Modeling and Optimization of Complex Building Energy Systems with Deep Neural Networks

Yize Chen, Yuanyuan Shi and Baosen Zhang∗

Department of Electrical Engineering, University of Washington, Seattle, WA, USA
{yizechen, yyshi, zhangbao}@uw.edu

Abstract—Modern buildings encompass complex dynamics of multiple electrical, mechanical, and control systems. One of the biggest hurdles in applying conventional model-based optimization and control methods to building energy management is the huge cost and effort of capturing diverse and temporally correlated dynamics. Here we propose an alternative approach which is model-free and data-driven. By utilizing high volume of data coming from advanced sensors, we train a deep Recurrent Neural Networks (RNN) which could accurately represent the operation's temporal dynamics of building complexes. The trained network is then directly fitted into a constrained optimization problem with finite horizons. By reformulating the constrained optimization as an unconstrained optimization problem, we use iterative gradient descents method with momentum to find optimal control inputs. Simulation results demonstrate proposed method’s improved performances over model-based approach on both building system modeling and control.

Index Terms—Building energy management, deep learning, gradient algorithms, HVAC systems

I. INTRODUCTION

According to a recent United Nations Environment Programme (UNEP) report, buildings are responsible for 40% of the global energy consumption [1]. Consequently, managing the energy consumption of buildings has significant economical, social, and environmental impacts, and has received much attention from researchers. Many approaches have been proposed to control building systems (e.g., commercial and office buildings, data centers) for energy efficiency, such as nonlinear adaptive control, Model Predictive Control (MPC) and decentralized control for building heating, ventilation, and air conditioning (HVAC) systems [2], [3], [4]. However, most previous research on building energy management are either based on the detailed physics model of buildings [5] or simplified RC circuit models [2], [3], [6]. The former often involves tedious and complex modeling processes with a huge number of variables and parameters, whereas the latter cannot fully capture the long term dynamics of large commercial buildings.

With the advance of sensing, communication and computing, detailed operation data are being collected for many buildings. These data along with future weather forecasts can be utilized for data-driven real-time optimization approaches. In [7], the authors developed a data predictive control method to replace the traditional MPC controller by using data to build a regression tree that represent the dynamical model for a building. However, regression trees still results in a linear model that can be far away from the true dynamics of building systems. While in [8], [9], reinforcement learning was proposed to learn control policies without any explicit modeling assumptions, but computational costs for searching through large state and action spaces is hight. Ill-defined reward functions (e.g., sparse, noisy and delayed rewards) could also prevent reinforcement learning algorithm finding the optimal control solutions [10]. Furthermore, large commercial buildings may have quality of service constraints that prevent the deep exploration of some states in reinforcement learning.

In this work, we address these challenges by proposing a data-driven method which closes the loop for accurate predictive model and real-time control. The method is based on deep recurrent neural networks that leverage rich volumes of sensor data [11]. Though neural network has previously been adopted as an approach for designing controllers, the lack of large datasets and computation capabilities have prevented it from being deployed in real-time applications [12]. Firstly in a supervised learning manner, our Recurrent Neural Networks (RNN) firstly learns the complex temporal dynamics mapping from various measurements of building operation profiles to energy consumption. Next we formulate an optimization problem with the objective of minimizing build-
ing energy consumption, which is subject to RNN-modeled building dynamics as well as physical constraints over a finite horizon of time. To solve the constrained optimization problem in a block-splitting approach, we take iterative gradient descent steps on the set of controllable inputs (e.g., zone temperature setpoints, heat rejected/added into each zone) at the current timestep. It thus finds the control inputs for each timestep. Fig. 1 illustrates our model framework. Our approach does not need analysis on complex interactions within conduction, convection or radiation processes. In addition, it can be easily scaled up to large buildings and distributed algorithms.

The main contributions of our paper are:

- We model the building energy dynamics using recurrent neural networks, which leverages large volumes of data to represent the complex dynamics of buildings.
- We propose an input/output optimization algorithm which efficiently find the optimal control inputs for the model represented by RNN.
- The proposed modeling and optimization approaches open door to the integration of complex system dynamics modeling and decision-making.

The contents of the paper are as follows. The rolling horizon control problem formulation and model-based method are firstly presented in Section II. In Section III we show the design of a deep RNN which models the dynamics of complex building systems. We then reformulate the control problem as an unconstrained optimization problem, and propose the algorithm to find optimal control inputs in Section IV. Finally, simulation results on large building HVAC system are evaluated and compared with model-based control method in Section V.

II. PROBLEM FORMULATION & PRELIMINARIES

A. Problem Formulation

We consider a building energy system which includes several subsystems and zones with potentially complex interactions between them. No information about the exact system dynamics is known. At time \( t \), we are provided with the building’s running profile \( X_t := [X_{uc}^t, X_c^t, X_{phy}^t]^T \), where \( X_{uc}^t \) denotes a collection of uncontrollable measurements such as zone temperature measurements, system node temperature measurements, lighting schedule, in-room appliances schedule, room occupancies and etc; and \( X_c^t \) denotes a collection of controllable measurements such as zone temperature setpoints, appliances working schedule and etc; \( X_{phy}^t \) denotes the set of physical measurements or forecasts values, such as dry bulb temperature, humidity and radiation volume. There are some physical constraints on some of \( X_t \) and \( X_{phy}^t \); for example the temperature setpoints as well as real measurements should not fall out of users’ comfort regions. Without loss of generality, we denote the constraints as \( X_{c}^t \leq X_{c}^t \leq X_{c}^t \) and \( X_{uc}^t \leq X_{uc}^t \leq X_{uc}^t \). Building System operators have a group of past running profile \( X = \{X_t\} \) along with the collection of energy consumption metering at each time step \( P = \{P_t\} \).

We are interested in firstly learning a model \( f(X_{t-T}, ..., X_{t}) = P_t \), where \( f(\cdot) \) denotes the predictive model with known parameters representing building’s physical dynamics. \( f(\cdot) \) maps past \( T \) timestep’s running profile to energy consumption at timestep \( t \).

With a model \( f(\cdot) \) representing the building dynamics, we formulate an optimal finite-horizon predictive control problem, and propose an efficient algorithm to find the group of optimal control inputs \( X_{uc}^t \). At timestep \( t \), the control input \( X_{uc}^t \) minimizes the energy consumption of the building for future \( T \) steps. Meanwhile, previous \( T \) steps’ control inputs would affect current energy consumption. The objective of the controller is to minimize the energy consumption with a rolling horizon \( T \), while maintaining some variables within comfortable intervals. Mathematically, we formulate the general control problem as

\[
\begin{aligned}
\text{minimize} & \quad \sum_{t=0}^{T} P_{t+\tau}^2 \\
\text{subject to} & \quad P_{t+\tau} = f(X_{t-T+\tau}, ..., X_{t+\tau}), \forall \tau \\
& \quad X_{c}^{t+\tau} \leq X_{c}^{t+\tau} \leq X_{c}^{t+\tau}, \forall \tau \\
& \quad X_{uc}^{t+\tau} \leq X_{uc}^{t+\tau} \leq X_{uc}^{t+\tau}, \forall \tau \\
& \quad X_{uc}^{t+\tau} = h(X_{t-T+\tau}, ..., X_{t-1+\tau}, X_{c}^{t+\tau}, X_{phy}^{t+\tau}), \forall \tau
\end{aligned}
\]

where (1b) \( h(\cdot) \) denotes the rolling horizon predictive model; (1c) and (1d) are the constraints on controllable and uncontrollable variables respectively; \( h(\cdot) \) in (1e) denotes a rolling horizon predictive function for uncontrollable variables based on past \( T \) steps’ observations as well as current step control inputs and physical forecasts.

B. First-Order Thermal Dynamic Model

For building HVAC system, one popular method used in finite-horizon MPC to model the thermal dynamics is the reduced Resistance-Capacitance (RC) model [2], [3], [6]. Here we use a rolling horizon MPC controller as a benchmark for comparison.

Denote \( \mathcal{N}(i) \) as the neighboring zones for zone \( i \), the first-order RC model modeling HVAC dynamics is formulated as

\[
C_i \dot{T}_{i,t} = \frac{T_{o,t} - T_{i,t}}{R_i} + \sum_{j \in \mathcal{N}(i)} \frac{T_{j,t} - T_{i,t}}{R_{ij}} + P_{i,t} \tag{2}
\]

where \( C_i, T_{i} \) are the thermal capacitance and room temperature for each zone \( i \), while \( T_{o} \) is the outside dry bulb temperature, and \( R_i, R_{ij} \) are the thermal resistance for zone \( i \) against the outside and the neighboring zone \( j \). The schematic of RC network for modeling HVAC system is shown in Fig. 2.

Once we find \( C_i, R_i, R_{ij} \) for all the zones, we have a 1st-order system to model the thermal dynamics. Since \( T_i \in X_{uc}, T_o \in X_{phy} \), by reformulating (2) and taking a sum of \( P_i \) for all zones, we reformulate and write the building overall thermal dynamics

\[
P_t = f_{RC}(X_{t-T}, ..., X_t) \tag{3}
\]
which is further used in the optimal control problem defined in (1a)-(1e). MPC for building HVAC system under different model settings has been implemented in [2], [3]. We focus on the performance comparison of RC model to our proposed method in both model fitting and optimization tasks.

III. RECURRENT NEURAL NETWORKS

Since the 1st-order thermal dynamic model defined in (2) does not either capture complicated nonlinear dynamics, nor model the long-term temporal dependencies of building HVAC system, the deep RNN model becomes a good replacement.

RNN is a class of artificial neural networks specially designed for sequential data modeling. Unlike fully-connected neural networks where inputs are fed into the neural networks as a full vector, RNN feeds input sequentially into a neural network with directed connections. It uses its internal memory to process time-series inputs. In Fig. 3 we show the structure of an RNN model.

We specifically design the RNN model to solve a time-dependent regression problem. That is to say, we want RNN automatically learn the relationship between sequential input \( x_t, t = 0, ..., T \) and output \( o_T \). At timestep \( t \), RNN is provided with hidden state vector \( h_t \) and input vector \( x_t \), and outputs its computation vector \( \hat{o}_t \). The \( t \)-step RNN cell is composed of three groups of neurons, \( \theta_{x,t}, \theta_{h,t}, \theta_{o,t} \). They are associated with input, hidden state and output respectively, and are organized in function \( f_{\theta_{x,t}, \theta_{h,t}, \theta_{o,t}} \) to complete the following computations:

\[
\begin{align*}
\hat{o}_t &= f_{\theta_{x,t}, \theta_{o,t}}(x_t, h_t), \\
h_{t+1} &= f_{\theta_{h,t}, \theta_{o,t}}(x_t, h_t)
\end{align*}
\]

where \( \hat{o}_t \) is the RNN’s prediction output, while \( h_{t+1} \) is passed into next neuron group and takes part in \( t + 1 \) step’s computation.

After concatenating all the neurons cells from 0 to \( T \), we get the chain function to compute \( h_T \). Thus the RNN compute the final prediction value \( \hat{o}_T \):

\[
\hat{o}_T = f_{\theta_{x,T}, \theta_{o,T}}(x_T, h_T)
\]

Since \( h_T \) captures information from past inputs \( x_{t-1} \), we trace hidden states back into functions of previous steps’ hidden states and inputs. Thus final output \( \hat{o}_T \) is eventually a function of the sequential inputs \( x_t, t = 0, ..., T \). For simplicity, let’s denote \( \theta = \{ \theta_{x,t}, \theta_{o,t}, \theta_{z,t} \}, t = 0, ..., T \) to be the set of neurons used in modeling the \( T \)-length temporal data, and wrap up all neural-composed functions of \( \{ f_{\theta_{x,t}, \theta_{o,t}}, f_{\theta_{h,t}, \theta_{o,t}} \} \) to get the overall function \( f_{RNN} \), which utilizes \( \theta \) to find the output predictions with length \( T \) time-series input:

\[
\hat{o}_T = f_{RNN}(x_0, ..., x_T)
\]

We set up the RNN model and initialize neuron weights \( \theta \) by sampling from a normal distribution. During batch-training process, with a group of sequential input \( x_t, t = 0, ..., T \), \( \hat{o}_T \) is firstly computed, and by doing back-propagation using stochastic gradient descent (SGD) with respect to all neurons [13], \( \theta \) is optimized to minimize the regression loss defined in mean-square-error (MSE) form:

\[
L_{\text{training}}(\theta) = ||\hat{o}_T - o_T||_2^2, \tag{7a}
\]

\[
\theta^* = \arg \min_{\theta} L_{\text{training}}(\theta). \tag{7b}
\]

We then set up a length-\( T \) RNN accordingly for our building dynamics modeling problem. With the training sets of input vectors of historical building operating profiles \( \{ X_{t-T}, ..., X_t \} \) and an output energy consumption \( P_t \), our RNN model \( f_{RNN} \) is trained to represent the system dynamics

\[
\hat{P}_t = f_{RNN}(X_{t-T}, ..., X_t)
\]

Our RNN is totally data-driven, and can process and represent temporal dependencies. With a rich volume of historical building operating data \( X \) and \( P \) provided as the training datasets, we train a deep RNN, which accurately models the nonlinear, complex temporal dynamics of building system. We will show in Section V that our deep RNN model outperforms RC model in fitting the dynamics of a large-scale building HVAC system.
IV. INPUTS OPTIMIZATION FOR BUILDING CONTROL

In this section we describe our control algorithm which is based on our pre-trained deep learning model. We demonstrate how it is able to incorporate (8) into the optimization problem (1). We also illustrate how to solve such optimization problem to find a collection of optimal control sequential inputs.

By substituting $f(\cdot)$ in (1) with $f_{RNN}$, and denote $X_i^{var} = [X_i^l, X_i^{uc}]$, the finite horizon control problem for building energy management is written as

\begin{align}
\text{minimize} & \quad \sum_{t=0}^{T} P_{t+\tau}^2 \\
\text{subject to} & \quad P_{t+\tau} = f_{RNN}(X_{t-T+\tau}, \ldots, X_{t+\tau}), \forall \tau \\
& \quad X_{t+\tau}^{var} \leq X_{t+\tau}^{var} \leq X_{t+\tau}^{var}, \forall \tau \\
& \quad X_{t+\tau}^{uc} = h(X_{t-T+\tau}, \ldots, X_{t-1+\tau}, X_{t+\tau}^{c}, X_{t+\tau}^{phy}), \forall \tau
\end{align}

Since $X_{t+\tau}^{uc}, \tau = 1, \ldots, T$ is directly controlled by control inputs of previous time. For all the uncontrollable variables with constraints we model, they also possess pairing controllable variables, e.g., the temperature measurements-temperature set-points. We then choose $X_{t+\tau}^{uc} = X_{t+\tau}^{c}, \tau = 1, \ldots, T$, since such uncontrollable values are the control outputs corresponding to the previous step’s control inputs. Thus we diminish constraint (9d).

Since the constrained optimization problem (9) includes a non-convex deep neural network in the constraints, we use log barriers functions to rewrite the problem in an unconstrained form:

\begin{align}
\min_{X_t^{l}, \ldots, X_{t+T}^{l}} L_{opt}(X_t^{l}, \ldots, X_{t+T}^{l}) =
\sum_{\tau=0}^{T} f_{RNN}(X_{t-T+\tau}, \ldots, X_{t+\tau}) \\
- \lambda \sum_{\tau=0}^{T} \log(X_{t+\tau}^{var} - X_{t+\tau}^{var}) \\
- \lambda \sum_{\tau=0}^{T} \log(X_{t+\tau}^{var} - X_{t+\tau}^{var})
\end{align}

where $\lambda$ is a tuning parameter, and $L_{opt}(X_t^{l}, \ldots, X_{t+T}^{l})$ defines a loss function with inputs $X_t^{l}, \ldots, X_{t+T}^{l}$. We solve this loss minimization problem by iteratively taking gradient descents of (10). Note that during RNN model training, we are taking gradients $\nabla_{\theta} L_{training}(\theta)$ with respect to all the neurons. Once training is done, $L_{training}(\theta)$ is converged. The RNN model serves as the temporal physical model, and is always modeling the building system dynamics accurately. Here we are taking gradients with this fixed, pre-trained RNN model, and find gradients $\nabla X_{t+\tau}^{c} L_{opt}(X_t^{l}, \ldots, X_{t+T}^{l}), \tau = 0, \ldots, T$ with respect to the group of controllable variables. Once $L_{opt}(X_t^{l}, \ldots, X_{t+T}^{l})$ is converged, and we find $X_t^{c}$ that is a local optimal solution. $X_t^{c}$ is also the solution of controllable inputs for the finite horizon optimal control problem at timestep $t$.

The $k$-step gradient descent method is working as follows:

\begin{align}
g_{t+\tau,k} = \eta \nabla X_{t+\tau,k}^{c} L_{opt}(X_{t,k-1}, \ldots, X_{t+T,k-1}) \\
X_{t+\tau,k} = X_{t+\tau,k-1} - g_{t+\tau,k}, \tau = 0, \ldots, T
\end{align}

where $\eta$ is the learning rate, and $X_{t+\tau,k}$ denotes the value for $X_{t+\tau}^{c}$ after $k$ step’s update.

Throughout our modeling and optimization approach, we do not make any physical model assumptions, and directly utilize a deep RNN to extract the model dynamics as well as finding the optimal actions to take at each time step to cut down energy consumption. We summarize the proposed method in Algorithm 1, which closes the loop for building dynamics modeling and control inputs optimization. In our implementation, we improve the algorithm performance by adding momentum to gradient descents (MomentumGD), which is shown to get over some local minima during optimization iterations as well as accelerating the convergence [14]. The MomentumGD is realized as follows:

\begin{align}
g_{t+\tau,k} = \gamma g_{t,k-1} + \eta \nabla X_{t+\tau,k}^{c} L_{opt}(X_{t,k-1}, \ldots, X_{t+T,k-1}) \\
X_{t+\tau,k} = X_{t+\tau,k-1} - g_{t+\tau,k}, \tau = 0, \ldots, T
\end{align}

where $\gamma$ is a momentum term determining how much previous gradients are incorporated into current step’s update.

**Algorithm 1** Input Optimization for Building Control

**Input:** Pre-trained RNN $f_{RNN}$, learning rate $\eta$, momentum $\gamma$, input optimization iterations $N_{iter}$

**Input:** Control window-size $T$

**Input:** Sensor measurements $X_t^{uc}$, weather forecasts $X_t^{phy}$

**Initialize:** $X_t^{l}, \ldots, X_{t+T}$

**Initialize:** Optimal control inputs $X_t^{c} \leftarrow \emptyset$

for iteration $= 0, \ldots, N_{iter}$ do

Update $X_t^{c}$ using gradient descent:

for $\tau = 0, \ldots, T$ do

$g_{t+\tau} \leftarrow \nabla X_{t+\tau}^{c} L_{opt}(X_t^{l}, \ldots, X_{t+T}^{l})$

$X_{t+\tau}^{c} \leftarrow X_{t+\tau}^{c} - \eta \cdot \text{MomentumGD}(X_{t+\tau}^{c}, g_{t+\tau}, \gamma)$

end for

Update $X_t^{uc}$ using gradient descent:

for $\tau = 0, \ldots, T$ do

$X_{t+\tau}^{uc} = X_{t+\tau}^{c}$

end for

end for

$X_t^{c}, \text{insert}(X_t^{c})$

V. CASE STUDY

In this section, we set up a realistic model in standard building simulation software EnergyPlus [15]. We demonstrate the effectiveness of our data-driven approach for both system dynamics modeling and building energy management. In order
to compare with the model-based approach, we focus on the
HVAC system for a large building complex. But our method
is a general regression and optimization approach, which
could be easily applied to overall building energy management
problem.

A. Experimental Setup

We set up our EnergyPlus simulations using a 12-storey
large office building (in Fig. 4) listed in the commercial
reference buildings from U.S. Department of Energy (DoE
CRB) [16]. The building has a total floor area of 498,584
square feet which is divided into 16 separate zones. We
simulate the building running through the year of 2004 in
Seattle, WA, and record (X_t, P_t) with a resolution of 10
minutes. We shuffle and separate 2 months’ data as our
stand-alone testing dataset for both regression and control
performance evaluation, while the remaining 10 months’ data
is used to for RNN training. The processed datasets have 55
input features, which include controllable variables such as
zone temperature setpoints, and uncontrollable variables such
as zone occupancies and temperature measurements. Output
is a single feature for energy consumption at each timestep.
We directly use historical weather data records into both RC
model and RNN model. For future work, the forecasts model
should also be considered into the pipeline. A finite horizon
of 4 hours is set for both MPC method and proposed method.

We set up our deep learning model using Tensorflow, a
Python open-source package. Our RNN model is composed of
1 recurrent layer with 3 subsequent fully-connected layers. We
adopt rectified linear unit (ReLU) activation functions, dropout
layers and Stochastic Gradient Descent (SGD) optimizer to
improve our neural network training.

B. Simulation Results

We first compare the model fitting performance for 1st order
model and RNN model, and the fitting result for two weeks’
ergy consumption is shown in Fig. 5. To quantitatively
compare the model fitting error, we calculate the Root-Mean-
Square-Error (RMSE) value for normalized energy consump-
tion on test dataset. RMSE for the first-order RC model
is 0.240. The RNN model improves RC model by 68.33%
with an RMSE of 0.076. It is also interesting to notice that
this large office building actually has an energy consumption
dropdown on weekdays’ noon due to the occupancy schedule.
RC model fails to capture this dynamic characteristics, while
RNN model is able to fit noon values given past 4 hours’
input measurements. Moreover, RC model performs poorly on
weekend regression task, which hardly represents the HVAC
dynamics. This inaccurate model would make subsequent
MPC algorithm fail to operate on correct model space.

Next we show the constrained optimization problem formu-
lated in (10) is efficient in finding optimal inputs X_t^c for
the HVAC system. In Fig. 6 we show a group of 3 plots cor-
responding to different zone temperature setpoint constraints.
We keep setpoint constraints the same for all the 16 zones.
Compare the results of X_t^c ∈[18°C,26°C] and the results of
X_t^c ∈[19°C,24°C], we observe that our approach is able to
find sharper control inputs with less energy consumption when
constraint intervals are bigger. When there is no constraint
on temperature, our approach simply finds extreme control
inputs such that the energy consumption is nearly same as the
midnight consumption.

We then compare the optimization performance for RC
model and RNN model. Fig. 7 illustrates a Monday-Friday
energy consumption profile with temperature setpoint con-
straints X_t^c ∈[18°C,26°C]. By using RNN model and taking
the gradient steps, we find a sequence of control inputs that
could reduce 30.74% of energy consumption. On the other
hand, the solution found by RC model only gives us a 4.07%
reduction of energy consumption. This further illustrates that
RC model is not good at modeling large-scale building system
dynamics.

Fig. 8 demonstrates how our proposed approach is able to
find a group of control inputs for the building system globally.
All of four zones’ setpoint schedule exhibit daily patterns.
Yet they are set to different values and evolution patterns.
These setpoint schedule can provide to building operators, and
it remains to be examined in real buildings if such optimized
schedules could benefit the complex system as a whole.
VI. CONCLUSION

In this work, we are exploiting Recurrent Neural Networks’ ability of learning complex temporal interactions among high-dimensional building dynamics. Our proposed method consists Recurrent Neural Networks regression and sequence optimization steps, which could both be solved efficiently. Our proposed approach is easily to be deployed for any building unit provided with rich historical running data. Simulation results show that our method outperforms existing ones both in capturing the thermal dynamics of the building as well as providing effective control solutions.

REFERENCES

[1] C.-C. Cheng, S. Pouffary, N. Svenningsen, and J. M. Callaway, “The kyoto protocol, the clean development mechanism and the building and construction initiative,” 2008.

[2] Y. Ma, A. Kelman, A. Daly, and F. Borrelli, “Predictive control for energy efficient buildings with thermal storage: Modeling, simulation, and experiments,” IEEE Control Systems, vol. 32, no. 1, pp. 44–64, 2012.

[3] X. Zhang, W. Shi, B. Yan, A. Malkawi, and N. Li, “Decentralized and distributed temperature control via hvac systems in energy efficient buildings,” arXiv preprint arXiv:1702.03308, 2017.

[4] Y. Shi, B. Xu, B. Zhang, and D. Wang, “Leveraging energy storage to optimize data center electricity cost in emerging power markets,” arXiv preprint arXiv:1606.01536, 2016.

[5] M. Tück and J. L. Hensen, “Overview of hvac system simulation,” Automation in Construction, vol. 19, no. 2, pp. 93–99, 2010.

[6] Y. Ma, F. Borrelli, B. Henvey, A. Packard, and S. Bortoff, “Model predictive control of thermal energy storage in building cooling systems,” in Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on. IEEE, 2009, pp. 392–397.

[7] A. Jain, M. Behl, and R. Mangharam, “Data predictive control for building energy management,” in BuildSys, 2017.

[8] L. Yang, Z. Nagy, P. Goffin, and A. Schlueter, “Reinforcement learning for optimal control of low exergy buildings,” Applied Energy, vol. 156, pp. 577–586, 2015.

[9] T. Wei, Y. Wang, and Q. Zhu, “Deep reinforcement learning for building hvac control,” in The Design and Automation Conference, 2017.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” in International conference on machine learning, 2013.

[11] K.-i. Funahashi and Y. Nakamura, “Approximation of dynamical systems by continuous time recurrent neural networks,” Neural networks, vol. 6, no. 6, pp. 801–806, 1993.

[12] D. H. Nguyen and B. Widrow, “Neural networks for self-learning control systems,” IEEE Control systems magazine, vol. 10, no. 3, pp. 18–23, 1990.

[13] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in Proceedings of COMPSTAT 2010. Springer, 2010, pp. 177–186.

[14] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in International conference on machine learning, 2013, pp. 1139–1147.

[15] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte et al., “Energyplus: creating a new-generation building energy simulation program,” Energy and buildings, vol. 33, no. 4, pp. 319–331, 2001.

[16] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, M. Rosenberg et al., “Us department of energy commercial reference building models of the national building stock,” 2011.