

Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques

Mengmeng Cai*, Manisa Pipattanasomporn, Saifur Rahman

Advanced Research Institute, Virginia Tech, Arlington, VA, USA

HIGHLIGHTS

- Reduced forecasting errors compared to conventional time-series model.
- Capable of handling high level uncertainty in the building load.
- Multi-step formulated convolutional neural network provides the highest accuracy.
- High computational efficiency is also offered by convolutional neural network.

ARTICLE INFO

Keywords:

Time-series building-level load forecasts
Deep learning
Gating mechanism
Seasonal ARIMAX

ABSTRACT

Load forecasting problems have traditionally been addressed using various statistical methods, among which autoregressive integrated moving average with exogenous inputs (ARIMAX) has gained the most attention as a classical time-series modeling method. Recently, the booming development of deep learning techniques make them promising alternatives to conventional data-driven approaches. While deep learning offers exceptional capability in handling complex non-linear relationships, model complexity and computation efficiency are of concern. A few papers have explored the possibility of applying deep neural networks to forecast time-series load data but only limited to system-level or single-step building-level forecasting. This study, however, aims at filling in the knowledge gap of deep learning-based techniques for day-ahead multi-step load forecasting in commercial buildings. Two classical deep neural network models, namely recurrent neural network (RNN) and convolutional neural network (CNN), have been proposed and formulated under both recursive and direct multi-step manners. Their performances are compared with the Seasonal ARIMAX model with regard to accuracy, computational efficiency, generalizability and robustness. Among all of the investigated deep learning techniques, the gated 24-h CNN model, performed in a direct multi-step manner, proves itself to have the best performance, improving the forecasting accuracy by 22.6% compared to that of the seasonal ARIMAX.

1. Introduction

The growing energy demand has raised concerns worldwide over the issue of environmental degradation. According to the U.S. Energy Information Administration Monthly Energy Review [1], 40% of the total energy consumption comes from buildings. In the smart grid paradigm, a number of innovative techniques have been introduced targeting improvements in both the power grid and building energy efficiency, including demand response (DR) [2] and demand-side management [3,4]. Buildings have been transformed from pure customers into prosumers who actively participate in power grid operation, providing various kinds of grid services, including peak demand curtailment [5,6], operation reserve [7] and frequency regulation [8].

Such a transformation has led to the shifted interest from system-level load forecasts to building-level load forecasts [9]. From the perspective of electric utilities, accurate building-level load forecasts ensure the effectiveness of both pre-DR resource allocation and post-DR performance evaluation (i.e., baseline identification) [10,11]. From the perspective of building owners, understanding how a building load profile fluctuates across time is the first step before conducting a building energy retrofit [12].

However, compared with the system-level load forecasting, an accurate building-level load prediction can be quite challenging, especially for those buildings with a large proportion of Heating, ventilation, and air-conditioning (HVAC) loads and relatively irregular usage patterns [13]. Apart from the complex nature of building thermal

* Corresponding author.

E-mail address: mmcai@vt.edu (M. Cai).

<https://doi.org/10.1016/j.apenergy.2018.12.042>

Received 2 October 2018; Received in revised form 30 November 2018; Accepted 9 December 2018

Available online 20 December 2018

0306-2619/© 2018 Elsevier Ltd. All rights reserved.

Nomenclature

\bar{Act}	mean value of the actual building load series	p	non-seasonal AR order
Act	actual load value	Pre	predicted load value
b_C	bias of candidate neuron	Q	seasonal MA order
b_f	bias of forget gate	q	non-seasonal MA order
b_i	bias of input gate	r	learning rate
b_o	bias of output gate	S	time span of the repeating seasonal pattern
bs	batch size	W_C	weight of candidate neuron
c_t	memory cell	W_f	weight of forget gate
\tilde{c}_t	candidate memory cell	W_i	weight of input gate
d	non-seasonal differencing	W_o	weight of output gate
D	seasonal differencing	X	input of the model
f_t	forget gate	x^A	air pressure series
h_t	hidden unit output	x^H	humidity series
i	index of observations in the testing dataset, $i \in [1, N]$	x^L	building load series
i_t	input gate	x_t^L	load value at time t
j	index of hours, $j \in [1, n]$	x^O	outdoor temperature series
ks	kernel size	x_t^O	outdoor temperature at time t
L	layer structure	x^W	wind speed series
m	length of the input series X , equals to 24	y	output of the model
N	number of observations in the testing dataset	z_t	white noise at time t
n	length of the output vector y , equals to 1 or 24	Φ_k	coefficient of the seasonal AR backshift operation at order $k \in [1, P]$
NN	number of neurons in each layer	ϕ_k	coefficient of the non-seasonal AR backshift operation at order $k \in [1, p]$
$O_{Conv1D_2 \times k}$	final output from the Conv1D $2 \times k$ layers	Θ_k	coefficient of the seasonal MA backshift operation at order $k \in [1, Q]$
$O_{Conv1D_2 \times 1}$	final output from the Conv1D 2×1 layers	θ_k	coefficient of the non-seasonal MA backshift operation at order $k \in [1, q]$
O_G	output from the linear gates		
o_t	output gate		
P	seasonal AR order		

physics, a large amount of uncertainties also complicate the problem. While weather variance impacts the building load externally, irregular occupants behavior adds even more randomness to the load internally [14].

Luckily, buildings nowadays are not only energy-intensive but also information-intensive due to the deployment of building automation system (BAS) [15,16]. Such plentiful building-level data help with the data-driven analysis of building load behaviors. High-speed evolution of analytics tools, on the other hand, continuously increases the effectiveness of data-driven models. Most existing load forecasting models use non-DL techniques, ranging from the simplest multiple linear regression (MLR) [17,18], autoregressive integrated moving average (ARIMA) [19,20], support vector regression (SVR) [21,22] to the most sophisticated artificial neural network (ANN) [23,24]. Hybrid prediction model combining the capabilities of these non-DL techniques is also discussed in [25]. Apart from load forecasting, these techniques have also been widely applied for predicting solar PV generation [26], wind generation [27] and electricity price [28]. With the huge success DL techniques made in solving complex statistical problems, researchers also started searching for the DL-driven solutions for load forecasting applications [29,30]. Essentially, deep neural networks boost the power of ANN via deepening its layers and leveraging its structures. These learning methods have been widely implemented to solve natural language processing and speech/image recognition problems. However, there is a limited discussion on topics related to time-series day-ahead building-level load forecasts.

Within the non-DL scope, ARIMAX has overall competitive advantages for load forecasting. Compared to MLR, it is more capable of capturing the temporal dependency; compared to SVR and ANN, it shows better interpretability. Unlike other supervised learning models, ARIMAX is tailored for time-series modeling, where the sequence of inputs matters. As the counterpart of ARIMAX in the DL community, recurrent neural network (RNN) is used to tackle datasets with sequential correlations. Authors in [31] proposed two RNN structured

models for medium-to-long term predictions. Authors in [32] proposed a pooling deep recurrent neural network (PDRNN) model to forecast household load. Nevertheless, computational efficiency was not compared. Also, this method requires an access to neighbors' load data, which is not practical. Authors in [33] innovatively created an autoregressive-like double-CNN model for asynchronous time-series forecasts. Experimental results demonstrated its predominant capability at handling various kinds of asynchronous datasets, including household load data. However, prior knowledge on the next day's weather condition is missing in the problem context [34]. Also, authors in [33] only analysed the performance of single-step building-level load forecasts, yet did not examine multi-step performance. The application of deep learning on feature extraction was also discussed in [35], indicating a significant improvement in prediction accuracy after applying an unsupervised deep learning model. It is also found in [35] that supervised deep learning did not show obvious advantages over other conventional tools. However, the discussion was only limited to the perceptron deep neural network, excluding the families of RNN and CNN networks.

Based on the literature outlined above, this study proposes DL-driven models specific for day-ahead building-level load forecasts. Only load forecasts on weekdays are discussed as most DR events happen on weekdays. Contributions made in this study can be summarized as follows:

1. Designed the gated RNN/CNN deep learning models for day-ahead building-level load forecasts with knowledge of next-day weather prediction.
2. Formulated the day-ahead building-level load forecasting problem under both: direct multi-step and recursive multi-step manners.
3. Thoroughly analyzed properties of Seasonal ARIMAX (SARIMAX), gated RNN and gated CNN models from aspects of accuracy, computational efficiency, generalizability and robustness.

The paper is organized as follows: Section 2 discusses the design and

development of SARIMAX, gated RNN and gated CNN models. Section 3 briefly introduces the experimental setup, including the real-world building load datasets, feature selection, data pre-process and model configuration. Case studies are discussed in Section 4 with detailed performance analysis. Conclusions and suggested future work are summarized in Section 5.

2. Methodology

All of the models compared in this study fall into the time-series category. The fact that time-series methods do not require additional time indexing parameters differentiates them from other supervised learning methods, e.g., MLR, SVR, and ANN, when handling time-series data. As a result, time-series methods are able to detect the time dependency (including hour-of-day and day-of-week) inherently embedded in the input data and avoid potential issues brought about by inappropriate time-index labeling. Candidate models discussed in this paper are the most representative time-series methods from both DL and non-DL scopes: the SARIMAX, the gated RNN and the gated CNN models.

2.1. Problem formulation

To ensure a fair comparison, the three investigated models are formulated into the same supervised learning framework. Where there is an input matrix X that integrates the information of historical load profile (x^L) and outdoor temperature profile (x^O), and also an output vector y that refers to the predicted load profile with configurable prediction horizon. In this study, only the outdoor temperature is selected as the weather relevant feature as it shows a dominant influence towards building load in comparison with other weather relevant variables (i.e., air pressure, humidity and wind speed). Detailed clarification for such selection is provided in Section 3.2. The day-ahead building-level load forecast is basically a multi-step forecasting problem. It requires the model to predict the next 24-h load profile at midnight (12 AM) of each day given the historical load behaviors and prior knowledge about the next day's weather forecast.

The day-ahead building-level load forecasting can be performed in two different ways: recursive and direct.

- For the recursive multi-step, a one-hour prediction model is developed and implemented recursively 24 times to yield the day-ahead load forecast. The predicted value from the previous time step is fed as one of the inputs to the prediction model of the subsequent step.
- For the direct multi-step, on the other hand, a 24-h prediction model is developed to generate the day-ahead load forecast at once.

To fully test the functionality of different time-series models, both the one-hour prediction models and the 24-h prediction models are developed and compared.

2.2. Seasonal ARIMAX model

Seasonal ARIMAX (SARIMAX) identifies the time-series patterns inside the series while capturing the linear covariance between target

variable and exogenous variables. Such time-series patterns and covariance are summarized in Table 1, with regard to building-level load profiles.

The standard seasonal ARIMAX model follows the notation of $ARIMAX(p, d, q) \times (P, D, Q)S$, where p = non-seasonal auto-regressive (AR) order, d = non-seasonal differencing, q = non-seasonal moving average (MA) order, P = seasonal AR order, D = seasonal differencing, Q = seasonal MA order, S = time span of the repeating seasonal pattern. Let x_t^L denotes the load value at time t , then the ARIMAX model can be mathematically expressed in Eqs. (1) and (2) with the backshift operator B . Where z_t and x_t^O represent the white noise and the exogenous outdoor temperature covariate at time t , respectively. $\nabla_s x_t^L = x_t^L - x_{t-s}^L$ and $\nabla x_t^L = x_t^L - x_{t-1}^L$ represent differencing operations.

$$\Phi(B^S)\phi(B)\nabla_s^D \nabla^d x_t^L = \beta x_t^O + \Theta(B^S)\theta(B)z_t \tag{1}$$

Where

$$\Phi(B^S) = 1 - \Phi_1 B^S - \dots - \Phi_P B^{PS}$$

$$\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$$

$$\Theta(B^S) = 1 + \Theta_1 B^S + \dots + \Theta_Q B^{QS}$$

$$\theta(B) = 1 + \theta_1 B + \dots + \theta_q B^q \tag{2}$$

2.3. Gated RNN model

RNN is a class of neural networks that are able to recurrently process sequential inputs. Such a capability is enabled by the internal time loops at each hidden layer unit, where the output of the unit at time step t is taken as the input for the next step $t + 1$. Long short-term memory units (LSTMs) have been proposed for addressing the gradient vanishing problem faced by vanilla RNN hidden units [36]. The intuition behind this approach is to maintain the memory property of the vanilla RNN units while being able to filter out redundant or misleading information through a long short-term gating mechanism.

Apart from the original hidden unit output (h_t) that is utilized to restore short-term memory immediately passed from previous time step, an internal memory cell (c_t) is introduced for restoring the long-term memory. Unlike the vanilla RNN unit, which has only one neuron, there are four neurons in the LSTM unit. One neuron works as a tangent function providing the candidate memory cell (\tilde{c}_t), computed using Eq. (3); the other three neurons work as sigmoid functions, controlling the flow of information, i.e., forget gate (f_t), input gate (i_t), and output gate (o_t), as given in Eq. (4). Based on the outputs of three gating functions, the internal memory cell (c_t) and the hidden layer output (h_t) subsequently update themselves according to Eq. (5). Where x_t is the input vector at each time step; W_c, W_f, W_i, W_o and b_c, b_f, b_i, b_o represent the weights and biases of the candidate neuron, the forget gate, the input gate, and the output gate.

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Table 1
Time-series characteristics of building-level load profiles.

Category	Name	Definition
Patterns	Trend	Long-term trending behaviors describing how the overall power consumption increases or decreases
	Periodicity	Periodical patterns of load profiles that mainly result from the seasonal energy usage preference and daily routine of occupants
	Temporal dependence	Correlation between the lagged load values and future load values
	Uncertainty	Randomness resided in load profiles due to non-routine behaviors of occupants
Covariance	Weather covariance	A part of load pertinent to space cooling/heating needs, which is sensitive to the outdoor temperature variance

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{4}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

$$h_t = o_t * \tanh(c_t) \tag{5}$$

Fig. 1 illustrates the architecture of the gated RNN model after being unrolled along the time axis. Where m and n denote the lengths of lagged values and prediction horizon; y_1, \dots, y_n denote forecasted load values at each time step. Note that weather prediction information is appended at the end of the input series (highlighted in yellow circles in Fig. 1), which differentiates this study from other existing work.

Unlike the seasonal ARIMAX model which only works as a one-hour prediction model, one advantage of the DL-based models is their flexibility on the lengths of input and output vectors. Parameters m and n in Fig. 1 can be set to 24 and 1 to obtain the one-hour gated RNN model (i.e., GRNN1). On the other hand, to obtain the 24-h gated RNN model (i.e., GRNN24), both m and n can be set to 24.

2.4. Gated CNN model

CNN is made up of neurons that apply convolution computation to the inputs hierarchically. It is invented for image processing. The first time when CNN was used to solve sequential problems was in [37]. After introducing a novel gating mechanism, the developed gated CNN model is able to outperform RNN for the language sequential modeling task. Inspired by the autoregressive model and work done in [37], authors in [33] designed a double-channel gated CNN model, called Significance Offset CNN (SOCNN). This model is taken as the prototype for designing the gated CNN model in this study. Following modifications have been made to fit the context of the day-ahead building-level load forecasts:

1. Synchronous input series: The input variables are embedded in a synchronous manner instead of asynchronous manner.
2. Concatenated weather prediction: Weather prediction information is appended to the end of historical series (highlighted in yellow circles in Fig. 2). It is a $2 \times n$ matrix with the first row filled with all zeros and the second row filled with predicted outdoor temperature values.

Fig. 2 illustrates the structure of the gated CNN model designed for day-ahead building-level load forecasts. The formats of input variables (X) and the output variable (y) are identical to that of the gated RNN model. According to Fig. 2, each input variable is processed simultaneously using two different convolutional operations, either operated together with the neighboring time stamps for detecting the temporal dependency or operated alone for identifying the weather correlation at

each time stamp. What differentiates these two operations is the kernel size of the filters: They are Conv1D filters with size $2 \times k$ (marked by blue triangles in Fig. 2) and Conv1D filters with size 2×1 (marked by blue thick arrows in Fig. 2). On top of the convolutional layers are the linear gates following the gating function in Eq. (6), where $O_{Conv1D_{2 \times k}}$ and $O_{Conv1D_{2 \times 1}}$ represent outputs of the last Conv1D $2 \times k$ layer and the last Conv1D 2×1 layer.

$$O_G = \sigma(O_{Conv1D_{2 \times k}}) \otimes O_{Conv1D_{2 \times 1}} \tag{6}$$

Then the output from linear gates (O_G) is mapped into the final output series (y) through a fully connected neural network layer, see Eq. (7). Where $W_{(m+n) \times n}$ and b represent the weight and bias of the fully connected layer.

$$y = W_{(m+n) \times n} O_G + b \tag{7}$$

Note that a $k - 1$ length padding (marked by dashed rectangular frames in Fig. 2) has been applied to the input sequences at each layer in order to make sure that $O_{Conv1D_{2 \times k}}$ and $O_{Conv1D_{2 \times 1}}$ share the same dimension.

The whole framework works as a data-dependent autoregressive model, where the Conv1D 2×1 filters provide the regressor at a single time step, and the Conv1D $2 \times k$ filters generate the coefficient for each regressor. Unlike the fixed coefficient applied in the autoregressive model, the output of Conv1D $2 \times k$ filters varies as the input data change. Therefore, the gated CNN model is much more capable of capturing the non-linear temporal relationship exist in the data.

Similarly to the gated RNN model, the one-hour gated CNN model (i.e., GCNN1) is obtained by setting $m = 24$ and $n = 1$; The 24-h gated CNN model (i.e., GCNN24) is obtained by setting both m and $n = 24$.

3. Experiment setup

Performances of the above five models (i.e., SARIMAX, GRNN1, GCNN1, GRNN24 and GCNN24) have been compared using case studies of three commercial buildings. Variances on locations and consumption levels of these buildings help substantiate the models' generalizability.

3.1. Datasets

Datasets used in this study were collected in one-hour intervals for a one-year period with 5% of missing data, consisting of five time series: hourly electrical load (kW), outdoor temperature ($^{\circ}$ F), air pressure (in), humidity (%) and wind speed (mph). Note that all buildings under study are gas-heated. Since hourly natural gas consumption data are not available, this paper focuses on predicting the electricity demand.

Building A, located in Alexandria, VA, is an academic building with the area of around 30,000 square feet. Buildings B, located in Shirley,

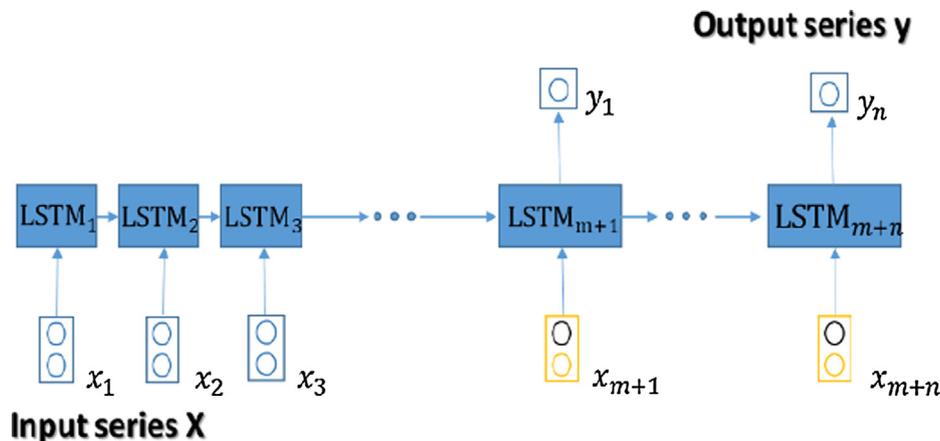


Fig. 1. Gated RNN model for day-ahead building-level load forecasts.

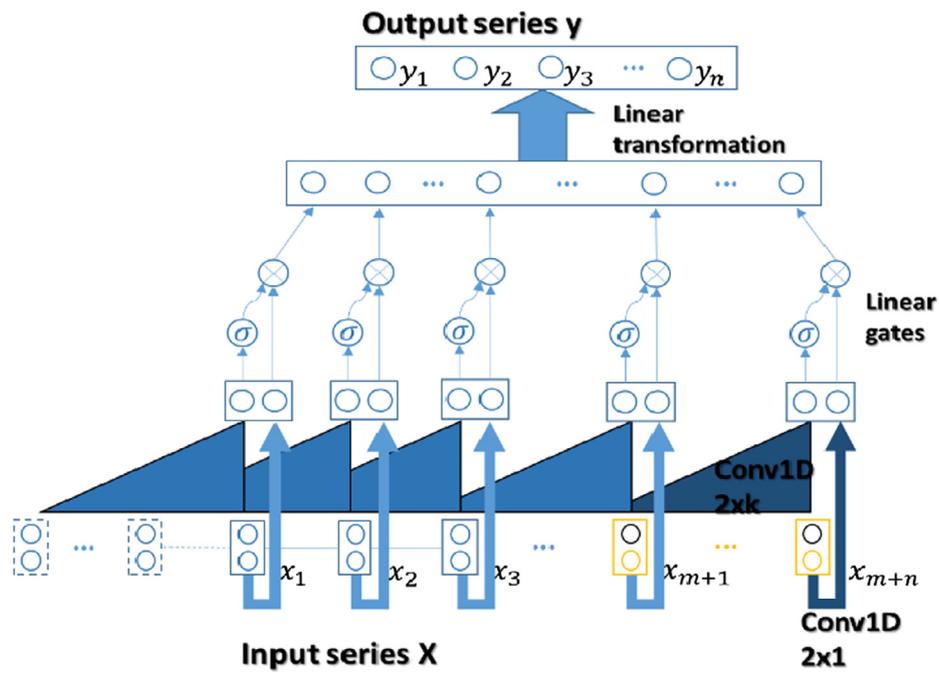


Fig. 2. Gated CNN model for day-ahead building-level load forecasts.

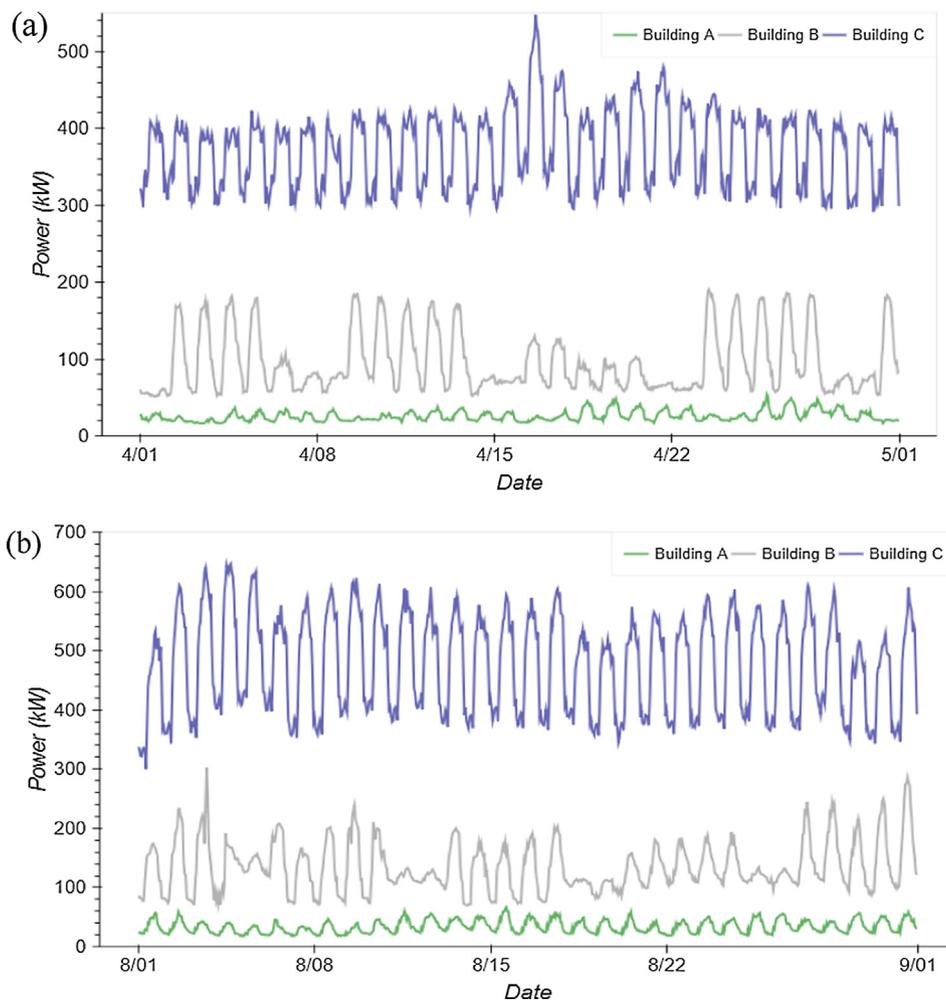


Fig. 3. Load profiles of three buildings in (a) April and (b) August.

NY, is a primary/secondary school with the area of around 80,000 square feet. And, Building C, located in Uxbridge, MA, is a grocery store with the area of around 55,000 square feet. Building A’s dataset was collected through the BEMOSS project [38]. Buildings B and C’ datasets are provided by the public EnerNOC Commercial building dataset [39]. Weather data were fetched from the *weather underground website* [40]. Due to different building load characteristics when HVAC is set to different modes (e.g., COOL or OFF), datasets of each building were divided into: the COOL period (i.e., HVAC mode is set to COOL, May–October) and the NON-COOL period (i.e., HVAC mode is set to OFF, November–April). Peak electricity demands of buildings A, B and C are approximately 40 kW, 180 kW and 450 kW during the NON-COOL period, and 60 kW, 380 kW and 650 kW during the COOL period. Fig. 3 depicts load profiles of these three buildings in April (NON-COOL) and August (COOL).

According to Fig. 3, daily load patterns of each building are relatively constant in April (due to weak weather covariance), whereas vary greatly in August (due to strong weather covariance). There is a trend of decreased share of uncertainty when the building load-scale increases. Despite the relatively constant daily load patterns exhibited by buildings B and C, there still exist some unpredictable irregular patterns.

3.2. Weather relevant feature selection

In order to select the most relevant weather indicators as the model features, a feature selection work is conducted prior to model construction. Here the weather variables accessible from weather underground website are considered as candidate features. These are outdoor temperature (x^O), air pressure (x^A), humidity (x^H) and wind speed (x^W). Pearson correlation coefficient is applied as the score function due to its widely recognized ability of measuring the correlation between two continuous variables. It can be computed based on Eq. (8), where cov represents the covariance, $\sigma_{x^{weather}}$ and σ_{x^L} are the standard deviations of any weather variable ($x^{weather}$) and building load series (x^L) respectively. Table 2 summarizes the Pearson correlation coefficient statistics between each weather variables and building load for buildings A–C. According to Table 2, a strong positive correlation is found between outdoor temperature and building load across all buildings, whereas the correlation between the other weather variables (i.e., air pressure, humidity and wind speed) and building load are either zero or insignificant (with absolute values of less than 0.3). Adding these weakly correlated variables into the feature set will tend to overfit the model. Therefore, in this study, only outdoor temperature (x^O) is selected as the weather relevant feature.

$$\rho_{x^{weather}, x^L} = \frac{cov(x^{weather}, x^L)}{\sigma_{x^{weather}} \sigma_{x^L}}, x^{weather} \in \{x^O, x^A, x^H, x^W\} \quad (8)$$

3.3. Data preprocessing

The following data-preprocessing procedures have been implemented:

Data Cleaning: In order to eliminate the influence of missing data, list-wise deletion was conducted. Raw datasets were then divided into multiple sections bounded by the missing points so as to maintain the continuity of time. Then, weekend data were removed, since the irregular load pattern during weekends has proven to have significant influence on the prediction accuracy on weekdays.

Data Segmentation: After filtering out the missing data and weekend profiles, cleaned datasets for each building were then segmented into: training dataset (90%), validation dataset (5%) and testing dataset (5%).

Time-series to supervised learning dataset: All training, validation and testing datasets were converted into the supervised learning format, where time-series sequences became input-output pairs through

the process of sliding windows.

Normalization: Finally, to stabilize the learning process, input variables in each training dataset, together with corresponding validation and testing datasets, were carefully normalized. Normalization helps prevent dramatic changes on the gradient, so as to smoothen the convergence.

3.4. Model validation

After all datasets were properly preprocessed, model parameters (i.e., weights and bias in DL-based models) and hyper-parameters (i.e., layers and number of neurons in DL-based models) were tuned using training and validation datasets. Once the optimal model parameters (different for different buildings) and hyper-parameters (same for each building) were obtained, testing datasets of each building were fed into the optimized models for comparing the performance. The Train-Validation-Test split was applied here to verify the capability of each model to be generalized to unseen data.

3.5. Algorithm implementation

All experiments were carried out in the Python compiling environment using an Intel Core i7-4770 CPU machine. Implementation of the SARIMAX model relied on the Python package: StatsModels. All DL-based models were developed using the TensorFlow [41].

Optimized hyper-parameters used in each model are summarized in Table 3. Where L , NN , r , bs , and ks represent the layer structure, number of neurons in each layer, learning rate, batch size, and kernel size (only available in CNN models), respectively. Adam optimizer was applied to all of the DL-based models.

4. Performance comparison

High prediction accuracy, high computational efficiency, robustness against weather forecast error and generalizability are considered as the most important qualities of a day-ahead load forecasting model. In this study, all above properties have been compared.

4.1. Accuracy of one-hour prediction models

Accuracy of the one-hour prediction models is presented in Tables 4 and 5, summarizing mean absolute percentage error (MAPE) and coefficient of variance (CV) of the single-step predictions. MAPE and CV were calculated using Eqs. (9) and (10), where Pre_i and Act_i represent the predicted and actual loads. N is the number of observations in the testing dataset. \bar{Act} is the mean of actual values.

$$MAPE = \frac{\sum_{i=1}^N \left| \frac{Pre_i - Act_i}{Act_i} \right|}{N} \times 100 \quad (9)$$

$$CV = \frac{\sqrt{\frac{\sum_{i=1}^N (Pre_i - Act_i)^2}{N-1}}}{\bar{Act}} \times 100 \quad (10)$$

The best performance among all three models is observed in Building C — the building with the largest peak load (450 kW/650 kW

Table 2
Pearson correlation coefficients between different weather relevant variables and building load.

	Building A (40–60 kW _p)	Building B (180–380 kW _p)	Building C (450–650 kW _p)
Outdoor Temperature	0.74	0.51	0.76
Air Pressure	0	0	0
Humidity	−0.26	−0.07	−0.10
Wind speed	−0.02	−0.04	0.02

Table 3
Model configuration.

non-DL model							
Model	<i>p</i>	<i>d</i>	<i>q</i>	<i>P</i>	<i>D</i>	<i>Q</i>	<i>S</i>
SARIMAX	24	1	0	0	1	0	24
DL models							
Model	<i>L</i>	<i>NN</i>	<i>r</i>	<i>bs</i>	<i>ks</i>		
GRNN_1	2	3	0.005	50	–		
GCNN_1	3 conv + 3 conv	8,5,1 + 8,5,1	0.005	50	6,3,3 + 1,1,1		
GRNN_24	3	32	0.005	50	–		
GCNN_24	3 conv + 3 conv	10,8,1 + 10,8,1	0.005	50	6,3,3 + 1,1,1		

Table 4
MAPE (%) of one-hour prediction models under different testing cases.

	Building A (40–60 kW _p)		Building B (180–380 kW _p)		Building C (450–650 kW _p)	
	NON-COOL	COOL	NON-COOL	COOL	NON-COOL	COOL
SARIMAX	13.54	16.13	10.57	8.04	2.95	3.85
GRNN1	8.67	8.99	7.91	5.95	2.36	2.41
GCNN1	8.02	8.53	5.73	5.85	2.23	2.38

Table 5
CV (%) of one-hour prediction models under different testing cases.

	Building A (40–60 kW _p)		Building B (180–380 kW _p)		Building C (450–650 kW _p)	
	NON-COOL	COOL	NON-COOL	COOL	NON-COOL	COOL
SARIMAX	18.71	22.12	13.40	11.14	3.39	5.28
GRNN1	12.20	12.01	8.63	8.10	2.99	3.15
GCNN1	10.97	10.86	6.85	7.68	2.66	3.03

during NON-COOL/COOL period). This building has MAPE values of less than 2.5% and 4.0% for one-hour DL-based models and the SARIMAX model. As the building load-scale become smaller, performances of all three models keep going down (see results for Buildings A and B). It is as expected because larger buildings usually have more occupants. As there are more people in a building, it is more likely that the uncertainty of their aggregated behavior will be averaged, leading to a more regular and predictable building load pattern.

For all three buildings being tested, the SARIMAX model performs significantly poorer than the DL-based models, and its model performance deteriorates significantly as the building peak load goes down. The performance of one-hour DL-based models is not heavily degraded compared to that of the SARIMAX for smaller buildings, as MAPE values remain at around 9% (compared to 17% of the SARIMAX) when dealing with the building with peak load of less than 60 kW. This indicates a much better generalizability of the one-hour DL-based models as compared to the SARIMAX model. Within the DL scope, it is also found that the GCNN1 model performs slightly better than the GRNN1 model and contributes the best performance for all cases. It brings down the MAPE errors of the SARIMAX model by 37.0% and 37.5% for NON-COOL and COOL cases on average.

4.2. Accuracy of 24-h prediction models

Accuracy of the 24-h prediction models are analyzed in Fig. 4, where the boxplots of RMSE tested on buildings A, B and C are shown in subplots Fig. 4(a)–(c). Since the SARIMAX model can only be formulated as the one-hour model, it is not considered in this subsection. Only two 24-h prediction models (i.e., GRNN24 and GCNN24) are

compared. Output of the 24-h model is a vector instead of a scalar as in the case of the one-hour model. Therefore, RMSE shown in Eq. (11) is applied to evaluate their prediction accuracy. Where *n* = 24 represents length of the output vector.

$$RMSE_i = \sqrt{\frac{\sum_{j=1}^n (Pre_j - Act_j)^2}{n}} \tag{11}$$

According to Fig. 4, the accuracy of the GRNN24 model is unsatisfactory, as the mean RMSE values of the GRNN24 model for all COOL cases are 6.6 kW, 25.0 kW and 44.0 kW of buildings A, B and C, all being above 6.6% of the building peaks. With lower RMSE means and variances, the GCNN24 performs much better compared to the GRNN24. Averaged RMSE values of the GCNN24 model are 4.0 kW, 13.4 kW and 15.2 kW for buildings A, B and C, which are around 6.7%, 3.5% and 2.3% of the building peaks, respectively.

4.3. Performance comparison of day-ahead building-level load forecasts

In this subsection, all developed one-hour/24-h prediction models are compared when performing the day-ahead building-level load forecasts. Fig. 5 plots the forecasting results of different models (including SARIMAX, GRNN1, GCNN1, GRNN24 and GCNN24) using one NON-COOL week and one COOL week of buildings A, B and C (only for weekdays). Averaged RMSE values for all five testing weekdays are summarized in Table 6. For each case, the experiment was conducted to perform day-ahead forecasts at midnight for five consecutive weekdays.

According to Fig. 5, despite the poorer one-hour prediction accuracy of the SARIMAX model compared to that of the one-hour DL-based models, it generates better multi-step forecasting results for all NON-COOL cases than the one-hour DL-based models. However, GCNN24 outperforms it for NON-COOL cases of Buildings A and B. Also, its performance on COOL cases expose its sensitivity to both outdoor temperature covariance and load uncertainty. For outdoor temperature covariance, it is observed that when there is a strong weather

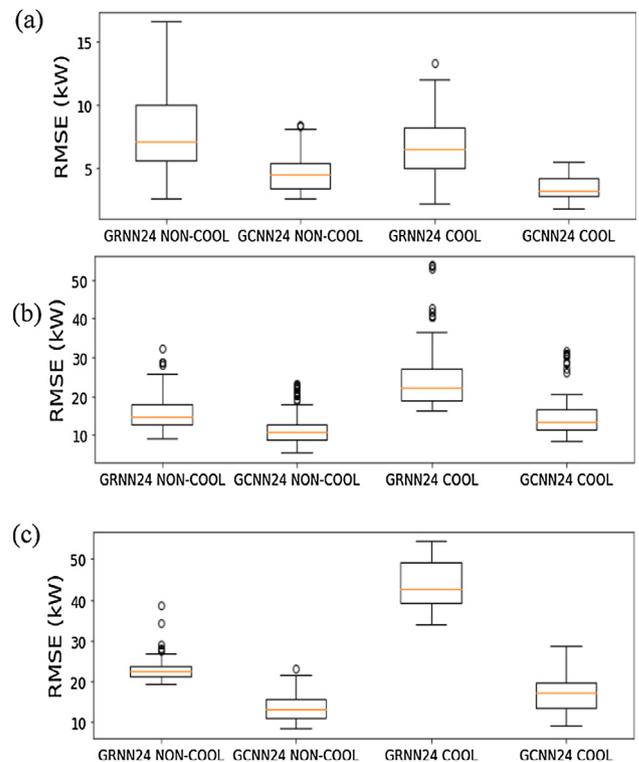


Fig. 4. Comparison of the RMSE boxplots of 24-h DL-based models across all scenarios: (a) Building A-40/60kW_p; (b) Building B-180/380kW_p; (c) Building C-450/650kW_p.

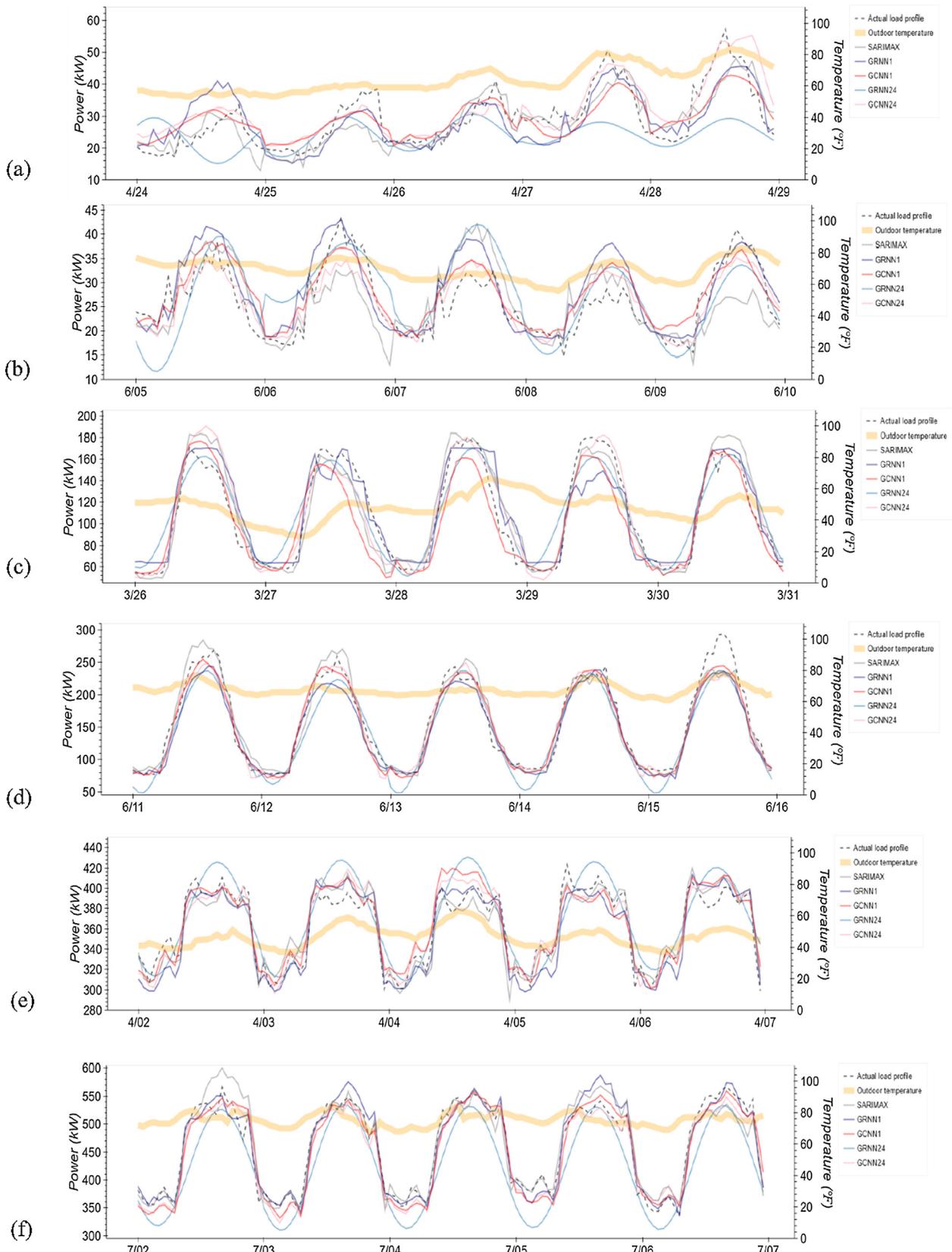


Fig. 5. Comparing day-ahead load forecasting results of different models (SARIMAX, GRNN1, GCNN1, GRNN24 and GCNN24). (a) Building A NON-COOL week; (b) Building A COOL week; (c) Building B NON-COOL week; (d) Building B COOL week; (e) Building C NON-COOL week; (f) Building C COOL week.

Table 6
Comparison of the averaged RMSE (kW) for testing datasets.

Building & Case	SARIMAX	GRNN1	GCNN1	GRNN24	GCNN24
A NON-COOL	5.2	5.5	5.2	9.0	5.1
	COOL	5.7	4.7	3.8	5.9
B NON-COOL	14.3	15.3	15.6	15.8	12.6
	COOL	17.5	17.1	14.3	23.1
C NON-COOL	11.4	12.8	13.0	21.4	12.5
	COOL	21.2	17.9	19.0	45.0

*Best performances of each case are highlighted in bold.

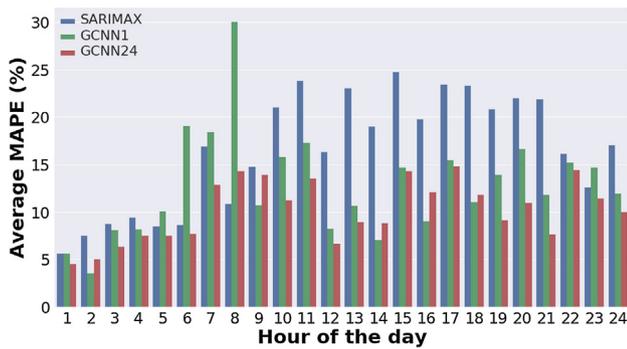


Fig. 6. Hour-of-day indexed error analysis for SARIMAX, GCNN1 and GCNN24.

Table 7
Computation Time for Each Model (in Seconds).

One-hour models	24-h models		
SARIMAX	87		
GRNN1 (850 epochs)	123	GRNN24 (1000 epochs)	139
GCNN1 (400 epochs)	92	GCNN24 (240 epochs)	80

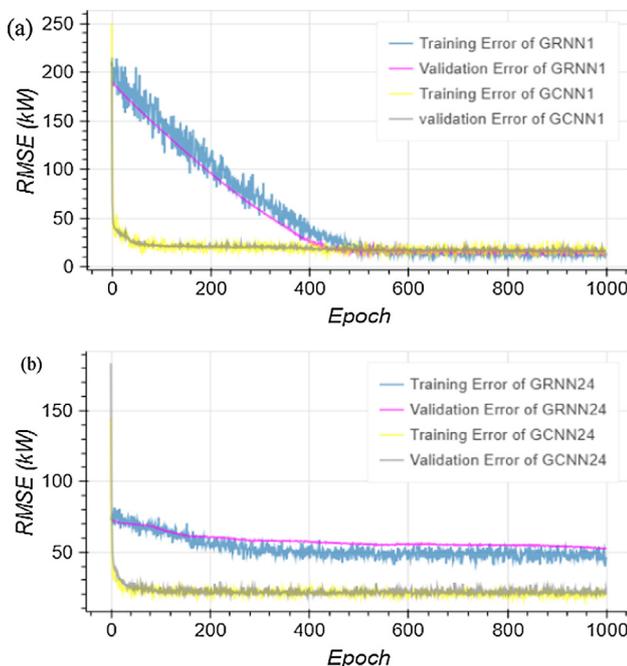


Fig. 7. Learning curves comparison between GRNN and GCNN models: (a) Learning curves of one-hour prediction models (GRNN1, GCNN1); (b) Learning curves of 24-h prediction models (GRNN24, GCNN24).

covariance (COOL period), the day-ahead load forecasts generated by the SARIMAX model heavily deviates from the actual values. Such influence, on the other hand, can be successfully handled by the DL-based

Table 8
Averaged RMSE (kW) for five testing days under different noise levels.

Percentage of introduced noise	GCNN24	GRNN1	GCNN1
0%	17.26	17.88	18.99
20%	17.62	18.21	19.53
40%	17.55	18.38	19.23
60%	17.84	18.52	19.47

models. For load uncertainty, the performance gap between the SARIMAX model and DL-based models for both NON-COOL and COOL cases are found to be the largest for Building A, which has the lowest peak load and highest share of uncertainty. This is intuitively understandable, as the SARIMAX model is built on an assumption of linearity, whereas the real temporal relationship and covariance are mostly non-linear. Also, the substantial amount of uncertainty contained in the time-series building load data may heavily degrade the performance of SARIMAX, since regression-based methods assume that both input and output variables follow the Gaussian distribution.

As for the DL models, despite the better one-hour prediction accuracy of the GCNN1 model compared to that of the GRNN1 model, indicated in Section 4.1, the GCNN1 model not necessarily wins the day-ahead load forecasting tasks. Their performances are comparative and it is hard to say which one outperforms the other. As for the 24-h models, the superior performance of the GCNN24 model over the GRNN24 model is observed as already proved in Section 4.2. When comparing across all one-hour and 24-h models, despite the high one-hour prediction accuracy of the one-hour models, their multi-step forecasting results are proven to be slightly worse than that of the GCNN24. This is due to the accumulated errors through recursive operations.

Overall, the GCNN24 model demonstrates the most promising ability of handling the day-ahead building-level load forecasts. The predicted curves provided by it closely follow the load shapes of different weekdays and capture the peak load change against various weather conditions. According to Table 6, GCNN24 outperforms other models for 4 out of 6 cases. It is found that for the GCNN24 model, the RMSE values for the testing cases with peak load of 380 kW or above are approximately 4.2% or less of the building peak. However, when the peak load decreases to 180 kW or less, the RMSE values are increased to be above 5.7% of the building peak.

In order to analyze how the GCNN24 model provides improvement in day-ahead load forecasting as compared to the other models, an hour-of-day indexed error analysis across SARIMAX, GCNN1 and GCNN24 is conducted, taking Building A, COOL case as an example. Averaged MAPE values at each hour of the day of different models are plotted in Fig. 6. As illustrated in Fig. 6, although the hourly prediction errors of the three models are similar in the first hour, these values evolve quite differently across the day. A significant increase in error happens starting from hour 10 when the SARIMAX model is applied. This is mainly due its low single-step prediction accuracy as well as the error accumulation during the recursive operations. By replacing the SARIMAX model with the GCNN1 model, single-step prediction accuracy has been largely improved. However, error accumulation problem still exists as an abrupt error peak is observed during hours 6–8. Finally, by converting the one-hour CNN model into the 24-h model, the model performance is further improved as the histogram of GCNN24 appears to be the most flattened across the day. Instead of fitting the model hour by hour, GCNN24 is able to learn the daily load shape as a whole so as to avoid abrupt deviation from the regular daily load pattern in its prediction. It largely prevents the negative impact of the error accumulation and contributes to a better multi-step performance.

4.4. Computational efficiency

Table 7 summarizes the computation time of each model in seconds,

based on the experiments conducted using the COOL dataset of Building C. Epoch numbers of each DL-based model were determined based on the convergence time shown on the learning curves (Fig. 7). According to Table 7, the GCNN24 model demonstrates the best computational efficiency. Although the SARIMAX shows higher computation efficiency compared to most DL-based models, it is less efficient than the GCNN24 model. Compared to the GRNN models, GCNN models have speeded up the computation by 25–42%.

Fig. 7(a) and (b) specifically compare the learning curves among different DL-based models, using the COOL dataset of building C.

Based on Fig. 7, the following observations are made:

1. With regard to one-hour prediction models, it takes the GCNN1 model less than 400 epochs to converge, while around 850 epochs are required for the GRNN1 model. However, the validation error curve of GRNN1 goes down more smoothly as compared to that of GCNN1.
2. With regard to 24-h prediction models, the GCNN24 model converges at around 240 epochs, but it takes the GRNN24 around 1000 epochs to converge. Also, the validation error curve of GRNN24 goes down more smoothly as compared to that of GCNN24.

Overall, GCNN24 outperforms all other models, including its traditional time-series counterpart SARIMAX, in terms of the computational efficiency. Such an advantage attributes to its capability of parallelizing the computation. Unlike the chain-like structure that the GRNN models apply for processing the time-series data, the GCNN models are able to parallel the sequence and process each section synchronously.

4.5. Robustness analysis

So far it has been proven that the GCNN24 and one-hour DL-based models own the top performances among the investigated models for day-ahead building-level load forecasts. However, the predicted weather profiles used in the testing datasets are assumed to be 100% accurate. An additional concern is raised as the accuracy of next-day weather prediction may not always be guaranteed. In order to examine the robustness of GCNN24, GRNN1 and GCNN1 for handling the day-ahead building-level load forecasts in presence of weather prediction error, a further robustness analysis was conducted.

Three sets of manipulated testing datasets were generated by introducing three different levels of noises into the day-ahead weather data. To be specific, for the five-day hourly weather profile ($5 \times 24 = 120$ data points), 20%, 40% and 60% of the original data points were added with a Gaussian distributed white noise. The GCNN24, GRNN1 and GCNN1 models were re-run on the COOL dataset of Building C using three noise-introduced weather profiles. Table 8 presents the averaged RMSE values for five testing days of GCNN24, GRNN1 and GCNN1 models under different noise levels.

It is shown that with the introduction of up to 60% weather prediction white noise, the averaged RMSE of the GCNN24, GRNN1 and GCNN1 models slightly increase from 17.26 kW to 17.84 kW (by 3.36%), from 17.88 kW to 18.52 kW (by 3.58%) and from 18.99 kW to 19.47 kW (by 2.53%). Therefore, it is verified that all three models: GCNN24, GRNN1 and GCNN1 models are robust against the error in the next-day weather forecasts.

5. Conclusion

In this study, the performances of three different time-series approaches (i.e., SARIMAX, GRNN and GCNN) in the application of day-ahead building-level load forecasts are systematically compared. The value of deep learning techniques in such an application is comprehensively verified.

Two popular deep learning networks, RNN and CNN, have been utilized and constructed into the day-ahead building-level load

forecasting framework. Their performance under both recursive multi-step and direct multi-step forecasting manners are analysed. Experimental results indicate that most of the proposed deep learning-based models (except GRNN24) achieve promising results as compared to its traditional counterpart SARIMAX. From the aspect of prediction accuracy, the day-ahead multi-step forecasting errors of the proposed GCNN24 model decrease by 22.6% on average for cases with strong weather covariance, compared to those of the SARIMAX model. Such decrement reaches 40% when the building load patterns are highly uncertain (i.e., for buildings with smaller peak electrical load). From the perspective of computational efficiency, when the GCNN24 model is applied, the operation time is speeded up by 8% compared to that of the SARIMAX model. From the aspect of generalizability, it is proven that the impact of building load uncertainty on the performances of GRNN1, GCNN1 and GCNN24 is less than that of the SARIMAX model.

Within the deep learning scope, the GCNN24 model outperforms all other deep learning-based models investigated in this study. It offers better accuracy compared to the GCNN1 model, and significant superiority to the GRNN models in both accuracy and computational efficiency. From the practical point of view, the multi-thread design of modern hardware well supports the models with parallelized structure. Therefore, it is expected that the GCNN24 model can play a dominant role in future day-ahead building-level load forecasting work, especially when dealing with load forecasts of a large number of buildings.

To sum up, the emergence of deep learning techniques provides us the opportunity to bring the performance of day-ahead building-level load forecasts to an even higher level. Compared to the conventional approaches, a well-design hierarchically-structured deep learning network may be more capable of capturing the data-dependent uncertainty and may even increase the computational efficiency for large-scale application. The GCNN24 model proposed in this study demonstrates competitive capabilities. It can be applied as a good start for the deep learning-based network investigation of day-ahead building-level load forecasts. For future work, building-level heating demand prediction using the proposed deep learning techniques can be validated when gas consumption data are available. Also, extended models involving the prediction of power consumption of individual load by type can be investigated.

References

- [1] "U.S. Energy Information Administration Monthly Energy Review." [Online]. Available: <http://www.eia.gov/totalenergy/data/monthly/#consumption>.
- [2] Wang F, Xu H, Xu T, Li K, Shafie-khah M, Catalão JPS. The values of market-based demand response on improving power system reliability under extreme circumstances. *Appl Energy* 2017;193:220–31.
- [3] Bian D, Pipattanasomporn M, Rahman S. A human expert-based approach to electrical peak demand management. *IEEE Trans Power Deliv* 2015;30(3):1119–27.
- [4] Shi Q, Li F, Hu Q, Wang Z. Dynamic demand control for system frequency regulation: concept review, algorithm comparison, and future vision. *Electr Power Syst Res* 2018;154:75–87.
- [5] Zhang X, Pipattanasomporn M, Rahman S. A self-learning algorithm for coordinated control of rooftop units in small- and medium-sized commercial buildings. *Appl Energy* 2017;205(June):1034–49.
- [6] Adhikari R, Pipattanasomporn M, Rahman S. An algorithm for optimal management of aggregated HVAC power demand using smart thermostats. *Appl Energy* 2018;217(January):166–77.
- [7] Hui H, Ding Y, Liu W, Lin Y, Song Y. Operating reserve evaluation of aggregated air conditioners. *Appl Energy* 2017;196:218–28.
- [8] Lin Y, Barooah P, Meyn S, Middelkoop T. Experimental evaluation of frequency regulation from commercial building HVAC systems. *IEEE Trans Smart Grid* 2015;6(2):776–83.
- [9] Chitsaz H, Shaker H, Zareipour H, Wood D, Amjadi N. Short-term electricity load forecasting of buildings in microgrids. *Energy Build* 2015;99:50–60.
- [10] Mathieu JL, Price PN, Kiliccote S, Piette MA. Quantifying changes in building electricity use, with application to demand response. *IEEE Trans Smart Grid* 2011;2(3):507–18.
- [11] Chen Y, Xu P, Chu Y, Li W, Wu Y, Ni L, et al. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. *Appl Energy* 2017;195:659–70.
- [12] Behl M, Smarra F, Mangharam R. DR-Advisor: a data-driven demand response recommender system. *Appl Energy* 2016;170:30–46.
- [13] Yildiz B, Bilbao JJ, Sprull AB. A review and analysis of regression and machine

- learning models on commercial building electricity load forecasting. *Renew Sustain Energy Rev* 2017;73(March 2016):1104–22.
- [14] Walter T, Price PN, Sohn MD. Uncertainty estimation improves energy measurement and verification procedures. *Appl Energy* 2014;130:230–6.
- [15] Heimgaertner F, Hettich S, Kohlbacher O, Menth M. Scaling home automation to public buildings: a distributed multiuser setup for OpenHAB 2. In: *GIoTS 2017 - Glob. Internet Things Summit, Proc.*; 2017.
- [16] Zhang X, Adhikari R, Pipattanasomporn M, Kuzlu M, Bradley SR. Deploying IoT devices to make buildings smart: Performance evaluation and deployment experience. In: *2016 IEEE 3rd world forum internet things, WF-IoT 2016*; 2017. p. 530–35.
- [17] Papalexopoulos AD, Hesterberg TC. A regression-based approach to short-term system load forecasting. *IEEE Trans Power Syst* 1990;5(4):1535–47.
- [18] Vu DH, Muttaqi KM, Agalgaonkar AP. Short-term load forecasting using regression based moving windows with adjustable window-sizes. In: *Ind. Appl. Soc. Annu. Meet.*; 2014. p. 1–8.
- [19] Lee CM, Ko CN. Short-term load forecasting using lifting scheme and ARIMA models. *Expert Syst Appl* 2011;38(5):5902–11.
- [20] Yun K, Luck R, Mago PJ, Cho H. Building hourly thermal load prediction using an indexed ARX model. *Energy Build* 2012;54:225–33.
- [21] Jain RK, Smith KM, Culligan PJ, Taylor JE. Forecasting energy consumption of multi-family residential buildings using support vector regression: investigating the impact of temporal and spatial monitoring granularity on performance accuracy. *Appl Energy* 2014;123:168–78.
- [22] Fan C, Xiao F, Wang S. Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. *Appl Energy* 2014;127:1–10.
- [23] Yu F, Xu X. A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network. *Appl Energy* 2014;134:102–13.
- [24] Javed F, Arshad N, Wallin F, Vassileva I, Dahlquist E. Forecasting for demand response in smart grids: an analysis on use of anthropologic and structural data and short term multiple loads forecasting. *Appl Energy* 2012;96:150–60.
- [25] Duan M, Darvishan A, Mohammaditab R, Wakil K, Abedinia O. A novel hybrid prediction model for aggregated loads of buildings by considering the electric vehicles. *Sustain Cities Soc* 2018;41(April):205–19.
- [26] Abedinia O, Raisz D, Amjadi N. Effective prediction model for Hungarian small-scale solar power output. *IET Renew Power Gener* 2017;11(13):1648–58.
- [27] Amjadi N, Abedinia O. Short term wind power prediction based on improved kriging interpolation, Empirical Mode Decomposition, and closed-loop forecasting engine". *Sustainability* 2017;9(11).
- [28] Khosravi A, Nahavandi S, Creighton D. Quantifying uncertainties of neural network-based electricity price forecasts. *Appl Energy* 2013;112:120–9.
- [29] Suryanarayana G, Lago J, Geysen D, Aleksiejuk P, Johansson C. Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods. *Energy* 2018;157:141–9.
- [30] Qiu X, Ren Y, Suganthan PN, Amaratunga GAJ. Empirical Mode Decomposition based ensemble deep learning for load demand time series forecasting. *Appl Soft Comput J* 2017;54:246–55.
- [31] Rahman A, Srikumar V, Smith AD. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Appl Energy* 2018;212(October 2017):372–85.
- [32] Shi H, Xu M, Li R. Deep learning for household load forecasting – a novel pooling deep RNN. *IEEE Trans Smart Grid* 2017;3053:1.
- [33] Binkowski M, Marti G, Donnat P. Autoregressive convolutional neural networks for asynchronous time series; 2017.
- [34] Taylor JW, Buizza R. Neural network load forecasting with weather ensemble predictions. *Trans Power Syst* 2002;17(3):626–32.
- [35] Fan C, Xiao F, Zhao Y. A short-term building cooling load prediction method using deep learning algorithms. *Appl Energy* 2017;195:222–33.
- [36] Hochreiter S, Unger Schmidhuber J. Long short-term memory. *Neural Comput* 1997;9(8):1735–80.
- [37] Dauphin YN, Fan A, Auli M, Grangier D. Language modeling with gated convolutional networks; 2016.
- [38] BEMOSS (An open source platform for building energy management). [Online]. Available: www.bemoss.org.
- [39] “EnerNOC commercial building dataset.” [Online]. Available: <https://open-enernoc-data.s3.amazonaws.com/anon/index.html>. [accessed: 09-Oct-2018].
- [40] “Weather underground.” [Online]. Available: <https://www.wunderground.com/> [accessed: 09-Oct-2018].
- [41] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: A system for large-scale machine learning; 2016.