Hindawi
Journal of Control Science and Engineering
Volume 2017, Article ID 3035892, 18 pages
https://doi.org/10.1155/2017/3035892

Research Article

Online Model Learning of Buildings Using Stochastic Hybrid Systems Based on Gaussian Processes

Hamzah Abdel-Aziz and Xenofon Koutsoukos

Department of Electrical Engineering and Computer Science, Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37235, USA

Correspondence should be addressed to Hamzah Abdel-Aziz; hamzah.abdelaziz@vanderbilt.edu

Received 17 March 2017; Revised 12 June 2017; Accepted 2 July 2017; Published 23 August 2017

Academic Editor: Tushar Jain

Dynamical models are essential for model-based control methodologies which allow smart buildings to operate autonomously in an energy and cost efficient manner. However, buildings have complex thermal dynamics which are affected externally by the environment and internally by thermal loads such as equipment and occupancy. Moreover, the physical parameters of buildings may change over time as the buildings age or due to changes in the buildings' configuration or structure. In this paper, we introduce an online model learning methodology to identify a nonparametric dynamical model for buildings when the thermal load is latent (i.e., the thermal load cannot be measured). The proposed model is based on stochastic hybrid systems, where the discrete state describes the level of the thermal load and the continuous dynamics represented by Gaussian processes describe the thermal dynamics of the air temperature. We demonstrate the evaluation of the proposed model using two-zone and five-zone buildings. The data for both experiments are generated using the EnergyPlus software. Experimental results show that the proposed model estimates the thermal load level correctly and predicts the thermal behavior with good performance.

1. Introduction

Heating, ventilation, and air conditioning (HVAC) in buildings is a major source of energy consumption. Annual reports show that it is the highest cause of energy consumption in residential buildings. Moreover, HVAC along with miscellaneous electric loads accounts for the highest two energy consumption sources in commercial buildings [1]. Therefore, there is a high demand for developing advanced HVAC control methodologies that can reduce the energy consumption of HVAC units without compromising users' comfort. Many of these advanced control methodologies are model-based and rely on accurate thermal models [2].

Developing accurate dynamical models is a challenging task because of the complex behavior of buildings. Thermal dynamics of buildings are stochastic nonlinear dynamics which are affected externally by the environment (e.g., ambient temperature) and internally by interactions between adjacent thermal zones. Additionally, the thermal dynamics of buildings behave differently based on the thermal load (e.g., occupancy) which may not be measured. Developing a unified parametric model for buildings is usually not practical because thermal dynamics differ from one building to another since each building has different construction materials, size, layout, and location. Moreover, the model parameters may change over time as the buildings age or due to changes in the configuration or structure. Such challenges can be addressed using nonparametric modeling approaches based on online model learning.

In this paper, we introduce a novel modeling framework to learn multizone data-driven dynamical models for buildings in an online fashion. The nonparametric nature of the model supports creating unified data-driven models for buildings. Furthermore, we do not assume that the thermal load can be measured, and we estimate it as a latent discrete variable. The proposed approach incorporates the effects of the thermal load, and as a result it increases the model accuracy. Additionally, the online learning adapts the model to the time variability during the system operation. We also include the ambient temperature prediction to incorporate the environment effects in the model.

The main contributions of this work are as follows:

(1) We introduce a nonparametric SHS model based on Gaussian processes which can be used to model complex coupled discrete and continuous dynamics required to develop data-driven models of smart buildings.

(2) We develop an online clustering-based learning methodology for SHS when the discrete dynamics cannot be measured. In the proposed methodology, we use the K-means clustering algorithm to identify the discrete states of the system and to segment the data into each corresponding discrete state. Then, each segment is used to build a distinct GP model to abstract the continuous dynamics. Learning these distinct models is more efficient and mitigates the computation limitations of using a single GP model with every variable as inputs.

(3) We utilize the data-driven modeling approach to provide an accurate thermal model for multizone buildings when the buildings' thermal load is latent. We evaluate the performance, and the efficiency of the proposed model is evaluated using datasets of two-zone and five-zone buildings. In both experiments, we show the model accuracy of estimating the level of the thermal load and predicting the zone air temperature.

The organization of this paper is as follows. Section 2 presents the related work of modeling thermal dynamics and model learning of dynamical systems using GPs. Section 3 illustrates a brief theoretical background of GPs. The proposed model is defined in Section 4 along with the model learning problem. The proposed model learning approach is presented in Section 5. Finally, the experimental results are discussed in Section 6 with the evaluation.

2. Related Work

2.1. Thermal Modeling of Buildings. Developing thermal models for buildings has been investigated extensively in the literature. Many studies proposed linear models for single-zone buildings [3, 4], while other studies use lumped thermal models to approximate multizone buildings as a single-zone model [5, 6]. Although these methods are useful, they typically result in very simple models. On the other hand, multizone models have been investigated to identify the thermal dynamics of each zone in multizone buildings [7–9]. These models are more accurate and they can be used for advanced control design (e.g., model predictive control [8]).

Most of the above-mentioned work relies on parametric models to represent the thermal behavior of buildings. However, parametric models require the building's detailed structure and/or the thermal equations to be known a priori. Also, they typically use linear dynamics and are not very accurate. Recently, nonparametric models have drawn attention to construct accurate nonlinear thermal models. A nonparametric thermal model based on recurrent neural network (RNN) architecture has been developed to learn a compact thermal model for a single thermal zone [10]. Models based on RNNs are very useful to represent the nonlinearity of thermal dynamics; however, they typically do not consider the stochasticity in the thermal dynamics. Moreover, these models require a large set of the training data to learn the model with the desired quality. To mitigate these limitations, a nonparametric probabilistic approach based on Gaussian processes (GPs) has been developed to learn thermal models in an online adaptive learning framework [11]. Both the RNN and the GP models assume that the thermal load of buildings (e.g., occupancy, heating rate from light and equipment) can be measured and they used it as a model input. However, this assumption may not be valid in many real scenarios. Another modeling approach is developed to learn the effect of the thermal load when it cannot be measured [12]. This model is based on using a gray box parametric model for the thermal dynamics and combining it with a latent force model based on Gaussian processes. The latent force model improves the parametric model accuracy by learning the dynamical effects of the latent thermal load.

Existing nonparametric modeling approaches for buildings assume that the thermal load can be measured and used as a model input. This assumption may limit the use of these approaches in many systems; therefore, we propose a nonparametric SHS model for multizone buildings when the thermal load is latent. The proposed model estimates the level of the thermal load and learns a distinct model for each level in order to improve the model efficiency and accuracy.

2.2. Model Learning of Dynamical Systems Using GPs. In this paper, the proposed modeling approach is based on Gaussian processes [13]. Gaussian processes have shown a great success in modeling many modern systems because of their attractive features [14]. GPs are nonparametric probabilistic models which can express the uncertainty in the system dynamics and the model confidence through the model predictive variance. Furthermore, models based on GPs are very simple since they have few hyperparameters and relatively require small datasets to learn the model. These features allow GPs to build efficient, robust, and adaptive dynamical models. GPs have been used recently to develop many time-series models for short-term and multistep forecasting [15, 16]. For instance, GPs are used to build forecast models for the ambient temperature and carbon intensity. These models are used to develop an adaptive heating control algorithm that minimizes the energy cost and carbon emissions [4]. Also, GPs are used to develop a load forecasting model for power systems in [17] and a short-term traffic volume prediction model with a high performance [18]. In addition to time-series models, GPs are used to identify state-space models with a control input in order to support robust filtering, smoothing, and prediction of the system state [19]. For instance, in robotics application, GPs have been used to build a model-based data-efficient learning framework for control policy search, known as PILCO. In this framework, the robot can learn the system dynamics and the control policy efficiently in an online fashion [20].

Gaussian processes have been used widely to model stochastic continuous systems. However, they typically do not consider systems with coupled discrete/continuous dynamics (e.g., SHS). In this paper, we utilize GP models to represent the continuous dynamics of SHS. Furthermore, we consider systems with latent discrete dynamics. This allows us to efficiently integrate the state-space modeling techniques for stochastic continuous systems into the proposed data-driven SHS modeling framework.

2.3. System Identification of SHS. Stochastic hybrid systems (SHS) are dynamical systems that integrate continuous and discrete dynamics. Moreover, the continuous and/or the discrete dynamics exhibit a probabilistic behavior [21]. System identification of hybrid systems (HS) has been investigated in the literature substantially to develop system identification methods for various classes of HS. Typically, these methods aim to estimate the system model parameters for a given model complexity [22]. For HS with unknown model complexity, a kernel-based approach is developed to identify a popular class of HS, known as piecewise affine systems, where GPs are used to model the impulse response of each submodel of the HS [23].

System identification of SHS has an additional level of complexity because of the presence of uncertainty in the model behavior along with the coupled continuous/discrete dynamics. For a given model structure, many methods have been developed to learn the model parameters (i.e., parameter identification) such as simulation-based approaches. Simulation-based approaches use the simulated trajectories to identify the model parameters based on randomized optimization techniques such as Markov chain Monte Carlo (MCMC) [24]. Most of these approaches assume a known model structure with unknown parameters. However, finding the model structure is very crucial in many complex systems. Alternatively, online model learning with reachability analysis framework based on Gaussian processes is developed to learn nonparametric models for SHS with deterministic and known discrete dynamics [25].

Recent nonparametric models for SHS are either limited to deterministic discrete dynamics or developed for offline learning. In this paper, we propose a nonparametric clustering-based modeling framework based on GPs which utilizes sensory data to learn SHS in an online fashion.

3. Background

3.1. Gaussian Process Model. Gaussian processes (GPs) are nonparametric probabilistic models that require only high-level knowledge about the system behavior and use the observed data to model the behavior of the underlying system [13]. Generally, GPs build a Gaussian distribution over functions, by which a function index variable is mapped to an infinite-dimensional function space. GPs are identified by mean and covariance functions. The mean function represents the expected value before observing any data and the covariance function (also called kernel) identifies the expected correlation between the observed data. For instance, the mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ are defined as

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))\left(f(\mathbf{x}') - m(\mathbf{x}')\right)\right]. \end{aligned} \tag{1}$$

The function modeled by the GP can be written as

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \tag{2}$$

We typically use a zero mean function for simplicity and squared exponential (SE) covariance kernel for its expressiveness combined with a noise kernel. Therefore, the mean and covariance functions can be expressed as

$$\begin{aligned} m(\mathbf{x}) &= 0, \\ k(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^T \Lambda^{-1} (\mathbf{x} - \mathbf{x}')\right] \\ &\quad + \delta_{\mathbf{x},\mathbf{x}'} \sigma_\omega^2, \end{aligned} \tag{3}$$

where σ_f is the kernel signal variance, $\Lambda := \operatorname{diag}([l_1^2, \ldots, l_D^2])$ is the characteristic length-scales matrix, δ is the Kronecker delta, and σ_ω is the noise variance. The above GP model builds a probability distribution over the functions $p(f(\mathbf{x}))$ by mapping n-samples $\mathbf{X}$ of a continuous variable $\mathbf{x}$ to a vector of random variable $\mathbf{f}$ with a Gaussian joint distribution, such that

$$p(\mathbf{y}) \sim \mathcal{N}(0, \mathbf{K}(\mathbf{X}, \mathbf{X})), \tag{4}$$

where $\mathbf{K}$ is nD-by-nD covariance matrix generated by (3).

We are interested in the GP posterior distribution given some test inputs and observations (training data). We define the set of test inputs where we want to predict the function value as $\mathbf{X}_*$. After observing data $\mathscr{D}$ and according to (4), the joint distribution of the known $\mathbf{y}$ and the unknown $\mathbf{y}_*$ function values is

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right). \tag{5}$$

Therefore, the posterior distribution $p(\mathbf{y}_* \mid \mathbf{X}_*, \mathbf{X}, \mathbf{y})$ is also a conditional Gaussian distribution with a mean and a covariance given by

$$\begin{aligned} \mathbb{E}[\mathbf{y}_* \mid \mathbf{y}, \mathbf{X}, \mathbf{X}_*] &= \mathbf{K}_*^T \boldsymbol{\beta}, \\ \operatorname{var}[\mathbf{y}_* \mid \mathbf{y}, \mathbf{X}, \mathbf{X}_*] &= \mathbf{K}_{**} - \mathbf{K}_*^T \left(\mathbf{K} + \sigma_\omega^2 \mathbf{I}\right)^{-1} \mathbf{K}_*, \end{aligned} \tag{6}$$

where $\mathbf{K}_* := k(\mathbf{X}, \mathbf{X}_*)$, $\mathbf{K}_{**} := k(\mathbf{X}_*, \mathbf{X}_*)$, $\mathbf{K} := k(\mathbf{X}, \mathbf{X})$, and $\boldsymbol{\beta} := (\mathbf{K} + \sigma_\omega^2 \mathbf{I})^{-1} \mathbf{y}$.

The hyperparameters of the above GP model are defined as $\Theta := (\sigma_f, l_1^2, \ldots, l_D^2, \sigma_\omega)$. The training data ($\mathscr{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, n\}$) are used to identify the model hyperparameters which best represent the training data. The learning process can be expressed as an optimization problem, where the optimal hyperparameters ($\widehat{\Theta}$) maximize the marginal likelihood such that

$$\widehat{\Theta} = \arg\max_{\Theta} \log p\left(\mathbf{y} \mid \Theta, \mathscr{D}\right), \tag{7}$$

where the marginal likelihood is given by

$$p\left(\mathbf{y} \mid \Theta, \mathscr{D}\right) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}\mathbf{y} - \frac{1}{2}\log\left(|\mathbf{K}|\right) - \frac{n}{2}\log\left(2\pi\right). \tag{8}$$

Optimization algorithms based on conjugate gradients have been developed to optimize the hyperparameters [13, 26]. Also, the popular quasi-Newton optimization method for nonlinear functions has been used to learn the GPs' hyperparameters effectively.

4. Stochastic Hybrid Systems

We introduce in this section a nonparametric stochastic hybrid systems (SHS) model based on GPs. To formalize the SHS model, we define $\mathcal{Q}$ as the set of discrete states and denote the continuous state space by $\mathbb{R}^D$ for each discrete state $q \in \mathcal{Q}$ with dimension D. The hybrid state space is defined as $\mathcal{S} := \mathcal{Q} \times \mathbb{R}^D$. For each discrete mode $q \in \mathcal{Q}$, its corresponding continuous dynamics evolve according to a stochastic process modeled by a GP model. The discrete state may also change based on a stochastic process. Furthermore, we consider systems with two types of inputs: (1) control input and (2) external uncontrolled input (disturbance) from the environment. The control input usually affects the system dynamics based on a control policy ($\pi(\mathcal{S}) : \mathcal{S} \to \mathcal{U}$) which maps the hybrid state space ($\mathcal{S}$) into the control input space ($\mathcal{U}$). On the other hand, the external uncontrolled input ($v \in \mathcal{V}$) affects the system dynamics and represents the interaction with the environment. Therefore, we propose to model the external input as a time-series disturbance model ($E : \mathbb{N} \to \mathcal{V}$). The model is formalized by the following definition.

Definition 1 (nonparametric SHS). A nonparametric SHS is defined as a tuple $\mathscr{H} = (\mathcal{Q}, X, \mathrm{Init}, \mathcal{U}, \mathcal{V}, A, \delta)$:

(i) $\mathcal{Q} := \{q_1, q_2, \ldots, q_m\}$, for some $m \in \mathbb{N}$, represents the discrete state space.

(ii) X is a set of continuous variables in the Euclidean space $\mathbb{R}^D$.

(iii) Init: $\mathscr{B}(\mathcal{S}) \to [0, 1]$ is an initial probability measure on the Borel space $\mathscr{B}(\mathcal{S})$ where $\mathcal{S} := \mathcal{Q} \times \mathbb{R}^D$.

(iv) $\mathcal{U} \subset \mathbb{R}^E$, for some $E \in \mathbb{N}$, represents the control input space.

(v) $\mathcal{V} \subset \mathbb{R}^F$, for some $F \in \mathbb{N}$, represents the external uncontrolled input space.

(vi) $\mathscr{A}$ assigns to each discrete state $q \in \mathcal{Q}$ a random function ($x_{k+1} = f_q(x_k, u_k, v_k)$) modeled by a GP which represents the evolution of the continuous state given the predecessor continuous state $x_k \in \mathbb{R}^D$, a control input $u_k \in \mathcal{U}$, and an external uncontrolled input $v_k \in \mathcal{V}$.

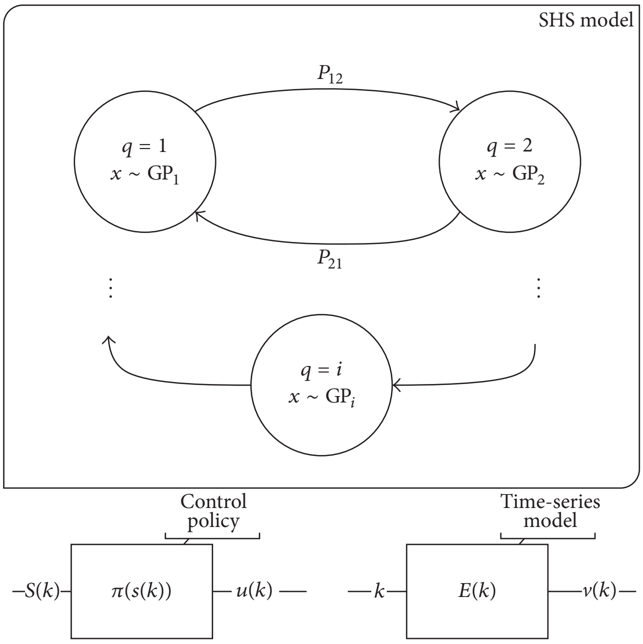


Figure 1: Nonparametric SHS components and interconnections.

A graphical representation of the model is shown in Figure 1.

The goal of model learning is to identify the discrete state space (i.e., $\mathcal{Q} := \{q_1, q_2, \ldots, q_m\}$ and the number of the discrete states m) and the continuous dynamics (i.e., $\mathscr{GP}_q(\cdot)$ for all $q \in \mathcal{Q}$) from a given dataset ($\mathscr{D}$). The data consists of the continuous state, the control input, and the external uncontrolled input. Formally, we can define the training data as

$$\mathscr{D} := \left\{\left(\widehat{\mathbf{x}}_k, \mathbf{y}_k\right) : k = T_s, \ldots, T_e\right\}, \tag{9}$$

where $\mathbf{y}_k$ is the successor continuous state (i.e., $\mathbf{y}_k = \mathbf{x}_{k+1}$), $\widehat{\mathbf{x}}_k$ is defined as $(\mathbf{x}_k, u_k, v_k)$, and $[T_s, T_e]$ is the time period at which the data have been collected. For many complex systems such as buildings, achieving the model learning objective is a challenging task, because the sensory data do not necessarily include information about the discrete state explicitly. For instance, learning the level of the thermal load in buildings is a major challenge because the thermal load cannot be measured in many real scenarios. Moreover, the system dynamics vary over time due to the variability of the system parameters.

In summary, model learning of SHS encompasses the following challenges:

(1) Identifying the discrete state space from data causes a major challenge, because the training data may not have explicit information about the discrete state.

(2) The hybrid dynamics add a level of complexity to the learning algorithm because identifying the continuous dynamics requires segmenting the data for each corresponding discrete state, so that a distinct GP model can be learned using each data segment.

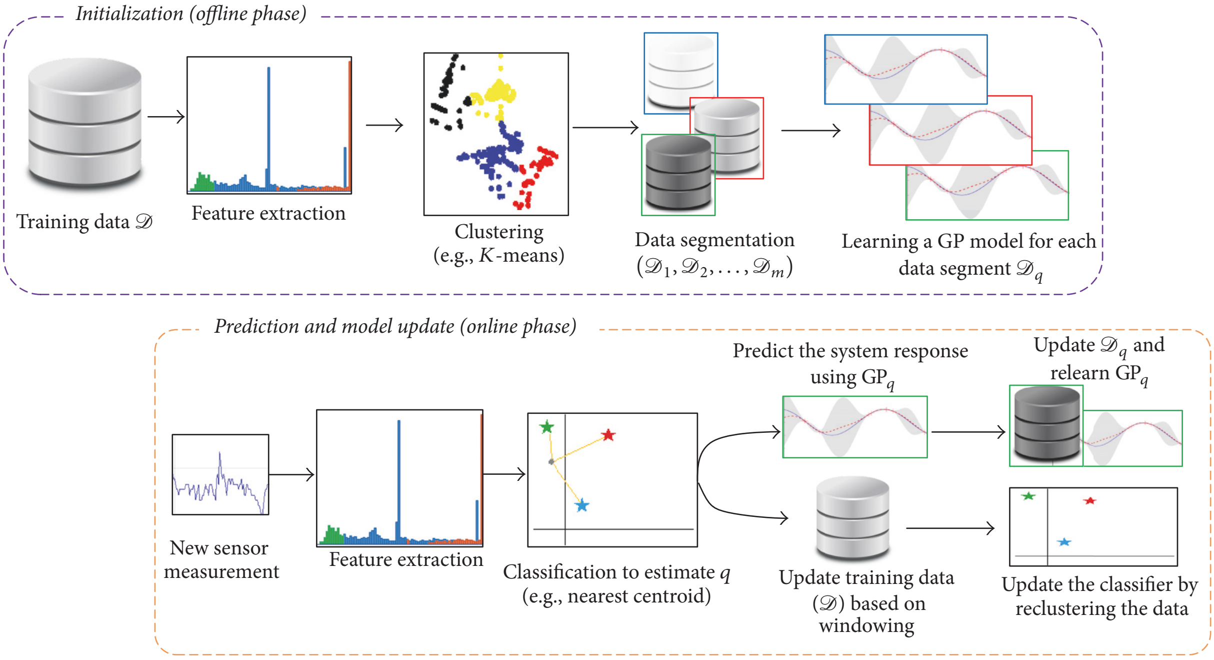

Initialization (offline phase)
Training data $\mathscr{D}$
Feature extraction
Clustering (e.g., K-means)
Data segmentation ($\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_m$)
Learning a GP model for each data segment $\mathscr{D}_q$
Prediction and model update (online phase)
New sensor measurement
Feature extraction
Classification to estimate q (e.g., nearest centroid)
Predict the system response using GP$_q$
Update $\mathscr{D}_q$ and relearn GP$_q$
Update training data ($\mathscr{D}$) based on windowing
Update the classifier by reclustering the data

Figure 2: Clustering-based online model learning for SHS.

(3) As mentioned earlier, the system dynamics depend on an external uncontrolled input v. Therefore, an appropriate time-series model $E(k)$ should be learned and integrated into the SHS modeling framework.

(4) Because of the variability of the system parameters, the learning algorithm must adapt the model to these changes, and therefore the model learning needs to be performed in an online fashion.

5. Online Clustering-Based Model Learning for SHS

In this section, we present a novel online learning approach which can be used to learn a nonparametric SHS model of complex systems with latent discrete dynamics. Moreover, the approach is used for online learning in order to adapt to system changes. The proposed learning approach consists of two phases, offline phase and online phase, as depicted in Figure 2. In the offline phase, we initialize the SHS model by identifying its discrete space first. We determine the number of the discrete states heuristically using Silhouette analysis method, and then we identify the discrete state of each data point by clustering the data. The clustering is also used to label and segment the training data to the corresponding discrete states. This allows us to learn the continuous dynamics for each discrete state using a distinct GP model. In the online phase, we estimate the discrete state in order to determine the GP model used to predict the continuous dynamics. Then, we update the model accordingly. This section provides a detailed discussion of the major steps in the proposed approach.

5.1. Initialization (Offline Phase)

5.1.1. Feature Extraction. Feature extraction is a technique used to transform the training data into a set of features. A set of features (usually referred to as a feature vector) contains the useful data needed for the clustering stage where the irrelevant information is discarded. Feature extraction is needed to distinctively filter the training data. Generally, the feature vector can be computed based on time-domain features (e.g., mean, root mean square) or frequency-domain analysis (e.g., transfer function, Fourier transforms). In this paper, the feature vector is computed based on time-domain features, such as mean and rate of change, because of the simplicity and the efficiency of these features.

5.1.2. Data Clustering for Discrete Space Identification. The goal of data clustering is to estimate the discrete state of each data point. Various clustering algorithms can be used to learn the discrete state such as K-means, Gaussian mixture model (GMM), and hierarchical clustering algorithms. The choice of the appropriate algorithm depends on the nature of the application and the collected data. In this section, we describe the K-means clustering algorithm which calculates the optimal centroids ($\widehat{\mathscr{C}}$) for K clusters, so that $\widehat{\mathscr{C}}$ minimizes the following potential function:

$$\widehat{\mathscr{C}} = \arg\min_{\mathscr{C}} \sum_{g(x)\in\chi} \min_{c\in\mathscr{C}} \left\| g(\hat{\mathbf{x}}) - c \right\|^2, \tag{10}$$

where $\chi \in \mathbb{R}^{n\times d}$ is the feature matrix extracted from the training data ($\mathscr{D}$) with size n and feature dimension d, $g(\hat{\mathbf{x}}) \in \mathbb{R}^d$ is the feature vector for the data point $\hat{\mathbf{x}} \in \mathscr{D}$, and $\widehat{\mathscr{C}} \in \mathbb{R}^{K\times d}$ represents the optimal K centroids.

The clustering-based learning approach considers the data that lie close to each other to probably belong to the same discrete state. However, it requires the number of clusters (i.e., the number of discrete states $m \in \mathbb{N}$) to be known a priori. We identify the number of discrete states m using a heuristic algorithm known as Silhouette analysis method [27]. The Silhouette analysis method determines the best number of clusters ($\widehat{K}$) which results in the best clustering consistency. Silhouette analysis evaluates the clustering consistency for a given K by calculating a scoring coefficient for each clustered data point. The Silhouette scoring coefficient has a range of $[-1, 1]$ where scores near +1 are assigned to data points that lie far from the neighboring clusters. On the other hand, scores near 0 are assigned to data points that lie very close to the boundary between their cluster and a neighboring one. Negative Silhouette scores are assigned to data points which might have been allocated to the wrong cluster. Therefore, clusters with a higher average Silhouette score have better consistency than clusters with a lower average Silhouette score. Formally, the Silhouette score $s(i)$ for a given data point i can be obtained using the following formula:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{11}$$

where $a(i)$ is the average distance from the data point i and the other points in its cluster and $b(i)$ is the minimum, minimized over clusters, average distance between the data point i and other points in a different cluster.

We use Silhouette analysis method to identify the number of discrete states, such that the average Silhouette score is maximized:

$$m = \widehat{K} = \arg\max_{K} \bar{s}(i), \quad K \in G, \tag{12}$$

where G is a finite set of potential values of m.

5.1.3. Data Segmentation and Learning GP Models for the Continuous Dynamics. As shown in the previous subsection, data clustering enables us to identify the discrete states ($\{q_1, q_2, \ldots, q_m\}$) and to label each data point in $\mathcal{D}$ with the corresponding discrete state. The labeled data is used to segment the training data into m datasets:

$$\mathcal{D}_q := \left\{ (\widehat{\mathbf{x}}, \mathbf{y})_i : q = \arg\min_{q' \in \mathcal{Q}} \left\| g(\widehat{\mathbf{x}}_k) - c_{q'} \right\|^2, \ \forall (\widehat{\mathbf{x}}, \mathbf{y})_i \in \mathcal{D} \right\}, \quad \forall q \in \mathcal{Q}, \tag{13}$$

where $c_{q'}$ is the cluster centroid of the discrete mode q' and $g(\widehat{\mathbf{x}})$ is the feature vector of the data point $\widehat{\mathbf{x}}$. We use each data segment $\mathcal{D}_q$ to learn a distinct GP model in order to learn the continuous dynamics for the discrete state $q \in \mathcal{Q}$. As discussed in Section 3, learning a GP model is an optimization process that calculates the optimal hyperparameters $\widehat{\Theta}_q$ of the GP in order to maximize the log likelihood function:

$$\widehat{\Theta}_q = \arg\max_{\Theta_q} \log p\left(\mathbf{y} \mid \Theta_q, \mathcal{D}_q\right). \tag{14}$$

The offline phase learns the SHS model using the initial training dataset, and then the model is updated online whenever a new measurement is received in order to improve the model quality and to adapt to the variation in the underlying system.

5.2. Prediction and Model Update (Online Phase)

5.2.1. Classification for State Estimation/Prediction. During the system operation at each time step k, we classify the new measured data point ($\widehat{\mathbf{x}}_k$) to estimate the current discrete state q_k. The new data point is classified using a nearest centroid classifier. This classifier assigns to the new data point the label of the class with the nearest centroid. The discrete state can be estimated:

$$q_k = \arg\min_{l} \left\| g(\widehat{\mathbf{x}}_k) - c_l \right\|^2, \tag{15}$$

where c_l is the centroid of class l.

5.2.2. Online Learning of SHS with Windowing. Online learning is very useful in order to adapt the model to the variability of the system parameters; however, it requires a proper data selection method to update the training dataset. We use a moving window method to update the model dataset (i.e., $\mathcal{D}_q$ for all $q \in \mathcal{Q}$) based on first-in-first-out (FIFO) policy where the new data point is inserted and the earliest one is dequeued. The dataset ($\mathcal{D}_q$) for each discrete state q is updated independently, and then it is used to update its corresponding ($\mathcal{GP}_q$) model and to reoptimize its hyperparameters ($\widehat{\Theta}_q$). We also use the moving window method to update the training dataset for the clustering (i.e., $\mathcal{D}$), and then we update the classifier centroids accordingly. Both the classifier centroids and the GP associated with the current discrete mode are updated by repeating the learning process of those models using the updated datasets.

We maintain a fixed window size for all the m datasets of the GP models and use a different window size of the training dataset for the clustering algorithm. The sizes of the training datasets affect the model learning running complexity and determine the forgetting weight of the model. Typically, there is a trade-off between selecting large and small dataset size. Online algorithms with large dataset size suffer from high running complexity and they adapt the updated models to the system variability in a slow pace (i.e., low forgetting weight of old information). However, these algorithms can learn models with high quality. On the other hand, online algorithms with small dataset size have efficient running complexity and the updated models adapt to the system variability faster. However, these algorithms may miss useful information from the discarded old data especially in models with large input space. Therefore, we consider the datasets sizes as configuration parameters because they depend on the nature of the modeled system and the requirements and the constraints of the system application. The updated models adapt to the variation in the systems, so that the model accuracy is improved and reflects the behavior of the underlying system.

5.2.3. Prediction of the System Response. We predict the continuous state of the underlying system using the GP model corresponding to the estimated discrete state (i.e., $\mathscr{GP}_{q_k}$). Hence, the predictive distribution of the continuous state can be formalized as

$$p\left(\mathbf{x}_{k+1} \mid \widehat{\mathbf{x}}_k\right) = f_{q_k}\left(\widehat{\mathbf{x}}_k\right) \sim \mathscr{GP}_{q_k}\left(m\left(\widehat{\mathbf{x}}\right), k\left(\widehat{\mathbf{x}}, \widehat{\mathbf{x}}\right)\right), \quad (16)$$

where $\widehat{\mathbf{x}}_k$ is the tuples $(\mathbf{x}_k, u_k, v_k)$. The prediction and the learning steps in the online phase are repeated iteratively each time when new data measurements arrive.

The quality of the learned model is evaluated by measuring the prediction performance. The prediction performance represents the agreement between the model and the physical process output. This comparison can be measured quantitatively using the root mean square error (RMSE) and the mean relative square error (MRSE) metrics. The RMSE and MRSE are defined as

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{N}\sum_{k=1}^{N} e_k^2}, \\ \text{MRSE} &= \sqrt{\frac{\sum_{k=1}^{N} e_k^2}{\sum_{k=1}^{N} y_k^2}}, \end{aligned} \quad (17)$$

where y_k is the process output and $e_k = \widehat{y}_k - y_k$ is the prediction error of the kth time step.

Multistep Prediction within a Discrete Mode. Multistep prediction of the system response is required in many verification and optimization algorithms. For the proposed SHS model, multistep prediction can be achieved by propagating the predictive distribution in (16) using the GP model of the current estimated mode (i.e., $\mathscr{GP}_q$). However, this propagation faces a major challenge where it requires to predict the system response at an uncertain input defined by a Gaussian distribution (i.e., $p(\mathbf{X}_*) \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$). The GP posterior of this predictive distribution can be obtained by

$$p\left(\mathbf{y}_*\right) = \iint p\left(\mathbf{y}_* \mid \mathbf{X}_*\right) p\left(\mathbf{X}_*\right) d\mathbf{y}_* d\mathbf{X}_*. \quad (18)$$

However, the posterior distribution shown in (18) is analytically intractable [16]. To overcome this limitation, we approximate the posterior distribution as Gaussian distribution using a linearization methodology [16, 20], where the approximated posterior $(p(\mathbf{y}_*) \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y))$ can be obtained by

$$\begin{aligned} \boldsymbol{\mu}_y &= \mathbb{E}\left[\mathbf{y}_* \mid \boldsymbol{\mu}_*\right], \\ \boldsymbol{\Sigma}_y &= \text{var}\left[\mathbf{y}_* \mid \boldsymbol{\mu}_*\right] + \mathbf{V}\boldsymbol{\Sigma}_*\mathbf{V}^T, \end{aligned} \quad (19)$$

where $\mathbb{E}[\mathbf{y}_* \mid \boldsymbol{\mu}_*]$ and $\text{var}[\mathbf{y}_* \mid \boldsymbol{\mu}_*]$ is the mean and covariance of the GP posterior calculated at the mean $\boldsymbol{\mu}_*$ of the input distribution as in (6) and $\mathbf{V}$ is defined by

$$\mathbf{V} = \frac{\partial \boldsymbol{\mu}_y}{\partial \boldsymbol{\mu}_*} = \beta^T \frac{\partial k\left(\mathbf{X}, \boldsymbol{\mu}_*\right)}{\partial \boldsymbol{\mu}_*}. \quad (20)$$

5.3. Learning Time-Series Model for the Uncontrolled Input. As mentioned earlier, the proposed approach identifies a time-series model $E(k)$ for the uncontrolled input v using single Gaussian processes model. The time-series model is learned online and independently of the SHS model learning algorithm. Therefore, the time-series model $E(k)$ of an observed time-dependent variable v_k is modeled as

$$v_k = f\left(k\right) \sim \mathscr{GP}\left(m\left(k\right), K\left(k, k'\right)\right). \quad (21)$$

We also use the moving window method to learn the above time-series model online.

6. Evaluation

In this section, we evaluate the performance of the proposed framework and its efficacy to learn thermal models for buildings when the applied thermal load (e.g., occupancy) cannot be measured. Thermal models are used to capture the dynamics of the buildings' air temperature, which are essential for developing many advanced control methodologies [2]. In this case, we use the proposed SHS to build a thermal model for buildings such that the zone air temperature represents the model continuous state, the HVAC heating/cooling rate represents the control input, the ambient temperature represents the uncontrolled input, and the thermal load represents the latent discrete state.

We evaluate the thermal model learning for (1) a two-zone data center building and (2) a five-zone office building. The actual behaviors for both buildings are generated using simulations based on the EnergyPlus software [28]. EnergyPlus is an open-source cross-platform building energy simulator engine funded by the U.S. Department of Energy (DOE) and Building Technologies Office (BTO) and managed by the National Renewable Energy Laboratory (NREL). EnergyPlus is used by engineers, architects, and researchers for high fidelity simulation of buildings. EnergyPlus requires two inputs: (1) the ambient temperature and the environment data and (2) the building description. The building description defines its structure and layout, the construction materials, the thermal zones with their dimensions and area, the HVAC system, the control strategies, and more. It also defines the building thermal loads with their schedules such as occupancy, lights, and electrical equipment. These detailed descriptions are used to construct several models (e.g., airflow network model, pollution model, and on-site power model) by which EnergyPlus simulates the building behavior.

We use EnergyPlus to represent the system response and use the data collected from EnergyPlus simulation (the building state and the control input values) to learn and to evaluate the proposed model learning framework. We also use the weather data used by EnergyPlus to learn a time-series model in order to forecast the uncontrolled input. At each time step k, we collect the system state (the zone air temperatures) and the control input (the applied heating/cooling rate from the HVAC unit). The collected data are then used to learn/update the model online as described earlier. Moreover, we compare the model prediction against

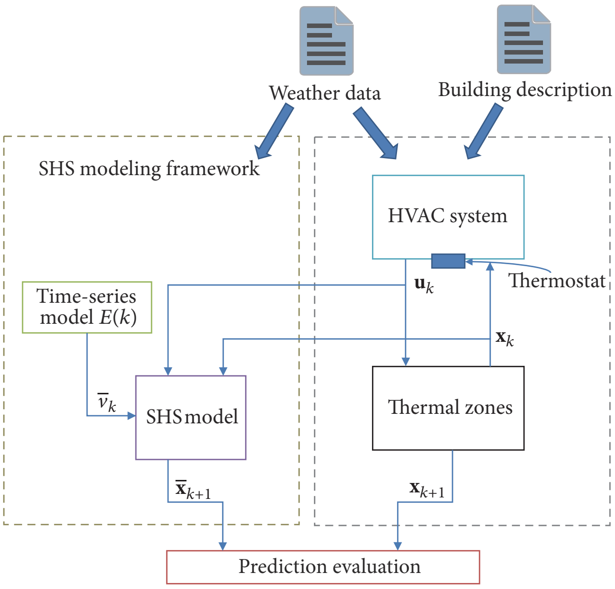


Figure 3: Block diagram of the experiment setup.

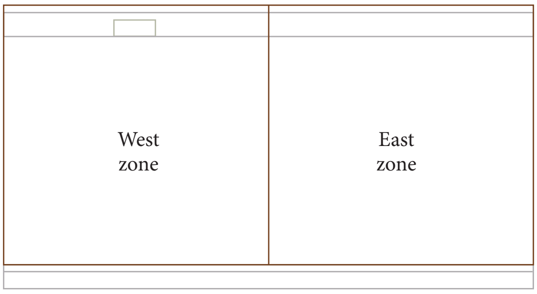


Figure 4: Two-zone data center building.

Table 1: The average Silhouette scores for different numbers of clusters.

m	Average Silhouette score	
	West zone	East zone
2	0.70	0.55
3	0.71	0.65
4	0.62	0.60
5	0.65	0.65

the system response (i.e., zone air temperatures for the next time step $k+1$) in order to evaluate the model learning performance. A general block diagram of the experiment setup is shown in Figure 3.

The proposed framework has been implemented using MATLAB® and Statistics and Machine Learning Toolbox Release 2016a [29]. Gaussian processes models are learned using a quasi-Newton optimization algorithm in order to optimize the model hyperparameters. The following subsections discuss the experiment results for a two-zone data center building and a five-zone office building.

6.1. Two-Zone Data Center Building. In this experiment, we use a dataset of a two-zone data center building to learn and to evaluate the building thermal model using the proposed SHS modeling framework. This dataset is synthetic data generated by EnergyPlus and is used to evaluate buildings' modeling in [10]. The data center building consists of two zones, the west zone and the east zone, as shown in Figure 4. The dataset contains hourly data for one-year simulation. Each data point consists of zone air temperature, ambient temperature, thermal load heating rate from IT equipment, lights, and electrical equipment, and HVAC unit cooling rate. The heating rate from the building thermal load depends on the building activities (e.g., how utilized or idle the IT equipment is). Many models in the literature assume that the thermal load can be measured and therefore can be used as a model input (e.g., [10]); however, this is very expensive in real data centers. Therefore, we consider the thermal load as a latent discrete state of the system and we use our proposed approach to estimate it from the measurable thermal data (i.e., zone temperature and HVAC unit cooling rate).

To formalize the model, we define the SHS model as follows: the continuous state ($\mathbf{x} \in \mathbb{R}^2$) represents the air temperature for both zones, the discrete state q represents the thermal load level, the uncontrolled input ($v \in \mathbb{R}$) represents the ambient temperature, and the control input ($u \in \mathbb{R}$) represents the HVAC cooling rate. Therefore, the predictive distribution of the zone air temperature can be represented as

$$\mathbf{x}_{k+1} \sim f_{q_k}\left(\mathbf{x}_k, u_k, v_k\right) : q_k \in \mathcal{Q}. \tag{22}$$

In the offline phase of the model learning approach, we used the first four weeks of data (i.e., 672 data points) to initialize the model and to learn its discrete states using the K-means clustering algorithm. Data clustering starts by extracting the time-domain features from the data. In this experiment, the features are the average cooling rate (i.e., $u(k)$) and the zone air temperature difference (i.e., $\Delta x = x_k - x_{k-1}$) (note that the values of these features are normalized in order to unify their scale). Then, the number of discrete states is estimated using the Silhouette analysis method. The calculated average Silhouette scores of this experiment are shown in Table 1. Based on this analysis, the number of the discrete states is three (i.e., $m = 3$). Since the discrete state represents the thermal load level, we identified three thermal load levels which correspond to low, medium, and high heat gained from the thermal loads. Figure 5 depicts the three clusters of the training data for each zone. Finally, we segment the data into three datasets and use them to learn three distinct GP models for each zone.

As explained earlier, the ambient temperature is considered as an uncontrolled input, and thus a time-series GP model is used to forecast its value for the next hour. We also update the forecast model every hour when we receive a new measurement. Figure 6 shows the forecasting results of the ambient temperature for three days.

In the online phase, we predict the zone temperature for the next hour using the learned SHS model and the predicted ambient temperature. Further, we update the training datasets every hour when we receive a new data point using

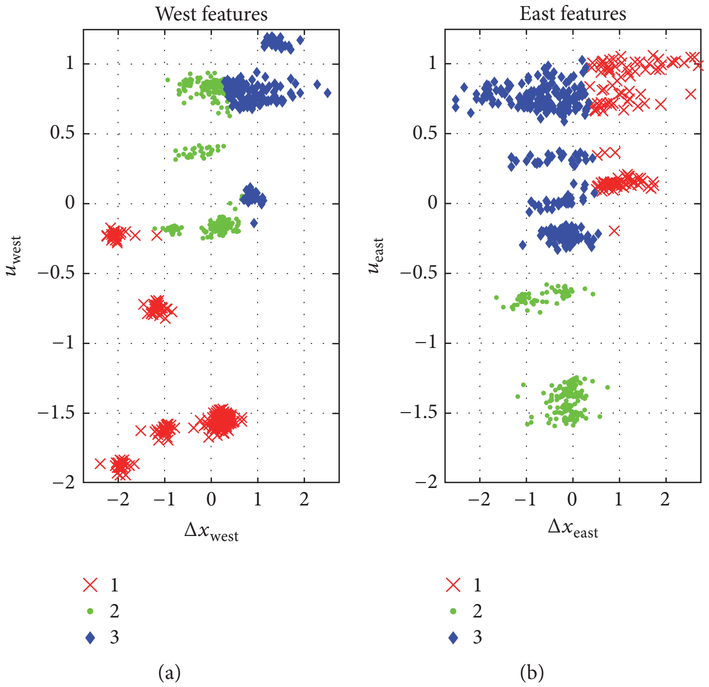


Figure 5: Clustering of the training data using K-means algorithm.

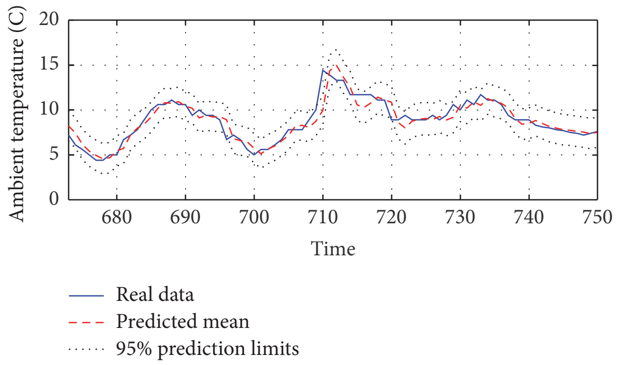


Figure 6: Prediction of ambient temperature using a time-series GP model.

the moving window method with a size of one-week data (i.e., 168 points). The updated datasets are then used to relearn the corresponding models. We run the prediction/learning steps iteratively for almost 11 months and use the results to evaluate the proposed SHS model learning approach. The results for the first three days of the discrete mode estimation (i.e., thermal load level) and the west zone temperature prediction are shown in Figures 7 and 8, respectively. The depicted results present the performance of the online model learning approach to predict the continuous state accurately and to estimate the discrete state successfully. As shown in Figure 8, we compare the model prediction with a full GP model. The full GP model assumes that the thermal load can be measured and the thermal model is defined as

$$\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, u_k, v_k, l_k\right), \tag{23}$$

Table 2: Performance metrics for the prediction of the zone air temperature, with a comparison between the SHS model and the full GP model.

	West zone		East zone	
	Full GP	SHS	Full GP	SHS
RMSE	0.05	0.06	0.07	0.06
MRSE	0.008	0.009	0.01	0.009
Max.	0.6	1.03	0.6	0.8

where l_k is the total heating rate from the thermal load. As shown in the results, the proposed SHS model and the full GP model have a similar performance. However, the learning of the full GP is computationally more expensive because of the large dimension of the input data. On the other hand, SHS model segments the data into smaller distinct models for each estimated level of the thermal load.

We evaluated the prediction performance of the continuous state using the root mean square error (RMSE) and the mean relative square error (MRSE) metrics, defined in (17). The error metrics statistics for the SHS model and the full GP model are shown in Table 2. The results show that our SHS model predicted the zones' temperature with a good performance. Also, the performance of the SHS model is similar to that of the full GP model which assumes that the thermal load is known and can be measured.

To evaluate the improvement of the online learning approach, we compared the RMSE metric for the proposed online learned model against another offline model, which is learned once offline only. In this experiment, we considered

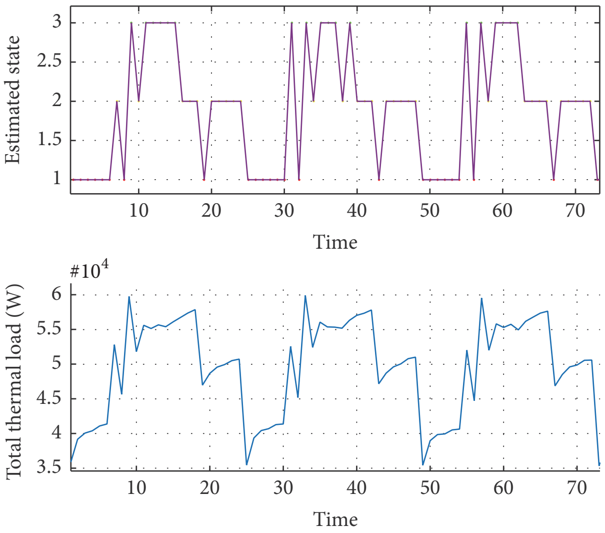


Figure 7: Discrete state estimation of the west zone versus the actual total thermal load.

the changing of the building climate due to seasons changing as an example to represent the variability in the system. Figure 9 shows the RMSE statistics for both models calculated for each week of the year. The results indicate that online learning allows our model to adapt to the variation in the system with a good performance. On the other hand, the offline model failed to adapt to these variations.

6.1.1. Evaluating Different Clustering Algorithms. There are many clustering algorithms which can be used to estimate the mode of the system. The choice of such algorithm depends on the nature of the data and the application. In this experiment, we studied three different clustering algorithms which are K-means, Gaussian mixture model (GMM), and hierarchical clustering algorithms. Figure 10 shows the clustering of the training data using each algorithm. Both K-means and GMM algorithms require the number of the clusters to be given a priori, where we used Silhouette analysis method to estimate it. Hierarchical algorithm, on the other hand, has the advantage of determining the number of the clusters based on its stopping criteria (e.g., cluster inconsistency), but it has many design parameters and suffers from run time complexity $O(n^2 \log(n))$ for large-scale data as indicated in Figure 11.

Figure 12 shows the boxplot of the heating rate caused by the thermal load (i.e., light, occupancy, and equipment) for each estimated discrete mode (i.e., cluster). Despite the differences between these algorithms, the results show that they identify the discrete state with distinctive levels of the thermal load. GMM estimated the thermal load levels with the lowest accuracy (i.e., low, high). Hierarchical algorithm has the highest accuracy (i.e., low, medium-low, medium-high, and high) of the thermal load levels.

6.1.2. Multistep Prediction within the Same Mode. As described earlier, multistep prediction requires propagating the uncertainty of the predictive distribution for each time step. In addition, the control policy of the HVAC is required to predict the control sequence of the HVAC unit (i.e., cooling rate). To do so, we used GP to learn the control policy of the HVAC as a function of the hybrid system state (i.e., zone air temperature and discrete mode) and the ambient temperature; that is,

Table 3: Performance metrics for multistep prediction of both SHS model and unimodal GP model.

	SHS model		Unimodal GP	
	West	East	West	East
RMSE	0.23	0.3	0.56	0.3
MRSE	0.009	0.013	0.023	0.013
LD	51	6	27	257

$$\mathbf{u}_{k+1} \sim f(\mathbf{s}_k, \nu_k) : \mathbf{s}_k = (q_k, \mathbf{x}_k). \tag{24}$$

Furthermore, we use the RMSE and MRSE metrics as a performance measure in order to evaluate the predicted mean. We also use another performance metric known as log predictive density (LD) to evaluate the prediction performance of the predicted mean and the predicted variance as well [14]. LD is defined as

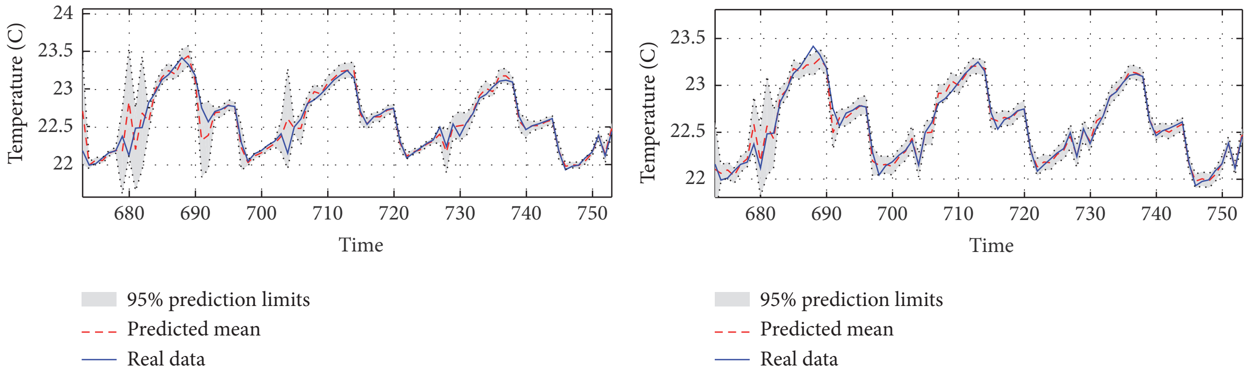
$$\mathrm{LD} = \frac{1}{2}\log(2\pi) + \frac{1}{2N}\sum_{i=1}^{N}\log(\sigma_i^2) + \frac{e_i^2}{\sigma_i^2}, \tag{25}$$

where σ_i is the prediction variance in ith step and e_i is the error between the system output and the predicted mean. Figure 13 shows the multistep prediction of the zone air temperature for both west and east zones using the proposed SHS model. We also implemented the multistep using a typical single GP model which does not estimate the discrete mode of the buildings (i.e., thermal load level). Figure 14 shows the multistep prediction of the zone air temperature for both west and east zones using a typical unimodal GP model. Table 3 shows the performance metrics statistics for the multistep prediction within the same mode of the SHS and the unimodal GP.

The performance metric shows that the SHS model provides a more accurate prediction than the unimodal GP as far as the system does not change its discrete mode. Moreover, the variance of the unimodal GP model is falsely optimistic.

6.2. Five-Zone Office Building. In this experiment, we evaluate the scalability of the framework using a five-zone office building dataset. The dataset is a synthetic dataset generated by EnergyPlus for a single story office building. The office building is a rectangular building with five zones and a return plenum. The building has four windows in each facade, and there are two interior glazed doors between the west and the core zone and between the east and the core zone, as shown in Figure 15.

The main thermal sources for all the five zones are the HVAC unit heating and cooling supply air, the office lights and equipment, and the office occupancy. The dataset measures the building thermal behavior hourly for one year using



(a) (b)

Figure 8: The predicted temperature of the west zone using (a) the proposed SHS model and (b) the full GP model.

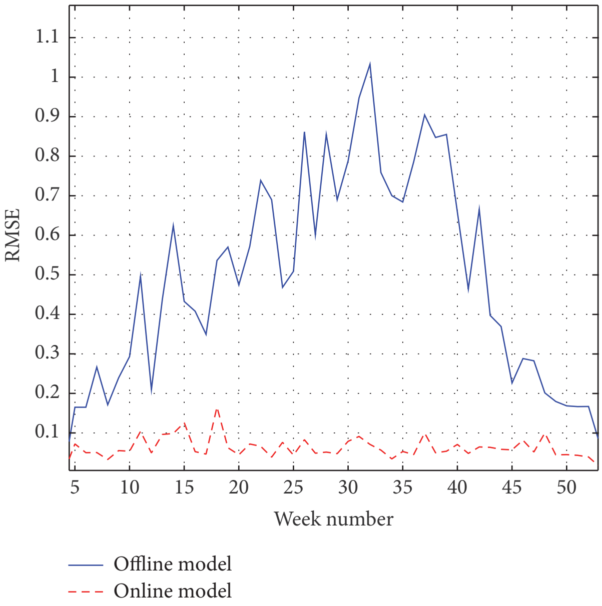


Figure 9: Error metric (RMSE) of the prediction of the zone air temperature for both the online and the offline learned SHS model.

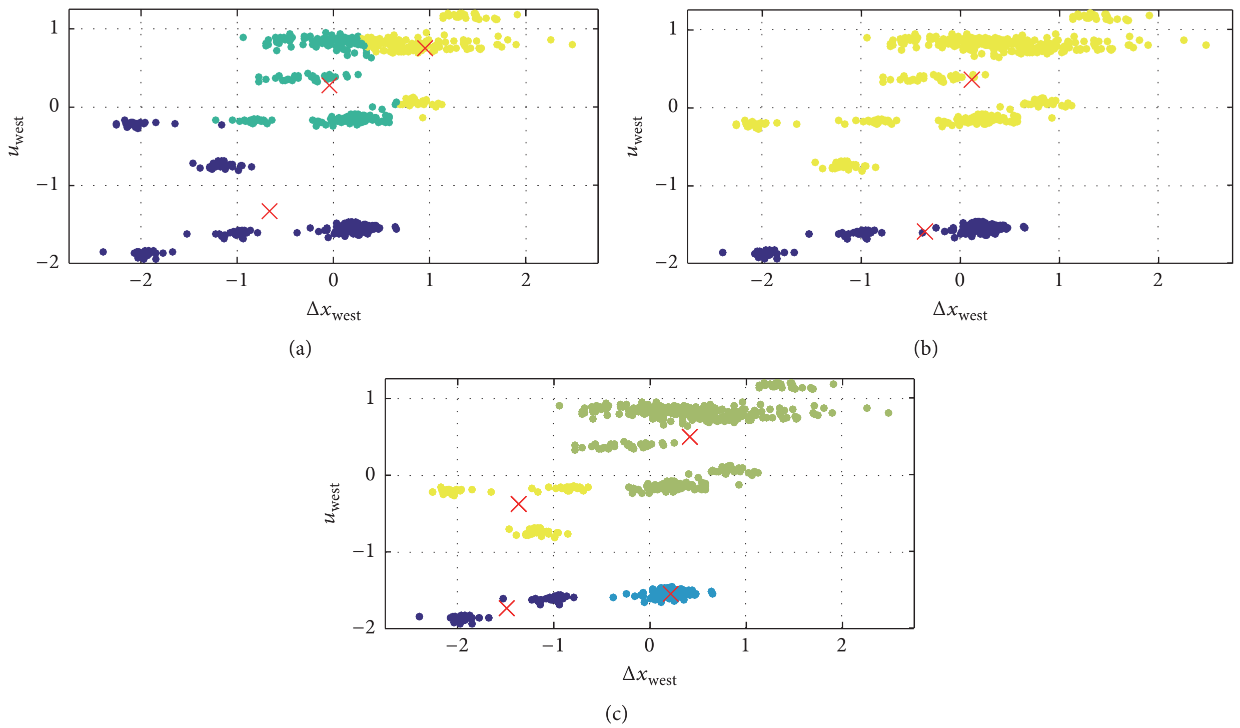
weather data of Chicago, IL. The measurements consist of the ambient temperature, zone air temperatures, cooling/heating rate from the HVAC unit, and heating rate from the thermal load (lights, occupancy, and office equipment) aggregated and averaged for every hour.

The thermal model of the building is represented by the following SHS model: the continuous state ($\mathbf{x} \in \mathbb{R}^5$) represents the zone air temperatures. The discrete state q represents the latent thermal load. The uncontrolled input ($v \in \mathbb{R}$) represents the ambient temperature. Lastly, the control input ($u \in \mathbb{R}$) represents the HVAC heating/cooling rate. Therefore, the predictive distribution of the zone air temperatures is given by

$$\mathbf{x}_{k+1} \sim f_{q_k}(\mathbf{x}_k, u_k, v_k) : q_k \in \mathcal{Q}. \tag{26}$$

Like the two-zone building, we also use the first four weeks of data (i.e., 672 data points) to initialize the model and learn its discrete space (i.e., m) and use the heating/cooling rate (i.e., $u(k)$) and the zone air temperature difference (i.e., $\Delta x = x_k - x_{k-1}$) as time-domain features for the K-means clustering algorithm. We use the extracted feature to cluster the training data and to identify the discrete states based on the Silhouette analysis. The estimated number of the discrete states of the south, the east, the west, and the core zones is two and for the north zone is three. Figure 16 shows the clustering of the training dataset for the south and the north zones and Figure 17 shows the clustering results of the west, the core, and the east zones.

In the online phase, we use the learned model to predict the zone air temperatures for the next hour. Also, we update the training data and its corresponding models every hour when we receive new data points. We run the prediction/learning steps iteratively for the rest of the data (about 11 months). The results for the discrete mode estimation (i.e.,



(a)

(b)

(c)

Figure 10: Clustering of the training data for the west zone using (a) K-means algorithm, (b) GMM algorithm, and (c) hierarchical algorithm.

Table 4: Prediction error statistics per zone

	South zone		East zone		North zone		West zone		Core zone	
	Full GP	SHS	Full GP	SHS	Full GP	SHS	Full GP	SHS	Full GP	SHS
RMSE	0.32	0.28	0.25	0.36	0.28	0.29	0.28	0.27	0.22	0.11
MRSE	0.11	0.05	0.08	0.06	0.10	0.05	0.1	0.05	0.6	0.02
Max.	3.5	6.3	5.5	10.1	12.1	4.5	6.7	4.5	2.5	3.2

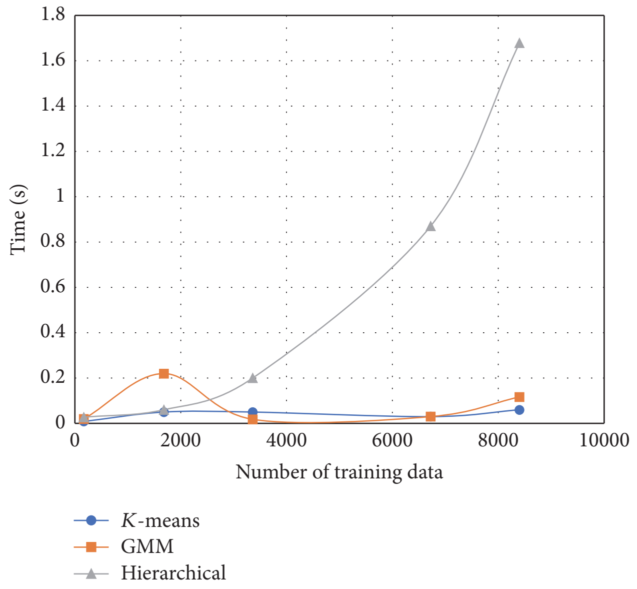
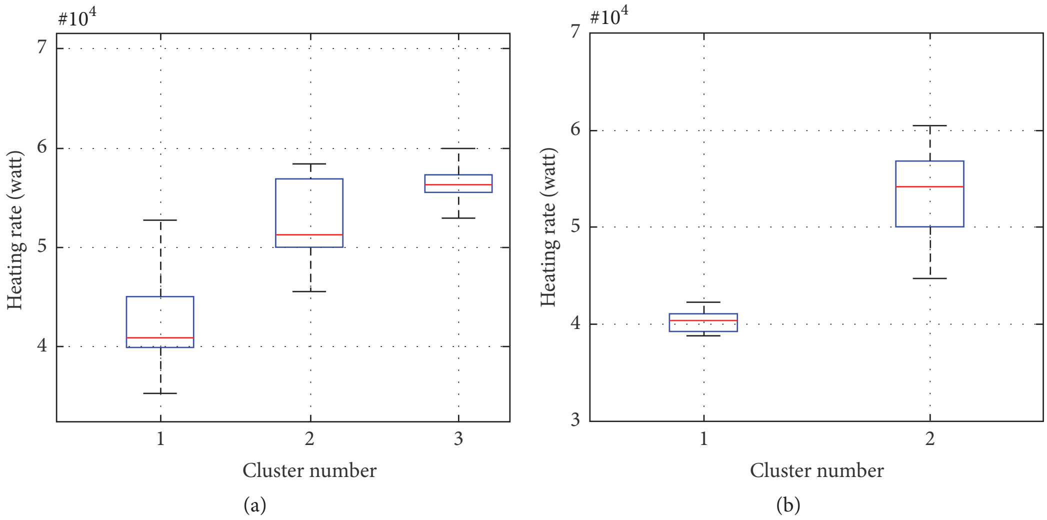


Figure 11: Clustering time for different data sizes of K-means algorithm, GMM algorithm, and hierarchical clustering algorithm.

thermal load level) and zone air temperature prediction of the north zone are shown in Figures 18 and 19, respectively. Typically, office environments have two modes, busy and idle, because people tend to go to their offices during the business hours and then the building becomes almost idle during the nights and holidays. The model learning approach estimated this office behavior successfully as shown in the results.

To evaluate the prediction performance, we use the root mean square error (RMSE) and the mean relative square error (MRSE) metrics defined in (17). The prediction error evaluation statistics are shown in Table 4. These results show that our SHS model predicted the zone air temperatures with a good performance where the average prediction error is less than or equal to 6%. Further, the proposed model shows a similar performance to the full GP model which assumes that the thermal load is known and can be measured.

6.3. Computation of Efficiency and Scalability of the Learning Algorithm. Despite the attractive features and properties of GPs, the running time becomes a major factor in the GP



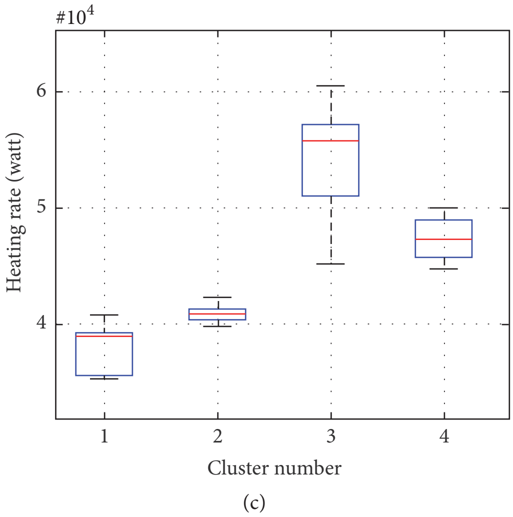


Figure 12: Average heating rate caused by west zone's thermal load for each discrete mode (in watt) estimated by (a) K-means algorithm, (b) GMM algorithm, and (c) hierarchical algorithm.

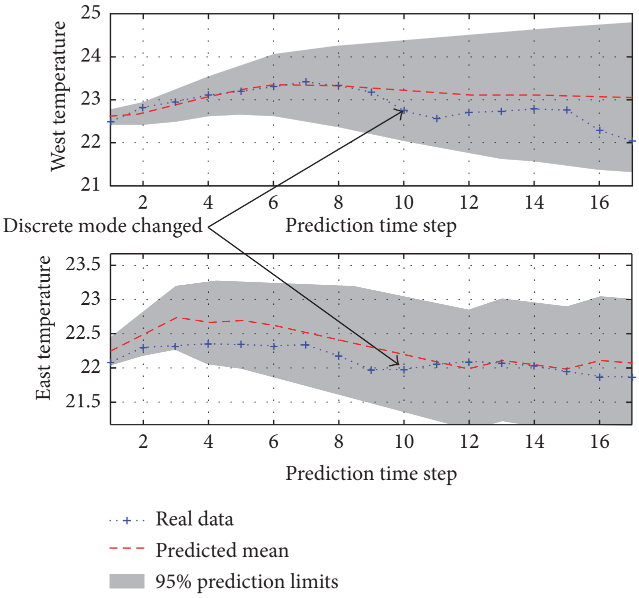


Figure 13: Multistep prediction of zone air temperature for both zones using the proposed SHS model.

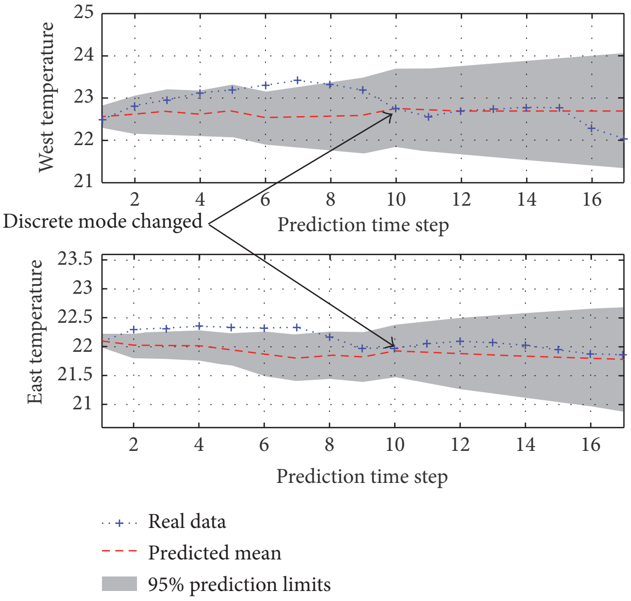


FIGURE 14: Multistep prediction of zone air temperature for both zones using a typical unimodal GP model.

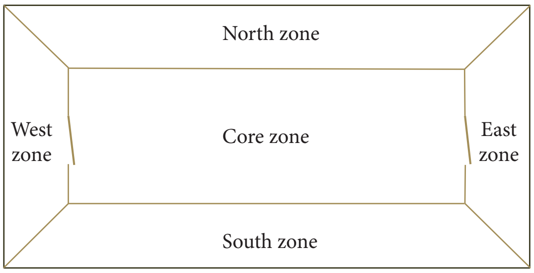


FIGURE 15: Five-zone office building layout.

performance when the dimension of the training data is large. Learning a GP model requires the inversion of the $n \times n$ covariance matrix where n is the size of the data. The matrix inversion has a complexity of $O(n^3)$. Therefore, GP learning becomes more computationally expensive when the dimension of the model (i.e., number of regressors) and/or the training dataset increase. To address this limitation, a typical solution is to divide or distribute the computation. We mitigate this limitation by approximating the thermal load as a discrete state. Therefore, we divide the training data to learn a distinct GP model independently for each mode. Further, in the context of buildings modeling, the model dimension can be decreased by using the adjacent zones only in the regressors regardless of the total number of zones in the buildings, (i.e., $x^i_{k+1} = f_{q_k}(\overline{\mathbf{x}}^i_k, u_k, v_k) : q_k \in \mathcal{Q}$, where $\overline{\mathbf{x}}^i_k$ is the zone air temperature of the adjacent zones of zone i). This approximation is acceptable because the zone air temperature is conditionally independent of other zones given the adjacent zones' air temperature.

To evaluate the model scalability, we used EnergyPlus to generate data for five different buildings with a different number of zones. For each building, we use the vector of zone air temperatures as the continuous state variable; therefore, the model dimension increases as the number of zones increases. We use a fixed training data size for all the buildings to learn a GP model of the thermal dynamics. Figure 20 shows the variation of the learning time of the GP model versus the number of zones in the building.

We also evaluated the performance and the efficiency of the model learning with respect to the training data size. To do so, we learn both the SHS and the full GP model for the north zone, in the five-zone building, online every hour for one week using different dataset sizes. Figure 21 compares the model learning time and the prediction error between the full GP model and the proposed SHS model. The results show that both models have a similar prediction performance; however, the proposed SHS model is faster than the full GP model. The SHS is faster because the training data are segmented into two smaller datasets for each level of the thermal load and then used to learn two distinct GP models. This segmentation distributes the computation of the GPs in the SHS model. On the contrary, the full GP uses a single GP with all data at once. Further, the SHS model approximates the thermal load as a discrete state instead of an input which reduces the model dimension. Despite these approximations, the SHS prediction performance is almost similar to the full GP performance.

7. Conclusion

In this work, we proposed a nonparametric SHS modeling framework with a clustering-based online learning approach. The proposed model can be used to build a thermal model for

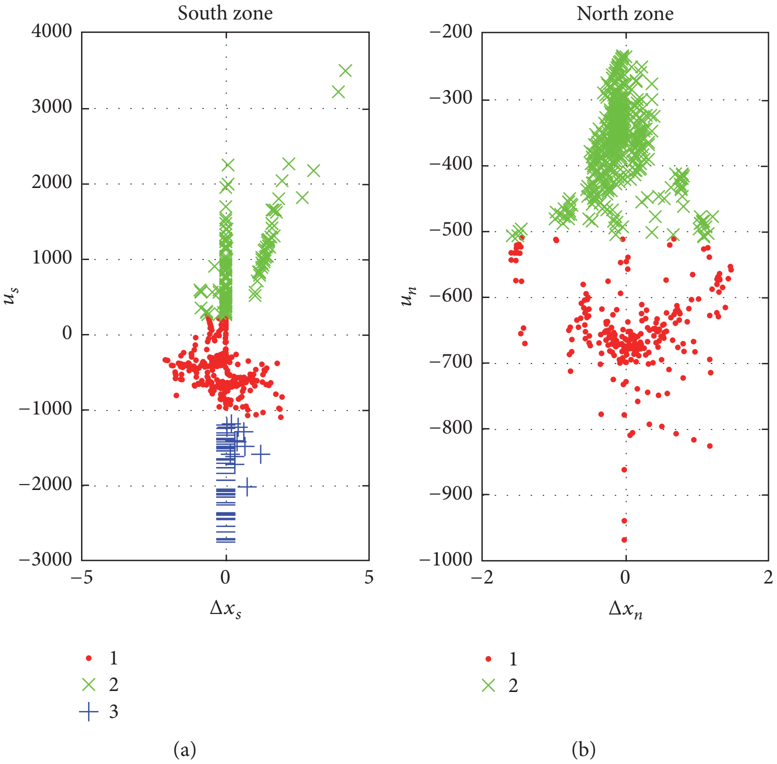


Figure 16: K-means clustering of the south and the north zones.

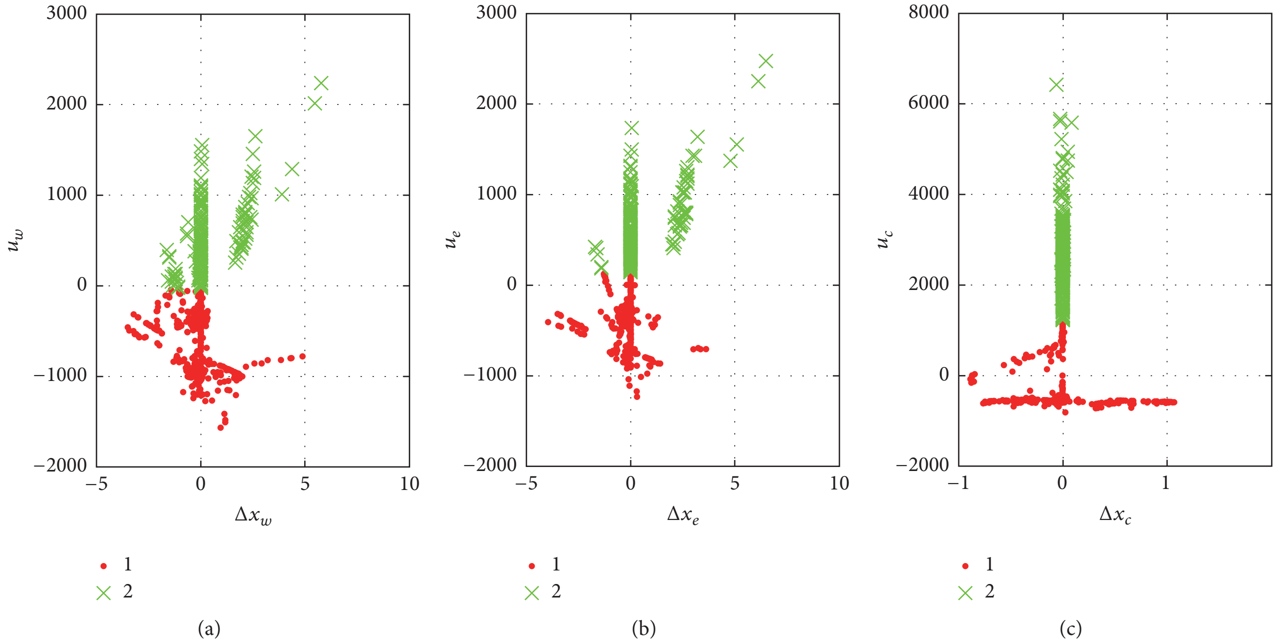


Figure 17: K-means clustering of the (a) west zone, (b) east zone, and (c) core zone.

buildings when the thermal load is latent (i.e., the thermal load cannot be measured). We evaluated the performance of our model by learning thermal models for buildings. Online learning is useful to adapt our model to nonstationary variations and to improve the model efficiency because the training data in the offline phase is not necessarily complete. Despite these advantages, online learning dictates real-time constraints. Experimental results demonstrate the efficiency

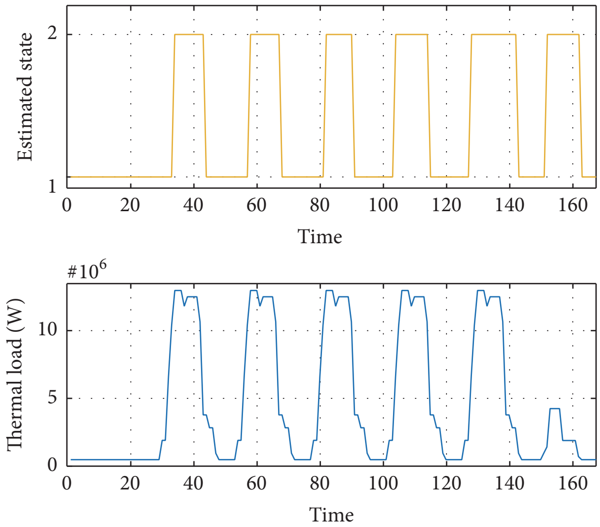


Figure 18: North zone discrete state estimation versus the actual total thermal load.

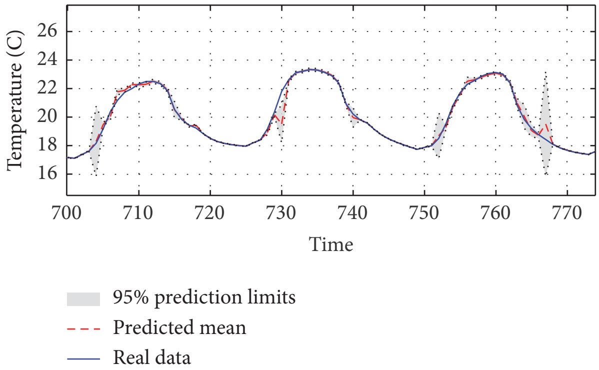


Figure 19: The predicted zone air temperature of the north zone.

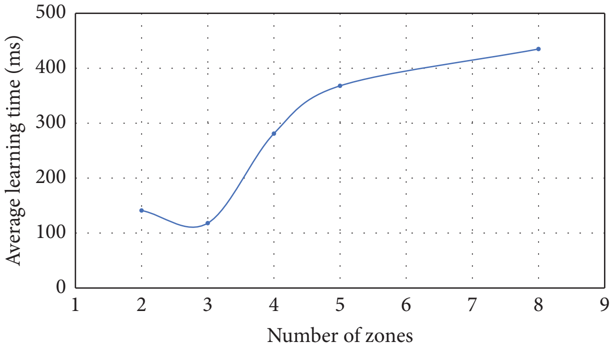
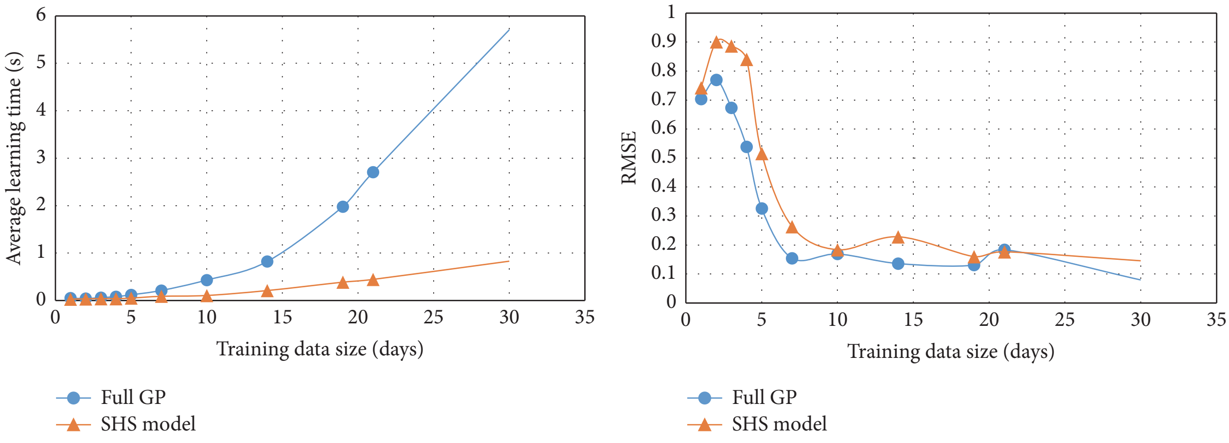


Figure 20: Average learning time of one GP model for different buildings with different numbers of zones.

of the proposed model learning approach. The learning process runs online with an adequate computation time. The average learning time for one GP model was 140 ms and 368 ms for the two-zone building and the five-zone building, respectively. Further, the average prediction time was 1 ms for both buildings.



(a)

(b)

Figure 21: Evaluating model learning efficiency and performance of the north zone in the five-zone building using different training dataset sizes, with a comparison between the full GP model and the SHS model. (a) Average learning time. (b) Prediction error statistics using RMSE.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported in part by the National Science Foundation under Award CNS-1238959 and by AFOSR DDDAS FA9550-13-1-0227.

References

[1] US Energy Information Administration, "The annual energy outlook 2015 with projection to 2040," Tech. Rep., 2015.

[2] S. Prívara, J. Cigler, Z. Váňa, F. Oldewurtel, C. Sagerschnig, and E. Žáčeková, "Building modeling as a crucial part for building predictive control," *Energy and Buildings*, vol. 56, pp. 8–22, 2013.

[3] A. Aswani, N. Master, J. Taneja et al., "Identifying models of HVAC systems using semiparametric regression," in *Proceedings of the 2012 American Control Conference, (ACC '12)*, pp. 3675–3680, IEEE, 2012.

[4] A. Rogers, S. Maleki, S. Ghosh, and J. Nicholas R, "Adaptive home heating control through gaussian process prediction and mathematical programming," in *Proceedings of the in Second International Workshop on Agent Technology for Energy Systems (ATES '11)*, pp. 71–78, event Dates, 2011.

[5] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, 2012.

[6] F. Oldewurtel, A. Parisio, C. N. Jones et al., "Energy efficient building climate control using Stochastic Model Predictive Control and weather predictions," in *Proceedings of the 2010 American Control Conference, (ACC '10)*, pp. 5100–5105, IEEE, July 2010.

[7] Q. Hu, F. Oldewurtel, M. Balandat, E. Vrettos, D. Zhou, and C. J. Tomlin, "Building model identification during regular operation - Empirical results and challenges," in *Proceedings of the 2016 American Control Conference, ACC 2016*, pp. 605–610, July 2016.

[8] D. Sturzenegger, D. Gyalistras, M. Morari, and R. S. Smith, "Semi-automated modular modeling of buildings for model predictive control," in *Proceedings of the 4th ACM Workshop on Embedded Systems for Energy Efficiency in Buildings, BuildSys 2012*, pp. 99–106, November 2012.

[9] B. Sun, P. B. Luh, Q.-S. Jia, Z. Jiang, F. Wang, and C. Song, "Building energy management: integrated control of active and passive heating, cooling, lighting, shading, and ventilation systems," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 588–602, 2013.

[10] H. Zhao, D. Quach, S. Wang et al., "Learning based compact thermal modeling for energy-efficient smart building management," in *Proceedings of the 34th IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2015*, pp. 450–456, November 2015.

[11] F. Massa Gray and M. Schmidt, "Thermal building modelling using Gaussian processes," *Energy and Buildings*, vol. 119, pp. 119–128, 2016.

[12] S. Ghosh, S. Reece, A. Rogers, S. Roberts, A. Malibari, and N. R. Jennings, "Modeling the thermal dynamics of buildings: A latent-force-model-based approach," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 1, Article ID 2629674, 2015.

[13] C. Rasmussen and C. Williams, *Gaussian Process for Machine Learning*, The MIT Press, Boston, Mass, USA, 2006.

[14] J. Kocijan, "Gaussian process models for systems identification," in *Proceedings of the 9th Int. PhD Workshop on Sys. and Cont*, pp. 8–15, 2008.

[15] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philosophical Transactions of the Royal Society of London. Series A. Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, 20110550, 25 pages, 2013.

[16] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting," *Advances in Neural Information Processing Systems*, pp. 545–552, 2003.

[17] H. Mori and M. Ohmi, "Probabilistic short-term load forecasting with Gaussian processes," in *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, ISAP'05*, pp. 452–457, November 2005.

[18] Y. Xie, K. Zhao, Y. Sun, and D. Chen, "Gaussian processes for short-term traffic volume forecasting," *Transportation Research Record*, no. 2165, pp. 69–78, 2010.

[19] R. Turner, M. P. Deisenroth, and C. E. Rasmussen, "State-space inference and learning with Gaussian processes," *Journal of Machine Learning Research*, pp. 868–875, 2010.

[20] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2015.

[21] L. M. Bujorianu, *Stochastic Reachability Analysis of Hybrid Systems*, Communications and Control Engineering Series, Springer Science & Business Media, 2012.

[22] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems a tutorial," *European Journal of Control*, vol. 13, no. 2-3, pp. 242–260, 2007.

[23] G. Pillonetto, "A new kernel-based approach to hybrid system identification," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 70, pp. 21–31, 2016.

[24] K. Koutroumpas, E. Cinquemani, P. Kouretas, and J. Lygeros, "Parameter identification for stochastic hybrid systems using randomized optimization: A case study on subtilin production by bacillus subtilis," *Nonlinear Analysis. Hybrid Systems*, vol. 2, no. 3, pp. 786–802, 2008.

[25] H. Abdel-Aziz and X. Koutsoukos, "Learning and reachability analysis for stochastic hybrid systems using mixtures of Gaussian processes," in *Proceedings of the 24th Mediterranean Conference on Control and Automation, MED 2016*, pp. 332–337, June 2016.

[26] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, New York, NY, USA, 2003.

[27] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344, John Wiley & Sons, New York, NY, USA, 1990.

[28] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann et al., "EnergyPlus: creating a new-generation building energy simulation program," *Energy and Buildings*, vol. 33, no. 4, pp. 319–331, 2001.

[29] "MATLAB and Statistics Toolbox Release 2016a," The MathWorks Inc, Natick, MA, USA, 2016.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration