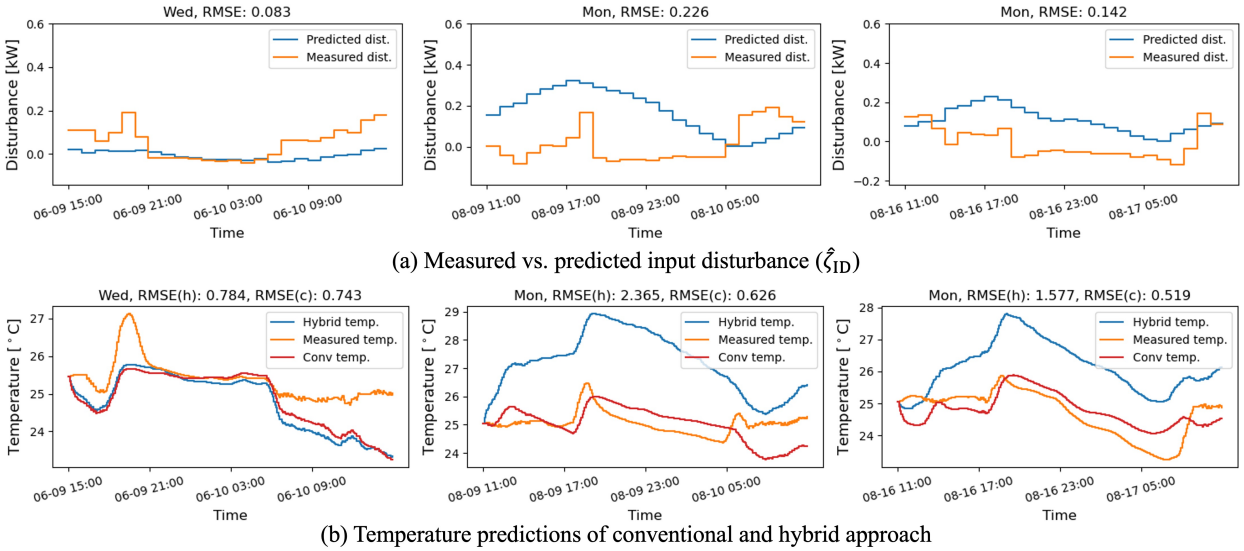
Figure 17: Prediction results of a good model on test data; Day information and RMSE are shown on the top



(a) Measured vs. predicted input disturbance ($\hat{\zeta}_{\mathrm{ID}}$)

(b) Temperature predictions of conventional and hybrid approach

Figure 18: Prediction results of a bad model on train data; Day information and RMSE are shown on the top



(a) Measured vs. predicted input disturbance ($\hat{\zeta}_{\mathrm{ID}}$)

(b) Temperature predictions of conventional and hybrid approach

Figure 19: Prediction results of a bad model on test data; Day information and RMSE are shown on the top

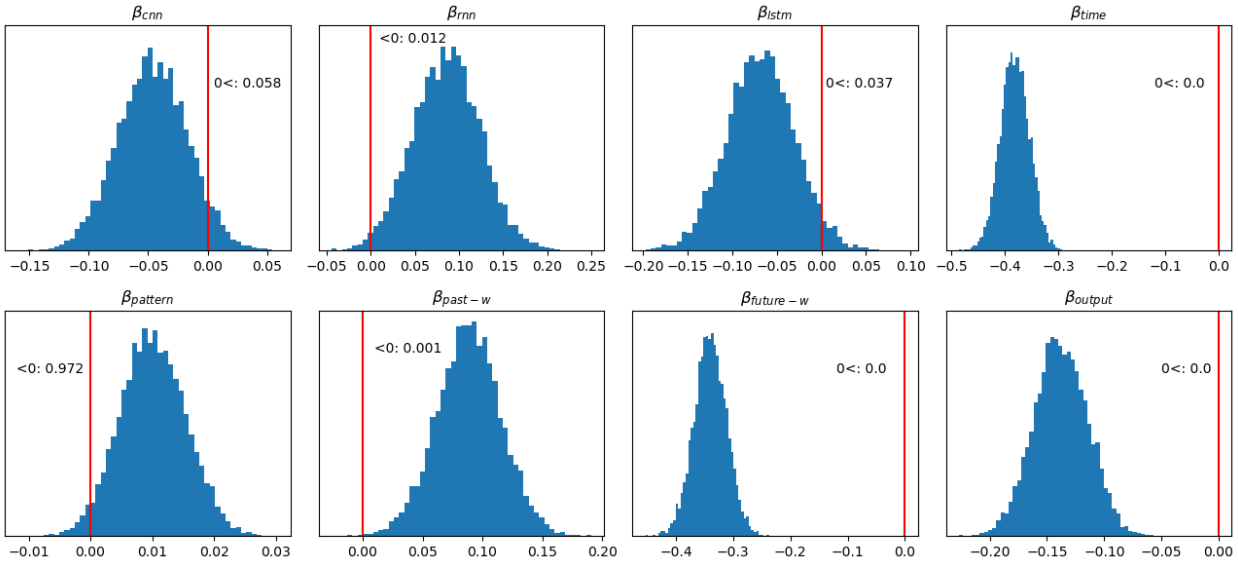


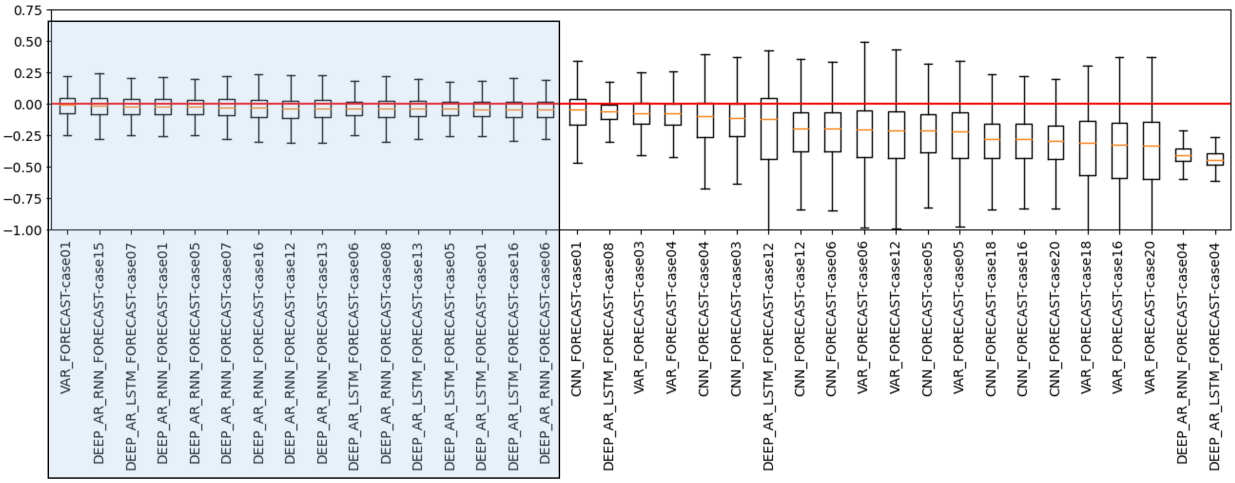
Figure 20: Model type and input feature's marginal effect on model performance

prediction error is smaller. Based on these findings, the time feature, future w, and ID model are selected as suitable model components.

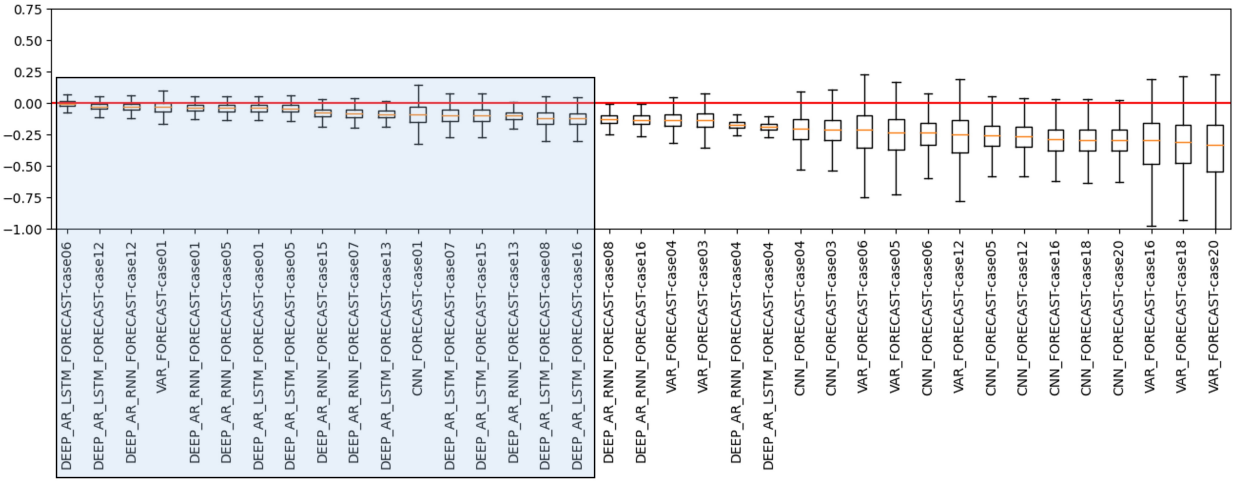
To evaluate robustness, we applied Eq. 24 to the selected regression cases. The model case with the lowest median prediction error across various model sizes was identified and compared to other cases. The comparison results, sorted by median values, are shown in Fig. 21, with details of each model case provided in the Appendix. Model cases that do not show statistically significant differences from the best case are highlighted in blue boxes.

One notable observation is that the dynamical models (i.e., RNN and LSTM) exhibit narrower error distributions, indicating greater robustness to changes in model size. Since all model cases in the blue boxes show similar performance, the final model structure is selected based on practical considerations. From Fig. 20, the LSTM is preferred over the RNN due to its stronger marginal effect. Furthermore, as discussed in Section 3.3, the LSTM requires less historical data, making it more efficient in terms of database usage. Finally, we found that the hour-of-the-week (how) feature is more effective in capturing weekly patterns than the day-of-the-week (dow) or weekday features. Based on these findings, the `DEEP_AR_LSTM_FORECAST-case01` is chosen as the final model structure.

We further investigated the impact of model size hyperparameters on the test prediction error (Fig. 22). As described in Eq. 22, we trained models with varying sizes and numbers of hidden layers, as well as different activation functions, and then applied the same regression analysis outlined in Eq. 23. The results indicate that increasing the number of hidden layers reduces the test prediction error, whereas the other variables either had no effect or a negative effect. Based on these findings, we selected the final model configuration as φ: ReLU and n_{layer}: 1. However, the optimal size of the hidden layers is determined during the training process by selecting the configuration that yields the lowest test RMSE.

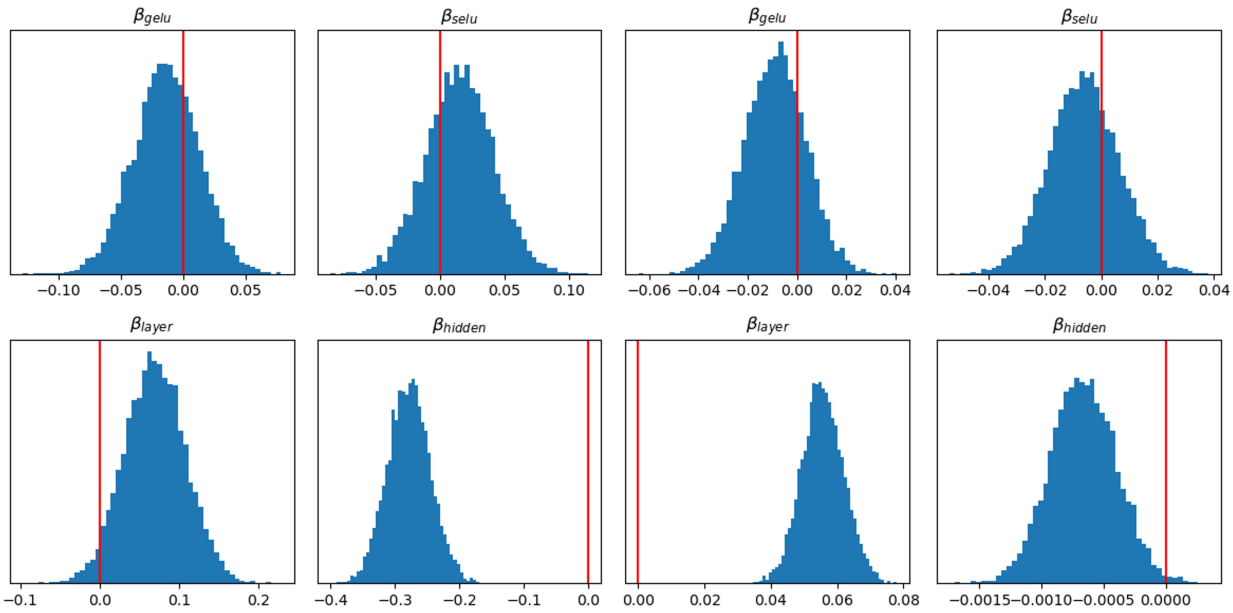


(a) Error differences against the best combination (DEEP_AR_LSTM_FORECAST-case15) in Berkeley weather



(b) Error differences against the best combination (DEEP_AR_RNN_FORECAST-case06) in Chicago weather

Figure 21: Robustness of each model type and input feature combination compared to the best combination; probability lower than zero (red line) indicates the probability of best combination is better than each case



(a) Berkely weather case

(b) Chicago weather case

Figure 22: Impact of model size hyperparameters on test prediction error

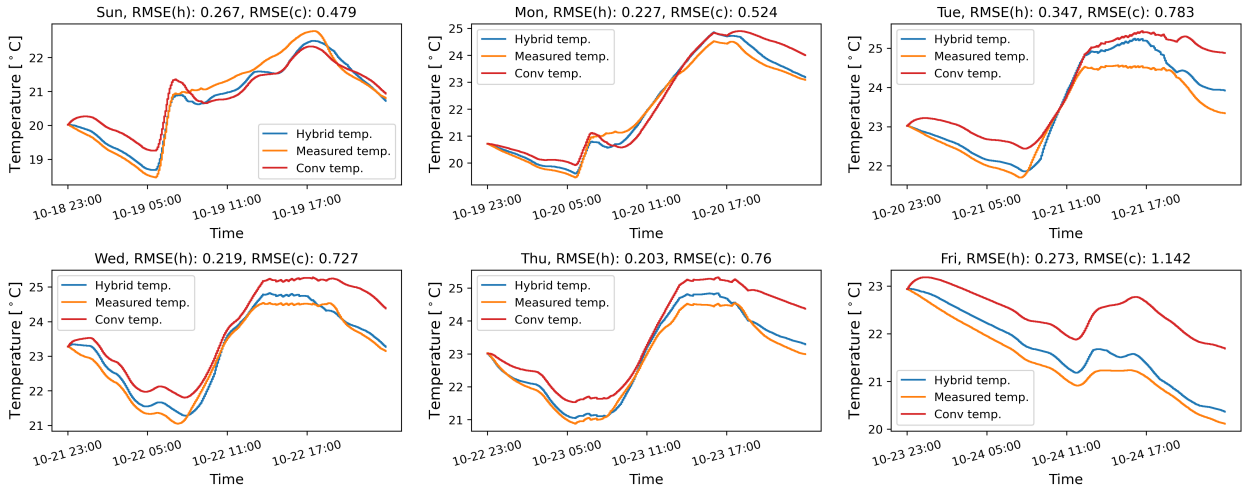


Figure 23: Temperature predictions of Hybrid and Conventional approaches for Berkeley data

4.3. Prediction performance on simulated data

Since the models are trained using cooling season data, it is important to evaluate their prediction performance on heating season data. Figures 23 and 24 present the one-day-ahead temperature predictions and the required heating load for Berkeley weather, respectively. Both the Hybrid and Conventional approaches capture the overall temperature profiles; however, the Conventional approach exhibits greater divergence over time, resulting in an RMSE increase of approximately 0.2–0.9°C. Similarly, both approaches capture the overall profile of the required heating load (as described in Eqs. 15–16), but the Hybrid approach achieves further error reduction, particularly on weekends. This improvement is attributed to the inclusion of unmeasured disturbance information in the gray-box model, as shown in Fig. 8. The temperature and required heating load predictions for the heating season in Chicago weather are shown in Figs. 25–26. Overall, both approaches perform worse than in the Berkeley case. However, the Hybrid approach achieves an RMSE reduction of approximately 0.3–2°C for temperature and 0.05–0.18 kW for the required heating load compared to the Conventional approach.

4.4. Prediction performance on experimental data

In this study, a Hybrid modeling approach was applied to a single-cell representation of a small office area within the research facility FLEXLAB. FLEXLAB [22], located at Lawrence Berkeley National Laboratory in Berkeley, California, USA, is a well-instrumented experimental test facility specifically designed for evaluating advanced building and grid technologies. For this experiment, a packaged heat pump rooftop unit (HP-RTU) was retrofitted to test an advanced control algorithm. The control experiment results, including the application of the Hybrid approach, are presented in a separate paper [39]. The focus of this paper is on evaluating the prediction performance of the Hybrid modeling approach.

The experimental cell represents a small office space with a floor area of 57 m^2 and a large north-facing window. The HP-RTU (AAON RQ 2-ton unit) features two-stage heating and cooling control, with nominal capacities of 6.53 kW for heating and 6.16 kW for cooling at 700 CFM (1190 CMH). The HP-RTU is controlled by a Schneider Electric SE8600 thermostat, with heating and cooling stages determined by the

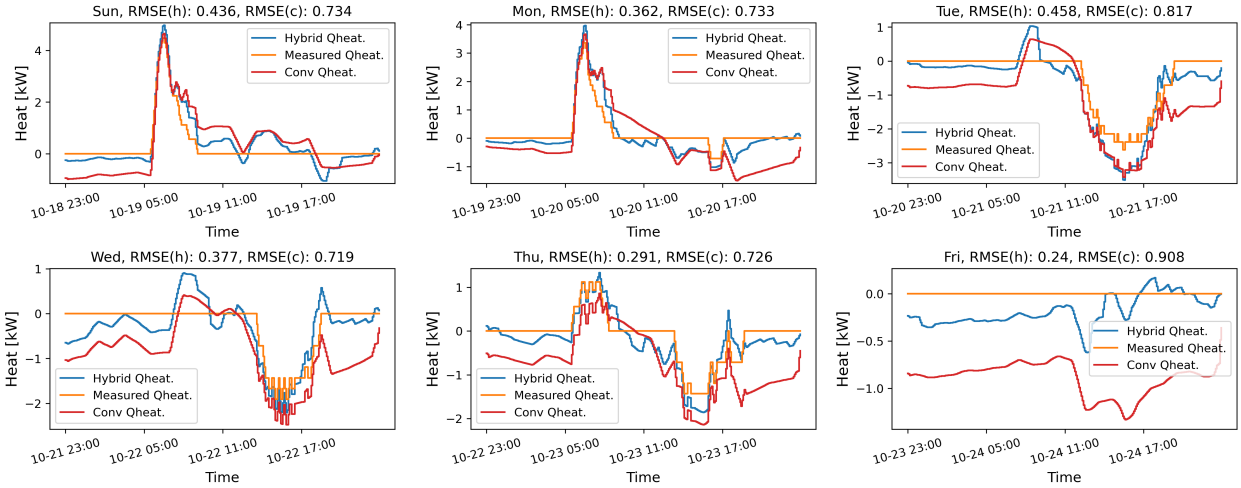


Figure 24: Amount of required heating rate of Hybrid and Conventional approaches for Berkeley data

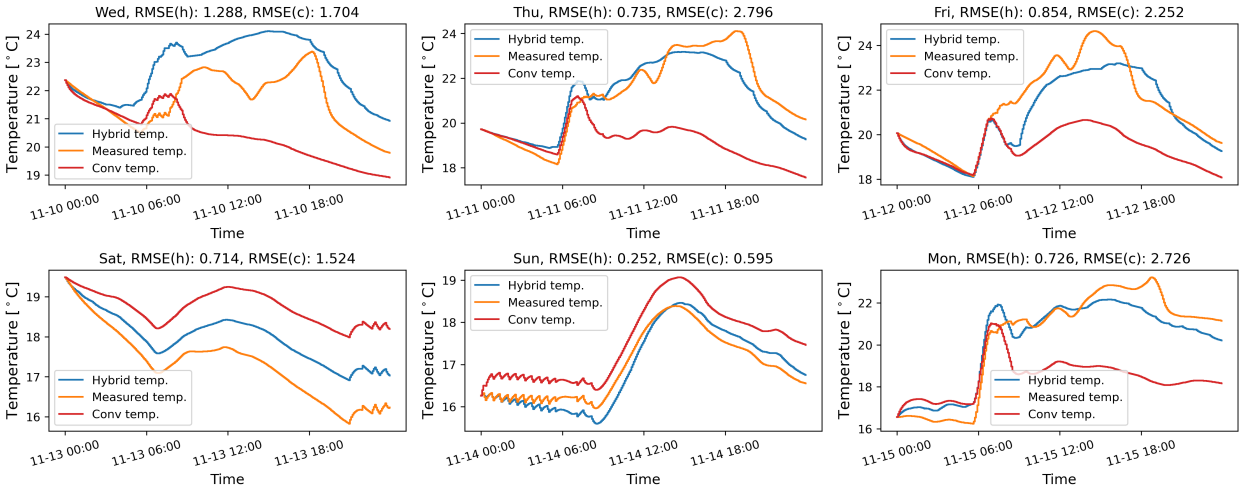


Figure 25: Temperature predictions of Hybrid and Conventional approaches for Chicago data

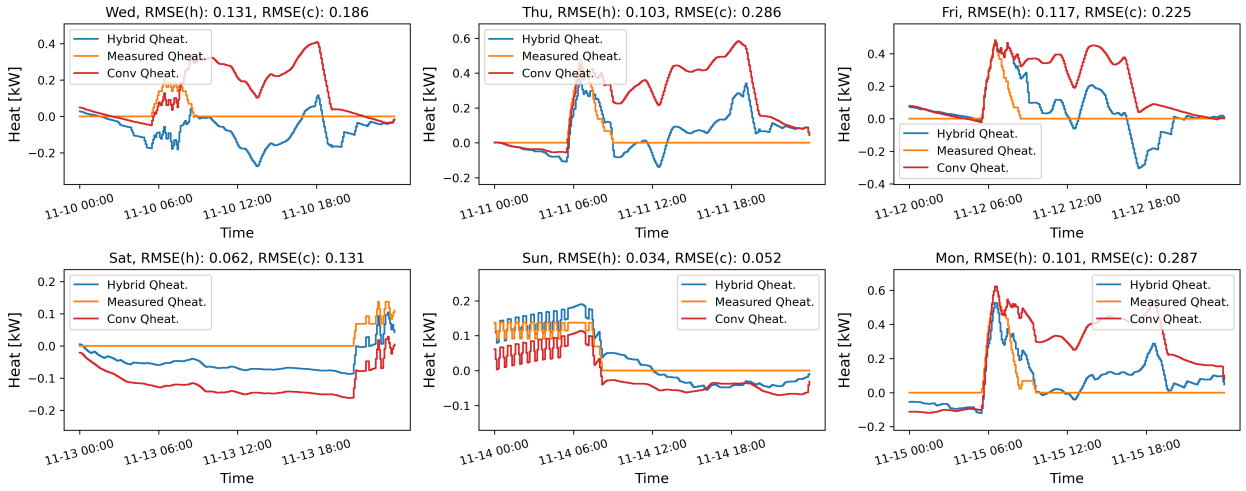


Figure 26: Amount of required heating rate of Hybrid and Conventional approaches for Chicago data

thermostat's internal control logic. Conditioned air is delivered through ceiling-mounted supply and return grilles via ducts connected to the HP-RTU. Although there is no dedicated exhaust fan, air is naturally exhausted through an exhaust grille connected to the ceiling plenum. Monitoring points are indicated in red text in Fig. 27(b).

The occupied setpoint ranged from 70°F (21.1°C) for heating to 74°F (23.3°C) for cooling during daytime hours (6:00–18:00 Pacific Time), while the unoccupied setpoint was 60°F (15.6°C) for heating and 80°F (26.7°C) for cooling during nighttime hours (18:00–6:00 Pacific Time). The HVAC system operated on a fixed weekday schedule (Monday–Friday, with weekends following weekday settings), and climate conditions were typical for Berkeley, CA. The supply fan operated at a fixed speed of 950 CFM (1614 CMH) (85% fan speed), with a minimum ventilation air supply of 155 CFM (263 CMH) set by a fixed outdoor air damper position. The heating supply air temperature was controlled by the thermostat, with a maximum limit of 100°F (37.8°C) to prevent hot air short-circuiting. The air-side economizer mode was disabled and kept at its minimum position.

Occupancy loads varied throughout the day: fully occupied periods (100%, 600 W) occurred from 08:00–12:00 and 13:00–17:00, while partially occupied periods (50%, 300 W) occurred from 07:00–08:00, 12:00–13:00, and 17:00–18:00. Lighting levels were adjusted accordingly, with full lighting (100%, 350 W) from 07:00–16:00 and partial lighting (50%, 175 W) from 06:00–07:00 and 16:00–20:00. A typical daily operational profile of the HP-RTU in the experimental cell is shown in Fig. 28.

After applying setpoint perturbations (Section 2.2), system identification was performed using the OD approach described in Section 2.2.3. Subsequently, one month of data was used to train the unmeasured disturbance model. The final model, `DEEP_AR_LSTM_FORECAST-case01`, selected in Section 4.2, was trained with a maximum of 1000 epochs and an early stopping patience of 150 epochs. Training for each feature set was repeated 10 times, and the configuration yielding the lowest mean squared error on one week of test data was selected.

Figure 29 compares the prediction performance of the Hybrid approach against measured data and the Conventional approach. The data is visualized in 6-hour intervals to assess prediction accuracy at different

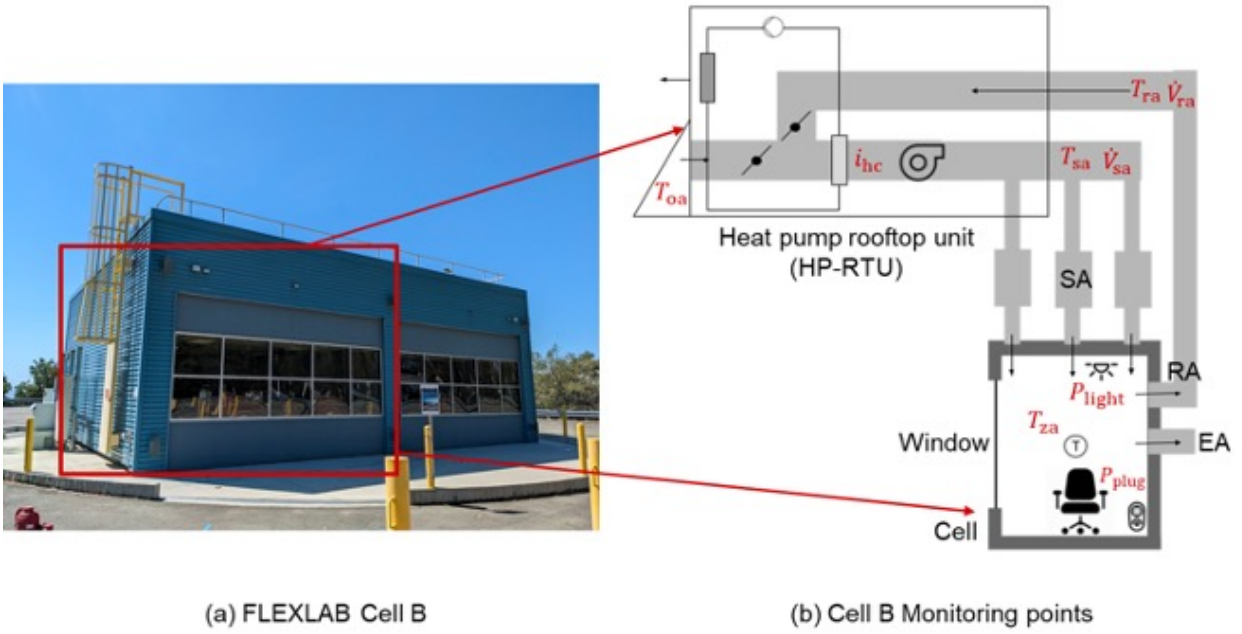


Figure 27: FLEXLAB Cell B and its mechanical schematic diagram

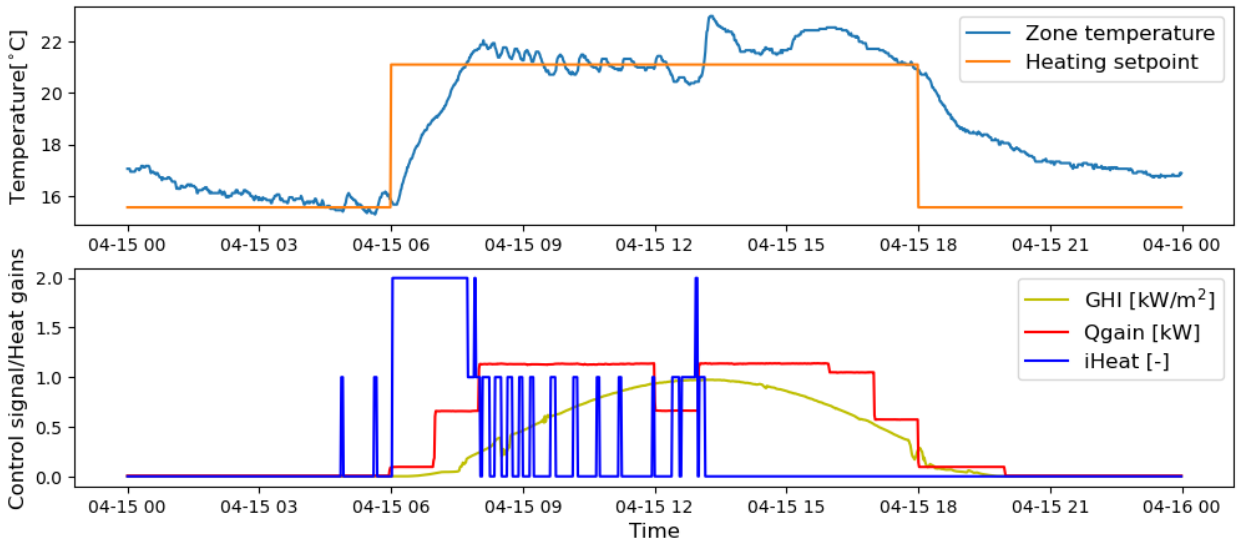


Figure 28: A daily operational profile of the HP-RTU in the experimental cell

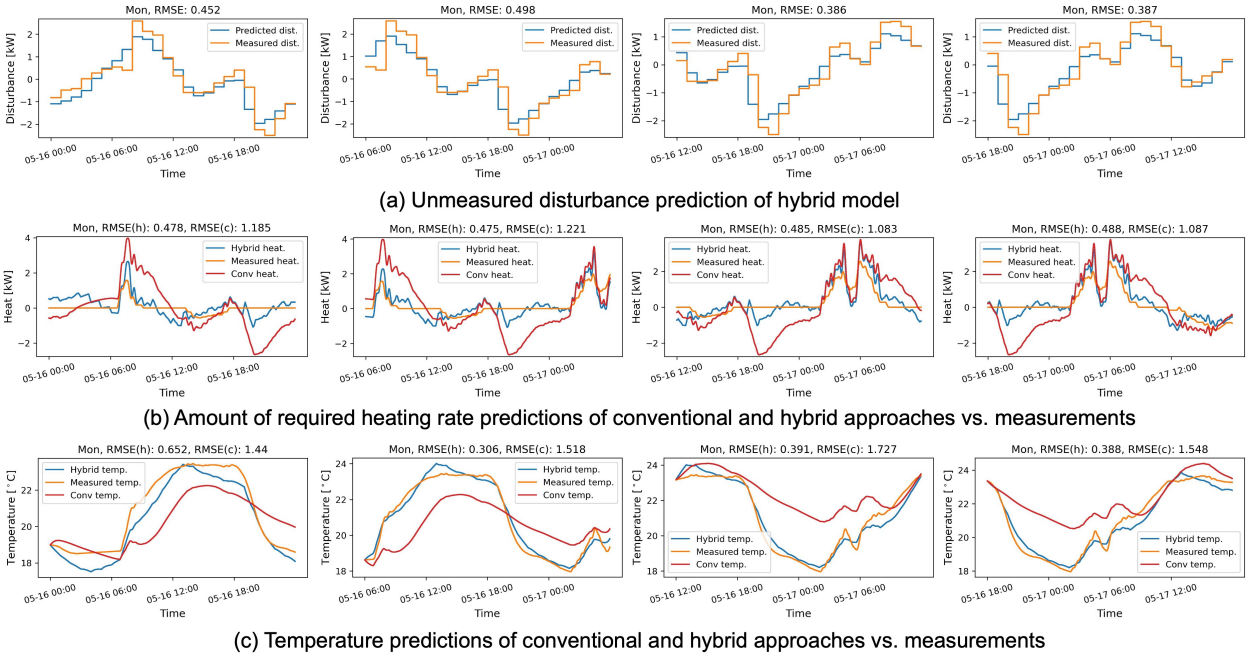


Figure 29: Prediction performance of Hybrid approach vs. measured data and Conventional approach

times of day. Results show that the Hybrid model effectively predicts future unmeasured disturbances, achieving an average RMSE reduction of 1.3°C in temperature prediction. In contrast, the absence of unmeasured disturbance prediction in the Conventional approach leads to increased inaccuracies in both temperature prediction and required heating rate, which can negatively affect the performance of predictive control applications.

5. Conclusion and limitations

In this paper, we present a Hybrid modeling approach that enhances the long-term temperature and load prediction capabilities of a gray-box model. While gray-box models are widely accepted for predictive applications such as MPC for energy efficiency and decarbonization, their system identification becomes challenging under unmeasured disturbances such as occupancy, lighting, appliances, and infiltration/exfiltration loads. Since directly measuring these loads is costly and often impractical in real buildings, we propose a neural network–based model for unmeasured disturbance loads to address the inherent limitations of the gray-box approach.

After comparing the performance and limitations of various system identification methods developed to handle unmeasured disturbances, we introduce the Hybrid modeling approach. To address the overfitting risk of neural network–based models, we develop a structured design and model selection process incorporating statistical tests. Using a calibrated building model, we generate two realistic datasets from different climates and use them to develop and evaluate the Hybrid models.

The Hybrid approach achieves RMSE reductions of approximately 0.2–0.9°C and 0.3–2°C for one-day-ahead temperature predictions in mild (Berkeley, CA) and cold (Chicago, IL) climates, respectively. Furthermore, when applied to experimental data from an office-setting laboratory building, the Hybrid approach reduces RMSE by an average of 1.3°C compared to the Conventional approach.

Despite these improvements, non-negligible prediction errors remain due to the stochastic nature of unmeasured disturbances, making the development of a perfect model impossible. In addition, when the gray-box model is inaccurate, the Hybrid model structure can be flawed if future control inputs are not incorporated, as illustrated in Fig. 12. While it is possible to include future control inputs in the Hybrid model, doing so introduces nonlinearity, which may impose computational burdens or degrade performance in predictive applications such as MPC.

Moreover, true unmeasured disturbances may vary over time due to changes in building operations or occupancy profiles. Although real-time modeling with sequential updates [40] can address this issue, it remains advisable to recalibrate the model regularly or after significant operational changes (e.g., seasonal transitions or extended building closures).

Overall, the Hybrid model offers superior predictive accuracy compared to the Conventional approach, particularly in compensating for inaccuracies in gray-box models affected by unmeasured disturbances. Since MPC makes optimal decisions at each control step, capturing the overall future thermal behavior of building operations is essential. Therefore, the Hybrid modeling approach can play a crucial role in enhancing energy efficiency and supporting decarbonization efforts by enabling long-term prediction applications such as load shifting and renewable energy integration.

6. Acknowledgements

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, by California Energy Commission through grant EPC-19-013, and by Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20212020800120).

References

[1] R. De Coninck, L. Helsen, Practical implementation and evaluation of model predictive control for an office building in brussels, Energy and Buildings 111 (2016) 290–298. doi:10.1016/j.enbuild.2015.11.014.

[2] D. Kim, J. E. Braun, Model predictive control for supervising multiple rooftop unit economizers to fully leverage free cooling energy resource, Applied energy 275 (2020) 115324. doi:10.1016/j.apenergy.2020.115324.

[3] D. Blum, Z. Wang, C. Weyandt, D. Kim, M. Wetter, T. Hong, M. A. Piette, Field demonstration and implementation analysis of model predictive control in an office HVAC system, Applied energy 318 (2022) 119104. doi:10.1016/j.apenergy.2022.119104.

[4] D. Kim, Z. Wang, J. Brugger, D. Blum, M. Wetter, T. Hong, M. A. Piette, Site demonstration and performance evaluation of MPC for a large chiller plant with TES for renewable energy integration and grid decarbonization, Applied energy 321 (2022) 119343. doi:10.1016/j.apenergy.2022.119343.

[5] J. Braun, N. Chaturvedi, An inverse Gray-Box model for transient building load prediction, HVAC&R Research 8 (1) (2002) 73–99. doi:10.1080/10789669.2002.10391290.

[6] S. Li, J. Joe, J. Hu, P. Karava, System identification and model-predictive control of office buildings with integrated photovoltaic-thermal collectors, radiant floor heating and active thermal storage, Solar Energy 113 (2015) 139–157. doi:10.1016/j.solener.2014.11.024.

[7] D. Kim, J. Cai, K. B. Ariyur, J. E. Braun, System identification for building thermal systems under the presence of unmeasured disturbances in closed loop operation: Lumped disturbance modeling approach, Building and environment 107 (2016) 169–180. doi:10.1016/j.buildenv.2016.07.007.

[8] D. Kim, J. Cai, J. E. Braun, K. B. Ariyur, System identification for building thermal systems under the presence of unmeasured disturbances in closed loop operation: Theoretical analysis and application, Energy and Buildings 167 (2018) 359–369. doi:10.1016/j.enbuild.2017.12.007.

[9] J. Joe, P. Karava, Agent-based system identification for control-oriented building models, Journal of Building Performance Simulation 10 (2) (2017) 183–204. doi:10.1080/19401493.2016.1212272.

[10] L. Ljung, System identification, in: A. Procházka, J. Uhlíř, P. W. J. Rayner, N. G. Kingsbury (Eds.), Signal Analysis and Prediction, Birkhäuser Boston, Boston, MA, 1998, pp. 163–173. doi:10.1007/978-1-4612-1768-8_11.

[11] A. R. Coffman, P. Barooah, Simultaneous identification of dynamic model and occupant-induced disturbance for commercial buildings, Building and environment 128 (2018) 153–160. doi:10.1016/j.buildenv.2017.10.020.

[12] T. Zeng, J. Brooks, P. Barooah, Simultaneous identification of linear building dynamic model and disturbance using sparsity-promoting optimization, Automatica: the journal of IFAC, the International Federation of Automatic Control 129 (2021) 109631. doi:10.1016/j.automatica.2021.109631.

[13] M. J. Ellis, Machine learning enhanced Grey-Box modeling for building thermal modeling, in: 2021 American Control Conference (ACC), IEEE, New Orleans, LA, USA, 2021, pp. 3927–3932. doi:10.23919/ACC50511.2021.9482715.

[14] P. Kumar, J. B. Rawlings, M. J. Wenzel, M. J. Risbeck, Grey-box model and neural network disturbance predictor identification for economic MPC in building energy systems, Energy and Buildings (2023) 112936doi:10.1016/j.enbuild.2023.112936.

[15] H. Madsen, J. M. Schultz, Short time determination of the heat dynamics of buildings, Technical University of Denmark, Department of Civil Engineering, 1993.

[16] ASHRAE, 2017 ASHRAE Handbook: Fundamentals, Chapter 18 SI: Nonresidential Cooling and Heating Load Calculations, Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc, 2017.

[17] E. J. H. Wilson, A. Parker, A. Fontanini, E. Present, J. L. Reyna, R. Adhikari, C. Bianchi, C. CaraDonna, M. Dahlhausen, J. Kim, A. LeBar, L. Liu, M. Praprost, L. Zhang, P. DeWitt, N. Merket, A. Speake, T. Hong, H. Li, N. Mims Frick, Z. Wang, A. Blair, H. Horsey, D. Roberts, K. Trenbath, O. Adekanye, E. Bonnema, R. El Kontar, J. Gonzalez, S. Horowitz, D. Jones, R. T. Muehleisen, S. Platthotam, M. Reynolds, J. Robertson, K. Sayers, Q. Li, End-Use load profiles for the U.S. building

stock: Methodology and results of model calibration, validation, and uncertainty quantification, Tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), United States (Mar. 2022). doi:10.2172/1854582.

[18] Z. O'Neill, S. Narayanan, R. Brahme, Model-based thermal load estimation in buildings, Proceedings of SimBuild 4 (1) (2010) 474–481.

[19] S. W. Ham, P. Karava, I. Bilionis, J. Braun, Real-time model for unit-level heating and cooling energy prediction in multi-family residential housing, Journal of Building Performance Simulation 14 (4) (2021) 420–445. doi:10.1080/19401493.2021.1968495.

[20] B. Dong, Z. Li, S. M. M. Rahman, R. Vega, A hybrid model approach for forecasting future residential electricity consumption, Energy and Buildings 117 (2016) 341–351. doi:10.1016/j.enbuild.2015.09.033.

[21] Z. E. Lee, K. M. Zhang, Scalable identification and control of residential heat pumps: A minimal hardware approach, Applied energy 286 (2021) 116544. doi:10.1016/j.apenergy.2021.116544.

[22] Lawrence Berkeley National Laboratory, FLEXLAB the world's most advanced integrated building and grid technologies testbed, https://flexlab.lbl.gov/, accessed: 2021-12-24 (2021).

[23] U.S. Department of Energy, EnergyPlus - weather data, https://energyplus.net/weather-location/north_and_central_america_wmo_region_4/USA/CA/USA_CA_Oakland.Intl.AP.724930_TMY3, accessed: 2021-12-24 (2021).

[24] S. W. Ham, D. Kim, T. Barham, K. Ramseyer, The first field application of a low-cost MPC for grid-interactive K-12 schools: Lessons-learned and savings assessment, Energy and Buildings 296 (2023) 113351. doi:10.1016/j.enbuild.2023.113351.

[25] S. Rouchier, M. J. Jiménez, S. Castaño, Sequential monte carlo for on-line parameter estimation of a lumped building energy model, Energy and Buildings 187 (2019) 86–94. doi:10.1016/j.enbuild.2019.01.045.

[26] P. Radecki, B. Hencey, Online model estimation for predictive thermal control of buildings, IEEE Transactions on Control Systems Technology 25 (4) (2017) 1414–1422. doi:10.1109/TCST.2016.2587737.

[27] T. Hong, D. Macumber, H. Li, K. Fleming, Z. Wang, Generation and representation of synthetic smart meter data, Building Simulation 13 (6) (2020) 1205–1220. doi:10.1007/s12273-020-0661-y.

[28] K. P. Murphy, Probabilistic Machine Learning: An introduction, MIT Press, 2022.

[29] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: a review, Data mining and knowledge discovery 33 (4) (2019) 917–963. doi:10.1007/s10618-019-00619-1.

[30] P. Lara-Benítez, M. Carranza-García, J. C. Riquelme, An experimental review on deep learning architectures for time series forecasting, International journal of neural systems 31 (03) (2021) 2130001. doi:10.1142/S0129065721300011.

[31] I. Yazici, O. F. Beyca, D. Delen, Deep-learning-based short-term electricity load forecasting: A real case application, Engineering applications of artificial intelligence 109 (2022) 104645. doi:10.1016/j.engappai.2021.104645.

[32] A. Nielsen, Practical Time Series Analysis: Prediction with Statistics and Machine Learning, "O'Reilly Media, Inc.", 2019.

[33] X. Hu, L. Chu, J. Pei, W. Liu, J. Bian, Model complexity of deep learning: a survey, Knowledge and information systems 63 (10) (2021) 2585–2619. doi:10.1007/s10115-021-01605-0.

[34] PyTorch Contributors, RNN, https://pytorch.org/docs/stable/generated/torch.nn.RNN.html, accessed: 2023-4-3 (2023).

[35] PyTorch Contributors, LSTM, https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html, accessed: 2023-4-3 (2023).

[36] T. Hastie, J. Friedman, R. Tibshirani, The Elements of Statistical Learning, Springer New York, 2009. doi:10.1007/978-0-387-21606-5.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, High-Performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.

[38] A. Gelman, Bayesian data analysis, third edition Edition, Chapman & Hall/CRC texts in statistical science, CRC Press, Boca Raton, 2014.

[39] S. W. Ham, D. Kim, A. Casillas, L. Paul, A. Prakash, M. Pritoni, R. Brown, Development and experimental demonstration of hybrid model-based predictive control approach: How can we predict and control future load without measuring heat gains?, Submitted to xx (2024).

[40] S. W. Ham, P. Karava, I. Bilionis, J. Braun, A scalable and practical method for disaggregating heating and cooling electrical usage using smart thermostat and smart metre data, Journal of Building Performance Simulation 15 (2) (2022) 251–267. doi:10.1080/19401493.2022.2032352.

Appendix A. Input features by model cases

The following input features are assigned for model cases. Table A.2 shows the input features of MLP and CNN models, and those of RNN and LSTM are presented in Table A.3

case	pattern features	past w	future w
case01	1 day	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$	how,T_{oa},$q_{\mathrm{sol,surface}}$
case02	4 days	ζ_{ID}	T_{oa},$q_{\mathrm{sol,win}}$
case03	4 days	ζ_{ID},how	how,T_{oa},$q_{\mathrm{sol,win}}$
case04	4 days	ζ_{ID},weekday	weekday,T_{oa},$q_{\mathrm{sol,win}}$
case05	4 days	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$	how,T_{oa},$q_{\mathrm{sol,win}}$
case06	4 days	ζ_{ID},weekday,T_{oa},$q_{\mathrm{sol,win}}$	weekday,T_{oa},$q_{\mathrm{sol,win}}$
case07	7 days	ζ_{ID}	T_{oa},$q_{\mathrm{sol,win}}$
case08	7 days	ζ_{ID},how	how,T_{oa},$q_{\mathrm{sol,win}}$
case09	7 days	ζ_{ID},weekday	weekday,T_{oa},$q_{\mathrm{sol,win}}$
case10	7 days	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$	how,T_{oa},$q_{\mathrm{sol,win}}$
case11	7 days	ζ_{ID},weekday,T_{oa},$q_{\mathrm{sol,win}}$	weekday,T_{oa},$q_{\mathrm{sol,win}}$
case12	4 days	ζ_{ID},dow,T_{oa},$q_{\mathrm{sol,win}}$	dow,T_{oa},$q_{\mathrm{sol,win}}$
case13	7 days	ζ_{ID},dow,T_{oa},$q_{\mathrm{sol,win}}$	dow,T_{oa},$q_{\mathrm{sol,win}}$
case14	4 days	ζ_{ID},T_{oa},$q_{\mathrm{sol,win}}$	T_{oa},$q_{\mathrm{sol,win}}$
case15	7 days	ζ_{ID},T_{oa},$q_{\mathrm{sol,win}}$	T_{oa},$q_{\mathrm{sol,win}}$
case16	4 days	ζ_{ID},weekday,T_{oa},$q_{\mathrm{sol,win}}$,i_{heat},i_{cool}	dow,T_{oa},$q_{\mathrm{sol,win}}$
case17	7 days	ζ_{ID},weekday,T_{oa},$q_{\mathrm{sol,win}}$,i_{heat},i_{cool}	dow,T_{oa},$q_{\mathrm{sol,win}}$
case18	4 days	ζ_{ID},dow,T_{oa},$q_{\mathrm{sol,win}}$,i_{heat},i_{cool}	dow,T_{oa},$q_{\mathrm{sol,win}}$
case19	7 days	ζ_{ID},dow,T_{oa},$q_{\mathrm{sol,win}}$,i_{heat},i_{cool}	dow,T_{oa},$q_{\mathrm{sol,win}}$
case20	4 days	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$,i_{heat},i_{cool}	dow,T_{oa},$q_{\mathrm{sol,win}}$
case21	7 days	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$,i_{heat},i_{cool}	dow,T_{oa},$q_{\mathrm{sol,win}}$

Table A.2: Input features of MLP and CNN models for model cases

case	pattern feature	past w	future w
case01	1 day	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$	how,T_{oa},$q_{\mathrm{sol,win}}$
case02	1 day	ζ_{ID},how	how
case03	1 day	ζ_{ID},T_{oa},$q_{\mathrm{sol,win}}$	T_{oa},$q_{\mathrm{sol,win}}$
case04	1 day	ζ_{ID},hod,T_{oa},$q_{\mathrm{sol,win}}$	hod,T_{oa},$q_{\mathrm{sol,win}}$
case05	1 day	ζ_{ID},dow,T_{oa},$q_{\mathrm{sol,win}}$	dow,T_{oa},$q_{\mathrm{sol,win}}$
case06	1 day	ζ_{ID},weekday,T_{oa},$q_{\mathrm{sol,win}}$	weekday,T_{oa},$q_{\mathrm{sol,win}}$
case07	2 days	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$	how,T_{oa},$q_{\mathrm{sol,win}}$
case08	4 days	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$	how,T_{oa},$q_{\mathrm{sol,win}}$
case09	7 days	ζ_{ID},how,T_{oa},$q_{\mathrm{sol,win}}$	how,T_{oa},$q_{\mathrm{sol,win}}$
case10	4 days	ζ_{ID},T_{oa},$q_{\mathrm{sol,win}}$	T_{oa},$q_{\mathrm{sol,win}}$
case11	7 days	ζ_{ID},T_{oa},$q_{\mathrm{sol,win}}$	T_{oa},$q_{\mathrm{sol,win}}$
case12	2 days	ζ_{ID},weekday,T_{oa},$q_{\mathrm{sol,win}}$	weekday,T_{oa},$q_{\mathrm{sol,win}}$
case13	4 days	ζ_{ID},weekday,T_{oa},$q_{\mathrm{sol,win}}$	weekday,T_{oa},$q_{\mathrm{sol,win}}$
case14	7 days	ζ_{ID},weekday,T_{oa},$q_{\mathrm{sol,win}}$	weekday,T_{oa},$q_{\mathrm{sol,win}}$
case15	2 days	ζ_{ID},dow,T_{oa},$q_{\mathrm{sol,win}}$	dow,T_{oa},$q_{\mathrm{sol,win}}$
case16	4 days	ζ_{ID},dow,T_{oa},$q_{\mathrm{sol,win}}$	dow,T_{oa},$q_{\mathrm{sol,win}}$
case17	7 days	ζ_{ID},dow,T_{oa},$q_{\mathrm{sol,win}}$	dow,T_{oa},$q_{\mathrm{sol,win}}$

Table A.3: Input features of RNN and LSTM models for model cases