Building Thermal Dynamics Modeling with Deep Learning exploiting Large Residential Smart Thermostat Dataset

Han Li
hanli@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, California, USA

Giuseppe Pinto
Politecnico di Torino
Torino, Italy
giuseppe-pinto@polito.it

Alfonso Capozzoli
Politecnico di Torino
Torino, Italy
alfonso.capozzoli@polito.it

Tianzhen Hong*
thong@lbl.gov
Lawrence Berkeley National Laboratory
Berkeley, California, USA

ABSTRACT

In this paper, we present a deep learning approach to model building thermal dynamics with large-scale smart thermostat data collected from residential buildings. We developed a Long Short-Term Memory (LSTM) model as a baseline and compared it to a CNN-LSTM model to predict indoor air temperature in a multi-step time horizon in 164 buildings. The study showed that the proposed CNN-LSTM achieved an average of 0.26 °C Mean Absolute Error (MAE) for one-hour-ahead (12 future steps) predictions, which is over 6% of improvement comparing with the baseline. Furthermore, the results indicated that the CNN-LSTM models achieved more robust performance across different building characteristics, system configurations and locations, with a standard deviation reduction of 22%, proving the effectiveness and generalizability of the proposed approach.

CCS CONCEPTS

• **Applied computing** → **Physical sciences and engineering**; • **Software and its engineering** → *Software notations and tools.*

KEYWORDS

Building thermal dynamics, data-driven model, deep learning

ACM Reference Format:
Han Li, Giuseppe Pinto, Alfonso Capozzoli, and Tianzhen Hong. 2022. Building Thermal Dynamics Modeling with Deep Learning exploiting Large Residential Smart Thermostat Dataset. In *The 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3563357.3564056

1 INTRODUCTION

Building thermal dynamics models which predict future indoor air temperature given control actions, are essential for optimizing HVAC system controls and building energy management. Such thermal dynamics models are usually classified into white-box, grey-box, and black-box models. White-box models are based on first principles that govern the energy and mass transfer in buildings. Those models are developed in a forward approach where information about building geometry, energy systems, occupant behaviors, and weather conditions are known. However, despite their effectiveness, this often requires extensive expertise in developing and calibrating white-box models [3]. In contrast, grey-box and black-box models use the inverse approach, exploiting measured data to identify a model that describes a building's thermal processes. Grey-box models use lumped parameters to represent buildings' thermal properties. For example, as an analog to the electric circuit, thermal resistance (R) and thermal capacity (C) are two types of parameters of reduced-order models that describe the heat transfer in buildings [10]. Although these parameters can be identified using regressions with measured data, they often require simplified assumptions about external and internal loads. Furthermore, they could not directly utilize data that is recorded by smart thermostat, such as occupant motion detection and HVAC system runtime, which leads to a waste of information. On the other hand, black-box models are purely data-driven as they often do not need prior-knowledge about the building and can fully exploit the data provided by smart thermostat. In recent years, Deep Learning (DL) techniques have been extensively used in the built environment due to their ability to approximate complex dynamics. Wang et al. compared 9 ML algorithms for building thermal load prediction, and found that LSTM could achieve lower than 0.4 °C MAE for one-hour ahead predictions [11]. Pinto et al. developed LSTM thermal dynamics models of multiple buildings to support reinforcement learning based energy management [8]. Mtibaa et al. compared LSTM-based model architectures for indoor air temperature predictions [5]. Elmaz et al. proposed a CNN-LSTM model architecture for indoor air temperature predictions [2]. However, most existing studies developed models only for specific buildings and did not examine the model generalizability and consistency for buildings with various characteristics across different climate regions. Moreover, very few studies used real measurements for model training and testing, limiting the effectiveness of the proposed approaches. In this paper, we introduce a methodology to pre-process, enrich and exploit smart thermostat data that spans over different buildings in three U.S. states, three space types and two heating system

*Corresponding author: thong@lbl.gov

BuildSys '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11.
https://doi.org/10.1145/3563357.3564056

configurations. Furthermore, we introduce a CNN-LSTM model trained and tested on over 150 residential buildings, with the aim to predict indoor air temperature over a multi-step horizon.

2 BACKGROUND

2.1 Deep Learning for Building Thermal Dynamics

Before the era of deep learning, most data-driven models for building thermal dynamics were linear and time-invariant [10]. As deep learning algorithms and computing resources became more and more available and mature in recent years, they have been increasingly applied in modeling building thermal dynamics due to their ability to approximate non-linearity and time-variance. The building indoor air temperature prediction problem involves multivariate inputs, sequential modeling, and multi-horizon outputs. A brief background of those three components is presented below:

- Feature extraction: it starts with the initial set of variables with the goal to extract a new set of processed variables that facilitate further pattern recognition. For multivariate time series data such as the thermostat measurements, 1D CNN is a commonly used technique.
- Sequential modeling: it receives sequential data such as time-series data, and outputs a single value or another sequence. Recurrent Neural Networks (RNNs), are a popular category of DL algorithms for sequential modeling. LSTM is a type of RNNs that specializes in both short-term and long-term memories [4]. It uses gating mechanisms that control non-linearity and information memory, and addresses the vanishing gradient issue in standard RNNs [9].
- Multi-horizon prediction: it forecasts the target variable for the next several steps at once. Depending on the model architecture, multi-horizon prediction can be classified into (1) iterative methods, where the single-step outputs and historical data are iteratively used for the next step prediction until the desired horizon is reached; (2) direct methods, where a complete sequence is output from the model and can be considered as sequence-to-sequence (seq2seq) methods. In this study, we used the direct methods to predict indoor air temperature.

3 METHODOLOGY

3.1 Data Processing

We used the smart thermostat data collected by ecobee's Donate Your Data (DYD) program where more than 190,000 households in the U.S. and Canada had voluntarily shared their data anonymously for research purposes as of 2022. Each thermostat has user-reported metadata about the building, including location (at city level), space type, gross floor area, number of floors, and time when the thermostat first connected. Figure 1 shows the three steps adopted to pre-process the data.

In step 1, we randomly sampled a subset of buildings using the building metadata. Specifically, the subset include buildings with three space types (i.e., apartment, townhouse, and detached single family houses), and two HVAC system configurations (i.e., with and without electric auxiliary heating) from three U.S. states with distinct climates (i.e., California, Texas, and New York). In step 2, we processed the time-series data for each building. To avoid the influence of behavioral change on building thermal dynamics due to COVID19 pandemic, we decided to only use a whole year of data in 2019. The raw time-series data includes information on the indoor environment and the HVAC systems, such as: indoor air temperature and humidity, cooling and heating setpoint temperature, supply fan runtime, cooling and heating system runtime, and occupant motion detections with five-minute temporal resolution. Furthermore, due to the high correlation of energy use with occupancy, we added temporal features such as time of the day, day of the week, and month of the year encoded as cosine and sine values, while differentiating between holidays with a binary encoding. Lastly, since the ecobee thermostat dataset does not include outdoor weather data, we added outdoor air temperature data to each thermostat using each thermostat's latitude and longitude to find the closest weather station listed by the National Oceanic and Atmospheric Administration (NOAA). In conclusion, a total of 23 features were used to describe the building thermal dynamics, which have been further standardized by scaling to unit variance using the scikit-learn package [7]. Table 1 shows the name, unit, and type of the variables used to train the deep learning models. The detailed descriptions and source code for data processing are available at the GitHub repository: https://github.com/tsbyq/EcoBee_BTD.

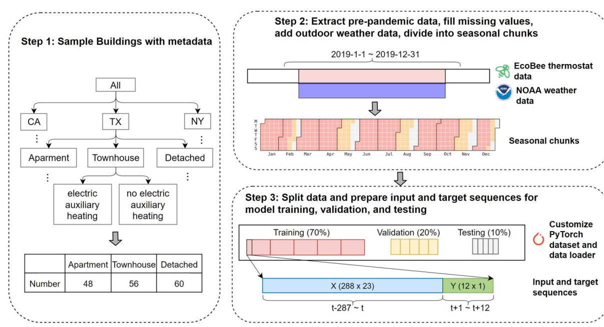

Figure 1: Data processing steps

3.2 Deep Learning Model Development

In this study, we implemented a simple LSTM model as the baseline, where data is directly input to the LSTM cell, followed by a linear layer for multi-horizon indoor temperature predictions. Then, we propose a modified version of the previous architecture, adding a 1D CNN module before the LSTM cell for feature enhancing. Figure 2 shows the proposed CNN-LSTM architecture. Both models are implemented using PyTorch [6]. We then performed a sensitivity analysis on models hyperparameters, optimizer, and learning rate scheduler, which will be presented in section 4.

3.3 Performance Metrics

In this study, we used MAE for evaluation purposes, since it has easy-to-interpret physical meaning (°C deviations) and increases steadily as the error grows. We also used performance improvement ratio (PIR) to quantify the relative performance improvement. The formula of MAE and PIR are shown below.

Variable Name	Meaning	Type	Unit
TemperatureExpectedCool	thermostat cooling setpoint	numerical	°C
TemperatureExpectedHeat	thermostat heating setpoint	numerical	°C
Humidity	relative humidity	numerical	%
auxHeat1	auxiliary heating system 1 runtime	numerical	seconds/5minutes
auxHeat2	auxiliary heating system 2 runtime	numerical	seconds/5minutes
auxHeat3	auxiliary heating system 3 runtime	numerical	seconds/5minutes
compCool1	cooling compressor 1 runtime	numerical	seconds/5minutes
compCool2	cooling compressor 1 runtime	numerical	seconds/5minutes
compHeat1	heating compressor 1 runtime	numerical	seconds/5minutes
compHeat2	heating compressor 1 runtime	numerical	seconds/5minutes
fan	supply air fan runtime	numerical	seconds/5minutes
Thermostat_Temperature	aggregated thermostat temperature	numerical	seconds/5minutes
Thermostat_Motion	occupant presence	binary	N.A.
T_out	outdoor air temperature from NOAA	numerical	°C
sin_hour	sine of an hour in a 24-hour day	numerical	N.A.
cos_hour	cosine of an hour in a 24-hour day	numerical	N.A.
sin_day_of_week	sine of an day in a 7-day week	numerical	N.A.
cos_day_of_week	cosine of an day in a 7-day week	numerical	N.A.
sin_month	sine of an day in a month	numerical	N.A.
cos_month	cosine of an day in a month	numerical	N.A.
sin_week_of_year	sine of a week in a 52-week year	numerical	N.A.
cos_week_of_year	cosine of a week in a 52-week year	numerical	N.A.
is_holiday	whether a day is holiday	binary	N.A.

Table 1: Time-series data variables

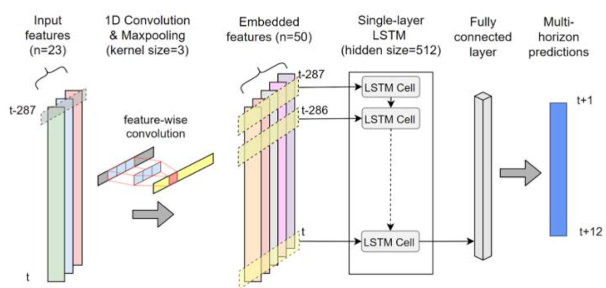

Figure 2: CNN + LSTM model architecture

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \quad (1)$$

$$PIR = \frac{MAE_{baseline} - MAE_{new}}{MAE_{baseline}} \times 100\% \quad (2)$$

4 EXPERIMENT AND RESULTS

4.1 Model Training

Deep learning model performance are highly influenced by hyperparameters settings; with this in mind, we used the Optuna hyperparameter optimization framework [1] to search for the best hyperparameters. The goal is to find the combination of hyperparameters that lead to the best MAE on a randomly selected subset from the whole dataset (10 homes). Among the hyperparameters selected for the optimization include neural network architecture (number of LSTM layers and convolutional kernel size), learning rate with a cosine annealing scheduler to gradually decrease the value, and Adam optimizer. The hyperparameter search space and training configurations for machine learning are shown in Table 2. The optimization process then starts with random sampling from the search space, and tries to improve using an evolutionary optimization approach minimizing a loss function (mean squared error (MSE)). The analysis highlighted how model architecture parameters: CNN kernel size, LSTM hidden size, together with learning rate, have the greatest influence on model performance, while the other hyperparameters only have marginal impacts. Therefore, we chose a single-layer LSTM with no dropout for the sequential model.

Hyperparameter	Distribution	Range	Selected
learning rate	log uniform	[2e-4, 2e-2]	2e-3
Adam optimizer weight decay	log uniform	[1e-6, 1e-4]	1e-5
Conv1D kernel size	discrete with step=32	[32, 256]	50
LSTM number of layers	discrete with step=1	[1, 4]	1
LSTM hidden size	discrete with step=128	[128, 1024]	512
LSTM dropout probability	discrete with step=0.1	[0, 0.8]	0
batch size	discrete with step=128	[128, 1024]	512
number of epochs	N.A.	N.A.	60

Table 2: Hyperparameter search space and selected values

We trained our models with an NVIDIA Titan RTX graphic card with 24GB graphics RAM, using mixed precision training with half precision floating point numbers enabled by PyTorch's automatic mixed precision (AMP) package, which provided 30% speedup compared with full-precision training, leading to a simulation time of minutes per model. Furthermore, due to the relatively simple CNN feature extraction, we did not observe significant time differences in training the vanilla LSTM and CNN-LSTM model.

4.2 Results and Discussions

We assessed the performance of machine learning models using the 10% test data discussed in section 3.2. The dataset contains 48 apartments, 56 townhouses, and 60 single family houses in California, Texas, and New York. We evaluated the performance of machine learning models from two aspects: (1) comparison of vanilla LSTM and CNN-LSTM model performance in general and by different prediction horizon, (2) prediction accuracy of CNN-LSTM models by different seasons, building locations, types, and HVAC system configurations.

Figure 3 shows the comparison of MAE distribution of different prediction horizons between the vanilla LSTM and CNN-LSTM models. It can be seen that except for the first three prediction steps (t+1 t+3), CNN-LSTM models achieved lower average MAE than vanilla LSTM, with 6.6% overall PIR for all prediction steps. Furthermore, the standard deviation of MAE of CNN-LSTM models are 22% lower than vanilla LSTM models except for the first prediction step (t+1), meaning they achieved more consistent performance than vanilla LSTM for most prediction steps. In summary, CNN-LSTM models performed better than vanilla LSTM models especially for longer-term predictions.

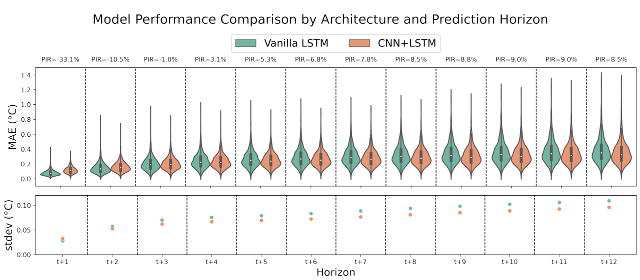

Figure 3: Vanilla LSTM vs CNN-LSTM performance comparison by prediction horizon

We then looked into the robustness of the CNN-LSTM models by breaking down the performance on all buildings by different seasons, locations, building types, and whether there is electric auxiliary heating in the building. The results shows that 98% of the CNN-LSTM models have an average MAE of less than 0.5 °C for the entire prediction horizon, which are adequate for applications such as thermal load prediction and optimal control. The proposed model generalized well as we do not see significant differences among seasons and building characteristics among most models.

5 CONCLUSIONS

In this study, we proposed a deep learning approach for multi-horizon indoor air temperature prediction using large scale smart thermostat data from residential buildings. The dataset we used in this study includes 164 buildings from three U.S. states with three building types and two HVAC system configurations. We developed a data processing pipeline which could support large-scale time-series forecasting and analytics using the ecobee dataset in the future. Overall, our proposed CNN-LSTM models achieved an average MAE of 0.26 °C for 1-hour-ahead (12-step-ahead) predictions, which is 6.6% better than vanilla LSTM models. We also investigated the model performance breakdown by different seasons, building types, locations, and HVAC system configurations. The results suggested that our proposed models can generalize well across residential buildings with different characteristics, maintaining the fast inference speed, that can support model predictive control (MPC) or deep reinforcement learning (DRL) control applications, where such speed is required.

Several future research opportunities exist beyond on this study. Firstly, we will look into DL models with longer-horizon predictions and extra covariates that influence the indoor thermal conditions such as occupancy and weather forecast. More sophisticated model architectures including attention-based seq2se2 models will be evaluated. Furthermore, we can investigate the effectiveness of transfer learning in presence of large amount of real building metadata and thermostat data, whose findings could help model-based controller deployments for buildings with newly connected thermostats.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[2] Furkan Elmaz, Reinout Eyckerman, Wim Casteels, Steven Latré, and Peter Hellinckx. 2021. CNN-LSTM architecture for predictive indoor temperature modeling. *Building and Environment* 206 (2021), 108327.

[3] Enrico Fabrizio and Valentina Monetti. 2015. Methodologies and advancements in the calibration of building energy models. *Energies* 8, 4 (2015), 2548–2574.

[4] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[5] Fatma Mtibaa, Kim-Khoa Nguyen, Muhammad Azam, Anastasios Papachristou, Jean-Simon Venne, and Mohamed Cheriet. 2020. LSTM-based indoor air temperature prediction framework for HVAC systems in smart buildings. *Neural Computing and Applications* 32, 23 (2020), 17569–17585.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[8] Giuseppe Pinto, Davide Deltetto, and Alfonso Capozzoli. 2021. Data-driven district energy management with surrogate models and deep reinforcement learning. *Applied Energy* 304 (2021), 117642.

[9] Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (2020), 132306.

[10] Zequn Wang and Yuxiang Chen. 2019. Data-driven modeling of building thermal dynamics: Methodology and state of the art. *Energy and Buildings* 203 (2019), 109405.

[11] Zhe Wang, Tianzhen Hong, and Mary Ann Piette. 2020. Building thermal load prediction through shallow machine learning and deep learning. *Applied Energy* 263 (2020), 114683.