Hybrid Modeling Framework for Systems with Unmeasured Time-Varying Disturbances: An Application to Buildings

By

Pranav Krishna

Thesis

Submitted in partial satisfaction of the requirements for the degree of

Master of Science

in

Chemical Engineering

in the

Office of Graduate Studies

of the

University of California

Davis

Approved:

2023

Acknowledgements

I would like to express my sincere gratitude to the members of my thesis committee, Dr. Matthew J. Ellis, Dr. Ahmet Palazoglu, and Dr. Nael El-Farra, for their guidance, encouragement, and support throughout my Master of Science degree journey in Chemical Engineering and the process of conducting my research. Their valuable feedback and insightful comments have helped me refine my ideas and produce a better-quality work.

I extend my heartfelt thanks to my parents, Mr. Krishna Kumar and Mrs. Swarnalatha Krishna, for their unconditional love, unwavering support, and sacrifices. Without them, this achievement would not have been possible. I also convey my special appreciation to my brother, Neerav Krishna, for his constant belief in me and his exceptional work ethic, which has inspired me to excel in my academic pursuits.

I am also thankful to all the professors, mentors, and colleagues who have contributed to my academic and personal growth, as well as the staff members of the Department of Chemical Engineering, especially Mr. Ryan Gorsiski, for providing me with the necessary resources and assistance throughout my graduate studies.

Lastly, I would like to express my gratitude to all my friends and family members who have been a constant source of inspiration and motivation during my academic journey. Your love and encouragement have kept me going, and I am forever grateful.

Thank you all for your support and encouragement.

Hybrid Modeling Framework for Systems with Unmeasured Time-varying Disturbances:

An Application to Buildings

Abstract

The energy consumed by the residential and commercial building sectors in the United States has been increasing at around 1.3% per year over the past decade, making efficient building operations more crucial than ever. Model predictive control (MPC), which is a model-based control method, has been proposed as a solution for the control and optimization of building operations due to its ability to optimize control actions based on constraints such as cost and energy. However, widespread adoption of MPC in buildings is limited by the challenges in developing and training a control-oriented building model. Building modeling is a challenging task due to the presence of unmeasured time-varying heat disturbances due to people, lighting, and electricity, and the lack of full state measurements, resulting in a coupled state, disturbance, and model parameter estimation problem.

Despite being unmeasured, these time-varying heat disturbances are correlated to certain time-features like the time of the day and the day of the week for several building types and occupancy patterns. Hence, hybrid models, which combine physics-based models, to capture the underlying dynamics of the system, and data-driven models, used to forecast the disturbances, have been proposed as a potential method for control-oriented modeling of buildings. In our previous work, a low-order thermal resistance-capacitance network was formulated to capture the dynamics of the building space and a feedforward neural network (FNN) was used to forecast the time-varying unmeasured disturbances.

This thesis presents a generalized hybrid modeling framework to identify models for systems that are subject to unmeasured time-varying disturbances. The proposed hybrid modeling framework combines a parameterized low-order physics-based model and a feedforward neural network (FNN) and utilizes a novel three-step training methodology to simultaneously estimate both the physics-based and FNN model parameters. The aim of the three-step training methodology is to provide better model predictions compared to the predictions made by the same model trained with alternative strategies. A model validation approach is also provided as part of the training methodology. The effectiveness of the proposed modeling and training approach is demonstrated by applying it to model the thermal dynamics of a building space. The time features, which provide the desired model predictions, are first determined. The superiority of the three-step training methodology is demonstrated by comparing the predictions generated by the models trained with alternative strategies to those generated by the model trained using the three-step training methodology. These results demonstrate that the hybrid modeling framework is suitable for modeling systems with unmeasured time-varying disturbances, and that the three-step training methodology results in models with minimal prediction errors, with fewer number of iterations as compared to its alternatives. The impact of unavailability of full state measurements is studied. Finally, the ability for the hybrid modeling framework to reproduce the results is evaluated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 2021, the combined end-use energy consumption of the residential and commercial building sectors in the United States accounted for about 21 quadrillion British thermal units [39]. This is responsible for about 39% of the end-use energy consumption for the calendar year, with the residential and commercial building sectors attributing to about 21% and 18%, respectively [39]. Globally, the building sector energy consumption increased on average by 1.3% per year, between 2010 and 2018 [23]. This increase in demand is driven by rapid urbanization, population growth, and increasing demand for energy services such as space heating, cooling, and lighting. With rising prices and demand for energy [4], the development of advanced modeling and control strategies for buildings is crucial.

Studies have indicated that the model predictive control (MPC) presents a promising solution for optimizing building operations [2, 13, 31], such as reducing building energy consumption cost while maintaining comfort, by using a model of the building space dynamics and accounting for external factors (e.g., time-varying prices) to determine the control actions that are optimized for a given objective [30, 34]. Researchers have shown that, by adding a supervisory MPC controller to the building HVAC systems, the energy consumption and operating cost can be reduced anywhere from 7% to more than 50% [18, 22]. Recently, a supervisory MPC was designed by [5] that takes into consideration both peak-load shaving and thermal comfort. The numerical results revealed that the optimal set-point temperature computed by the MPC reduced the total operating cost by 12.46% and the peak electricity demand by 13.43% compared to the nominal operation. In another study, an MPC was developed for a low-energy office building, resulting in 35% heating energy demand savings compared to using a rule-based controller ([17]).

Despite its advantages, MPC has not been widely adopted as a control and optimization approach for building operations, due to the challenges in building model identification [21]. Being a model-based control method, the performance of MPC relies heavily on the accuracy of the dynamic model. Building model identification is complex due to the presence of unmeasured time-varying heat disturbances, which evolve on similar timescales as the building dynamics. These disturbances result from changes in, for example, occupancy, solar radiation, and electrical equipment. Due to these disturbances, the building model identification is also coupled with estimating the unmeasured time-varying heat disturbances. Handling these unmeasured disturbances is one of the fundamental challenges in control-oriented modeling of buildings [9, 15, 24]. Since building model identification is a critical prerequisite for effective implementation of the MPC for building operations, overcoming the challenges of building model identification has become a major focus of research (refer, also, to the recent reviews [13, 27, 31] for a comprehensive review of building modeling for MPC).

Building spaces can be modeled in three ways: (1) physics-based models, (2) data-driven models, and (3) hybrid models [28]. Physics-based models are models that are derived from first principles such as laws of conservation of mass and energy. The primary advantage of using physics-based models is that they provide a fundamental understanding of the underlying physics of the system being studied. Physics-based models are further grouped into two categories. These are high-fidelity models and control-oriented models. High-fidelity models are formulated using detailed ordinary differential equations (ODEs) describing the dynamics of the system. For building modeling, building modeling software such as Trnsys [7] and EnergyPlus [38] are used to formulate high-fidelity models. The advantage of such models is that, provided the right parameters, they have the ability to capture the system dynamics accurately. However, high-fidelity physics-based models can be complex and computationally intensive as they require solving many coupled mathematical equations that describe the physical phenomena. They require detailed knowledge of the system parameters, for example, the materials of construction and thickness of walls, in the context of buildings. However, some of these parameters may be difficult to obtain in practice due to experimental limitations [28] or uncertainties. Due to their computational complexity and the large amount of engineering time required to develop, configure, and maintain these models, high-fidelity physics-based models are generally not used for the purpose of controls [27].

Control-oriented modeling, on the other hand, commonly adopts a low-order model or a reduced-order model as a means of representing the system dynamics. In the context of buildings, a common approach to model the system dynamics is by using a low-order resistance-capacitance (RC) network [28]. In the RC method, capacitance represents the thermal capacitance (capacity to store heat), while resistance represents the thermal resistance (resist heat flow through the material). This modeling approach yields a reduced-order model of the building space which is suitable for building controls. The advantage of these control-oriented models are that they are computationally inexpensive, and are easy to design and configure, which are qualities required for a model to be used in a control system [27].

Data-driven models are models that are based on purely statistical methods and machine learning algorithms. These models can capture nonlinear relationships between the input and output variables, and do not rely on any prior knowledge of the system dynamics. Some examples of data-driven models include artificial neural networks, support vector machines, and decision trees. In the context of buildings, neural networks have been shown to provide accurate predictions of building energy. A comprehensive review of machine learning and deep learning method for building performance has been conducted in [37]. For example, in [36], an artificial neural network (ANN) was developed to forecast the annual energy consumption of a building based on readily available energy performance certificates (EPC) and specific user defined characteristics such as the length of the heating period. In another study [40], a framework for achieving optimal control over air handling units (AHUs) using deep reinforcement learning (DRL) was proposed. In this study, the authors used a long-short-term-memory (LSTM) network to approximate real-world HVAC operations, to build training environments for the DRL. A physics constrained deep learning methodology was proposed for modeling building thermal dynamics by encoding the physics-based knowledge into a recurrent neural network (RNN) architecture [12]. The resulting model was able to accurately predict the dynamics of the buildings for the next thirty days, given only ten days worth of data. In [20], model identification of a multi-zone building HVAC system, employing transfer learning was proposed. The model included a recurrent neural network (RNN) augmented with a subspace identification-based residual model. The RNN is pre-trained using a large amount of data generated from a representative simulated zone. However, since data-driven models purely rely on the data from the system, they generally have poor extrapolating capabilities, lack interpretability, require high amounts of data, and have unbounded uncertainty in predictions that may not satisfy physical constraints [8].

Hybrid models are formed by a combination of physics-based and data-driven models. The advantage of hybrid models is that they can provide insights into the underlying physics of the system while capturing the nonlinearities and disturbances of the system [6, 8]. Due to this reason, hybrid models have been proposed as a solution for control-oriented building modeling, and hybrid modeling frameworks have been developed that incorporate control-oriented physics-based and disturbance models [14, 25, 26]. For example, a hybrid modeling framework that uses a resistance-capacitance (RC) model, to predict the mean temperature of a residential two-story building, and a data-driven model, to predict the temperature difference between the two floors, was proposed in [10]. The results indicated that the hybrid model was able to accurately predict the building space dynamics with only three weeks of building operation data. In [22], a modeling framework was proposed, in which the thermal dynamics of the building are modeled by a linearized thermal RC model, and the uncertainties or disturbances associated with the HVAC process were handled separately by an inverse neural network model. These two models were coupled using a feedback linearization method. In [11], a semi-parametric regression was utilized to identify the system model parameters of a first-order thermal model, and regression trees were utilized to develop a disturbance forecasting model. In [26], a hybrid model was proposed that modeled the building thermal dynamics using a physics-based model and forecasted the unmeasured disturbances using a neural network. However, the neural network and physics-based models were trained separately, which could result in compounding prediction errors. In our previous study [25], a physics-based thermal resistance-capacitance (RC) network model, derived from first principles, and a feedforward neural network (FNN), used to forecast the unmeasured heat disturbances, were integrated to form a parameterized hybrid model. Our approach involved simultaneously training the parameters of both the physics-based and FNN models. The advantage of simultaneously training the hybrid model parameters is that this approach minimizes overall prediction errors and can potentially avoid compounding prediction errors resulting from training models separately.

Despite our previous work, training hybrid models is not without challenges. The training problem is non-convex, resulting in the possibility that the trained model parameters can depend on the initial parameter estimates provided to the solver. Hence, in this context, alternative methods for training the parameters of the hybrid models may be required, which can mitigate the influence of the initial parameter estimates provided to the solver. Additionally, the process of validating the model needs careful consideration. Generally, the model is trained using the training dataset and its ability to generalize to new data is assessed by making predictions on a validation dataset. Since, in many real-world systems, it is not feasible to obtain full-state measurements, providing initial conditions for model validation is a challenging task. Since the validation is conducted on a different dataset as compared to the one on which the model was trained on, the initial states have to be estimated, which adds a degree of uncertainty that compounds throughout the validation process.

This thesis presents a generalized hybrid modeling framework to identify models for systems that are subject to unmeasured time-varying disturbances. The proposed hybrid modeling framework combines a parameterized low-order physics-based model, to model the underlying dynamics of the system, and a feedforward neural network (FNN) to forecast the unmeasured time-varying disturbances. The hybrid modeling framework utilizes a novel three-step training methodology, proposed in the thesis, to simultaneously train both the physics-based model and the FNN model parameters. The aim of the three-step training methodology is to provide better model predictions compared to the predictions made by the same model trained with alternative strategies. A model validation approach is also provided as part of the training methodology, which addresses the unavailability of full-state measurements. The effectiveness of the proposed modeling and training approach is demonstrated by applying it to model the thermal dynamics of a building space. The time features which provide the desired model predictions are first determined. The superiority of the three-step training methodology is then demonstrated by comparing the predictions generated by the models trained with alternative strategies to those generated by the model trained using the three-step training methodology. The impact of unavailability of full state measurements is studied. Finally, the ability for the hybrid modeling framework to reproduce the results is evaluated. The thesis aims to demonstrate that the hybrid modeling framework is suitable for modeling systems with unmeasured time-varying disturbances and that the three-step training methodology results in models with minimal prediction errors, with fewer number of iterations as compared to its alternatives.

Chapter 2

Preliminaries

2.1 Notation

Time dependent vectors are denoted by $x_k \in \mathbb{R}^n$ where $k \in \mathbb{Z}$. The predicted variable x at time step k is denoted by $\tilde{x}_k$.

2.2 Class of Parameterized System Models

A class of system models that are derived from first principles i.e., mass, momentum, and energy balances are considered. These models are expressed as a system of coupled first-order nonlinear ordinary differential equations (ODEs) with unknown parameters that capture the dynamics of the physical system, given by:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), d(t); \theta_{sys}) \\ y(t) &= h(x(t); \theta_{sys}) \end{aligned} \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^{m_u}$ is the manipulated inputs and measured disturbances, $d(t) \in \mathbb{R}^{m_d}$ is the unknown time-varying unmeasured disturbances, $y(t) \in \mathbb{R}^l$ is the measured output vector, and $\theta_{sys} \in \mathbb{R}^{n_{\theta_{sys}}}$ is the parameter vector. The functions f and h are vector valued functions. For the sake of brevity, $u(t)$ are called the measurable inputs and $d(t)$ are called the unmeasured disturbances. An additional assumption is imposed on the types of time-varying disturbances ($d(t)$) considered in this work (Discussed in Section 3.1).

To estimate the parameter values (e.g., θ_{sys} in (2.1)), data is sampled from the system, generally with a fixed sample period. Hence, the model presented in (2.1) is discretized in time. This results in a system model of the form:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, d_k; \theta_{sys}) \\ y_k &= h(x_k; \theta_{sys}) \end{aligned} \tag{2.2}$$

where x_k, u_k, and y_k represent the state, measured input, and unmeasured disturbance, respectively, at time step k. With slight abuse of notation, the function f in (2.2) is the state transition map.

2.3 Feedforward Neural Networks

A feedforward neural network (FNN) is a type of artificial neural network that consists of multiple layers of interconnected nodes, called neurons [29]. A neuron consists of inputs multiplied by a set of weights and a bias is added. The result is then passed through an activation function, which produces an output. The input layer of the FNN receives input data to the FNN. The input has a specific shape that depends on the number of input features or attributes. The output layer of the FNN produces the final prediction, which can be in the form of a scalar, a vector, or a matrix depending on the nature of the target data. The intermediate layers are known as hidden layers. These are composed of neurons that perform computations based on the inputs they receive from the previous layer. If each neuron in a hidden layer is connected to all nodes in the previous and next layer, the network is called a dense network. Activation functions are used to introduce nonlinearity into the FNN model, enabling it to model complex relationships between inputs and outputs. Two commonly used activation functions include the rectified linear unit (ReLU) and the hyperbolic tangent (tanh). ReLU is given by:

$$f(z) = max(0, z)$$

where z is the input to the activation function and tanh is given by:

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

The parameters of the FNN correspond to the weights and biases of the neural network. These parameters are usually trained in a supervisory learning fashion, meaning the parameters are adjusted based on a set of labeled examples. The FNN model is represented by:

$$z_{out} = f_d(z_{in}; \theta_d) \tag{2.3}$$

where $\theta_d \in \mathbb{R}^{n_{\theta_d}}$ includes the weights and biases of all neurons in the network, $z_{in} \in \mathbb{R}^{n_{z_{in}}}$ is the predictor variable vector, $z_{out} \in \mathbb{R}^{n_{z_{out}}}$ is the output of the FNN, and f_d is the resulting mapping between the predictor variable z_{in} to the output z_{out}.

Chapter 3

Hybrid Modeling Framework and Training Methodology

In this chapter, the development of a generalized hybrid modeling framework for systems with unmeasured time-varying disturbances is detailed, which is an abstraction of the problem of modeling buildings with unmeasured time-varying heat gains. The three-step training methodology for training the hybrid model is described.

3.1 Hybrid Model Framework

In the proposed hybrid modeling framework, the underlying system dynamics are modeled using a physics-based model, and the unmeasured disturbances are modeled using a data-driven model. An additional assumption is imposed on the class of time-varying unmeasured disturbances to formulate the data-driven model. The unmeasured disturbances are assumed to be correlated (in time) to certain known or predictable factors (as explained in Section 4.3, this assumption is satisfied for building modeling). Examples of such factors include time features and weather conditions. With this assumption, a dense FNN is used to model and predict the unmeasured disturbances from these factors, called

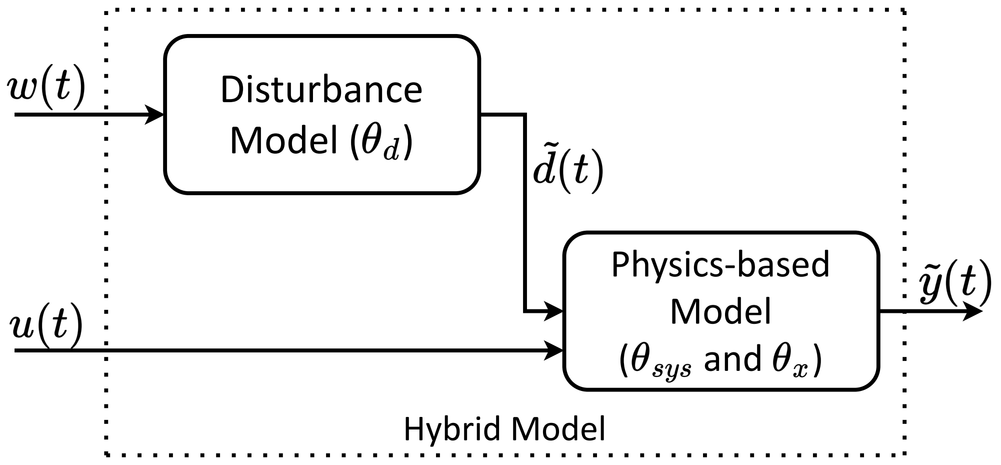

$w(t)$
Disturbance Model (θ_d)
$\tilde{d}(t)$
$u(t)$
Physics-based Model (θ_{sys} and θ_x)
$\tilde{y}(t)$
Hybrid Model

Figure 3.1: The architecture of the proposed hybrid modeling framework.

predictor variables. The FNN of the form (2.3) is used, given by:

$$\tilde{d}(t) = f_d(w(t); \theta_d) \tag{3.1}$$

where $w(t) \in \mathbb{R}^{m_w}$ is the predictor variable vector, $\tilde{d}(t)$ is the predicted unmeasured disturbance, and $\theta_d \in \mathbb{R}^{n_{\theta_d}}$ is a vector containing the weights and bias of the FNN.

With the physics-based model in (2.1) and the FNN in (3.1), the resulting hybrid model is given by:

$$\begin{aligned} \dot{\tilde{x}}(t) &= f(\tilde{x}(t), u(t), \tilde{d}(t); \theta_{sys}) \\ x(0) &= \theta_x \\ \tilde{d}(t) &= f_d(w(t); \theta_d) \\ \tilde{y}(t) &= h(\tilde{x}(t); \theta_{sys}) \end{aligned} \tag{3.2}$$

where $\tilde{x}(t) \in \mathbb{R}^n$ denotes the predicted state, $\tilde{y}(t) \in \mathbb{R}^l$ denotes the predicted output, and $\theta_x \in \mathbb{R}^{n_{\theta_x}}$ is the initial state parameter vector. The model architecture is depicted in Figure 3.1. The hybrid model receives two inputs: $u(t)$ and $w(t)$. Using $w(t)$, the data-driven model produces a prediction of the time-varying disturbances ($\tilde{d}(t)$). The physics-based model receives $u(t)$ and $\tilde{d}(t)$ as inputs. The physics-based model is initialized with the parameterized initial condition to compute a prediction of the output over time ($\tilde{y}(t)$), which corresponds to the output of the hybrid model.

Sampled data is used to train the hybrid model. Usually, this data is synchronously sampled. For training, the continuous-time model is converted to its discrete-time analog. The continuous-time model can be discretized in time by assuming constant inputs over sampling periods and applying a numerical integrator, giving the following discrete-time model:

$$\begin{aligned}
\tilde{x}_{k+1} &= f(\tilde{x}_k, u_k, \tilde{d}_k; \theta_{sys}) \\
\tilde{x}_0 &= \theta_x \\
\tilde{d}_k &= f_d(w_k, \theta_d) \\
\tilde{y}_k &= h(\tilde{x}_k, \theta_{sys})
\end{aligned} \tag{3.3}$$

While a discrete-time model is used for training, the underlying parameters are for the continuous-time model.

3.2 Training Problem

A training problem is formulated to train, i.e., estimate, the hybrid model parameters for a specific system using a dataset collected from the system. The training dataset consists of synchronously sampled input-output pairs where an excitation signal is used for the manipulated inputs, exciting the system dynamics. An additional dataset is collected for model validation. The training problem is formulated as a prediction error method (PEM), known as simulation PEM. Simulation PEM involves using the model and the sequence of input values from the training dataset to make a single prediction from the start of the training dataset until the end of the dataset. The prediction error at time step k is defined as:

$$\epsilon_k := y_k - \tilde{y}_k$$

where $\epsilon_k \in \mathbb{R}^l$ denotes the prediction error, $y_k \in \mathbb{R}^{n_l}$ is the measured output at time step k, and $\tilde{y}_k$ is the predicted output at the time step k. The loss function of the simulation PEM training problem is defined as:

$$V(\theta, Z^{N_{train}}) := \frac{1}{N_{train}} \sum_{k=0}^{N_{train}-1} \ell(\epsilon_k) + R(\theta) \tag{3.4}$$

where $\theta := \begin{bmatrix} \theta_{sys}^T & \theta_x^T & \theta_d^T \end{bmatrix}^T \in \mathbb{R}^{n_{\theta_{sys}}+n+n_{\theta_d}}$ is the parameter vector, which includes all parameters, $Z^{N_{train}} := \{(u_k, y_k)\}_{k=0}^{N_{train}}$ denotes the training dataset, $Z^{N_{validation}} := \{(u_k, y_k)\}_{k=0}^{N_{validation}}$ denotes the validation dataset, ℓ is a scalar-valued positive definite function that penalizes the prediction errors in the loss function, and $R(\theta)$ is the regularization term. Since the loss function uses an N_{train}-step output prediction, N_{train} is called the prediction horizon of the training problem. The $R(\theta)$ is a regularization term that helps to prevent overfitting (e.g., [32]). Standard regularization techniques include Lasso and ridge regression. With Lasso regularization, the sum of the absolute values of the parameters is used, i.e., $R(\theta) = \|\theta\|_1$. With ridge regularization, the sum of the squares of the parameters is used, i.e., $R(\theta) = \|\theta\|_2$.

With the loss function in (3.4) and hybrid model in (3.3), the training problem, i.e., parameter estimation problem, is given by:

$$\begin{aligned} \min_{\theta} \quad & V(\theta, Z^{N_{train}}) \\ \text{s.t.} \quad & \tilde{x}_{k+1} = f(\tilde{x}_k, u_k, \tilde{d}_k; \theta_{sys}) \\ & \tilde{x}_0 = \theta_x \\ & \tilde{d}_k = f_d(w_k; \theta_d) \\ & \tilde{y}_k = h(\tilde{x}_k) \\ & \epsilon_k = y_k - \tilde{y}_k, k \in \{0, \ldots, N_{train} - 1\} \end{aligned} \tag{3.5}$$

The resulting solution of (3.5) is denoted by θ^*. Simulation PEM allows for simultaneously training both the physics-based and FNN model parameters. The potential advantage of simultaneously training the hybrid model parameters is that this approach minimizes overall prediction errors and can potentially avoid compounding errors if the parameters are instead separately trained.

3.3 Three-step Training Methodology

Although the training problem in (3.5) is non-convex (even for a hybrid model with a linear physics-based model and a linear FNN), several solution methods can be applied to solve the training problem in (3.5). Models with neural networks are typically trained using first-order or pseudo-first-order methods, such as gradient descent or stochastic gradient descent because they can handle large datasets. However, the size of the training dataset in control applications is often limited due to practical constraints. In these cases, second-order or quasi-Newton methods can be considered. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is a quasi-Newton method that estimates the inverse of the Hessian matrix, i.e., can estimate the local curvature of the objective function, which can prevent the solver from getting stuck in narrow valleys or saddle points [16, 33]. It may be a suitable method for training models in control applications with smaller datasets.

Nonetheless, solving the training problem to obtain accurate model parameter values is challenging in practice, as demonstrated in Section 4.4. The non-convexity of the problem may result in the solver converging on a local minimum and ultimately, the parameter estimates obtained are dependent on the initial guess provided to the solver. Training multiple models with different initial guesses for the parameter values is one approach to overcome this challenge, but it increases computational time. Additionally, the inclusion of the initial state parameter (θ_x) in the training problem, seems to exacerbate the convergence issues based on extensive computational studies.

To address these challenges, a three-step training methodology to solve the training problem in (3.5) is proposed. This heuristic method leverages the strengths of both gradient descent and BFGS to obtain better parameter estimates with fewer iterations than using a single solver. The methodology consists of two gradient descent steps and one BFGS step. To obtain suitable initial guesses of θ_{sys} and θ_d, the initial state (θ_x) is fixed in the first gradient descent step. This first step is motivated by the observation that including the initial state parameter in the training problem can lead to convergence issues. Also, the effect of the initial condition will dissipate over the prediction horizon of the training problem. While BFGS can obtain better parameter estimates than gradient descent, it may be sensitive to the initial model parameter estimates, which can affect the solver's convergence and the quality of the final solution. Therefore, a second gradient descent is used where the initial condition θ_x is included in the training process to obtain a suitable initial guess of all parameters for training using BFGS.

To describe the three-step training methodology, the full parameter vector for the training step j at iteration i is denoted by: $\theta_{j,i}^T := [\theta_{sys,j,i}\ \theta_{x,j,i}\ \theta_{d,j,i}]$. Each training step will execute M_j iterations. The number of iterations that each step executes is a hyperparameter of the three-step training methodology. Prior to the first step, an initial guess for the parameter vector must be provided. To achieve this, an initialization method needs to be specified. The system parameters (θ_{sys}) can be initialized using domain knowledge or heuristics, since they are physics-based parameters. An example initialization strategy of the system parameters may involve specifying an approximate range of expected parameters and randomly selecting a parameter value as an initial guess from this range. The initialization of the initial condition (state) parameter may take the form of $\theta_{x,1,0} = f_{x_0}(y_0)$ where y_0 is the output measurement at time step $k = 0$. Some examples include $\theta_{x,1,0} = C^{-1}y_0$ for the case $y_k = Cx_k$ and C is an invertible matrix and $\theta_{x,1,0} = Hy_0$ where H is a specified matrix (this latter approach is used in Section 4.3). The FNN model parameters (θ_d) can be initialized randomly based on a distribution.

In the first step of the three-step training methodology, gradient descent is used to solve the training problem, using the initialized parameters. The parameter θ_x is fixed to the initial guess provided to it, i.e., does not get adjusted or trained. Hence, θ_x will retain the value $\theta_{x,0,0}$ upon completion of the first step. However, θ_{sys} and θ_d are trainable, i.e., their values are adjusted by the gradient descent algorithm to optimize the training problem. The number of iterations that the first training step will execute is M_1. The resulting parameter estimates after this step are as follows:

$$\theta_{1,M_1}^T := [\theta_{sys,1,M_1}\ \theta_{x,1,0}\ \theta_{d,1,M_1}]$$

In the second step, gradient descent is applied again, but all model parameters are trained. The trained parameter values from the first step are provided as the initial guesses in the second step, given as:

$$\theta_{2,0}^T := [\theta_{sys,1,M_1}\ \theta_{x,1,0}\ \theta_{d,1,M_1}]$$

The total number of iterations that the second step executes is M_2. The resulting parameter estimates after this step are given as:

$$\theta_{2,M_2}^T := [\theta_{sys,2,M_2}\ \theta_{x,2,M_2}\ \theta_{d,2,M_2}]$$

Finally, BFGS is used in the third step and all model parameters are trained. The parameters are initialized with the trained parameter values resulting from the second step, and the initial guess used in this step is given as:

$$\theta_{3,0}^T := [\theta_{sys,2,M_2}\ \theta_{x,2,M_2}\ \theta_{d,2,M_2}]$$

The total number of iterations that the third step executes is M_3. The resulting model parameter values are the parameter estimates returned by the three-step methodology and are denoted by:

$$\theta_{3,M_3} := [\theta_{sys,3,M_3}\ \theta_{x,3,M_3}\ \theta_{d,3,M_3}]$$

Often, one dataset is collected and divided into three sets: (1) training, (2) validation, and (3) testing datasets (e.g., [35]). Generally, the model is trained using the training dataset and its ability to generalize to new data is assessed by making predictions on the validation and testing datasets. However, predicting the output from the inputs in the testing and validation datasets is complicated by the fact that the initial state is

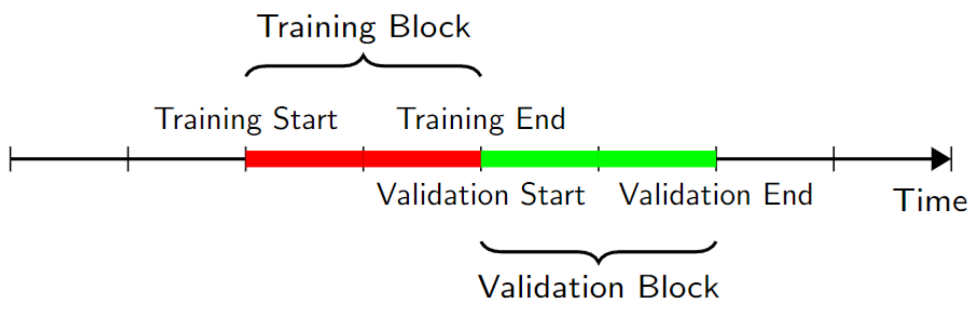


Figure 3.2: A diagram representing the sequential dataset used for training and validation.

not known. To address this, the dataset is assumed to consist of sequential data and a representation of the two datasets over time is provided in Figure 3.2. In this study, the same dataset is used for both model testing and validation. Thus, the validation dataset is immediately after the training dataset, such that the last data point in the training dataset and the first data point in the validation (or testing) dataset are the same. Since the data is sequential, the initial conditions for the testing and validation process can be generated from the trained model parameter values, obtained from the three-step training methodology. That is, the trained model uses the training dataset to generate a prediction of the states ($\tilde{x}_{N_{train}}$), which serves as the initial state for the testing/validation predictions. The outputs of the system over the validation dataset are predicted using the model in (3.3), the initial state estimates, and the inputs collected over the validation dataset.

Model validation criteria are defined. The criteria are dependent on the application, but can be based on a prediction accuracy metric (e.g., the mean squared error of the output predictions over the testing or validation dataset). Visual inspection of output predictions from the resulting models are also commonly used in practice. Since the three-step training methodology is a heuristic training strategy, the methodology may not return a suitable parameter estimate. If the model does not satisfy the criteria, the model is retrained by re-initializing the parameters with new initial guesses. A flow diagram of the methodology is shown in Figure 3.3 and is summarized in Algorithm 1.

Algorithm 1 Three-Step Training Methodology

```
1. Input: Initial parameter guesses ($\theta_{1,0}$), the number of iterations ($M_1$, $M_2$, and $M_3$), loss function, and validation criteria.
2. Output: Trained parameters $\theta_{3,M_3}$.
3. Step 1: Initialize parameters: $\theta_{1,0}^T = [\theta_{sys,1,0}\ \theta_{x,1,0}\ \theta_{d,1,0}]$. Train parameters $\theta_{sys,1,0}$ and $\theta_{d,1,0}$ using gradient descent starting from $\theta_{sys,1,0}$ and $\theta_{d,1,0}$ as the initial guesses. The initial condition parameter is fixed to $\theta_{x,1,0}$.
    (a) For $M_1$ iterations:
        i. Compute the gradient of the loss function $V(\theta_1)$ with respect to $\theta_1$ ($\theta_{x,1,0}$ fixed).
        ii. Update $\theta_{1,i+1}$ using gradient descent.
    (b) Trained parameter values at the end of Step 1: $\theta_{1,M_1}^T := [\theta_{sys,1,M_1}\ \theta_{x,1,0}\ \theta_{d,1,M_1}]$
4. Step 2: Initialize parameters with parameters from Step 1: $\theta_{2,0} = \theta_{1,M_1}$. Train parameter $\theta_2$ using gradient descent.
    (a) For $M_2$ iterations:
        i. Compute the gradient of the loss function $V(\theta_2)$ with respect to $\theta_2$.
        ii. Update $\theta_{2,i+1}$ using gradient descent.
    (b) Trained parameter values at the end of Step 2: $\theta_{2,M_2}^T := [\theta_{sys,2,M_2}\ \theta_{x,2,M_2}\ \theta_{d,2,M_2}]$
5. Step 3: Initialize parameters with parameters from Step 2: $\theta_{3,0} = \theta_{2,M_2}$. Train parameter $\theta_3$ using BFGS.
    (a) For $M_3$ iterations:
        i. Compute the hessian matrix of the loss function $V(\theta_3)$ with respect to $\theta_3$.
        ii. Update $\theta_{3,i+1}$ using BFGS.
    (b) Trained parameter values at the end of Step 3: $\theta_{3,M_3}^T := [\theta_{sys,3,M_3}\ \theta_{x,3,M_3}\ \theta_{d,3,M_3}]$
6. Validation: Trained parameters $\theta_{3,M_3}$ are provided as parameter values for model validation.
    (a) If the model predictions satisfy validation criteria, the trained parameter value are returned.
    (b) Else, the training process is repeated from Step 1 with a new set of random initial guesses.
```

Remark 1. *Although the proposed three-step training methodology uses a combination of gradient descent and BFGS, other solvers can be used (e.g., the Adam optimizer instead of gradient descent or the limited memory BFGS instead of BFGS).*

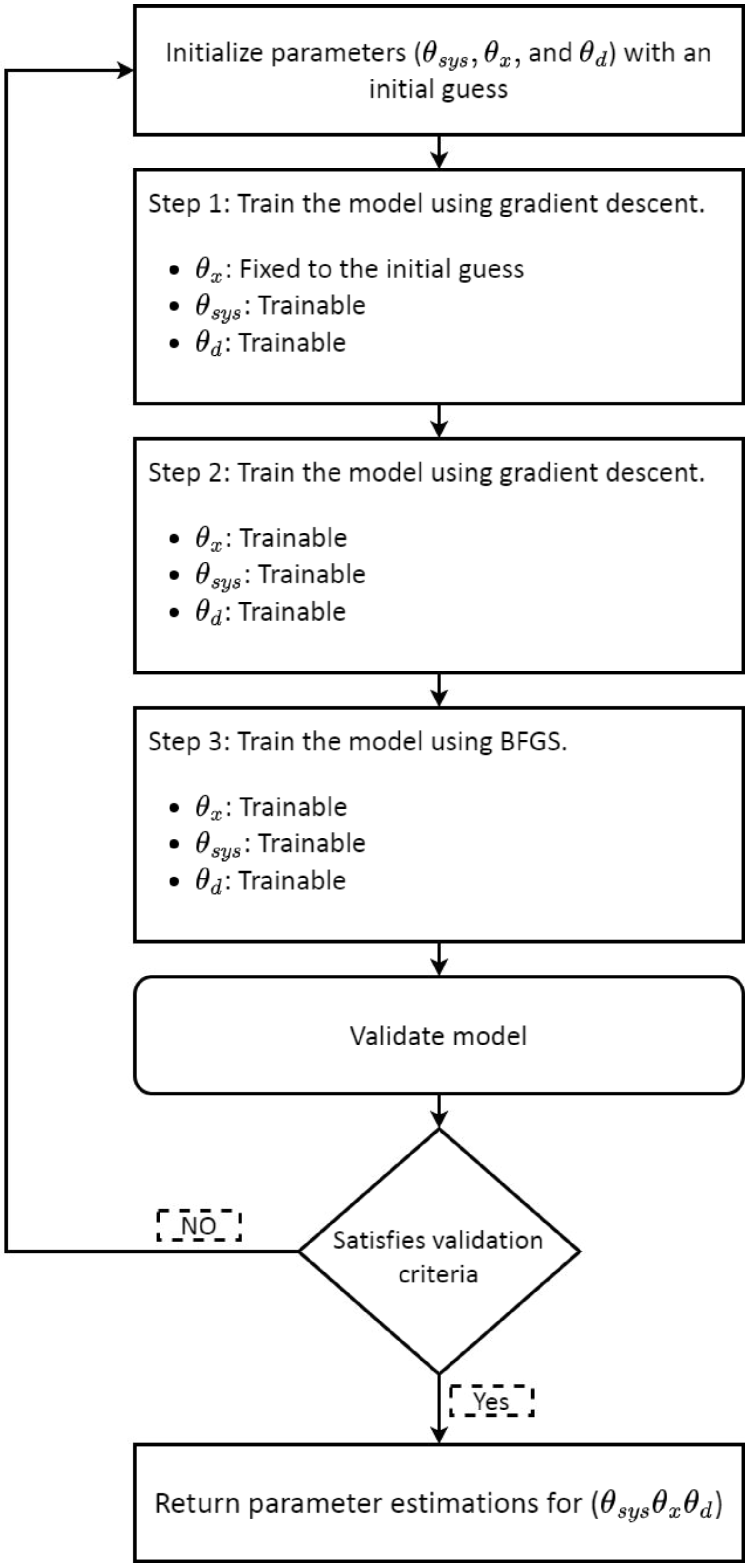

Initialize parameters (θ_{sys}, θ_x, and θ_d) with an initial guess
Step 1: Train the model using gradient descent.
• θ_x: Fixed to the initial guess
• θ_{sys}: Trainable
• θ_d: Trainable
Step 2: Train the model using gradient descent.
• θ_x: Trainable
• θ_{sys}: Trainable
• θ_d: Trainable
Step 3: Train the model using BFGS.
• θ_x: Trainable
• θ_{sys}: Trainable
• θ_d: Trainable
Validate model
Satisfies validation criteria
NO
Yes
Return parameter estimations for ($\theta_{sys}\theta_x\theta_d$)

Figure 3.3: Flowchart representing the three-step training methodology

Chapter 4

Application of the Hybrid Modeling Framework to a Building Space

In this chapter, the proposed hybrid modeling framework and the three-step training methodology are applied to model the thermal dynamics of a building space.

4.1 Description of the Building Space

In building modeling, interior spaces are areas which are enclosed by walls, floors, and ceilings. These interior spaces are divided into zones, where each zone represents a particular area with its own temperature control mechanism. In the context of modeling, it is common practice to consider an aggregated approach where building spaces, potentially comprising several building zones, are modeled as a single entity. The air temperature within the space is assumed to be spatially uniform (a standard assumption made in building modeling). For control-oriented physics-based modeling of building spaces, the thermal dynamics of the building spaces are represented as low-order thermal resistance-capacitance (RC) networks, where the thermal resistances and capacitances are estimated from data obtained for a specific building space [3, 13, 28]. The thermal resistance determines the ability of a material to resist heat transfer. Similarly, the thermal capacitance refers to the ability of a material to store thermal energy.

Consider a building thermal space modeled with two thermal masses given by:

$$\begin{aligned} C_{ia}\frac{dT_{ia}}{dt} &= \frac{1}{R_{oi}}(T_{oa}-T_{ia}) + \frac{1}{R_{mi}}(T_m - T_{ia}) - \dot{Q}_{clg} + \dot{Q}_{int} + \alpha_{ia}I_t \\ C_m\frac{dT_m}{dt} &= \frac{1}{R_{mi}}(T_{ia}-T_m) + \frac{1}{R_{mo}}(T_{oa}-T_m) + \alpha_m I_t \end{aligned} \tag{4.1}$$

where the subscript ia denotes indoor air, m denotes (lumped) building thermal mass, and oa denotes outdoor air. The notation T_j ($j = ia, oa, m$) denotes temperature, C_j ($j = ia, m$) denotes thermal mass capacitance, R_i ($i = oi, mi, mo$) denotes thermal resistance, $\dot{Q}_{clg}$ denotes the sensible cooling rate of the HVAC equipment, $\dot{Q}_{int}$ denotes the interior heat gains from all other sources (e.g., occupants, lighting, and equipment operating within the space), α_{ia} and α_m are constants, and I_t represents the solar radiation intensity. The heat gains in the building space due to solar radiation can be calculated as, $\dot{Q}_{solar,j} = \alpha_j I_t$ ($j = ia, m$). The thermal RC network describing the space thermal dynamics is shown in Figure 4.1.

The interior heat gains $\dot{Q}_{int}$ and the heat gains due to solar radiation $\dot{Q}_{solar,ia}$ cannot be separately measured. For modeling, these disturbances are lumped together into the term $\dot{Q}_{other}$. For the space considered, the heat gain due to solar radiation on the thermal masses ($\dot{Q}_{solar,m}$) is small and can be neglected. Further discussion on these disturbances is provided in the next section. The resulting model, upon incorporating these simplifications, is parameterized and given by:

$$\begin{aligned} \frac{dT_{ia}}{dt} &= \theta_1(T_{oa}-T_{ia}) + \theta_2(T_m - T_{ia}) - \theta_3\dot{Q}_{clg} + \theta_3\dot{Q}_{other}, \quad && T_{ia}(0) = \theta_6 \\ \frac{dT_m}{dt} &= \theta_4(T_{ia}-T_m) + \theta_5(T_{oa}-T_m), && T_m(0) = \theta_7 \end{aligned} \tag{4.2}$$

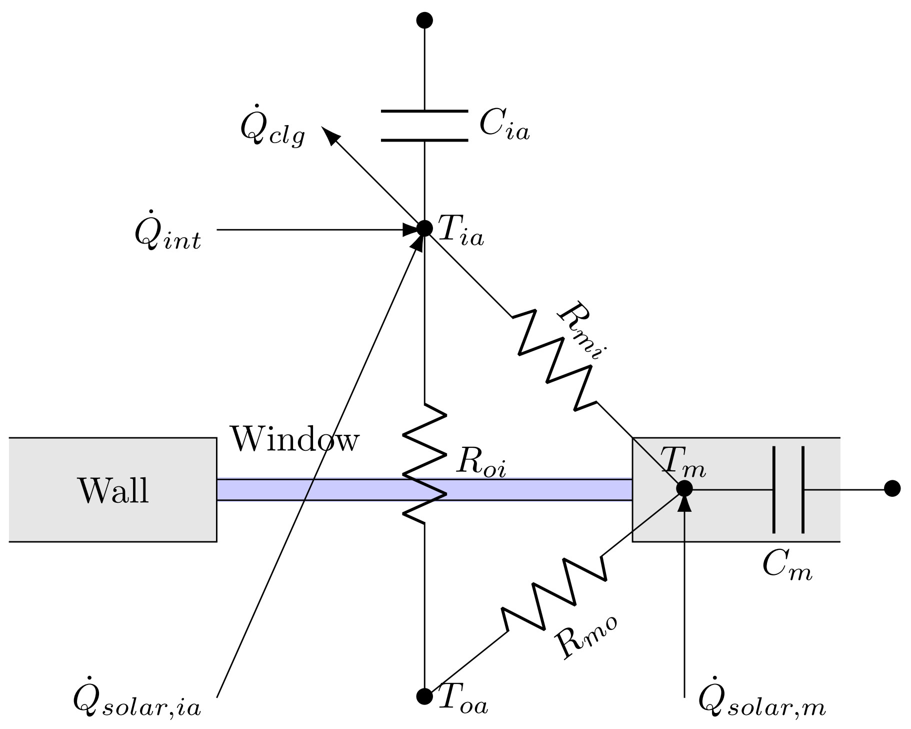


Figure 4.1: Thermal resistance-capacitance network of the building space.

where $\theta_{sys}^T := [\ \theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ ]$ and $\theta_x^T := [\ \theta_6\ \theta_7\ ]$. Detailed building space models are typically of higher order than the model in (4.2). However, the model in (4.2) is used to simulate the building space to avoid the issue of significant structural plant-model mismatch, which poses additional challenges that are beyond the scope of the present work. The parameter values used to simulate the building space are provided in Table 4.1. The sensible cooling rate of the HVAC system is modeled as being regulated by a proportional-integral (PI) controller with saturation and anti-windup via back-calculation. The nonlinear controller approximately models a variable air volume (VAV) terminal box and its control, whereby the discharge flow rate of a VAV box is manipulated by a regulatory controller to ensure that the indoor air temperature is maintained at the setpoint. The maximum cooling capacity is denoted by $\dot{Q}_{clg,max}$ and its value is provided in Table 4.1. The cooling rate, $\dot{Q}_{clg}$ is treated as a manipulated input to the system model.

Table 4.1: The simulated building thermal RC model parameters and their respective values.

Parameter	Value	Parameter	Value
C_{ia}	$196.8\,\mathrm{kJ\,K^{-1}}$	R_{mo}	$125.3\,\mathrm{K\,kW^{-1}}$
C_m	$2530\,\mathrm{kJ\,K^{-1}}$	α_{ia}	$3.8 \times 10^{-4}\,\mathrm{kW\,m^2\,W^{-1}}$
R_{mi}	$0.781\,\mathrm{K\,kW^{-1}}$	α_m	$1.391 \times 10^{-5}\,\mathrm{kW\,m^2\,W^{-1}}$
R_{oi}	$11.6\,\mathrm{K\,kW^{-1}}$	$\dot{Q}_{clg,max}$	$10\,\mathrm{kW}$

The parameter vectors are given by:

$$\theta_{sys}^T := \begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5\end{bmatrix} = \begin{bmatrix}\frac{1}{C_{ia}R_{oi}} & \frac{1}{C_{ia}R_{mi}} & \frac{1}{C_{ia}} & \frac{1}{C_m R_{mi}} & \frac{1}{C_m R_{mo}}\end{bmatrix}$$

$$\theta_x^T := \begin{bmatrix}\theta_6 & \theta_7\end{bmatrix} = \begin{bmatrix}T_{ia,0} & T_{m,0}\end{bmatrix}$$

The continuous-time model is converted into a discrete-time model by assuming zeroth-order hold on all inputs. The resulting parameterized discrete-time linear model, with $x^T = [T_{ia}\ T_m]$, $u^T = [\dot{Q}_{clg}\ T_{oa}]$, $d = \dot{Q}_{other}$, and $y = T_{ia}$, is given as:

$$\begin{aligned} x_{k+1} &= A(\theta_{sys})x_k + B(\theta_{sys})u_k + B_d(\theta_{sys})d_k \\ x_0 &= \theta_x \\ y_k &= Cx_k \end{aligned} \tag{4.3}$$

where $C = [1\ 0]$, and $A(\theta_{sys})$ and $B(\theta_{sys})$ are computed from (4.2).

4.2 Data Collection and Generation

The dynamic model in (4.1) with the air temperature regulated by the PI controller with anti-windup is simulated under a time-varying setpoint. From this simulation, data is collected to train the parameters of a hybrid model. The training period is defined as a specific period of time during which the model is trained on the training dataset. Similarly, the validation period is a period of time during which the performance of the trained model is evaluated on a separate dataset, called the validation dataset. The parameters of the hybrid building model are trained using a seven-day simulation of a building space, starting at midnight on Sunday, July 28, 2019 and ending at midnight on Sunday, August 4, 2019, corresponding to the training period. Similarly, the hybrid model is validated using a seven-day simulation of the same building space from midnight on Sunday, August 4, 2019 to midnight on Sunday, August 11, 2019, corresponding to the validation period.

During the simulation, a temperature setpoint excitation signal is used, generated from a pseudo-random binary signal (PRBS), to generate the training and validation data. The excitation setpoint trajectory has values of either 23°C or 25°C. This input-output data consisting of $\dot{Q}_{clg}$, T_{oa}, and T_{ia} is collected with a sampling time of 1 minute. All simulated trajectories generated to construct the training and validation datasets are depicted in Figures 4.2 and 4.3.

4.3 Three-Step Training Methodology with Predictor Variable Selection

The modeling objective is to develop a hybrid building model capable of forecasting the indoor air temperature, given a prediction of the outdoor air temperature, HVAC cooling rate, and predictor variables. In this section, the appropriate predictor variables to forecast the unmeasured heat disturbances associated with building spaces are determined. Following this, the proposed hybrid modeling framework is applied to formulate a building space model and the three-step training methodology is demonstrated by training hybrid building models and evaluating their predictions.

In buildings, the heat gains (heat disturbances) provided to the thermal masses result

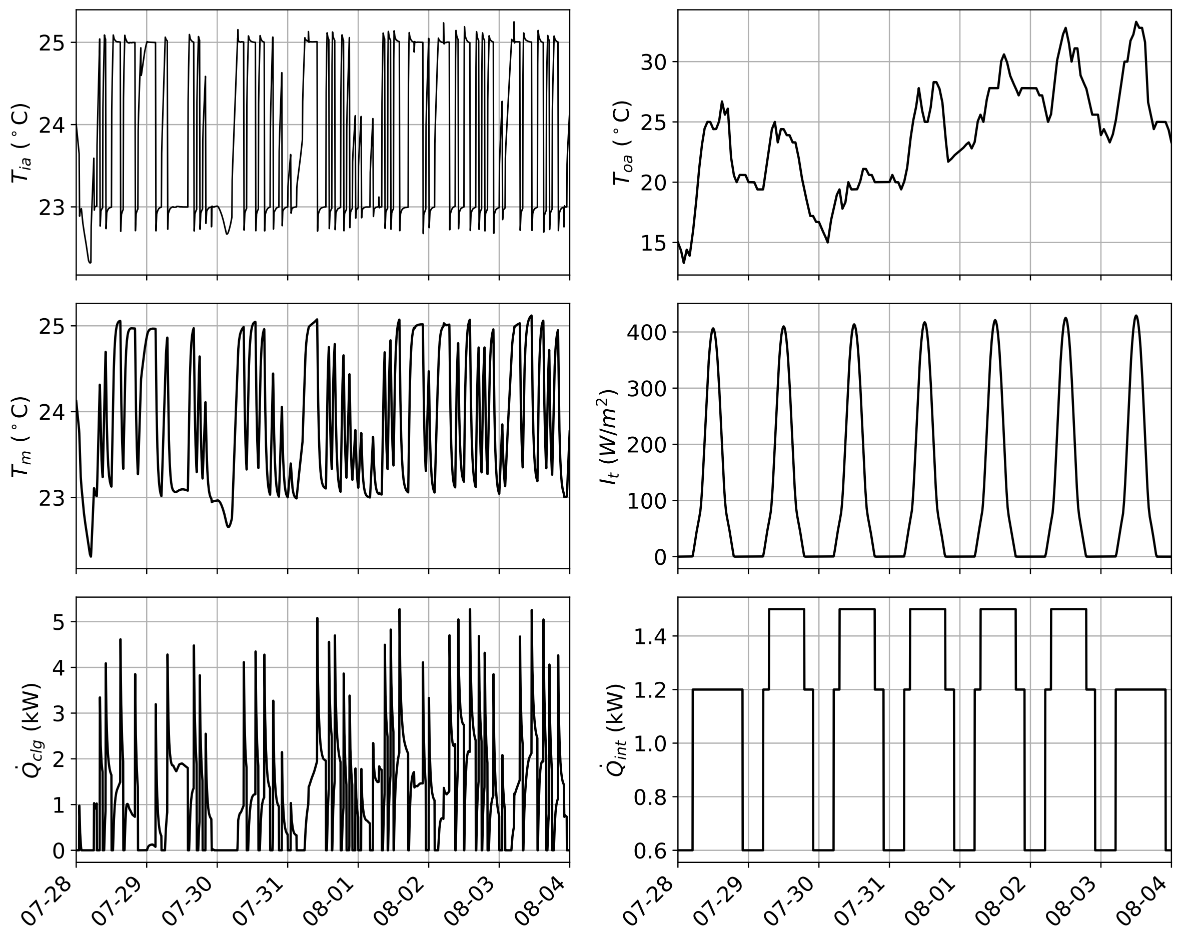


Figure 4.2: Resulting trajectories of the building space during the training period under the temperature setpoint trajectory generated to excite the system dynamics.

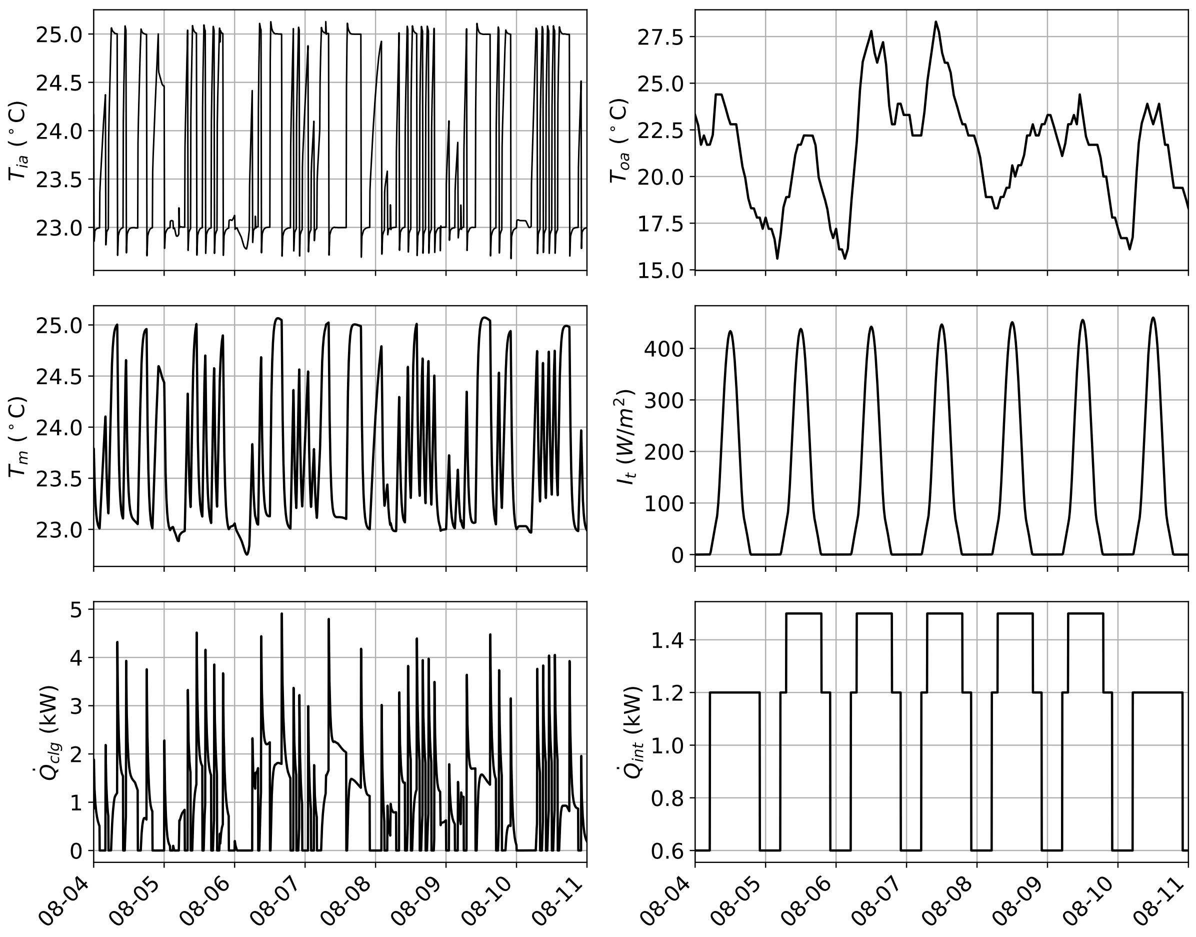


Figure 4.3: Resulting trajectories of the building space during the validation period under the temperature setpoint trajectory generated to excite the system dynamics.

from, for example, solar radiation, people, lighting, and electrical equipment. These disturbances are not generally measured but need to be forecasted for acceptable closed-loop performance under MPC (e.g., [26]). The heat gains from solar radiation ($\dot{Q}_{solar}$) are a function of the sun's position with respect to earth, which in-turn is a function of the time of the day (TOD) and the day of the year (DOY). Heat gains from people, lighting, and electrical equipment ($\dot{Q}_{int}$) are a function of the occupancy of the buildings. For buildings (e.g., commercial buildings), the occupancy pattern is dependent on the day type (e.g., day of the week, time of the day, weekday, and weekend). Therefore, $\dot{Q}_{int}$ is correlated to the day of the week (DOW), whether it is a weekday or a weekend (WW), and the time of the day (TOD). This can also be seen from Figures 4.2 and 4.3 where the values of $\dot{Q}_{int}$ are higher during the weekdays and lower during the weekends. Hence, the relevant time features that will be considered as predictor variables for forecasting the heat disturbances ($\dot{Q}_{other}$) are TOD, DOW, DOY, and WW.

The FNN of the form (3.1) is used to forecast the unmeasured heat disturbances ($\dot{Q}_{other}$) from the predictor variables. In particular, three choices of the predictor variables are considered:

- Time Of Day, Day Of Week, Day of Year
- Time Of Day, Day Of Week
- Time Of Day, Day Of Week, Weekday or Weekend

The proposed FNN has a dense network architecture which consists of one input layer, two hidden, layers and an output layer. The input to the network consists of the predictor variables. The activation function employed in the hidden layers is the hyperbolic tangent (tanh), while the activation function of the output layer is the rectified linear unit (ReLU).

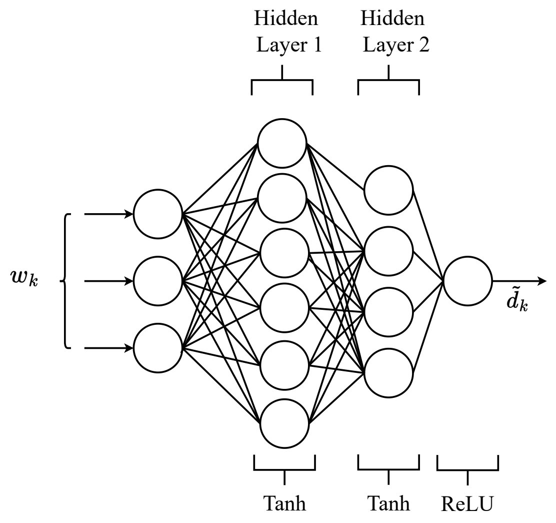
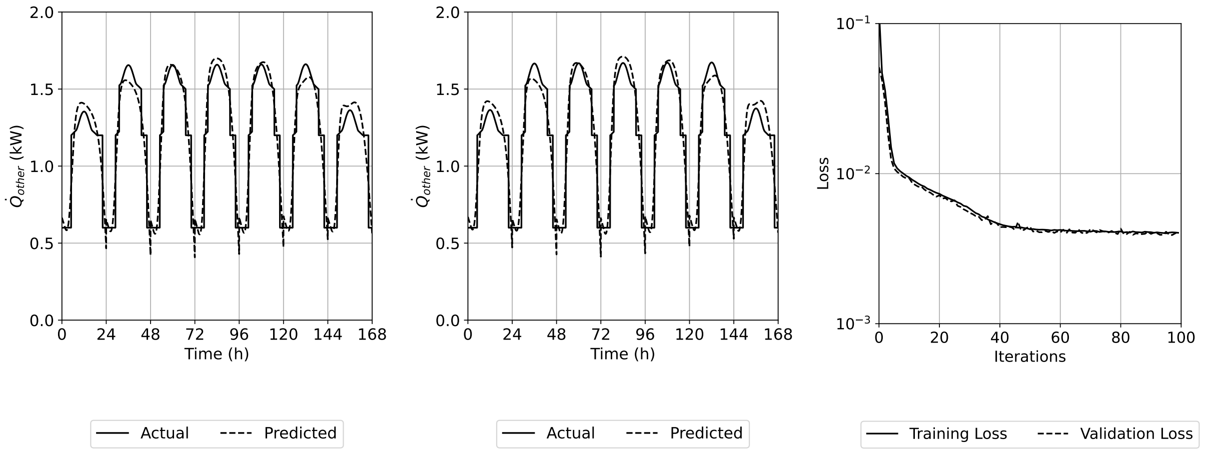


Figure 4.4: A diagram of the dense FNN architecture used to predict $\dot{Q}_{other}$

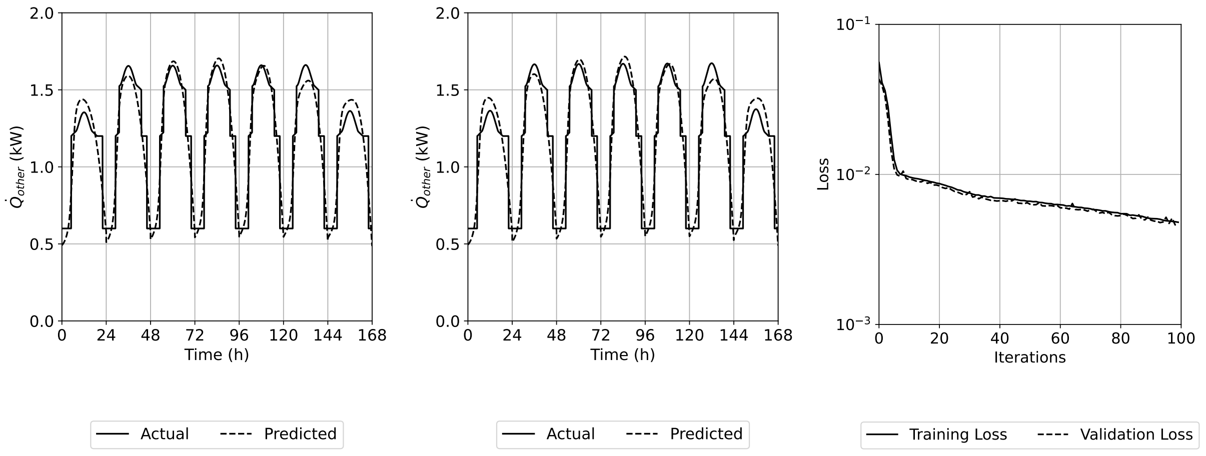
The first hidden layer consists of 6 neurons, while the second layer consists of 4 neurons. A graphical representation of the network architecture is provided in the accompanying Figure 4.4.

The choice of the FNN architecture is first validated. Three FNNs are developed using TensorFlow [1] and are trained in a supervisory manner by assuming that the heat disturbance ($\dot{Q}_{other}$) is known. Specifically, the Adam optimizer is used for 100 iterations, to train the weights and biases of the FNNs. The dataset used to train and validate the predictions of these FNNs consists of time features (TOD, DOW, DOY, WW), and the corresponding $\dot{Q}_{other}$ values calculated from the data represented in Figure 4.2. The training and validation datasets correspond to those defined in Section 4.2. Model 1 uses TOD, DOW, and DOY as the predictor variables, model 2 uses TOD and DOW as the predictor variables, and model 3 uses TOD, DOW, and WW as the predictor variables. The predictions of $\dot{Q}_{other}$ compared to the actual values of $\dot{Q}_{other}$ over the training and validation datasets generated by models 1, 2, and 3, are presented in Figures 4.5, 4.6, and 4.7, respectively.



(a) (b) (c)

Figure 4.5: Prediction of $\dot{Q}_{other}$ generated by FNN with TOD, DOW, and DOY as predictor variables over (a) the training dataset and (b) the validation dataset. (c) The training and validation loss function values.



(a) (b) (c)

Figure 4.6: Prediction of $\dot{Q}_{other}$ generated by FNN with TOD and DOW as predictor variables over (a) the training dataset and (b) the validation dataset. (c) The training and validation loss function values.

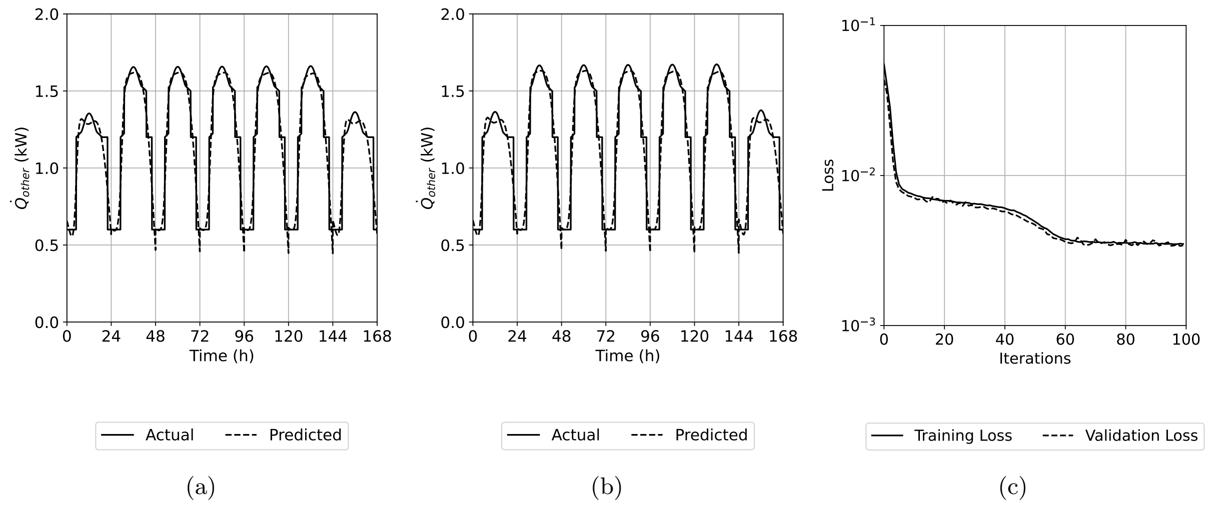


Figure 4.7: Prediction of $\dot{Q}_{other}$ generated by FNN with TOD, DOW, and WW as predictor variables over (a) the training dataset and (b) the validation dataset. (c) The training and validation loss function values.

Figures 4.5a, 4.6a, and 4.7a depict the predictions of $\dot{Q}_{other}$ generated by models 1, 2, and 3, respectively, on the training dataset. Figures 4.5b, 4.6b, and 4.7b depict the predictions of $\dot{Q}_{other}$ generated by the three FNN models on the validation dataset. Furthermore, Figures 4.5c, 4.6c, and 4.7c depict the training and validation losses for each of the three models. The maximum prediction error is used to evaluate the predictions generated by a model, and is defined as:

$$\max_{k\in\{0,\dots,N\}} |m_k - \tilde{m}_k| \tag{4.4}$$

where $N = N_{train}$(training dataset) or $N = N_{validation}$(validation dataset), m_k represents the actual value of variable at time step k, and $\tilde{m}_k$ represents the corresponding predicted value of the variable. From these figures, the maximum prediction error of $\dot{Q}_{other}$ is at most 0.1 kW across all three models. In addition, the average validation loss values across the three models with different predictor variables is less than 0.5×10^{-2}. Hence, the proposed FNN architecture is suitable for predicting $\dot{Q}_{other}$.

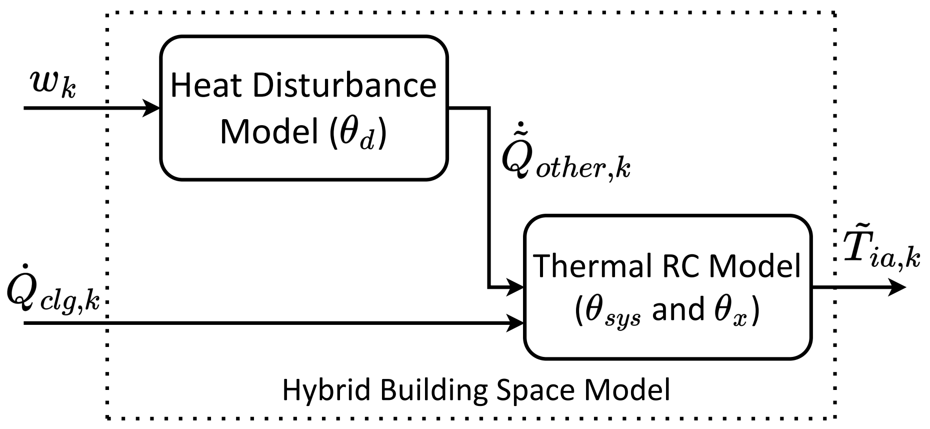

w_k
Heat Disturbance Model (θ_d)
$\dot{\tilde{Q}}_{other,k}$
$\dot{Q}_{clg,k}$
Thermal RC Model (θ_{sys} and θ_x)
$\tilde{T}_{ia,k}$
Hybrid Building Space Model

Figure 4.8: Block diagram of the hybrid modeling framework applied to the building space.

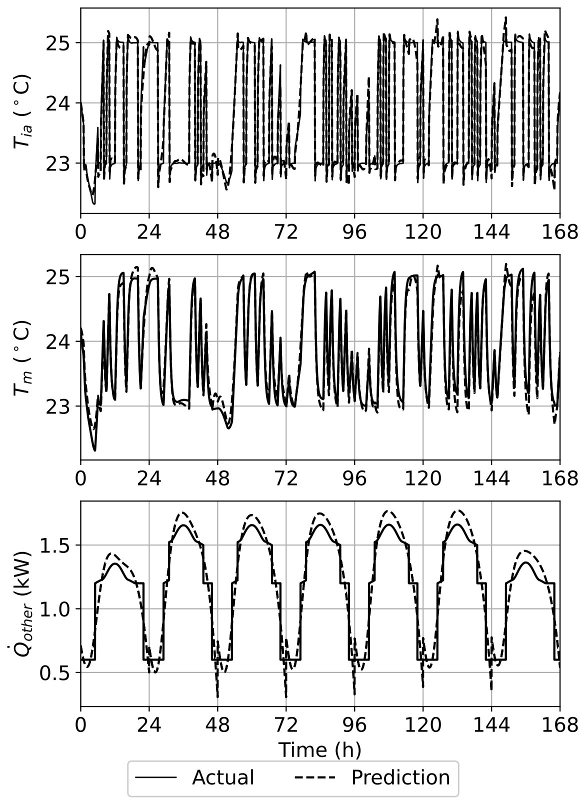
The proposed hybrid building model combines the thermal RC model described in (4.3), for modeling building space thermal dynamics, with the FNN model described above, to forecast the unmeasured heat disturbances. This results in a model of the form:

$$\begin{aligned}
\tilde{x}_{k+1} &= A(\theta_{sys})\tilde{x}_k + B(\theta_{sys})u_k + B_d(\theta_{sys})\tilde{d}_k \\
\tilde{x}(0) &= \theta_x \\
\tilde{d}_k &= f_d(w_k;\theta_d) \\
\tilde{y}_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \tilde{x}_k \\
\tilde{x}_k^T &= \begin{bmatrix} \tilde{T}_{ia,k} \tilde{T}_{m,k} \end{bmatrix}, u_k^T = \begin{bmatrix} \dot{Q}_{clg,k} T_{oa,k} \end{bmatrix}
\end{aligned} \tag{4.5}$$

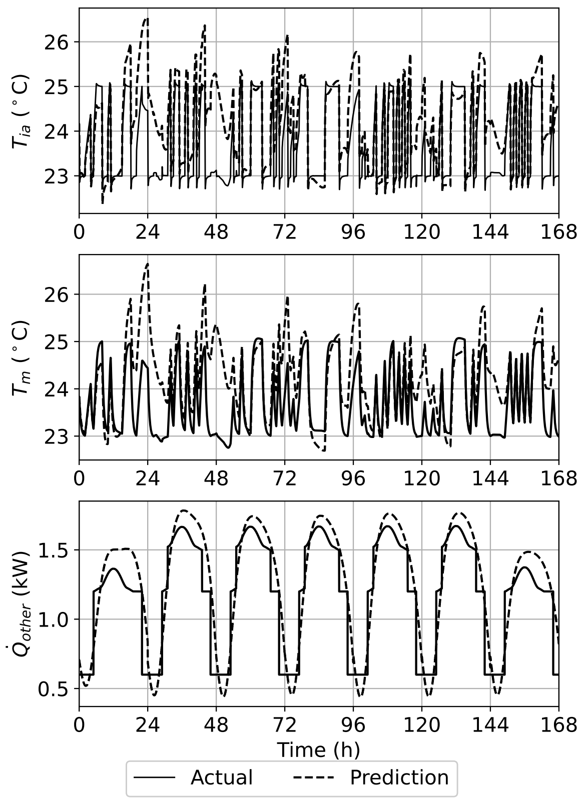
where $A(\theta_{sys})$, $B(\theta_{sys})$, and $B_d(\theta_{sys})$ are defined in Section 4.1. Figure 4.8 presents a block diagram of the proposed hybrid building space model. The inputs to the heat disturbance model are the predictor variables (w_k). The disturbance model forecasts the unmeasured heat gains ($\dot{\tilde{Q}}_{other,k}$). The forecasted heat gains along with the HVAC cooling rate ($\dot{Q}_{clg,k}$) are provided to the thermal RC model, which outputs the indoor air temperature $\tilde{T}_{ia,k}$.

The simulation PEM is employed to simultaneously estimate the thermal RC initial conditions and FNN model parameters. The training problem of the form, provided in (3.5), is adapted for estimating the parameters of the hybrid building space thermal model. The three-step training methodology, following Algorithm 1 adapted for this study, is utilized to solve the training problem. The hyperparameters for the three-step training methodology are as follows. Instead of gradient descent, the Adam optimizer is used. The exact gradients are computed using TensorFlow [1], and the BFGS algorithm is implemented using TensorFlow-Probability. The first step employees 3000 iterations, the second step uses 2500 iterations, and the third step involves 2500 iterations. In TensorFlow-Probability's implementation of BFGS, there is no method to extract the training and validation loss every iteration. In lieu of this, the training and validation loss function values are recorded every time the loss function is called, which can happen several times each iteration during the line search. These additional calls to the loss function are also recorded. Hence, the total number of iterations are normalized to match the set 2500 iterations. While this does not capture the true behavior of the training and validation loss, it still provides the overall behavior, which is found to be informative (e.g., to determine if the solver converged).

The collected input data $(T_{oa}, \dot{Q}_{clg})$ and the output data (T_{ia}) are normalized to a range between 0 and 1. Additionally, the time features to be used as predictor variables are also normalized. For the initialization strategy of the initial condition parameter (θ_x), the measured indoor air temperature at initial time is used for both states, i.e., the initialization strategy is $\theta_{x,0} = Hy_0 = [1\ 1]^T\, T_{ia,0}$. Correspondingly, the initial estimates for the system model parameters (θ_{sys}) are provided at random from a normal distribution with mean as 0.1 and standard deviation as 0.05. The initial estimates for the FNN model parameters (θ_d) are provided using the Glorot uniform initializer, which randomly initializes the weights where the value is drawn from a uniform distribution within the



(a)



(b)

Figure 4.9: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 1 with TOD, DOW, and DOY as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

limits $[-\theta_{init,max}, \theta_{init,max}]$ where $\theta_{init,max} = \sqrt{6/(n_{in}+n_{out})}$, n_{in} is the number of input units, and n_{out} is the number of output units [19].

A total of nine hybrid building models are formulated and trained, with three hybrid models being trained for each of the three predictor variable combinations described above, using the three-step training methodology. Models 1, 2, and 3 have TOD, DOW, and DOY as the predictor variables, models, 4, 5, and 6 have TOD and DOW as predictor variables, and models 7, 8, and 9 have TOD, DOW, and WW as predictor variables. The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by each of the nine hybrid building models are presented and discussed. All the hybrid models are initialized with different initial parameter estimates.

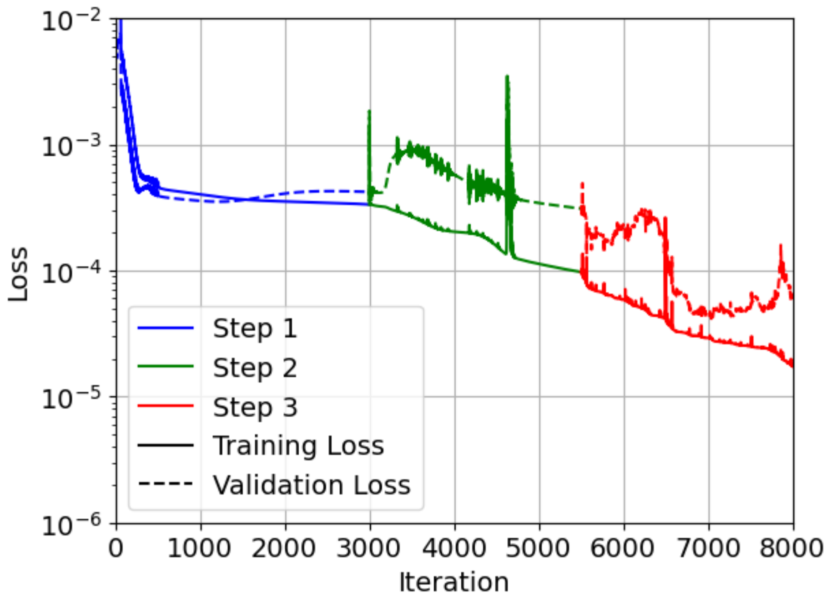
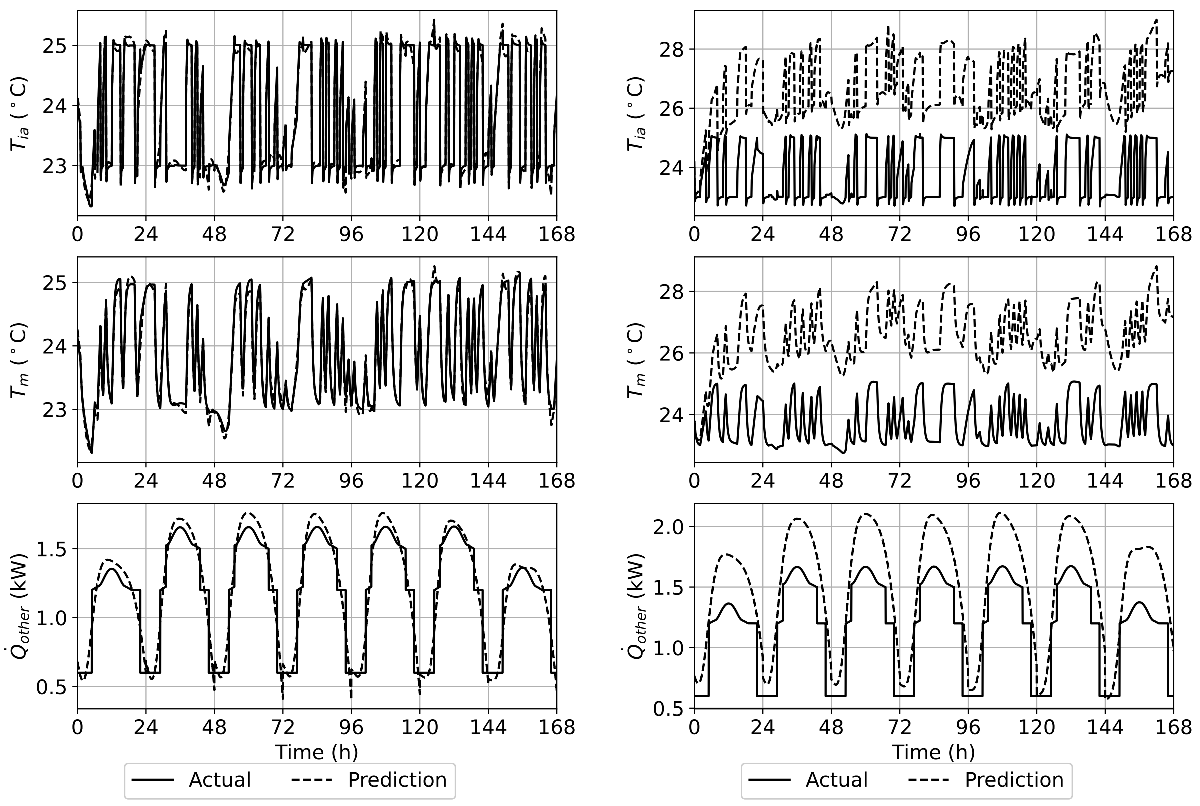


Figure 4.10: Training and validation loss for model 1 with TOD, DOW, and DOY as predictor variables



(a) (b)

Figure 4.11: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 2 with TOD, DOW, and DOY as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

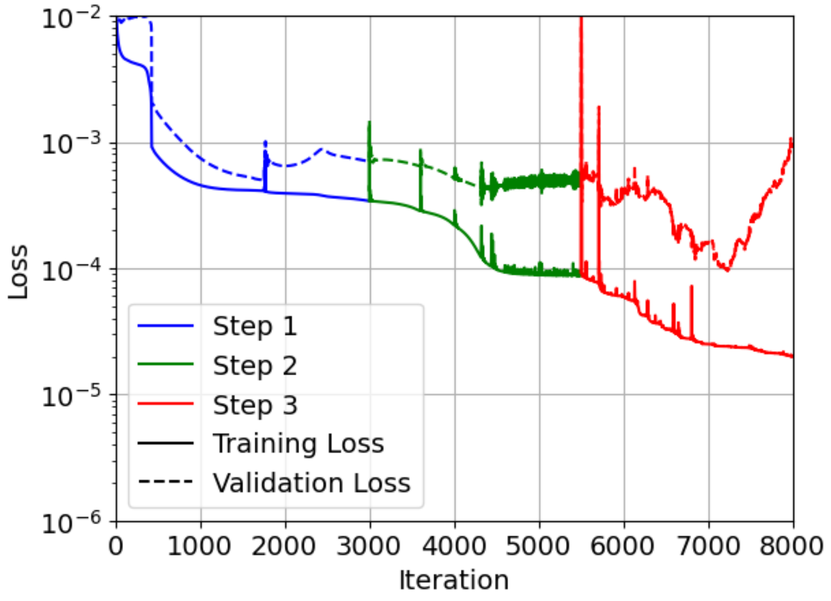
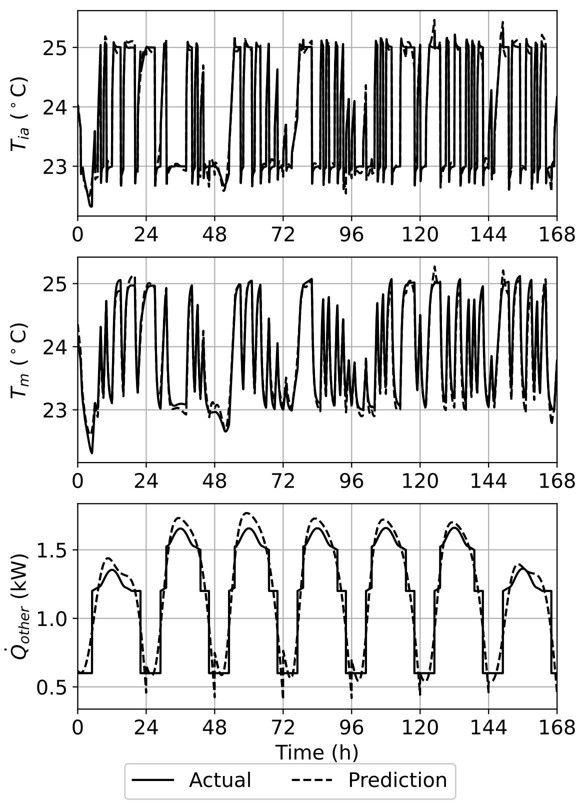
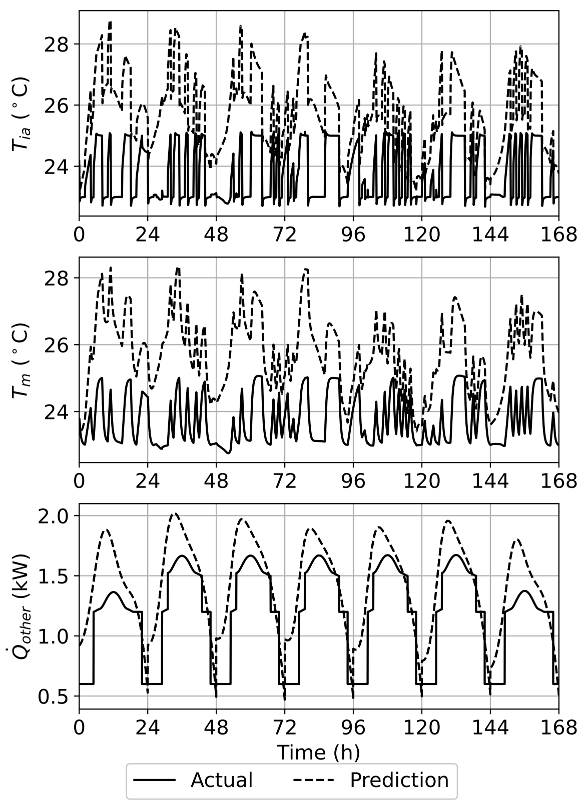


Figure 4.12: Training and validation loss for model 2 with TOD, DOW, and DOY as predictor variables



(a)



(b)

Figure 4.13: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 3 with TOD, DOW, and DOY as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

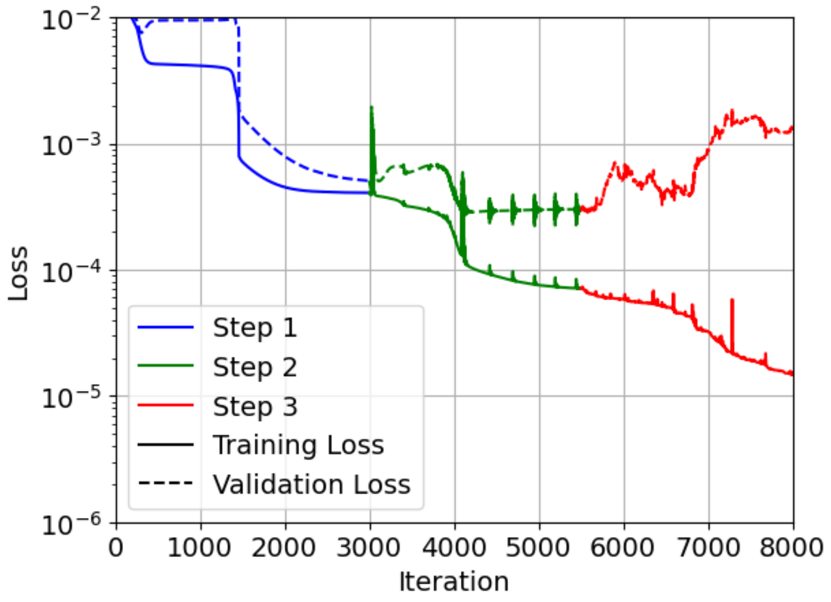


Figure 4.14: Training and validation loss for model 3 with TOD, DOW, and DOY as predictor variables

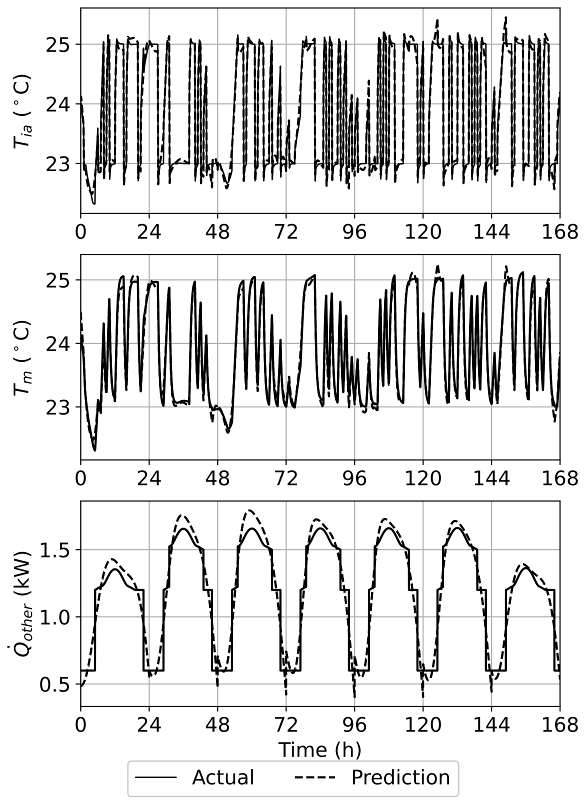
The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by hybrid building models 1, 2, and 3, over the training and validation periods are depicted in Figures 4.9, 4.11, and 4.13, respectively. Over the training period, the maximum prediction errors of the indoor air temperatures (T_{ia}) are 0.2°C, 0.2°C, and 0.19°C for models 1, 2, and 3, respectively, and the maximum prediction errors of the thermal mass temperatures (T_m) are 0.3°C, 0.29°C, and 0.3°C for models 1, 2, and 3, respectively. Over the validation period, the maximum prediction errors of the indoor air temperatures (T_{ia}) are 1.6°C, 4°C, and 4.3°C for models 1, 2, and 3, respectively, and the maximum prediction errors of the thermal mass temperatures (T_m) are 2.5°C, 4.3°C, and 4.1°C for models 1, 2, and 3, respectively. Additionally, the training and validation loss function values are presented in Figures 4.10, 4.12, and 4.14. The trained parameter values and the individual training and validation loss values for the three hybrid building models are presented in the Table 4.2. The average training loss of the three models is 7.42×10^{-6}, and the average loss over the validation period is 3.48×10^{-3}. Despite the small average training loss, the average validation loss

Table 4.2: The actual system parameter values compared to the trained system parameter values of the models with TOD, DOW, and DOY as predictor variables and the corresponding losses.

Parameters	Actual Parameter value	Model 1	Model 2	Model 3
θ_1	23.40	22.50	23.06	22.46
θ_2	1.58	1.97	1.53	1.94
θ_3	18.30	17.91	17.92	17.80
θ_4	1.82	1.85	1.82	1.85
θ_5	0.01	0.03	0.02	0.01
Training Loss		6.42×10^{-6}	8.76×10^{-6}	7.08×10^{-6}
Validation Loss		4.28×10^{-4}	6.54×10^{-3}	3.47×10^{-3}

is about 470 times greater than the training loss. One of the factors leading to greater prediction errors and loss values observed over the validation period is due to the model not being able to extrapolate beyond the training period. That is, the model is trained over the training period, which has range of DOY from 208 to 215, and validated over the validation period, which has range of DOY from 215 to 222. Hence, DOY may not be a good variable to be included as a predictor variable, as the predictions extrapolate beyond the training dataset. However, with the availability of more training data over a full year, DOY may be a useful predictor variable.

In the following three hybrid building models (models 4, 5, and 6), the absence of DOY as a predictor is studied. The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by hybrid building models 4, 5, and 6 over the training and validation periods are depicted in Figures 4.15, 4.17, and 4.19, respectively. Over the training period, the maximum prediction errors of the indoor air temperatures (T_{ia}) are 0.18°C, 0.1°C, and 0.19°C for models 4, 5, and 6, respectively, and the maximum prediction errors of the thermal mass temperatures (T_m) are 0.2°C, 0.18°C, and 0.2°C for models 4, 5, and 6, respectively. Over the validation period, the maximum prediction errors of the indoor air temperatures (T_{ia}) are 0.6°C, 0.3°C, and 0.6°C for models 4, 5, and 6, respectively, and the maximum prediction errors of the thermal mass temperatures (T_m) are 0.8°C, 0.7°C, and 0.7°C for models 4, 5, and 6,



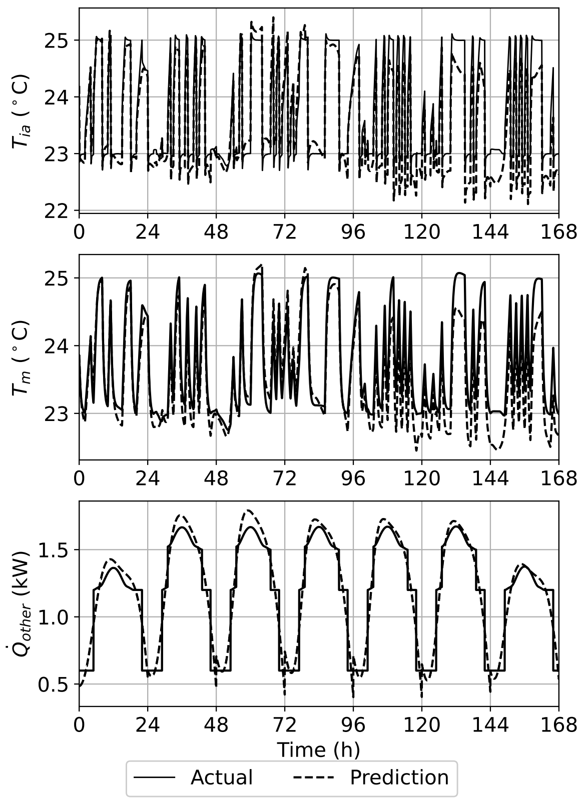


Figure 4.15: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 4 with TOD, and DOW as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

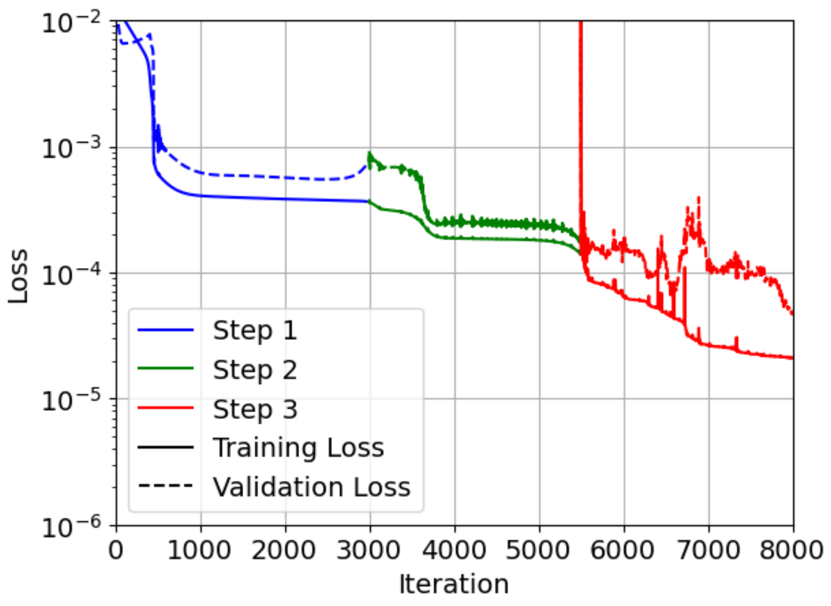


Figure 4.16: Training and validation loss for model 4 with TOD, and DOW as predictor variables

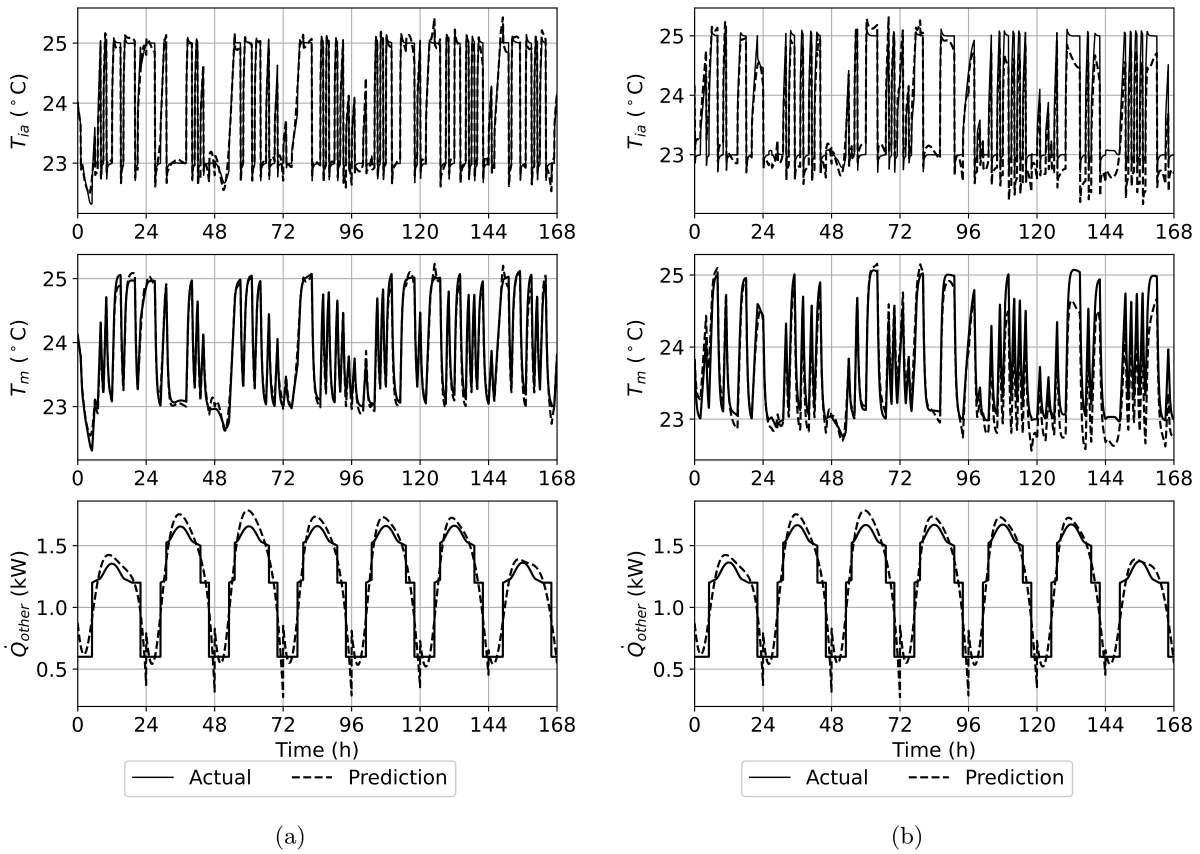


Figure 4.17: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 5 with TOD, and DOW as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

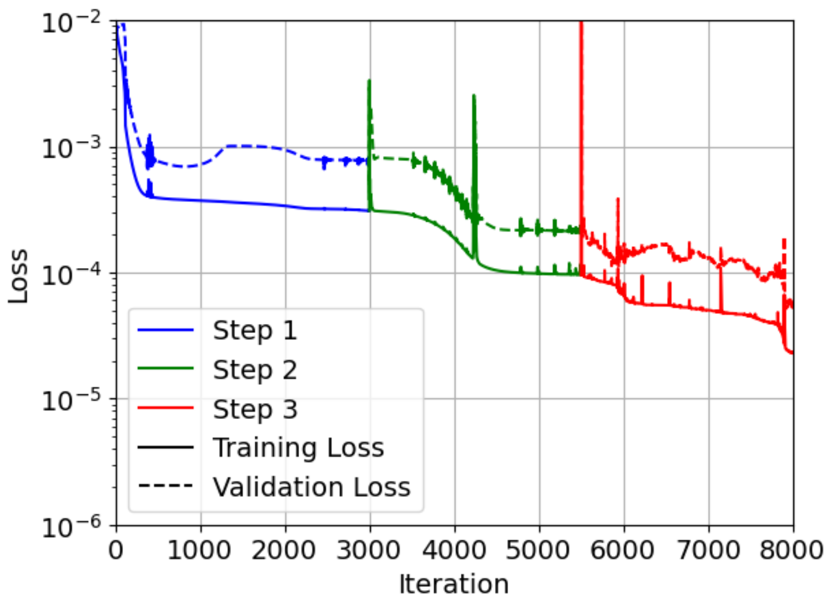
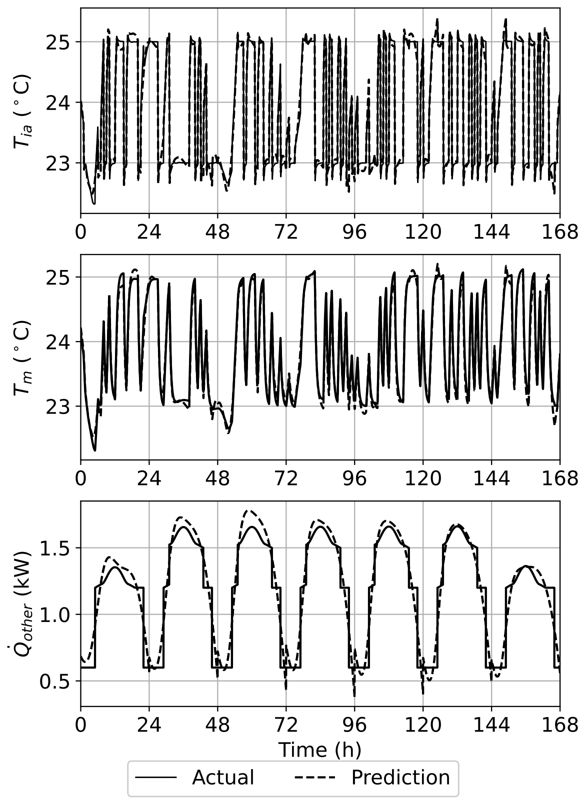
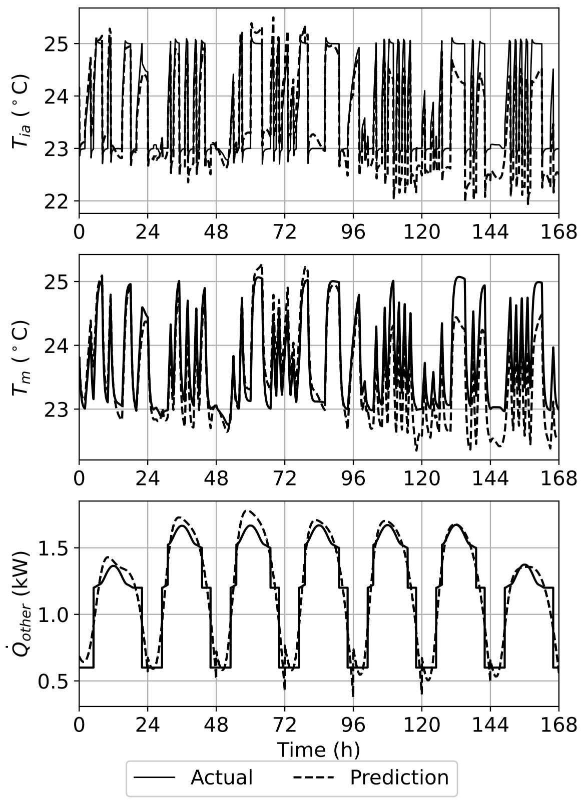


Figure 4.18: Training and validation loss for model 5 with TOD, and DOW as predictor variables



(a)



(b)

Figure 4.19: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 6 with TOD, and DOW as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

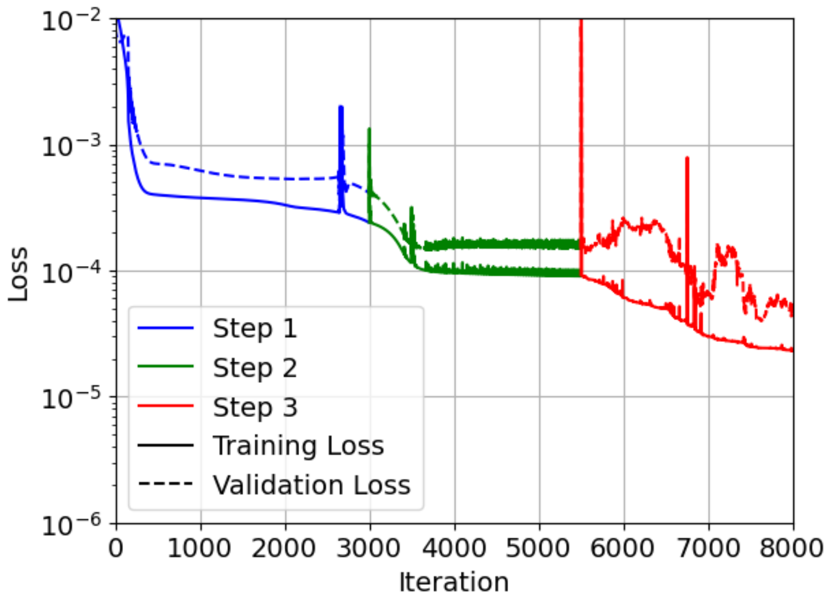


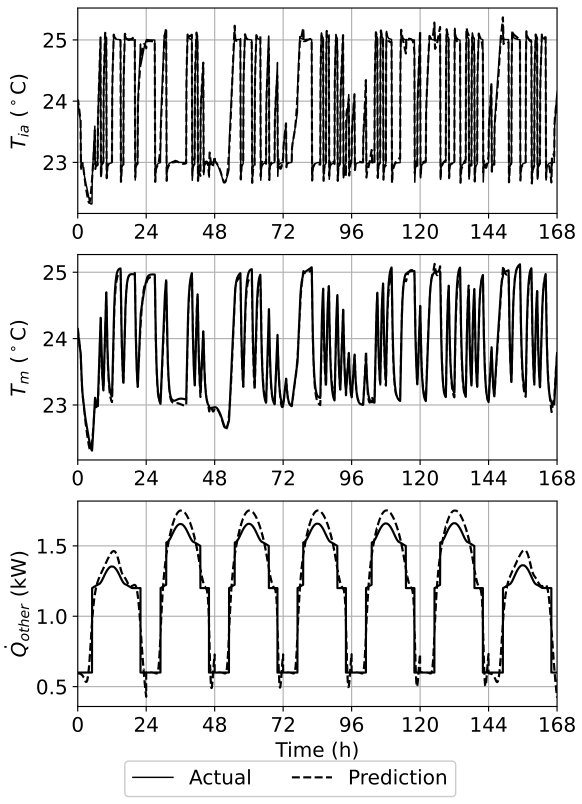
Figure 4.20: Training and validation loss for model 6 with TOD, and DOW as predictor variables

Table 4.3: The actual system parameter values compared to the trained system parameter values of the models with TOD and DOW as predictor variables and the corresponding losses.

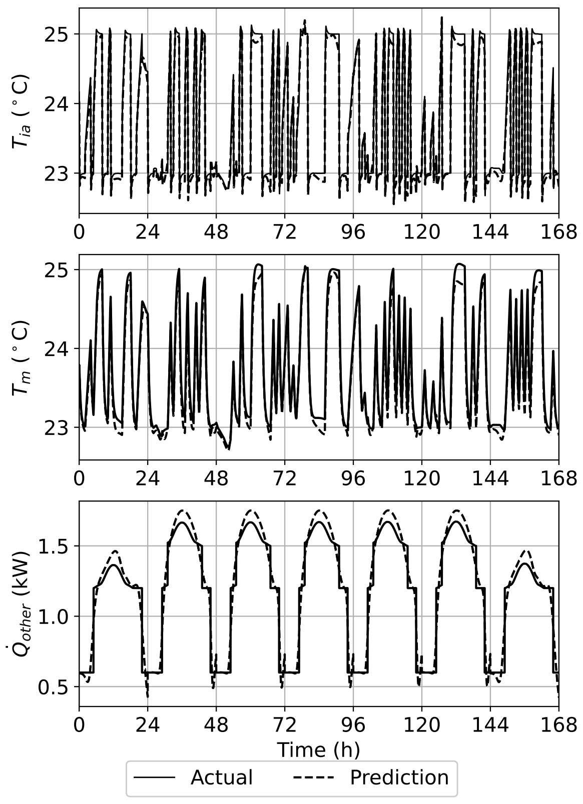
Parameters	Actual Parameters	Model 1	Model 2	Model 3
θ_1	23.40	22.50	23.06	22.46
θ_2	1.58	1.97	1.53	1.94
θ_3	18.30	17.91	17.92	17.80
θ_4	1.82	1.85	1.82	1.85
θ_5	0.01	0.03	0.02	0.01
Training Loss		7.03×10^{-6}	7.85×10^{-6}	7.27×10^{-6}
Validation Loss		5.67×10^{-5}	1.17×10^{-4}	7.62×10^{-5}

respectively. However, the maximum prediction errors over both training and validation periods for both T_{ia} and T_m are lower compared to when TOD, DOW, and DOY are used as predictor variables. Over the training period, the maximum prediction error of T_{ia} reduced by about 10% and maximum prediction error of T_m decreased about 60%. Over the validation period, the maximum prediction error of T_{ia} reduced by about 63% and maximum prediction error of T_m decreased by about 70%. Additionally, the training and validation loss function values are presented in Figures 4.16, 4.18, and 4.20. The trained parameter values and the individual training and validation loss values for the three hybrid building models are presented in the Table 4.3. The average training loss is 7.38×10^{-6} and the average validation loss is 8.32×10^{-5}. These average losses are lower than the losses when TOD, DOW, and DOY are used as predictor variables.

In the following three hybrid building models (models 7, 8, and 9) the predictions generated by models with predictor variables TOD, DOW, and WW are studied. The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by hybrid building models $7, 8$, and 9, over the training and validation periods are depicted in Figures 4.21, 4.23, and 4.25, respectively. Over the training period, the maximum prediction errors of the indoor air temperatures (T_{ia}) are 0.05°C, 0.04°C, and 0.05°C, for models 7, 8, and 9, respectively, and the maximum prediction errors of the thermal mass temperatures (T_m) are 0.08°C, 0.08°C, and



(a)



(b)

Figure 4.21: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 7 with TOD, DOW, and WW as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

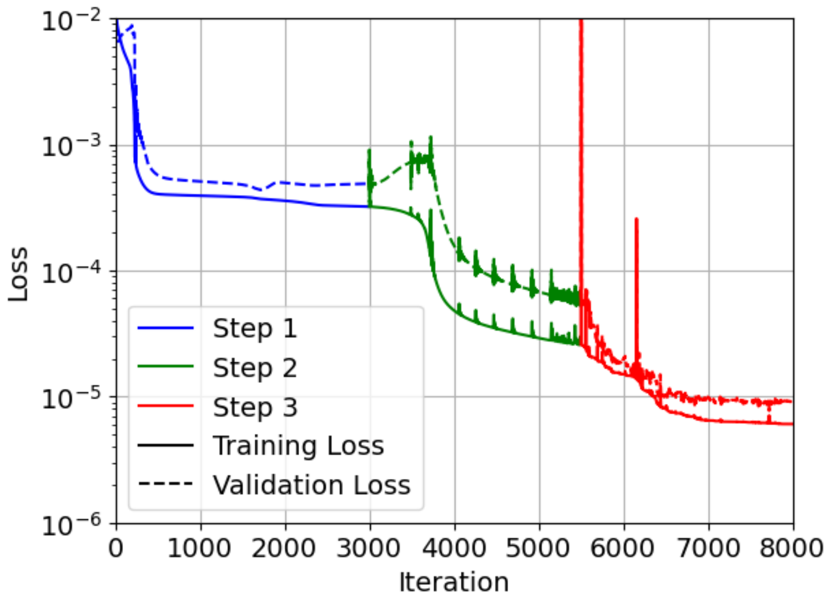
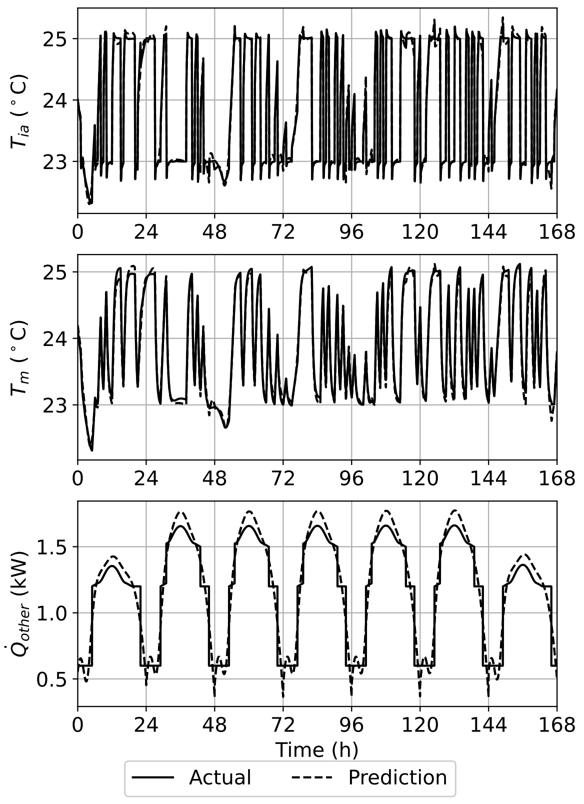
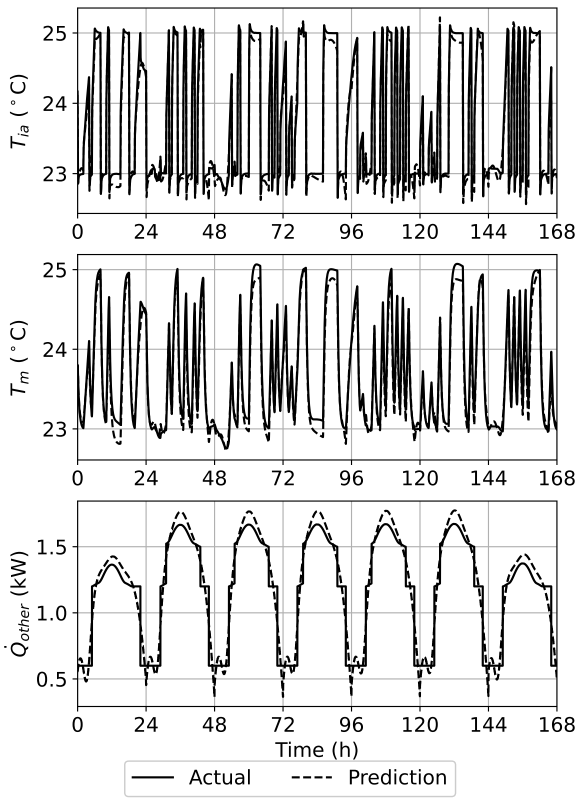


Figure 4.22: Training and validation loss for model 7 with TOD, DOW, and WW as predictor variables



(a)



(b)

Figure 4.23: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 8 with TOD, DOW, and WW as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

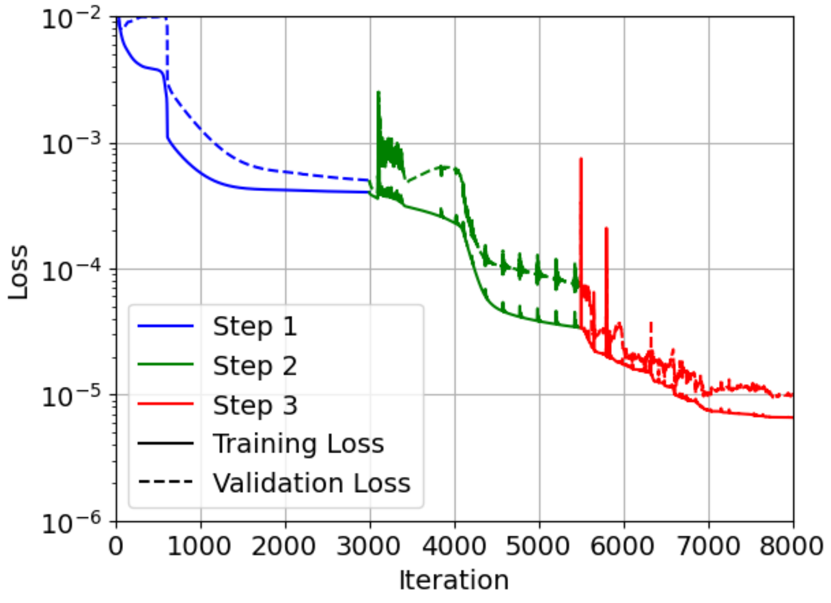


Figure 4.24: Training and validation loss for model 8 with TOD, DOW, and WW as predictor variables

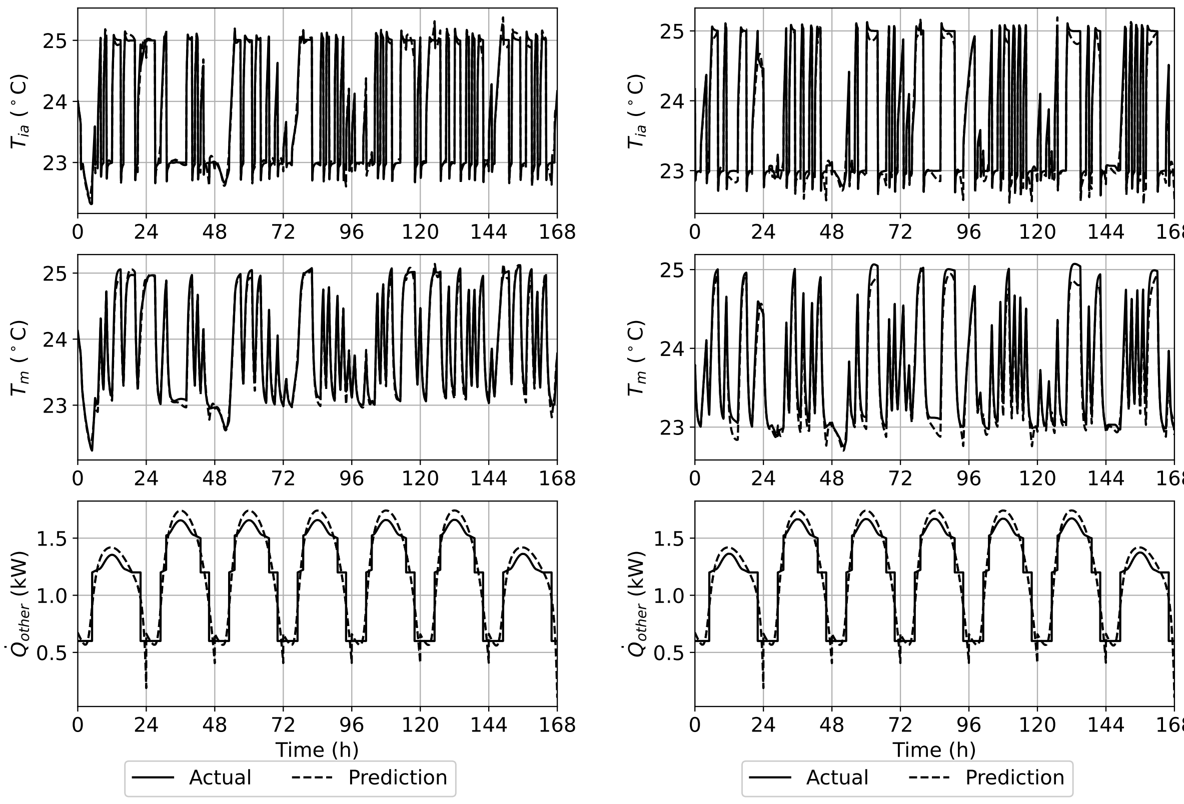


Figure 4.25: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by model 9 with TOD, DOW, and WW as predictor variables over (a) the training dataset and (b) the validation dataset compared to the actual values.

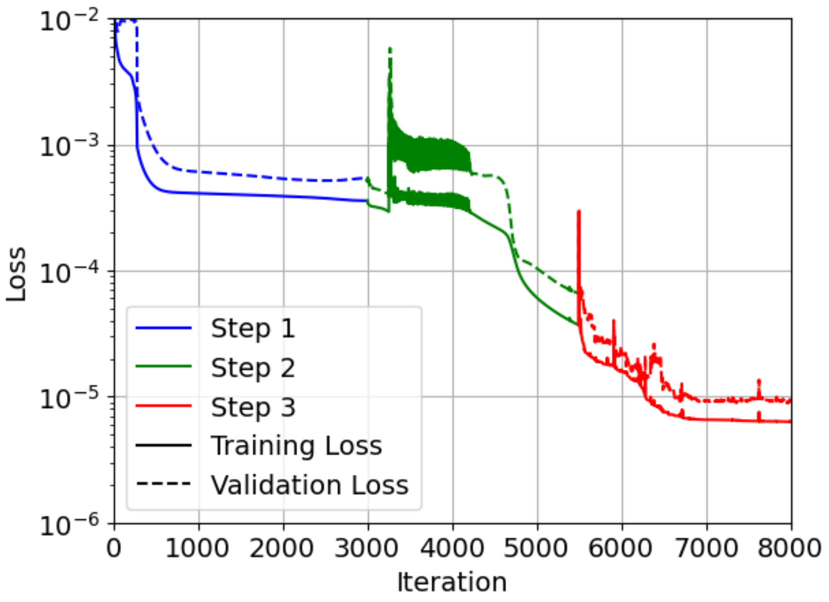


Figure 4.26: Training and validation loss for model 9 with TOD, DOW, and WW as predictor variables

Table 4.4: The actual system parameter values compared to the trained system parameter values of the models with TOD, DOW, and WW as predictor variables and the corresponding losses.

Parameters	Actual Parameters	Model 1	Model 2	Model 3
θ_1	23.40	22.50	23.58	22.85
θ_2	1.58	1.97	1.57	1.69
θ_3	18.30	17.91	18.36	17.94
θ_4	1.82	1.85	1.83	1.82
θ_5	0.01	0.03	0.01	0.00
Training Loss		7.03×10^{-6}	3.57×10^{-6}	6.37×10^{-6}
Validation Loss		5.67×10^{-5}	6.88×10^{-6}	9.38×10^{-6}

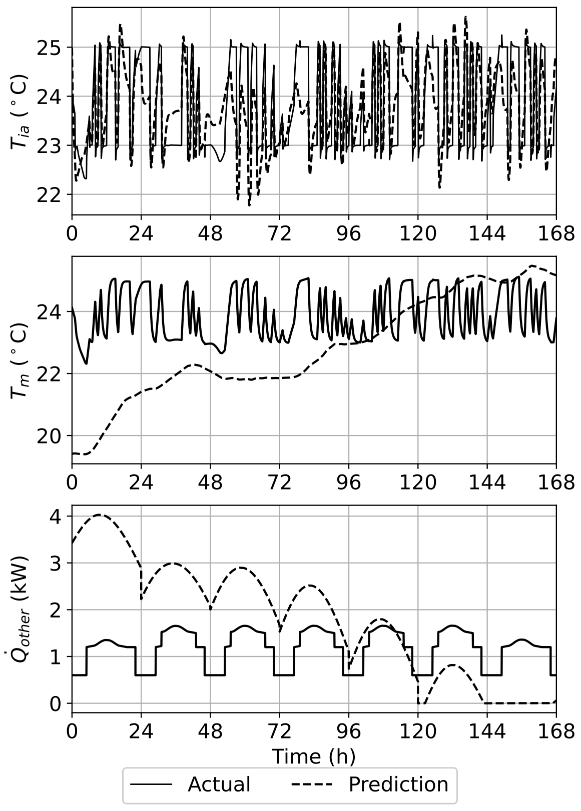
0.1°C, for models 7, 8, and 9, respectively. Over the validation period, the maximum prediction errors of the indoor air temperatures (T_{ia}) are 0.1°C, 0.12°C, and 0.13°C, for models 7, 8, and 9, respectively, and maximum prediction errors of the thermal mass temperatures (T_m) are 0.12°C, 0.13°C, and 0.13°C, for models 7, 8, and 9, respectively. The maximum prediction errors when TOD, DOW, and WW are used as predictor variables is lower than when TOD and DOW are used as the predictor variables. Additionally, the training and validation losses are presented in Figures 4.22, 4.24, and 4.26. The predicted parameter values and the training and validation loss values for all the three models are presented in Table 4.4. The difference between the training and validation loss values are lower than the differences when (TOD, DOW) and (TOD, DOW, DOY) are used as predictor variables. The average training loss is 5.65×10^{-6} and the average validation loss is 2.43×10^{-5}. These average validation loss values are lower than the loss values when TOD and DOW are used as predictor variables. Additionally, the maximum prediction errors over the validation and training periods are the lowest when compared to the other six cases. Hence, the combination of TOD, DOW, and WW as predictor variables performs the best among all the different combinations discussed above.

4.4 Comparison of Other Training Methodologies with the Three-Step Training Methodologies

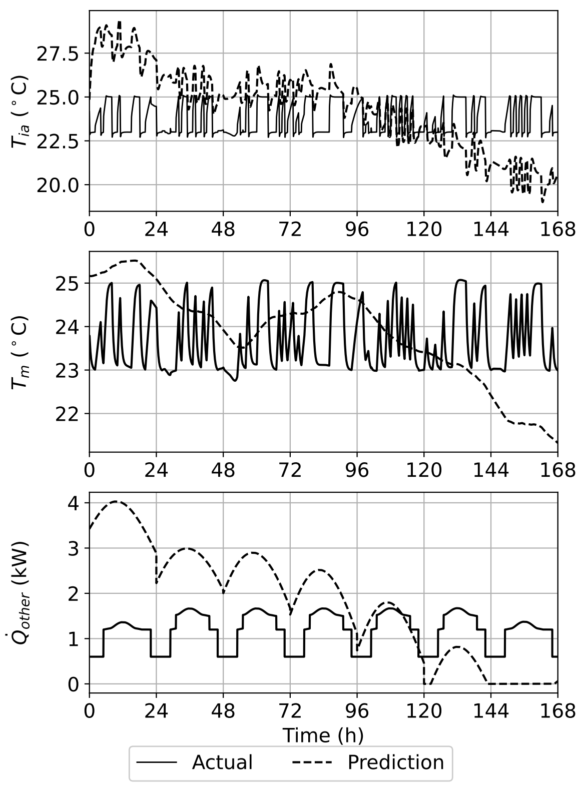
The aim of this section is to compare the three-step training methodology with alternative training methodologies. To achieve this, four hybrid building models are trained, all with identical initial parameter estimates. Since, the three-step training methodology utilizes both Adam optimizer and BFGS algorithms, the first model trained only uses the Adam optimizer, the second model trained only uses BFGS, the third model trained uses a single step of Adam optimizer (where all parameters including the initial condition parameters are trained) followed by a single step of BFGS, and the fourth model trained uses the three-step training methodology. The approach to train the third model is called the two-step training methodology.

The total number of iterations is 8000 in all cases. When the models are trained using only Adam optimizer or BFGS, the number of iterations for each of the solvers is 8000. In the case of training the model using the two-step training methodology, the total number of iterations for the gradient descent step is 5500 and that of BFGS step is 2500. When the model is trained using the three-step training methodology, the first gradient descent step utilizes 3000 iterations, the second gradient descent step utilizes 2500 iterations, and the third step of BFGS utilizes 2500 iterations. The predictor variables in all the cases are TOD, DOW, and WW.

The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model trained using only the Adam optimizer over the training and validation datasets are provided in Figure 4.27. Over the training period, the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 1.4°C, 4.2°C, and 3.9 kW, respectively. Over validation period, the maximum

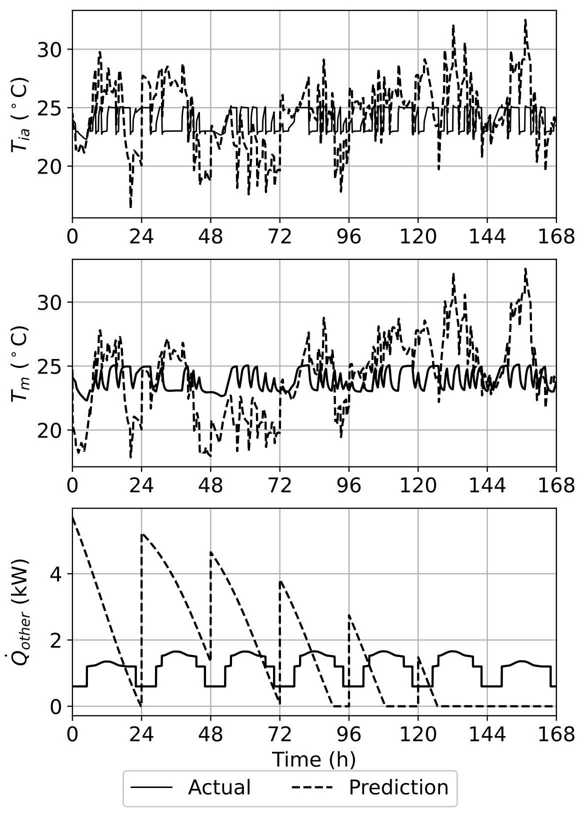


(a)

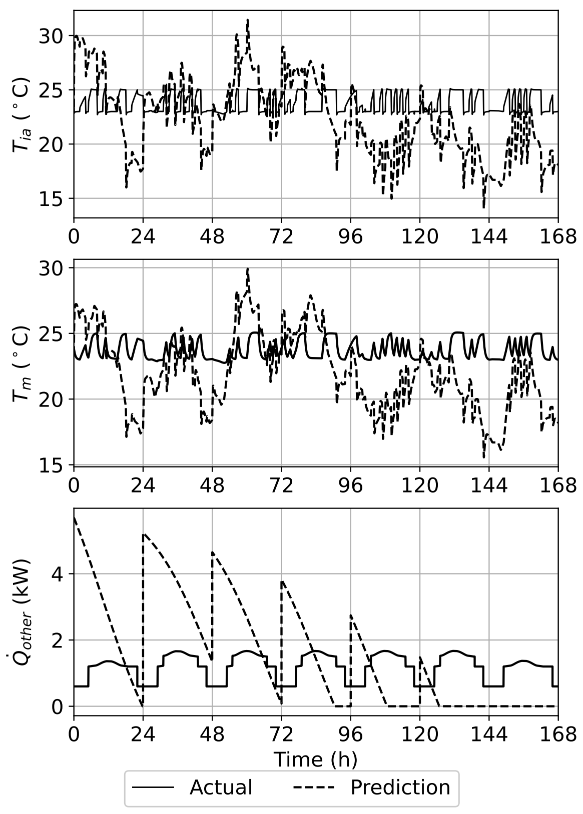


(b)

Figure 4.27: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model trained using the Adam optimizer over (a) training dataset, and (b) validation dataset compared to the actual values

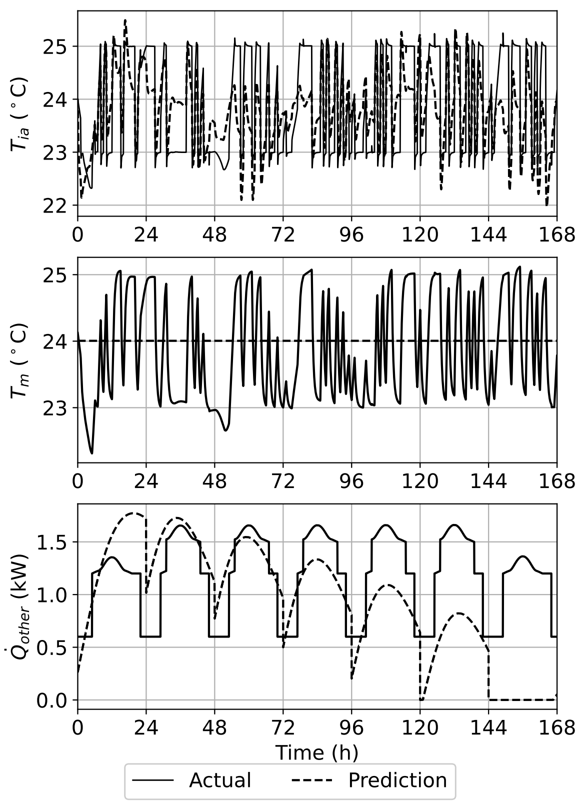


(a)

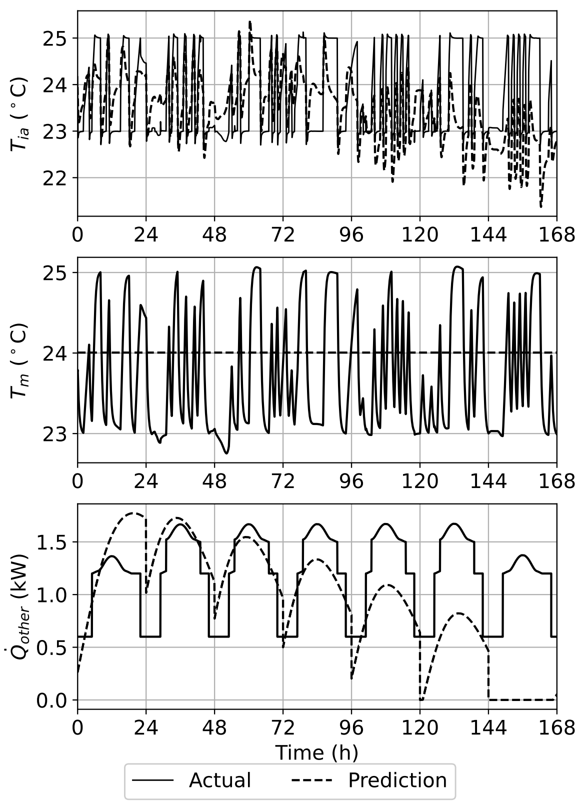


(b)

Figure 4.28: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model trained using the BFGS algorithm over (a) training dataset, and (b) validation dataset compared to the actual values

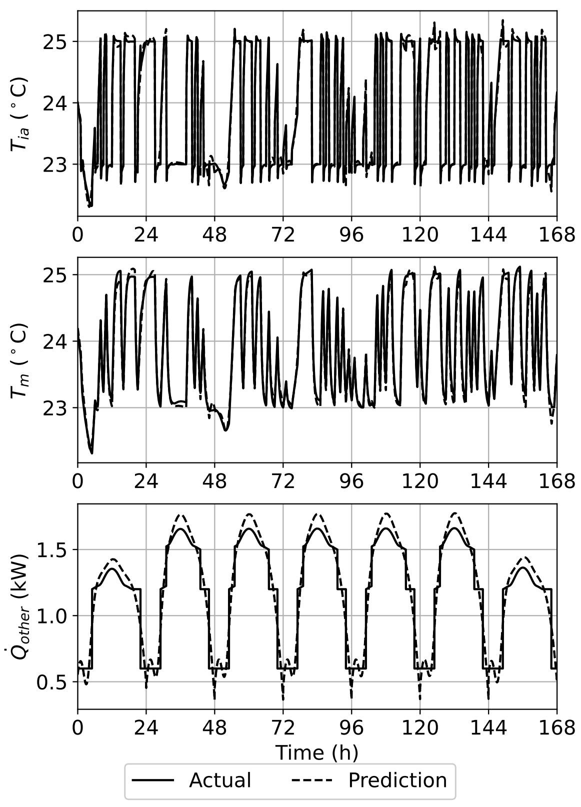


(a)

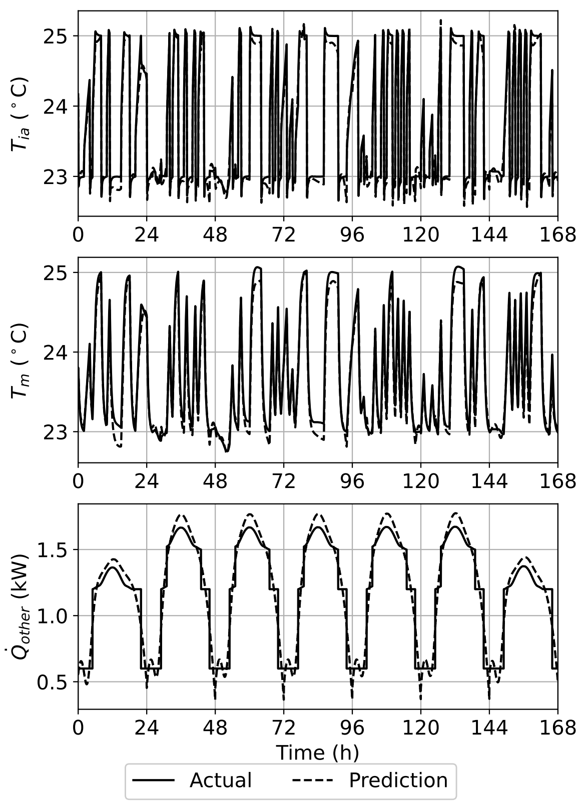


(b)

Figure 4.29: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model trained using the two-step training methodology over (a) training dataset, and (b) validation dataset compared to the actual values



(a)



(b)

Figure 4.30: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model trained using the three-step training methodology over (a) training dataset, and (b) validation dataset compared to the actual values

prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 5°C, 2°C, and 3.9 kW, respectively.

The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model trained using only the BFGS algorithm over the training and validation datasets are provided in Figure 4.28. Over the training period, the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 7.3°C, 7°C, and 3.9 kW, respectively. Over validation period, the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 8°C, 8.2°C, and 3.9 kW, respectively.

The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model trained using the two-step training methodology over the training and validation datasets are provided in Figure 4.29. Over the training period, the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 1.1°C, 2°C, and 1.4 kW, respectively. Over validation period, the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 6°C, 1.5°C, and 1.4 kW, respectively.

The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model trained using the three-step training methodology over the training and validation datasets are provided in Figure 4.30. Over the training period, the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 0.1°C and 0.11°C, and 0.2 kW, respectively. Over validation period, the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 0.11°C, 0.1°C, and 0.21 kW, respectively.

The prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ for models trained using alternative methodologies are about three times higher on average than the prediction errors of the model trained using the three-step training methodology. Additionally, the training loss for the model trained using the three-step training methodology is 7.38×10^{-6} and the validation loss is 8.32×10^{-5}. On the other hand, the training loss for the same model trained using the two-step training methodology is 2.91×10^{-4}, and the validation loss is

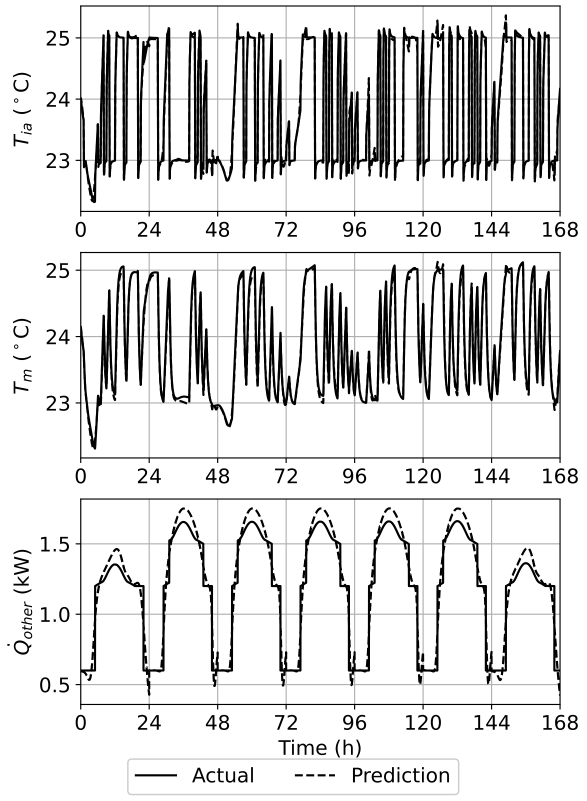
1.73×10^{-3}. The training and validation losses for the model trained using only BFGS is $5.3{\times}10^{-3}$ and $1.01{\times}10^{-2}$, respectively. The training and validation losses for the model trained using only Adam optimizer is $2.739{\times}10^{-4}$ and $4.479{\times}10^{-3}$, respectively. Hence, the three-step training methodology outperforms the single BFGS, single Adam optimizer, and the two-step training methodology in terms of lower maximum prediction errors and lower training and validation loss values.

4.5 Comparison of Model Training with State and Output Measurements

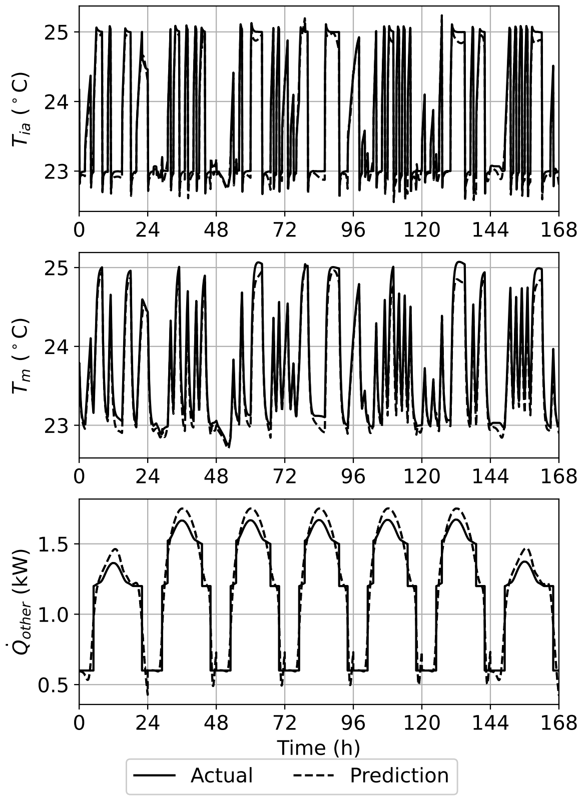
In this section the impact of the unavailability of full state measurements is considered. In this study, output measurements refers to the case when T_{ia} is measured while state measurements refer to the case when both T_{ia} and T_m are measured. Two hybrid building models are formulated such that the first model has output measurements and the second model has state measurements. Models in both cases use TOD, DOW, and WW as predictor variables, and are trained using the three-step training methodology.

The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model with output measurements on the training and validation datasets are depicted in Figure 4.31. Over the training period, the maximum prediction error of T_{ia} is 0.1°C and the maximum prediction error of T_m is 0.11°C. Over the validation period, the maximum prediction error of T_{ia} is 0.18°C and the maximum prediction error of T_m is 0.2°C. The training and validation loss values are depicted in Figure 4.32.

On the other hand, the prediction of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model with state measurements over the training and validation datasets are presented



(a)



(b)

Figure 4.31: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model with output measurements over (a) training dataset, (b) validation dataset compared to the actual values.

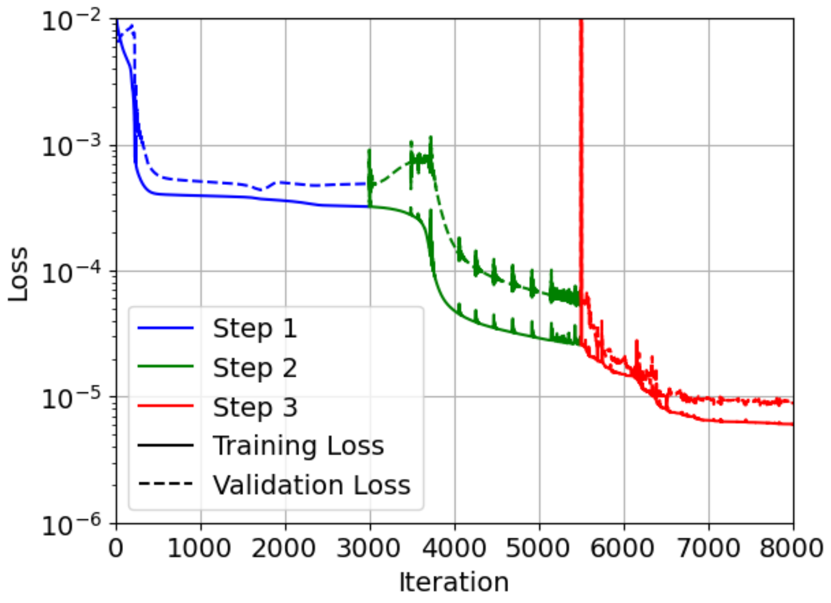
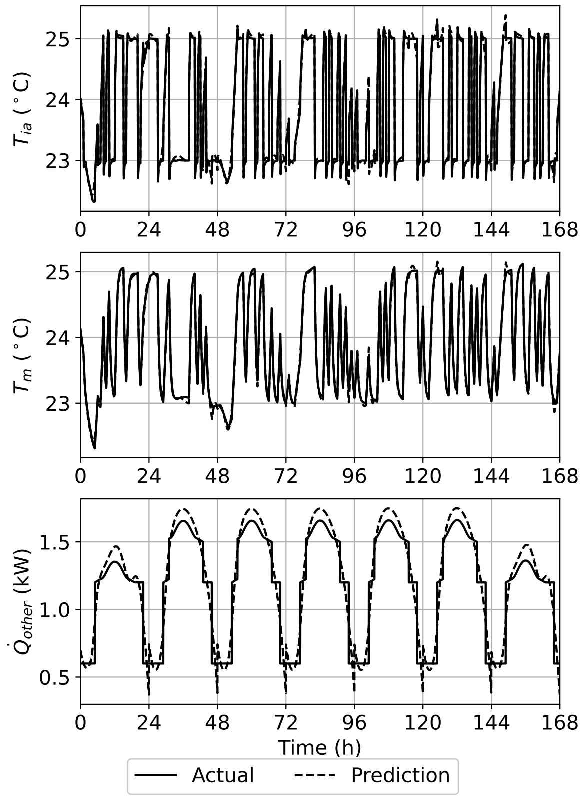
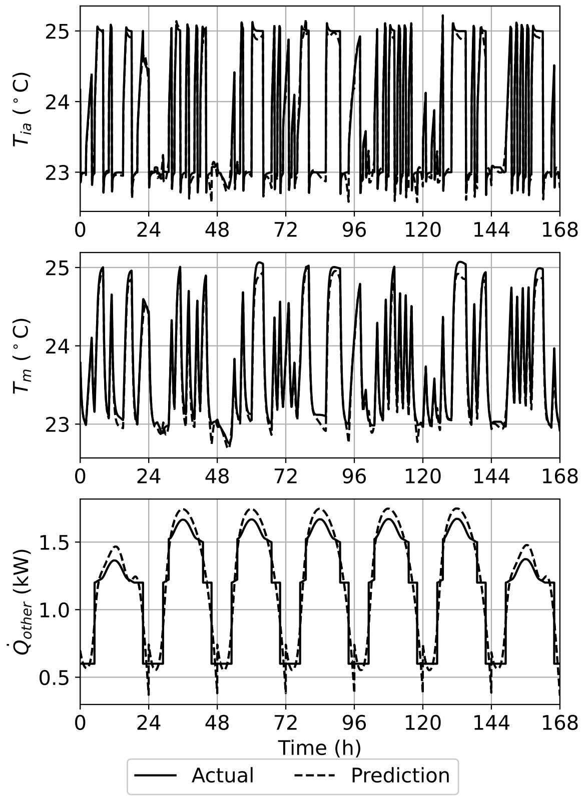


Figure 4.32: Training and validation loss for the hybrid building model with output measurements.



(a)



(b)

Figure 4.33: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the hybrid building model with state measurements over (a) training dataset, (b) validation dataset compared to the actual values.

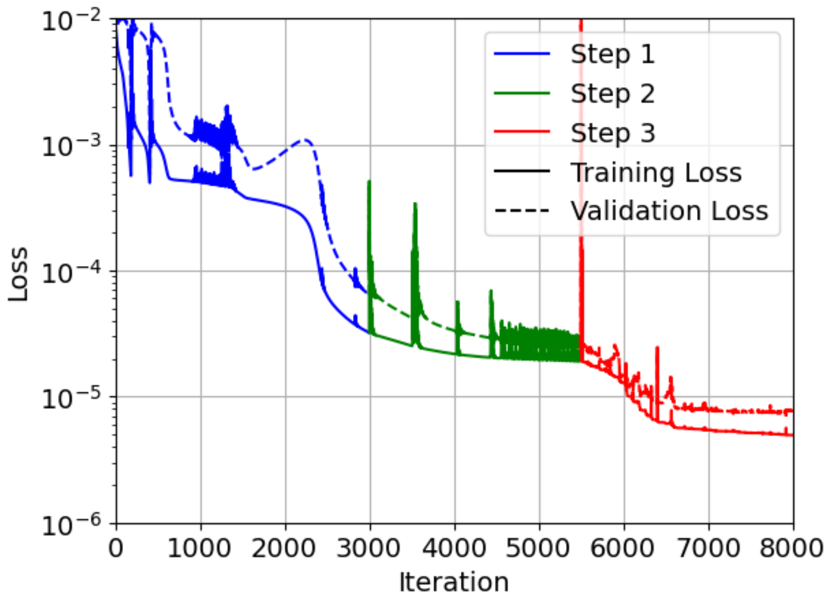


Figure 4.34: Training and validation loss for the hybrid building model with state measurements.

Table 4.5: The actual system parameter values compared to the trained system parameter values of the building models with state and output measurements.

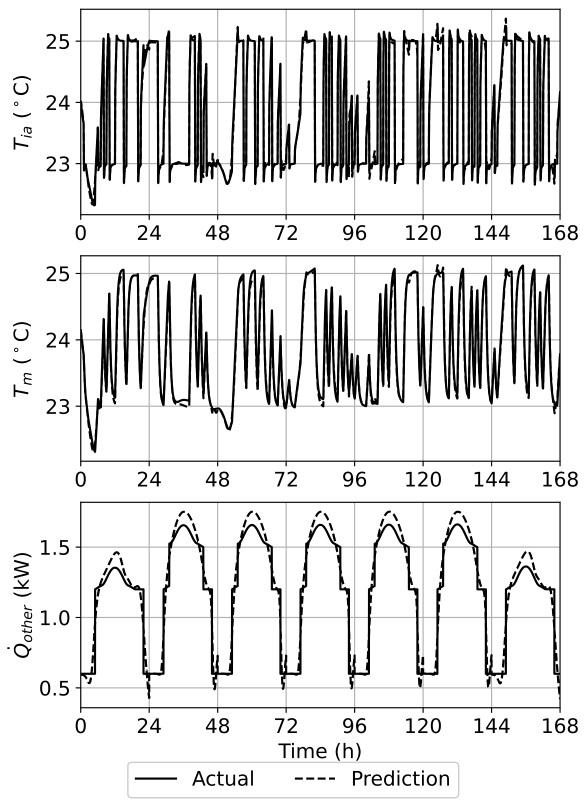
Parameter	Actual Parameter Value	Trained Parameter Value for State Measurement Case	Trained Parameter Value for Output Measurement Case
θ_1	23.40	23.22	23.59
θ_2	1.58	1.50	1.57
θ_3	18.30	18.14	18.36
θ_4	1.82	1.81	1.84
θ_5	0.01	0.01	0.01
Training Loss		3.25×10^{-6}	3.57×10^{-6}
Validation Loss		5.86×10^{-6}	6.88×10^{-6}

in Figure 4.33. Over the training period, the maximum prediction error of T_{ia} is 0.04°C and the maximum prediction error of T_m is 0.07°C. Over the validation period, the maximum prediction error of T_{ia} is 0.07°C and the maximum prediction error of T_m is 0.09°C. The training and validation losses are presented in Figure 4.34. Additionally, the trained system parameter values and the training and validation losses for both the models are presented in Table 4.5. The training and validation losses for models with state measurements are smaller compared to the loss values for models with output measurements.

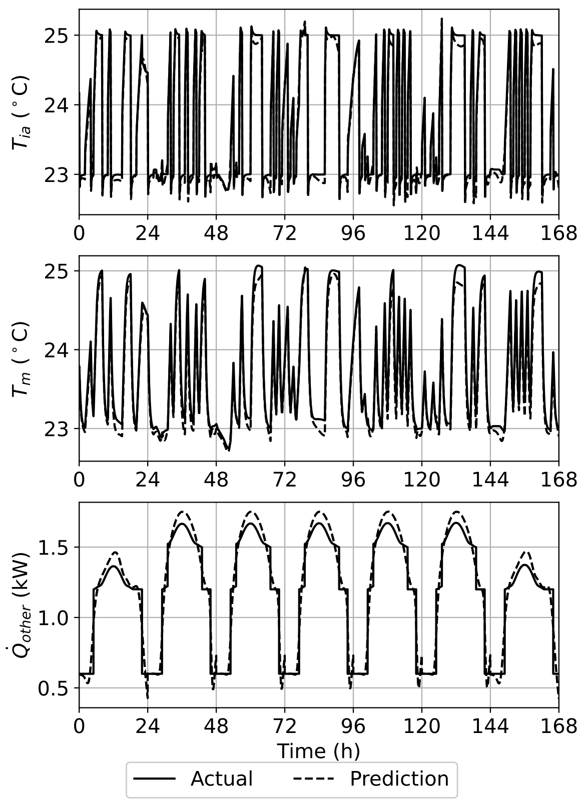
4.6 Reproducibility of Training Results

The aim of this section is to establish the reproducibility of the training results, i.e., if a second model is trained with identical initial parameter estimates as the original model, the predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ are identical over both training and validation periods. Two hybrid building models with TOD, DOW, and WW as predictor variables are trained using the three-step training methodology. Both the hybrid models are provided with the same initial parameter estimates.

The predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the first hybrid building model, over the training and validation periods are provided in Figure 4.35. The predictions of

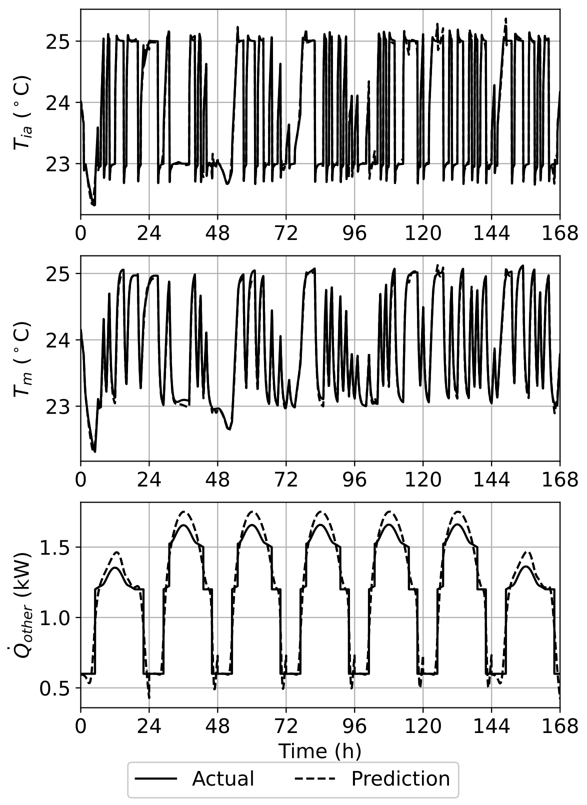


(a)

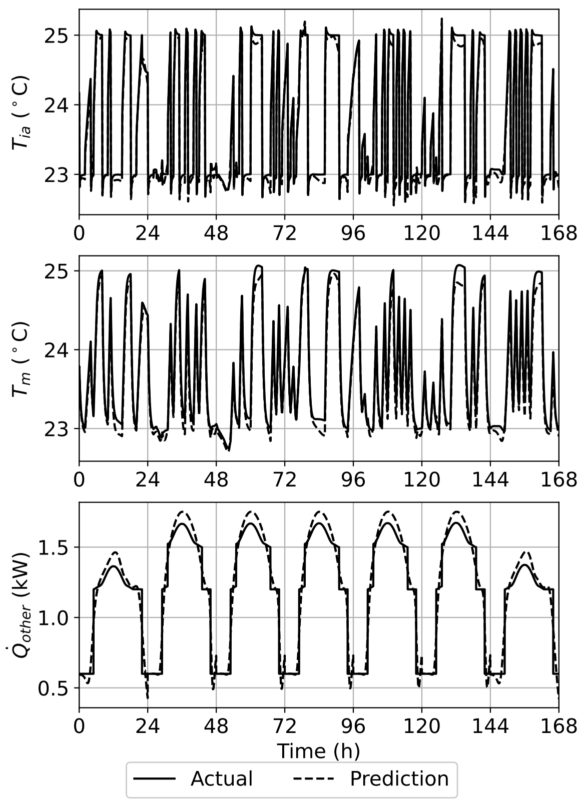


(b)

Figure 4.35: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the first hybrid building model over (a) the training dataset, and (b) validation dataset compared to the actual values.

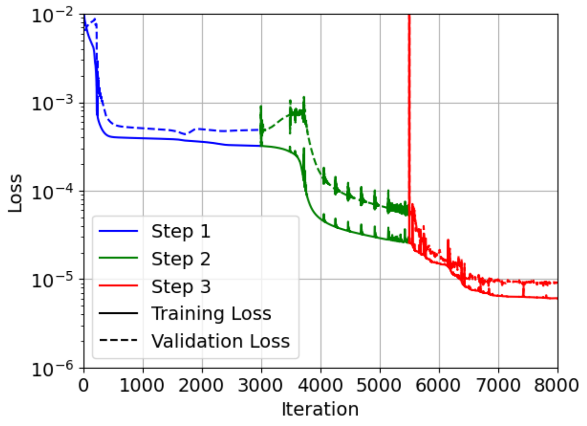


(a)

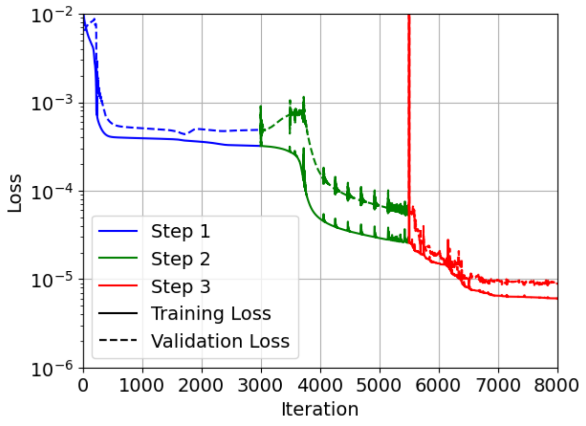


(b)

Figure 4.36: Predictions of T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the second hybrid building model over (a) the training dataset, and (b) the validation dataset compared to the actual values.



(a)



(b)

Figure 4.37: Training and validation loss function values, over the training iterations, corresponding to (a) original hybrid building model, and (b) second hybrid building model.

T_{ia}, T_m, and $\dot{Q}_{other}$ generated by the second model over the training and validation periods are provided in Figure 4.36. Over the validation period, the maximum prediction error of T_{ia}, T_m, and $\dot{Q}_{other}$ are 0.05°C, 0.1°C, and 0.25 kW, respectively, for both the models. Over the training period, the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ are 0.02°C, 0.1°C, and 0.25 kW, respectively, for both models. The loss plots for both the models are depicted in Figures 4.37a and 4.37b. The training losses for both the models are $7.03\ \times 10^{-6}$ and the validation losses are 5.67×10^{-5}. This demonstrates that the second model is a reproduced copy of the original model. The trained parameter values of both the hybrid building models are shown in Table 4.6. The parameter values are identical, therefore, the models can be reproduced when provided with identical initial estimates for the model parameters.

Table 4.6: The actual system parameter values compared to the trained system parameter values of the first and second building models.

Parameters	Actual Parameter value	Trained Parameter Value (Original)	Trained Parameter Value (Reproduced)
θ_1	23.4	22.33	22.33
θ_2	1.58	1.65	1.65
θ_3	18.3	17.66	17.66
θ_4	1.82	1.80	1.80
θ_5	0.0114	0.00	0.00

Chapter 5

Conclusion and Future Work

In this thesis, a generalized hybrid modeling framework to identify models for systems subject to unmeasured time-varying disturbances was presented. The framework utilized a parameterized low-order physics-based model to capture the underlying system dynamics and an FNN to forecast the unmeasured time-varying disturbances. The framework utilized the proposed novel three-step training methodology to simultaneously train both the physics-based and FNN model parameters. Model validation was developed as part of the three-step training methodology. The effectiveness of the hybrid modeling framework was demonstrated by applying it to model the thermal dynamics of a building space. Specifically, the proposed architecture of the FNN was evaluated by forecasting the unmeasured heat disturbances using the FNN with three combinations of predictor variables. The results indicated that the prediction errors of the disturbances were minimal and that the proposed neural network architecture was suitable to forecast the disturbances. Three hybrid building models for each of the three predictor variable combinations were trained using the three-step training methodology and the predictions generated by these models were evaluated. The results indicated that the maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$ were the lowest when the time of the day, the day of the week, and weekday weekend were used as predictor variables. The superiority of the three-step training methodology was then highlighted by comparing the predictions generated by the hybrid building model trained using the three-step training methodology to those trained using single solver strategies and alternative strategies. The three-step training methodology outperformed the alternatives in terms of training and validation loss values and maximum prediction errors of T_{ia}, T_m, and $\dot{Q}_{other}$. The impact of full state measurements was then studied, which resulted in hybrid building models with state measurements having slightly lower training and validation losses compared to the models with output measurements. Finally, the reproducibility of the model training results was established for the hybrid modeling framework. In all cases, the results indicated that the proposed three-step training methodology provided better model predictions, with minimal prediction errors and fewer number of iterations, compared to the predictions made by the same model trained with alternative strategies.

In future work, the hybrid modeling framework and the three-step training methodology can be considered for systems with measurement noise. Regularization techniques may be explored as potential mechanism to increase the robustness of the training process to measurement noise.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] A. Afram, F. Janabi-sharifi, A. S. Fung, and K. Raahemifar. Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system. *Energy and Buildings*, 141:96–113, 2017.

[3] Z. Afroz, GM. Shafiullah, T. Urmee, and G. Higgins. Modeling techniques used in building HVAC control systems: A review. *Renewable and Sustainable Energy Reviews*, 83:64–84, 2018.

[4] C.F. Alvarez and G. Molnar. What is behind soaring energy prices and what happens next? Technical report, International Energy Agency, 2021.

[5] C. Anuntasethakul and D. Banjerdpongchai. Design of supervisory model predictive control for building HVAC system with consideration of peak-load shaving and thermal comfort. *IEEE Access*, 9:41066–41081, 2021.

[6] M. S. F. Bangi and J. S. I. Kwon. Deep hybrid modeling of chemical process: Application to hydraulic fracturing. *Computers & Chemical Engineering*, 134:106696, 2020.

[7] W. A. Beckman, L. Broman, A. Fiksel, S. A. Klein, E. Lindberg, M. Schuler, et al. TRNSYS The most complete solar energy system modeling and simulation software. *Renewable Energy*, 5(1-4):486–488, 1994. Climate change Energy and the environment.

[8] W. Bradley, J. Kim, Z. Kilwein, L. Blakely, M. Eydenberg, J. Jalvin, C. Laird, and F. Boukouvala. Perspectives on the integration between first-principles and data-driven modeling. *Computers & Chemical Engineering*, 166:107898, 2022.

[9] A. R. Coffman and P. Barooah. Simultaneous identification of dynamic model and occupant-induced disturbance for commercial buildings. *Building and Environment*, 128:153–160, 2018.

[10] B. Cui, C. Fan, J. Munk, N. Mao, F. Xiao, J. Dong, and T. Kuruganti. A hybrid building thermal modeling approach for predicting temperatures in typical, detached, two-story houses. *Applied Energy*, 236:101–116, 2019.

[11] J. Dong, T. Ramachandran, P. Im, S. Huang, V. Chandan, D. L. Vrabie, et al. Online learning for commercial buildings. In *Proceedings of the 10th ACM International Conference on Future Energy Systems*, pages 522–530, New York, NY, 25–28 June 2019.

[12] J. Drgoňa, A. R. Tuor, V. Chandan, and D. L. Vrabie. Physics-constrained deep learning of multi-zone building thermal dynamics. *Energy and Buildings*, 243:110992, 2021.

[13] J. Drgoňa, J. Arroyo, I. C. Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, and L. Helsen. All you need to know about model predictive control for buildings. *Annual Reviews in Control*, 50:190–232, 2020.

[14] M. J. Ellis. Machine learning enhanced grey-box modeling for building thermal modeling. In *Proceedings of the American Control Conference*, pages 3927–3932, New Orleans, LA, USA, 2021.

[15] M. J. Ellis, M. J. Wenzel, and R. D. Turney. System identification for model predictive control of building region temperature. In *Proceedings of the 4th International High*

Performance Buildings Conference, pages 3583, 1–10, West Lafayette, IN, USA, 11-14 July 2016.

[16] R. Fletcher. *Practical methods of optimization.* John Wiley & Sons, 2013.

[17] S. Freund and G. Schmitz. Implementation of model predictive control in a large-sized, low-energy office building. *Building and Environment*, 197:107830, 2021.

[18] A. Garnier, J. Eynard, M. Caussanel, and S. Grieu. Predictive control of multizone heating, ventilation and air-conditioning systems in non-residential buildings. *Applied Soft Computing*, 37:847–862, 2015.

[19] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[20] H. Hassanpour, P. Mhaskar, and M. J. Risbeck. A hybrid machine learning approach integrating recurrent neural networks with subspace identification for modelling HVAC systems. *The Canadian Journal of Chemical Engineering*, , 2022.

[21] G. P. Henze. Model predictive control for buildings: a quantum leap? *Journal of Building Performance Simulation*, 6:157–158, 2013.

[22] H. Huang, L. Chen, and E. Hu. A new model predictive control scheme for energy and cost savings in commercial buildings: An airport terminal building case study. *Building and Environment*, 89:203–216, 2015.

[23] International Energy Agency. *Energy Efficiency 2020.* IEA, Paris, 2020. License: CC BY 4.0.

[24] D. Kim, J. Cai, J. E. Braun, and 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*, 167:359–369, 2018.

[25] P. Krishna and M. J. Ellis. Control-oriented hybrid modeling framework for building thermal modeling. In M. Li, editor, *Energy Systems and Processes: Recent Advances in Design and Control*, pages 9–1–9–28. AIP Publishing LLC, 2023.

[26] P. Kumar, J. B. Rawlings, M. J. Wenzel, and M. J. Risbeck. Grey-box model and neural network disturbance predictor identification for economic MPC in building energy systems. *Energy and Buildings*, 286:112936, 2023.

[27] X. Li and J. Wen. Review of building energy modeling for control and operation. *Renewable and Sustainable Energy Reviews*, 37:517–537, 2014.

[28] Y. Li, Z. O'Neill, L. Zhang, J. Chen, P. Im, and J. Degraw. Grey-box modeling and application for building energy simulations - A critical review. *Renewable and Sustainable Energy Reviews*, 146:111174, 2021.

[29] C. G. Looney. *Pattern recognition using neural networks: theory and algorithms for engineers and scientists*. Oxford University Press, Inc., 1997.

[30] J. Ma, J. Qin, T. Salsbury, and P. Xu. Demand reduction in building energy systems based on economic model predictive control. *Chemical Engineering Science*, 67(1):92–100, 2012.

[31] E. T. Maddalena, Y. Lian, and C. N. Jones. Data-driven methods for building control — A review and promising future directions. *Control Engineering Practice*, 95:104211, 2020.

[32] A. D. R. McQuarrie and C.-L. Tsai. *Regression and time series model selection*. World Scientific, 1998.

[33] J. Nocedal and S. J. Wright. Large-scale unconstrained optimization. *Numerical Optimization*, pages 164–192, 2006.

[34] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45:15–27, 2012.

[35] Y. M. Ren, M. S. Alhajeri, J. Luo, S. Chen, F. Abdullah, Z. Wu, and P. D. Christofides. A tutorial review of neural network modeling approaches for model predictive control. *Computers & Chemical Engineering*, 165:107956, 2022.

[36] J. Reynolds, Y. Rezgui, A. Kwan, and S. Piriou. A zone-level, building energy optimisation combining an artificial neural network, a genetic algorithm, and model predictive control. *Energy*, 151:729–739, 2018.

[37] P. W. Tien, S. Wei, J. Darkwa, C. Wood, and J. K. Calautit. Machine learning and deep learning methods for enhancing building energy efficiency and indoor environmental quality – a review. *Energy and AI*, 10:100198, 2022.

[38] U.S. Department of Energy. EnergyPlus Energy Simulation Software. `http://apps1.eere.energy.gov/buildings/energyplus/`, 2011.

[39] U.S. Energy Information Administration. March 2023 monthly energy review. `https://www.eia.gov/tools/faqs`, 2021. Accessed on April 7, 2023.

[40] Z. Zou, X. Yu, and S. Ergan. Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network. *Building and Environment*, 168:106535, 2020.