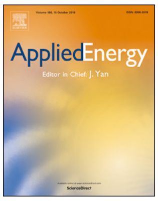
Applied Energy 278 (2020) 115410

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy



Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks

Gopal Chitalia[a,b], Manisa Pipattanasomporn[a,c,*], Vishal Garg[b], Saifur Rahman[c]

[a] *Smart Grid Research Unit, Department of Electrical Engineering, Chulalongkorn University, 10330, Thailand*
[b] *Center for IT in Building Science, IIIT-Hyderabad, 500032, India*
[c] *Bradley Department of Electrical and Computer Engineering, Advanced Research Institute, Virginia Tech, Arlington, VA 22203, USA*

HIGHLIGHTS

- A comprehensive framework for short-term electrical load forecasting is presented.
- RNN with attention reduced forecasting errors by 20–45% from the state of the art.
- Robust against different building types, locations, weather and load uncertainties.
- One month of data is enough to give satisfactory results.
- Clustering and 15-min data give better results in hour-ahead load forecasting.

ARTICLE INFO

Keywords:
Unsupervised clustering
Deep learning
Ensemble model
Sensitivity analysis
Short-term electrical load forecasting
Uncertainties in weather forecasts

ABSTRACT

This paper presents a robust short-term electrical load forecasting framework that can capture variations in building operation, regardless of building type and location. Nine different hybrids of recurrent neural networks and clustering are explored. The test cases involve five commercial buildings of five different building types, i.e., academic, research laboratory, office, school and grocery store, located at five different locations in Bangkok-Thailand, Hyderabad-India, Virginia-USA, New York-USA, and Massachusetts-USA. Load forecasting results indicate that the deep learning algorithms implemented in this paper deliver 20–45% improvement in load forecasting performance as compared to the current state-of-the-art results for both hour-ahead and 24-ahead load forecasting. With respect to sensitivity analysis, it is found that: (i) the use of hybrid deep learning algorithms can take as less as one month of data to deliver satisfactory hour-ahead load prediction, (ii) similar to the clustering technique, 15-min resolution data, if available, delivers 30% improvement in hour-ahead load forecasting, and (iii) the formulated methods are found to be robust against weather forecasting errors. Lastly, the forecasting results across all five buildings validate the robustness of the proposed deep learning framework for the short-term building-level electrical load forecasting tasks.

1. Introduction

According to the U.S. Energy Information Administration (EIA), the U.S. buildings sector consumed nearly 75% of the total electricity sales in 2018 [1]. With rapid population and economic growth, electricity used in the building sectors is projected to be doubled by 2050 from the 2018 consumption [2]. The buildings sector generates one-third of greenhouse gas, two-thirds of halocarbon and about one-third of black-carbon emissions [3]. Accurate building-level load forecasting can help deliver efficient building operations, thus mitigating such adverse effects. In addition, authors in [4] show that a 1% increase in load forecasting error can result in about £10 million increase in annual operating costs.

Typically, load forecasting horizons are either: (i) short-term, from one hour to one week, (ii) medium-term, from one week to a few months, or (iii) long-term, from months to years [5]. Short-term load forecasting (STLF) [6–8] is now possible thanks to widespread availability of Advanced Metering Infrastructure (AMI) and Internet of Things (IoT) devices that enable collection of electrical load and weather data at a more granular level [18,19]. STLF benefits real-time

[*] Corresponding author at: Smart Grid Research Unit, Department of Electrical Engineering, Chulalongkorn University, 10330, Thailand.
E-mail addresses: gopal.chitalia@research.iiit.ac.in (G. Chitalia), manisa.pip@chula.ac.th (M. Pipattanasomporn), vishal@iiit.ac.in (V. Garg), srahman@vt.edu (S. Rahman).

https://doi.org/10.1016/j.apenergy.2020.115410
Received 21 January 2020; Received in revised form 30 May 2020; Accepted 14 June 2020

control of building energy systems [9,10], including demand response [11], demand management [12,13], charging/discharging energy storage units [14], and energy transactions [15,16]. STLF is also a key element for generation dispatch and demand curtailments in a microgrid [17].

A large amount of building-level data has now become available, which has paved way for data-driven approaches [20], i.e., statistical/machine learning related approaches instead of physics-based approaches [21,22]. Previous studies have employed linear regression [23,24], non-linear regression [21], non-parametric regression [25,26], multi-layered perceptron [27], Auto Regressive Integrated Moving Average (ARIMA) [28], Extreme Machine Learning [29], Support Vector Regression (SVR) [30–34], fuzzy models [35], wavelet transform [36], random forests [37], Artificial Neural Network (ANN) [38,39] and hybrid methods for predicting building energy consumption [40]. Authors in [28] showcased different ARIMA- and ANN-based models and exponential smoothing methods for load forecasting. Authors in [41] proposed a technique that uses both multiple linear regression and a seasonal ARIMA model to forecast cooling and electric loads. Such traditional approaches, however, suffer from relatively irregular building usage patterns caused by irregular occupant behaviors [42], and high non-linearity in building dynamics/thermal physics added with uncertainties on weather [43–45].

The Recent development of deep learning methods, such as Deep Neural Network (DNN), including Convolution Neural Network (CNN), Recurrent Neural Network (RNN), has had a great impact in the fields of computer vision, Natural Language Processing (NLP) and speech recognition. DNN can model a complex function and can extract a variety of features from a large dataset. However, there has been limited use of deep learning methods for building-level load forecasting, and only recently, it is being applied. Authors in [46] proposed an RNN sequence-to-sequence model, i.e., an encoder-decoder architecture, to predict medium-to-long-term loads. Authors in [47] proposed RNN and CNN models to forecast commercial building loads hour-ahead and day-ahead. Authors in [48] used DNN/RNN methods for load forecasting, including Long Short Term Memory (LSTM) and LSTM encoder-decoder models. Authors in [49] proposed an RNN model for load forecasting with the time horizons of 24, 48 h, 7 days, and 30 days. In [50], the authors proposed an ANN model based on two back propagation algorithms for district level load forecasting. Authors in [51] exploited the potential of deep learning in unsupervised learning focusing on perceptron DNN for feature extraction and indicating a significant improvement in prediction accuracy. Authors in [52] proposed a recurrent inception CNN for load forecasting. A pooling based deep RNN for predicting household load was discussed in [53]. LSTM-based multi-input multi-output based window approach was discussed in [54].

The above studies mostly focus on RNN and CNN models and demonstrate that deep learning methods can deliver much better load forecasting accuracy than those achieved by traditional methods. Other deep learning models have not been much explored for load forecastings, such as attention model, BiLSTM and ConvLSTM. The attention model has been proven to be very useful and given state-of-the-art results in NLP [55]. The attention model accounts for all the past hidden states, while all the methods proposed in the literature use only the preceding hidden state. Hence, it is impossible to achieve a satisfactory forecasting accuracy with an incorrectly generated hidden state vector [52]. To address this problem, this paper investigates the use of the attention model with RNN, which accounts for all hidden states. Overall, this paper explores nine different deep learning models, including LSTM, LSTM with attention, BiLSTM, BiLSTM with attention, CNN + LSTM, CNN + BiLSTM, ConvLSTM, Conv BiLSTM, and Encoder-Decoder, for hour-ahead and 24-h ahead load forecasting. Also, unsupervised clustering with k-means based on historical load, weather, and schedule parameters is explored for hour-ahead load forecasting. To the best of the authors' knowledge, the deep learning algorithms that combine LSTM/BiLSTM models with attention or convolution mechanisms, i.e., LSTM with attention, BiLSTM with attention, ConvLSTM and ConvBiLSTM, are explored for the first time in building-level load forecasting.

The other limitations identified in the literature is the fact that most of the load forecasting methods were typically applied to buildings in the same geographic region, and of similar building functions. Hence, the methods proposed in the literature may not work well if different building types located in different geographic regions are used. In this paper, five buildings of different functions located in three countries, Thailand, India, and USA, are used to showcase the versatility of different deep learning models for building-level load forecasting.

Furthermore, as one of the major pain points identified in the literature was of that load forecasting at the building level requires a considerable amount of data [56], i.e., at least one year to get a good prediction, this study also explores the minimum length of historical data that can achieve acceptable forecasting accuracy.

Also, as peak load varies greatly among buildings due to different building types, sizes, locations, occupant behaviors and type of energy usage, researchers face challenges in having a fair comparison of forecasting errors among buildings using traditional error metrics, like Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). To the best of authors' knowledge, Root Mean Square Logarithmic Error (RMSLE) is used for the first time in the load forecasting problem as the error metric to allow fair comparison among buildings.

Overall, the major contributions and key findings from this work are summarized as follows:

- The deep learning methods formulated in this paper deliver RMSLE of 0.03 to 0.3 for all five buildings, and demonstrates 20–35% improvement in hour-ahead load forecasting accuracy, and 20–45% improvement in 24-h ahead forecasting accuracy as compared to the most recent state-of-the-art results [47].
- This paper provides a detailed discussion on the attention model, feature selection and hyperparameter fine-tuning, which are the main contributors to a major improvement in load forecasting accuracy.
- The load forecasting models presented here have been tested against buildings of different peak loads (ranging from 80 kW to 700 kW), functions (i.e., academic, research laboratory, office, school and grocery store) and weather conditions in three different countries (i.e., Thailand, India, and USA).
- The formulated deep learning models can deliver satisfactory load forecasting results even with as little as one month of data.
- Higher granularity data, i.e., 15-min intervals, could provide better forecasting accuracy as compared to one-hour intervals.
- The use of RMSLE provides deep insight that higher forecasting accuracy can be expected when dealing with buildings that do not have much load variations throughout a day.

This paper is organized as follows. Section 2 describes the research outline and all forecasting algorithms explored in this paper. Section 2.6 summarizes evaluation metrics used. Section 3 describes the building datasets and preprocessing methods. Section 4 discusses hour-ahead load forecasting, including input feature sets, hyperparameter fine-tuning and sensitivity analysis. Section 5 discusses algorithms for 24-h ahead load forecasting and model performance. All of the analysis above is discussed in the context of one building in Bangkok. Then, Section 6 discusses load forecasting for all other buildings and compares the results with the state of the art.

2. Framework and methodology for load forecasting

The research framework carried out in this paper is depicted in Fig. 1. It comprises the following processes: (i) data pre-processing; (ii) feature set selection; (iii) algorithm formulation; (iv) hyperparameter

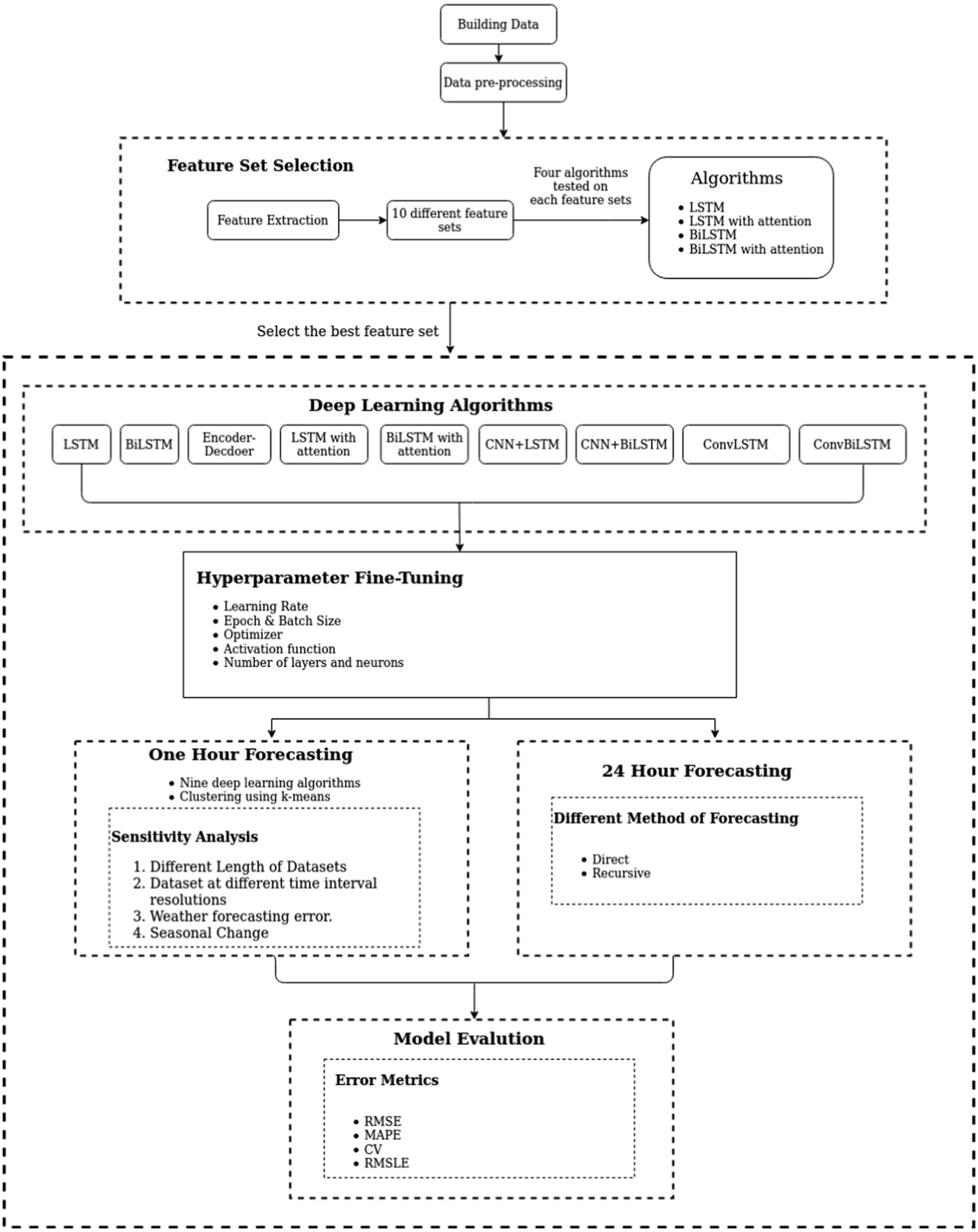


Fig. 1. Flowchart of proposed methodology for Load forecasting framework.

fine-tuning; and (v) building-level load forecasting at 1-h and 24-h ahead; and (vi) model evaluation.

2.1. Data pre-processing

After the building-level data are obtained, the data pre-processing involves filling in the missing values, detecting outliers and normalizing the data.

2.2. Feature set selection

The next step is to select relevant feature sets for a prediction model to be developed. Because redundant information is thrown out, feature set selection helps reduce the over-fitting problem, and helps in dimensionality reduction as only the relevant features are taken as model inputs. This in turn reduces the computation load of the model. In this study, the input matrix X comprises weather related parameters (x^W), scheduled related parameters (x^S) and historical loads (x^L), which are the three different kinds of input features vector. y being the output vector which refers to the predicted load. Pearson correlation is used to determine the most appropriate/relevant features for all the three kinds. Ten different feature sets are explored and tested with selected algorithms, namely, LSTM, LSTM with attention, BiLSTM, BiLSTM with attention. The feature set which gives the best results is then selected.

2.3. Deep learning algorithms

Nine different RNN/CNN and their hybrid algorithms are formulated for load forecasting, as described below. The focus is placed on formulating the LSTM/BiLSTM with attention and ConvLSTM/ConvBiLSTM algorithms, which are explored for the first time in building-level load forecasting.

2.3.1. LSTM

LSTM is a special kind of RNN. The main idea behind LSTM is to have better control over gradient flow, overcoming the RNN's problem of vanishing gradient [57], and to ensure better preservation of long-term dependencies by filtering out redundant or misleading information. As shown in Fig. 2(a), LSTM does this with the help of memory cell, input (i_t), output (o_t) and forget gates (f_t). Each of the LSTM units works in tandem, which is unlike vanilla RNN that has a single hidden layer. LSTM functions are expressed in Eq. (1):

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

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

$$\widetilde{C}_t = \tanh(W_c\cdot[h_{t-1}, x_t] + b_C) \tag{1c}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{1d}$$

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

$$h_t = o_t * \tanh(C_t) \tag{1f}$$

Where, W_f, W_i, W_c and W_o are weight and b_f, b_i, b_c and b_o are bias parameter vectors of LSTM, which are learned through back-propagation. Each sigmoid function (σ) outputs values in the range of $0 - 1$;0 means completely removing the information; and 1 means completely retaining the information. x_t is the current input at time t which integrates historical loads (x_t^L), weather (x_t^W) & schedule parameter (x_t^S) and h_{t-1} is the previous hidden layer output. The output vector $y_t = g(h_t)$ is the predicted load value at time t, where g can be any activation function (e.g., sigmoid, softmax, tanh, relu). The forget gate (f_t) in Eq. (1a) decides which information to retain or throw away. With the help of the input gate (i_t) in Eq. (1b) and the candidate memory cell ($\widetilde{C}_t$) in Eq. (1c), the decision is made to identify which values to update or to store in the cell state. The old cell state (C_{t-1}) is then updated to the new cell state (C_t) Eq. (1d). Using Eqs. (1e) and (1f), the output is determined. This process then continues to repeat.

2.3.2. BiLSTM (or Bidirectional LSTM)

A typical architecture of BiLSTM consists of two LSTM components, forward LSTM ($\overrightarrow{h_t}$) and backward LSTM ($\overleftarrow{h_t}$) as shown in Fig. 2(b). These two components are concatenated to form the output, as $y_t = \sigma(W\cdot[\overrightarrow{h_t}, \overleftarrow{h_t}, x_t] + b)$. The main intuition behind providing backward component is to provide the network a pathway so that it can look into future values as well. This might help the network in learning some dependencies of the load forecasting problem, which is a time series problem with periodic dependencies.

2.3.3. Encoder-decoder model

A typical architecture of an encoder-decoder sequence-to-sequence model is shown in Fig. 2(c). The *encoder model* comprises several RNN units that encode the input sequence into a fixed length context vector

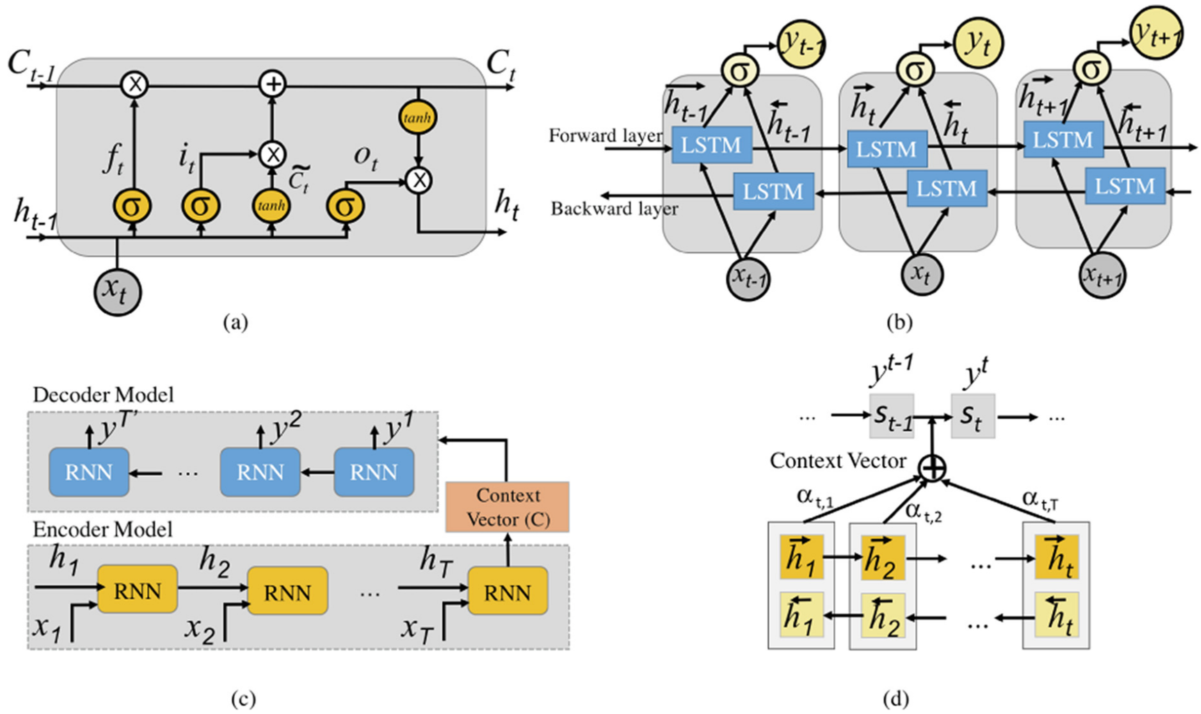


Fig. 2. (a) LSTM unit with forget gate (f_t), input gate (i_t) and output gate (O_t); (b) BiLSTM architecture with one forward and one backward LSTM layer; (c) Encoder-decoder architecture with context vector (C) out of the last hidden state is used; and (d) RNN with attention with context vector (C) a weighted combination of input states.

(C). The hidden states (h_i) are calculated based on the input vector (x_t) and its previous hidden state (h_{t-1}) as $h_t = f(x_t, h_{t-1})$. Where,f is any RNN function (e.g., LSTM or Gated recurrent units (GRU)). The *context vector* is the final hidden state output from the encoder model, where $C = h_T$. It acts as an initial hidden layer for the decoder, aiming to capture the information from the input, allowing the decoder to make accurate predictions. The *decoder model* decodes the context vector representation into another sequence. It is again a stack of several recurrent units, each accepting a hidden state from the previous unit. Here, $h_t = f(C, h_{t-1})$ and output $y_t = g(h_t)$. Where,f can be any RNN function, and g can be any activation function (e.g., sigmoid, softmax, and tanh). This model can learn patterns over a timescale and has been very successful in predicting a target sequence with all the information retained. However, as the length of input increases, it becomes very difficult for the context vector to retain previous information. The attention mechanism has been developed to solve this problem.

2.3.4. LSTM/BiLSTM with attention

The attention mechanism was designed to remember long source input. While in the encoder-decoder model, the context vector out of the last hidden state is used, the attention model creates a context vector, which is a weighted combination of the input states. Hence, the context vector has access to the entire input sequence, thus getting rid of the problem of forgetting. The attention technically only pays attention to those inputs that are critical for the next step. An architecture of bidirectional RNN + attention is shown in Fig. 2(d). Note: Any RNN, like LSTM or BiLSTM, can be combined with the attention model.

Let's assume a scoring function $f: \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$ which computes relevance between its input vectors and assigns weight to each of them. Then, the context vector (c_t) is a weighted sum of hidden states $H = \{h_1, h_2, \ldots, h_{t-1}\}$, representing the relevant information for the current time step. This is shown in Eq. (2).

$$c_t = \sum_{i=1}^{t-1} \alpha_i h_i \tag{2a}$$

$$\alpha_i = \frac{\exp(f(h_i, h_t))}{\sum_{j=1}^{t-1} \exp(f(h_j, h_t))} \tag{2b}$$

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}_i}^{\top}; \overleftarrow{\boldsymbol{h}_i}^{\top}]^{\top}, \ i = 1, \ldots, n \tag{2c}$$

The context vector then is integrated with previous hidden state unit s_{t-1} and previous target output y_{t-1} to predict the current hidden state s_t, shown in Eq. (3).

$$s_i = g(s_{i-1}, y_{i-1}, c_i) \tag{3}$$

2.3.5. CNN + LSTM, CNN + BiLSTM

Convolutional neural network (CNN) has been proven effective in time-series problems [58]. This paper explores CNN + LSTM and CNN + BiLSTM, a hybrid model that combines salient features of both. While CNN can extract and learn local/spatial features as well as reducing the number of parameters, LSTM/BiLSTM can capture variation in long/short term dependencies. Fig. 3 shows how this model works.

2.3.6. ConvLSTM, ConvBiLSTM

LSTM, whose gates perform convolutions, is ConvLSTM. That is, the convolutional reading of input is directly built into each unit of LSTM.

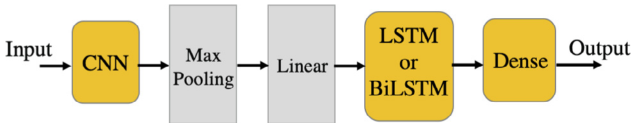


Fig. 3. CNN + LSTM/BiLSTM architecture.

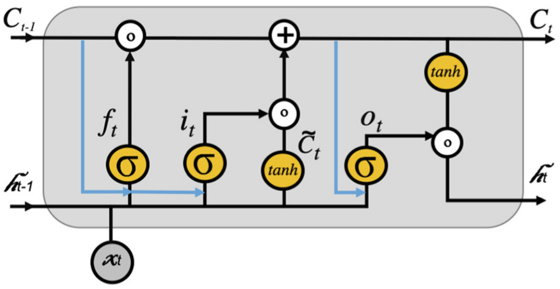


Fig. 4. ConvLSTM architecture.

Though developed for extracting underlying spatial features in 2-Dimensional input, they can be adopted for 1-D input problems. Architecture of ConvLSTM is depicted in Fig. 4.

Equations for ConvLSTM are summarized in Eq. (4), where '*' denotes the convolution operator and '∘' denotes the Hadamard product.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + W_{cf} \circ \mathscr{C}_{t-1} + b_f) \tag{4a}$$

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + W_{ci} \circ \mathscr{C}_{t-1} + b_i) \tag{4b}$$

$$\mathscr{C}_t = f_t \circ \mathscr{C}_{t-1} + i_t \circ \tanh(W_c * [h_{t-1}, x_t] + b_c) \tag{4c}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + W_{co} \circ \mathscr{C}_t + b_o) \tag{4d}$$

$$h_t = o_t \circ \tanh(\mathscr{C}_t) \tag{4e}$$

Any RNN architecture can be used with convolution, such as BiLSTM.

2.4. Hyperparameter fine-tuning

The performance of an algorithm is highly dependent on the choice of hyperparameters. This is especially true for the case of RNN, which has a more complex architecture than other traditional models and can achieve better and consistent performance with the right hyperparameters. Doing a grid search on each possible combination of hyperparameters is infeasible as it requires much time in evaluating unpromising areas of search space. Therefore, instead of tuning all the hyperparameters at once, a more informed decision can be made by selecting a set of hyperparameters to tune, and based on the information gathered, the best hyperparameters are fixed after every run. A generic method to fine-tune hyperparameters of all algorithms is described below.

- **Step 0:** Select the initial hyperparameter set.
- **Step 1:** Select the best learning rate
- **Step 2:** Select the best epoch and batch size.
- **Step 3:** Select the best optimizer.
- **Step 4:** Select the best activation function.
- **Step 5:** Select the best number of layers and neurons.

2.5. Load Forecasting: 1-h and 24-h ahead

All algorithms discussed in Section 2.3 have been used for 1-h ahead load forecasting in this paper.

For 24-h ahead forecasting, on the other hand, two approaches have been formulated: recursive and direct. In the recursive method, an iterative approach was adopted for 24-h ahead load forecasting using one-hour interval data. See Fig. 5.

In the first iteration, historical values of the past 24 h, consisting of building load, weather and scheduled related parameters at times $t, t-1 \cdots t-23$, are fed to the forecasting models as inputs. The models then predict the building load value at the next time step ($t+1$). Note that here t is every one hour. At the second iteration, the predicted load value ($t+1$) is then used, together with the 24-h historical values ($t, t-1 \cdots t-22$) to form the new input for building load prediction at

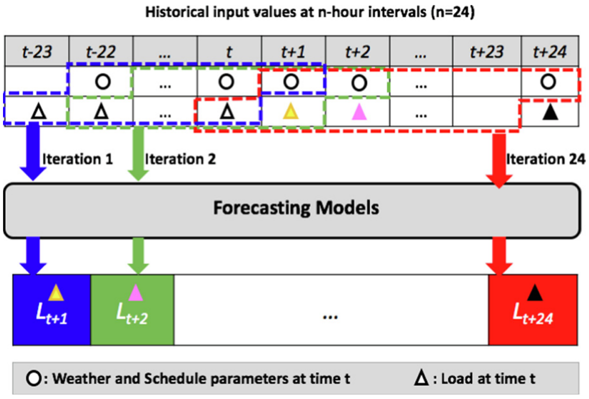


Fig. 5. Iterative process for 24-h ahead load prediction.

the next time step ($t + 2$). This recursive process continues until all the next day building load values are predicted, i.e., $t + 1, t + 2 \cdots t + 24$.

In the direct approach, on the other hand, given the past day input values, i.e, $t, t - 1 \cdots t - 23$ consisting of scheduled parameters, weather parameter and historical loads, the loads for the next 24 h are predicted all at once, i.e., $t + 1, t + 2 \cdots t + 24$.

2.6. Evaluation Metrics

This section discusses the evaluation metrics used for determining the accuracy of load forecasting.

2.6.1. Metrics for Hour-ahead Forecasting

Root mean squared error (RMSE), mean absolute percentage error (MAPE) and, coefficient of variance (CV) are used as the evaluation metrics. RMSE, MAPE and CV were calculated using Eq. 5(a)-(c), respectively.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (P_i - A_i)^2}{N}} \tag{5a}$$

$$\mathrm{MAPE} = \frac{\sum_{i=1}^{N} \left| \frac{P_i - A_i}{A_i} \right|}{N} \times 100 \tag{5b}$$

$$\mathrm{CV} = \frac{\sqrt{\frac{\sum_{i=1}^{N} (P_i - A_i)^2}{N-1}}}{\bar{A}} \times 100 \tag{5c}$$

Where, P_i and A_i represented predicted and actual load values. N is the total number of observations. $\bar{A}$ is the mean of the actual load values.

2.6.2. Metrics for 24-h ahead Forecasting

Unlike in the hour-ahead prediction, which gives a scalar output, the output of 24-h ahead prediction is a vector comprising 24 load values. Therefore, error metrics are calculated for each day and averaged into one final value, using Eq. 5(a)-(c) with N = 24, which is the length of output vectors.

2.6.3. Metrics for comparison among buildings

Since every building has different peak load as shown in Table 1, RMSE is not suitable to compare forecasting errors among buildings as it will penalise more for buildings with higher loads. So to have a fair assessment of results among buildings, log of predictions and actual values are taken. Hence, Root Mean Square Logarithmic Error (RMSLE) is used, as shown in Eq. (6).

$$\mathrm{RMSLE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\log(P_i + 1) - \log(A_i + 1))^2} \tag{6}$$

Note: RMSLE is the metric chosen for ASHRAE Great Energy Predictor III competition [59].

3. Dataset Description

This section discusses the datasets for load forecasting and data preprocessing.

3.1. Building datasets

In this paper, load forecasting was performed on five commercial buildings: one in Bangkok, Thailand; one in Hyderabad, India; and three in the USA (Virginia, New York, and Massachusetts). These buildings are of different types, i.e., academic, research laboratory, office, school and grocery store, ranging in size from 840 m^2 to 7,500 m^2. Their peak electrical loads range from 74 kW to 646 kW. The buildings' physical and electrical characteristics, together with outdoor temperature ranges, are summarized in Table 1.

Table 1 also summarizes the average load and the average off-peak (nighttime) load for each building to demonstrate the degree of difference in peak, average and off-peak loads. The building in Bangkok has the lowest average off-peak load, as this building has the lowest nighttime consumption. On the other hand, the average off-peak load is very close to its average load for the building in Massachusetts, USA. This is intuitive as this building is a grocery store, and there is a refrigeration load that needs to operate 24/7.

The electrical load data of the building in Bangkok were collected from Chulalongkorn University's Building Energy Management System (CU-BEMS) [60]. Electrical load data of the building in India were collected from [61]. Electrical load data of the building in Virginia, USA, were gathered from its smart meter, while those of the buildings in New York and Massachusetts, USA were collected from public EnerNOC Commercial building data set [62]. Except for the building in Bangkok, which has one-minute interval data, data of other buildings are available at one-hour intervals. Weather parameters including outdoor temperature (O, °C), humidity (RH, %), global solar radiation (GSR, W/m^2), air pressure (A, *inch*) and wind speed (WS, *mph*) were collected from Weather Underground [63].

3.2. Data Pre-processing

The weather data obtained from Weather Underground are at irregular intervals, i.e., several values per hour. Hence, they were averaged at one-hour intervals to obtain hourly data. Then, missing weather-related variables were filled with the combination of interpolation, as well as through the data from the timeanddate website [64].

For the building in Bangkok of which the data are in one-minute intervals, building load data were averaged at every one-hour intervals. For all buildings, missing electrical load values were filled with neural network with weather parameters (O_t, RH_t and GSR_t) and schedule variables (hour of day (H_i, 1–24) and day of week (W_i, 1–7) as input features, and L_t being the output.

All data were normalized and divided into 80–10-10 for training, validation, and testing.

4. Hour-ahead forecasting

This section discusses feature set selection, hyperparameter fine-tuning, ensemble learning-based clustering, forecasting results, and sensitivity analysis. All are in the context of hour-ahead load forecasting.

Table 1
Building load and weather data set characteristics.

Building characteristics				Electrical load			Weather
No.	Location	Building type	Area (m^2)	Peak load (kW)	Avg. load (kW)	Avg. off-peak load (kW)	Min/max temp. (°C)
Bldg 1	Bangkok, Thailand	Academic	2,700	104.3	19.0	7.9	16.1/37.4
Bldg 2	Hyderabad, India	Research lab	840	78.7	27.5	12.5	12.6/39.4
Bldg 3	Virginia, USA	Office	2,300	74.3	26.8	21.9	−10.6/37.8
Bldg 4	New York, USA	School	7,500	381.3	109.8	71.8	−10.6/36.7
Bldg 5	Massachusetts, USA	Grocery store	5,100	646.2	396.5	334.2	−14.4/35.6

Table 2
Pearson correlation coefficients between load and weather parameters of all buildings.

	Bldg 1	Bldg 2	Bldg 3	Bldg 4	Bldg 5
O	0.50	0.57	0.74	0.51	0.76
RH	−0.36	−0.35	−0.26	−0.07	−0.10
GSR	0.56	0.46	–	–	–
WS	–	–	−0.02	−0.04	0.02
A	−0.13	–	0	0	0

4.1. Feature set selection

To select the most relevant weather features, Pearson correlation (ρ) was determined between building load (L) and weather parameters (O, RH, GSR, WS, A). The Pearson correlation coefficients between loads and weather parameters for all five buildings are listed in Table 2. Note that not all weather parameters are available at all building locations. It can be seen that O is highly correlated with building load (kW) at all locations. RH are somewhat correlated with building loads for the first three buildings. GSR appears to be as highly correlated to building loads as O for the buildings in Bangkok and Hyderabad. However, WS and A have low correlation with building loads.

In addition, historical building loads (L_i, $i = t-1, t-2...$) are known to be highly correlated with the future building loads. Hence, any combination of historical load values can be taken as features for load forecasting. The schedule-related variables, consisting of: hour of the day (H, 1–24), day of the week (W, 1–7), and month number (M, 1–12), should be explored since they can capture repeating patterns in time-series problems. Thus, to identify the best input feature sets, which can provide the best load forecasting results, this study explored 10 different combinations of historical loads, weather parameters and scheduled related variables as different feature sets (FS_1,..., FS_{10}), as summarized in Table 3.

Fig. 6 summarizes RMSE, MAPE, and CV for the building in Bangkok, when these 10 feature sets were used as the input features for load forecasting based on the following four algorithms: LSTM, LSTM with attention, BiLSTM and BiLSTM with attention. As shown, FS_3, FS_9 and FS_{10} yield the best comparable results. Here, the hyperparameters initially taken are summarized in Table 4.

Table 3
Different feature sets (here t is every one hour).

No.	Input features
FS_1	L_{t-1}
FS_2	L_{t-1}, O_t
FS_3	L_{t-1}, O_t, H_t, W_t
FS_4	L_{t-1}, O_t, H_t, W_t, M_t
FS_5	L_{t-1}, O_t, H_t, W_t, RH_t, GSR_t
FS_6	L_{t-1}, O_t, H_t, W_t, RH_t, GSR_t, L_{t-2}, O_{t-1}, H_{t-1}, W_{t-1}, RH_{t-1}, GSR_{t-1}
FS_7	L_{t-1}, O_t, GSR_t
FS_8	L_{t-1}, O_t, GSR_t, W_t
FS_9	L_{t-1}, O_t, GSR_t, H_t
FS_{10}	L_{t-1}, O_t, H_t, W_t, GSR_t

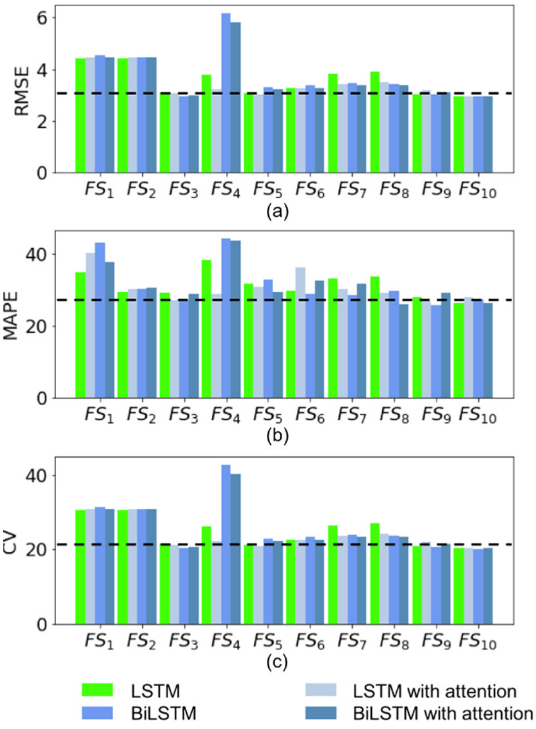
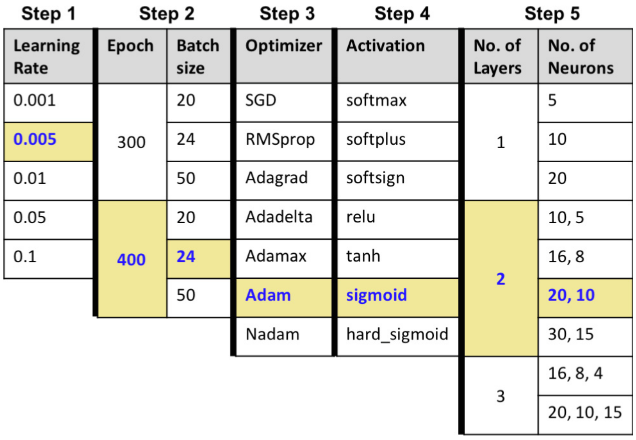


Fig. 6. (a) RMSE, (b) MAPE and (c) CV for different FS_S for the building in Bangkok.

Table 4
Hyperparameters initially taken for feature set selection.

Learning rate	0.001	Optimizer	Adam
Batch size	50	Layers	2
Epoch	300	Neurons	10, 5
Activation function	Sigmoid		

As evident from Pearson correlation (Table 2), O and GSR have higher correlation with load as compared to that of RH with load. Hence, feature sets with RH_t as a parameter, i.e., FS_5 and FS_6 do not perform better as they tend to overfit the model. Similarly, adding M_t as a feature in FS_4 tends to overfit and does not help improve the model performance. When schedule parameters are not used as inputs, i.e., FS_1, FS_2, and FS_7, the model performances are not as good as those when they are used. This shows the importance of the schedule parameters, on which building loads are dependent. When comparing FS_8, FS_9 and FS_{10}, using both H_t and W_t yield better results than using only one of them, as both help in classifying the building operation in different aspects. W_t gives information about weekdays and weekends, whereas H_t provides information about daytime and nighttime.



Step 1	Step 2		Step 3	Step 4	Step 5	
Learning Rate	Epoch	Batch size	Optimizer	Activation	No. of Layers	No. of Neurons
0.001	300	20	SGD	softmax	1	5
0.005		24	RMSprop	softplus		10
0.01		50	Adagrad	softsign		20
0.05	**400**	20	Adadelta	relu	**2**	10, 5
0.1		**24**	Adamax	tanh		16, 8
		50	**Adam**	**sigmoid**		**20, 10**
			Nadam	hard_sigmoid		30, 15
					3	16, 8, 4
						20, 10, 15

Fig. 7. Steps to fine tune hyperparameters with the best hyperparameter set for LSTM highlighted.

Even though FS_3, FS_9 and FS_{10} give comparable forecasting results, FS_3,consisting of L_{t-1}, O_t, H_t, W_t, was selected as the best feature set moving forward. This is because FS_3 relies on only outdoor temperature data, which are widely available in all locations, and it has the less number of features, hence making it computationally less heavy.

4.2. Fine-tuning hyperparameters

This study used the initial hyperparameters shown in Table 4. The learning rates, different combinations of epoch and batch sizes, optimizers, activation functions, and different combinations of layers and number of neurons, explored in this study, are summarized in Fig. 7.

The figure also highlights the hyperparameters that gave the best result for the LSTM model, which are: the learning rate of **0.005**; the epoch and batch size of **400** and **24**, respectively; **Adam** optimizer; **Sigmoid** activation function; number of layers of **2** with **20 and 10 neurons** on each layer respectively. The procedure above was followed to fine-tune all nine algorithms for each dataset. Then, these algorithms were used in ensemble learning-based clustering, discussed in the next section.

4.3. Ensemble learning-based clustering

In this study Euclidean distance was used as a similarity metric. Elbow method was used to determine the optimal number of clusters k. Using the building in Bangkok as an example, the relationship between the number of clusters and Within-Cluster Sum of Squares (WCSS) indicates that the optimal number of clusters is four. However, since load data are limited to 8760 values in a year or 8784 in a leap year, dividing them into four clusters would mean fewer data available in each cluster to train on. This would affect the prediction performance. Hence, the number of clusters (k) of two and three were chosen to experiment. Different combinations of input features, including historical load (L_{t-1}), O_t, H_t and W_t, were experimented to cluster the data as shown in Table 5. It is important to note that clustering was done using historical load at time $t-1$ as the load at time t is unknown and is the forecasting output.

Table 5
Feature sets for clustering.

Clustering Sets	Clustering Inputs
CS_1	L_{t-1}, H_t
CS_2	L_{t-1}, O_t
CS_3	L_{t-1}, O_t, H_t
CS_4	L_{t-1}, O_t, H_t, W_t

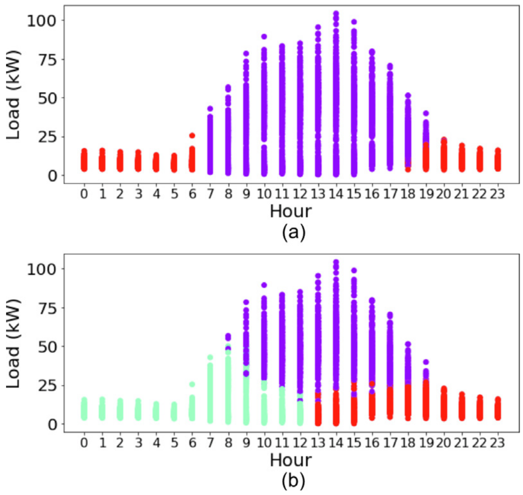


Fig. 8. (a) Two cluster plot; and (b) three cluster plot of Hour vs Load for CS_1 feature set (BKK building).

For the two clusters, as Euclidean distance could not differentiate the periodicity of hours in a day (0–23), the hours were shifted from 0–7 to 24–31 to group low-load hours together. After doing the k-means on CS_1 for two clusters, the plot is shown in Fig. 8(a). Notice K-means automatically separates the data into the low-load (nighttime) and high-load (daytime) groups. For three clusters in Fig. 8(b), shifting hours would not be required since the clustering process automatically classifies two low-load clusters and one high-load cluster.

4.4. Model performance: hour-ahead forecasting

Using the input feature set FS_3, all nine algorithms discussed in Section 2 were used to perform hour-ahead load forecasting on the building in Bangkok. The forecasting errors (RMSE, MAPE, and CV) of all models before and after fine-tuning their parameters are summarized in Table 6. The results indicate the overall 4–12% improvement in model performance, showing the importance of fine-tuning hyperparameters. It can be seen that all hybrid RNN algorithms gave comparable results for the Bangkok building with BiLSTM performing the best (RMSE of 2.89, MAPE of 25.33, and CV of 19.92).

All the nine algorithms with their hyperparameters fine-tuned were also run on all clustering feature sets (CS_{1-4}) for two and three clusters. The best case results with ensemble for each clustering set are summarized in Table 7.

As shown, the best results with the help of an ensemble is CS_1 for two clusters. In CS_1, BiLSTM with attention and LSTM with attention perform the best for low-load and high-load clusters, respectively, hence ensembled to produce the one final load forecasting model. It is also observed that the results for two clusters are better than those for

Table 6
Model performance before and after tuning the hyperparameters: hour-ahead load forecasting (BKK building).

	Before/after hyperparameter tuning		
Algorithm	RMSE	MAPE	CV
LSTM	3.28/2.91	31.0/25.05	22.63/20.06
LSTM w/attention	3.23/2.93	34.66/**24.76**	22.33/20.20
BiLSTM	3.07/**2.89**	27.81/25.33	21.22/**19.92**
BiLSTM w/attention	3.16/2.96	30.20/26.77	21.82/20.47
CNN + LSTM	3.28/3.01	32.35/27.92	22.65/20.79
CNN + BiLSTM	3.05/2.94	27.51/27.17	21.08/20.26
ConvLSTM	3.28/3.03	31.03/28.69	22.63/20.88
ConvBiLSTM	3.21/2.94	31.42/28.21	22.14/20.33
Encoder-Decoder	3.28/2.99	31.04/26.84	22.63/20.62

Table 7
Clustering Results.

	Two Clusters			Three Clusters		
	RMSE	MAPE	CV	RMSE	MAPE	CV
CS_1	**2.50**	**20.17**	**17.13**	2.74	23.58	20.87
CS_2	2.77	23.90	19.45	2.95	27.49	20.85
CS_3	2.57	20.22	17.50	2.77	22.83	20.66
CS_4	2.79	24.23	20.00	2.82	23.73	20.61

three clusters. This is because of the small amount of dataset available for three clusters. Table 7 also reveals a significant improvement in results with clustering as compared to without clustering, shown in Table 6. RMSE comes down to 2.50, representing the reduction in error of about 13.5%. MAPE comes down to 20.17, or an error reduction of about 20.4%. And, CV comes down 17.13, which indicates a reduction of about 14%.

The plot for actual vs. predicted loads for all algorithms, together with outdoor temperature, is shown in Fig. 9 for one week of the test dataset. It is evident that the algorithms are able to detect daily and weekly variations, i.e., low nighttime and weekend loads. Also, the reason for higher MAPE and CV is due to very low load values (0–10 kW) at nights and weekends, and hence it penalizes more.

4.5. Sensitivity analysis

This section investigates the impact of different lengths of the dataset, higher resolution data, weather forecasting errors and seasonal change on load forecasting accuracy.

4.5.1. Impact of different lengths of datasets

In most previous studies, the amount of historical data required for hour-ahead load forecasting is typically at least a year. This study explores the minimum amount of historical data required for building-level load forecasting that can result in comparable prediction accuracy to the use of 12 months of historical data. The following dataset lengths were explored: one, two, three, four and six months, and compared with the 12-month length of data. Each dataset was divided into 90 (training)- 10 (validation)-seven days (testing). To ensure a fair comparison, the test dataset was kept the same for all cases. The seven-day test dataset was chosen to be Sunday to Saturday, during the second last week of December before the holiday started. All the nine algorithms were tested, and results are shown in Table 8.

Table 8
Model performance results for different length of datasets.

	Length of dataset (months)					
Algorithm	One	Two	Three	Four	Six	Twelve
LSTM	3.37	3.40	3.31	3.41	3.38	3.04
LSTM with attention	3.37	3.48	**3.26**	**3.25**	**3.14**	**2.95**
BiLSTM	3.31	3.29	3.44	3.84	3.16	3.23
BiLSTM with attention	**3.28**	**3.22**	3.44	3.40	3.26	3.00
CNN + LSTM	3.45	3.28	3.41	3.39	3.34	3.12
CNN + BiLSTM	3.61	3.23	**3.26**	3.29	3.40	3.41
ConvLSTM	3.58	3.40	3.56	3.32	3.26	3.16
ConvBiLSTM	3.40	3.32	3.52	3.26	3.38	3.18
Encoder-Decoder	3.35	3.54	3.29	3.50	3.40	3.01

As shown, it is quite apparent that generally, the longer the historical data available for training, the better the load forecasting accuracy would be. With 12 months of historical data, the best RMSE is 2.95 for LSTM with attention. The best RMSE increases to 3.14 (LSTM with attention, 6.4% increase), 3.25 (LSTM with attention, 10.15% increase), 3.26 (LSTM with attention and CNN + BiLSTM, 10.5% increase), 3.22 (BiLSTM with attention, 9.2% increase) and 3.28 (BiLSTM with attention, 11.2% increase) when the dataset lengths of six, four, three, two and one months were used, respectively. From the results, it is observed that having six months of historical data performs reasonably well as compared to having 12-month historical data. With one month of data, forecasting errors (RMSE) increase by no more than 11%. This implies that one month of data collection is deemed to be sufficient for building-level load forecasting using the formulated deep learning method(s).

4.5.2. Impact of higher resolution data

In most of the studies, load forecasting is usually performed at one-hour intervals. Since CU-BEMS datasets are available at one-minute intervals, it is possible to explore the impact of higher resolution data on the hour-ahead building-level load forecasting. This study explores the accuracy of hour-ahead building-level load forecasting using 15-min interval data as compared to one-hour interval data.

An iterative approach was adopted in this study to predict the next hour building load using 15-min interval data. In the first iteration, historical values in the past one hour, shown in Table 9, consisting of building load, weather and scheduled related parameters at times t, $t-1$, $t-2$, $t-3$, are fed to the forecasting model which predicts the building load value at the next time step ($t+1$). Note that here t is

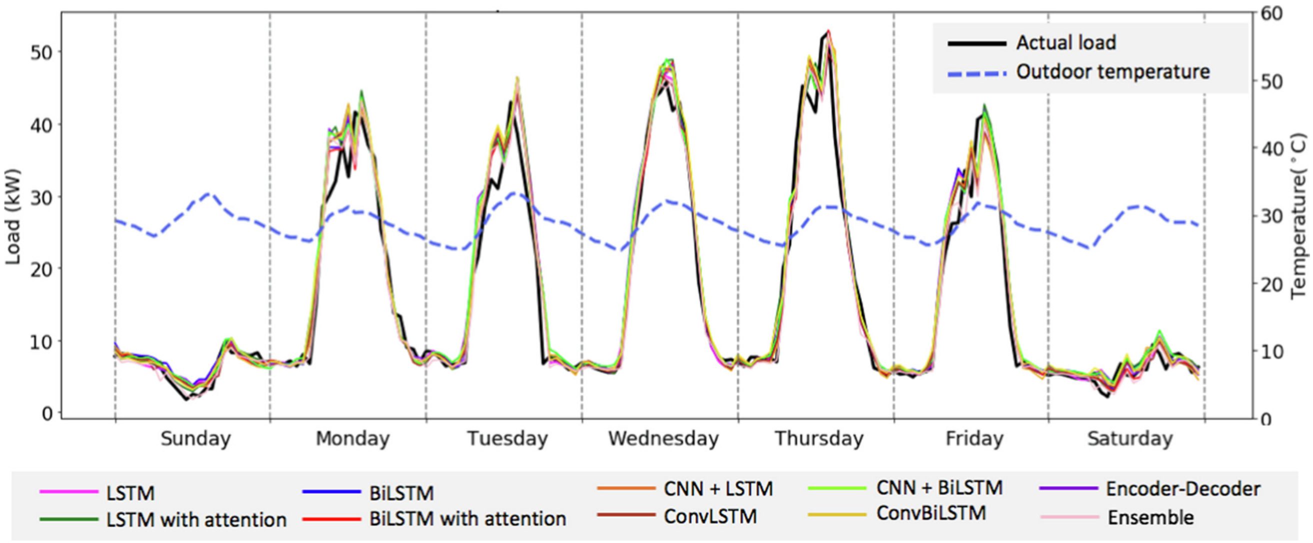


Fig. 9. Actual vs predicted load plot: hour-ahead load forecasting (BKK building).

Table 9
Feature set

FS	L_{t-1}, O_t, H_t, W_t, L_{t-2}, O_{t-1}, H_{t-1}, W_{t-1} L_{t-3}, O_{t-2}, H_{t-2}, W_{t-2}, L_{t-4}, O_{t-3}, H_{t-3}, W_{t-3}

Note: Here *t* is every 15 min.

Table 10
Model performance results at 15 min interval dataset.

Algorithm	RMSE	MAPE	CV
LSTM	2.54	20.96	17.66
LSTM with attention	**2.47**	**19.22**	**17.18**
BiLSTM	2.64	19.38	18.33
BiLSTM with attention	2.64	18.16	18.34

every 15 min.

At the second iteration, the predicted load value at time ($t + 1$), together with historical one-hour values (t, $t - 1$, $t - 2$), are used to form the new input for building load prediction at next time step ($t + 2$). This recursive process continues until all the next hour building load value are predicted, i.e., $t + 1$, $t + 2$, $t + 3$ and $t + 4$. These 15-min next hour building loads (in kWh) are then combined by simply summing them up to obtain the next hour building load (also in kWh).

The load forecasting accuracy indices (RMSE, MAPE, and CV) using 15-min data resolution are shown in Table 10. Since LSTM, LSTM with attention, BiLSTM, and BiLSTM with attention perform better than other models as shown in Table 6, only these models are selected for hour-ahead forecasting using 15-min interval data.

It can be seen that there was a significant improvement in load forecasting accuracy when 15-min data resolution was used as compared to one-hour resolution data. The table indicates that the forecasting performance has improved, and is comparable to that of clustering in Table 7.

4.5.3. *Impact of errors in weather forecast*

In most studies, the weather parameters used for load forecasting are assumed to be 100% accurate. Although weather forecasting services can give very accurate outdoor temperature prediction, a typical error of 5–15% of predicted next-hour temperature can be expected. Hence, noises were introduced in the weather data using Gaussian distribution with the mean of the actual temperature, and three standard deviations (3SD) of 5%, 10%, and 15%. The load forecasting results after introducing noises in predicted outdoor temperature are summarized in Table 11.

It can be seen that RMSE increases from 2.89 to 2.91 (LSTM with attention, 0.7% increase), 2.98 (LSTM with attention, 3.1% increase) and 2.95 (LSTM with attention, 2.1% increase) for 5%, 10% and 15% weather forecast errors, respectively.

Table 11
RMSE with weather forecasting errors.

	Weather Forecasting Errors		
Algorithm	**5%**	**10%**	**15%**
LSTM	2.94	3.11	3.12
LSTM with attention	**2.91**	**2.98**	**2.95**
BiLSTM	2.99	3.00	3.16
BiLSTM with attention	3.01	3.03	3.12
CNN + LSTM	3.08	3.14	3.17
CNN + BiLSTM	3.09	3.31	3.24
ConvLSTM	3.00	3.11	3.24
ConvBiLSTM	3.21	3.21	3.38
Encoder-Decoder	2.98	3.11	3.26

Table 12
Model performance on seasonal change.

	RMSE		—
Algorithm	**Summer**	**Rainy**	
LSTM	**3.38**	3.18	
LSTM with attention	3.42	3.33	
BiLSTM	3.45	3.29	
BiLSTM with attention	3.50	3.30	
CNN + LSTM	3.63	3.16	
CNN + BiLSTM	3.39	**3.13**	
Conv2D + LSTM	3.57	3.27	
Conv2D + BiLSTM	3.53	3.19	
Encoder-Decoder	3.51	3.59	

4.5.4. *Impact of seasonal change*

In most studies, the length of the dataset is at least a year and is divided into training-validation-testing with the ratio of 80–10-10 (or similar). Hence testing was typically done only the last 10% of the dataset, which is usually the month of December. This study also explores the impact of seasonal change on load forecasting accuracy.

In Bangkok, usually the Summer season is from March to around the end of June. Rainy season starts around the end of June to around the end of October, and then Winter season starts. In this study, four months of Summer (March-June) and four months of Rainy season (July-October) were explored. Both datasets were divided into 90% training, 10% validation and last week for testing. The last week of June and the last week of October were selected as the testing dataset, as both weeks represent transitional seasonal change from Summer to Rainy and Rainy to Winter respectively. All the nine algorithms were tested, and load forecasting accuracy results are shown in Table 12.

To ensure fair comparison, the results were compared using the case of equal dataset length of four months. As can be seen from Table 8 and Table 12 the RMSE increases from 3.25 to 3.38 (LSTM, 4% increase) during the transition from Summer to Rainy season, and decreases from 3.25 to 3.13 (CNN + BiLSTM, 3.7% decrease) during the transition from Rainy to Winter season.

Note: In this study, winter seasonal change (i.e. from November-February) could not be explored as it is a year-long dataset, hence no data of January, February of next year is available.

5. 24-h ahead Forecasting

As mentioned in Section 4.5.2, LSTM, LSTM with attention, BiLSTM, and BiLSTM with attention performs better than other models. Hence, they are selected for 24-h ahead forecasting. Table 13 summarizes results for 24-h ahead load forecasting for the Bangkok building.

Two algorithms were tested: recursive and direct. As shown, the recursive 24-h forecasting yields better forecasting accuracy overall with LSTM with attention performing the best.

The plot for actual vs. predicted loads with the direct and recursive forecasting methods, together with outdoor temperature, is shown in Fig. 10 for one week of the test dataset. As expected, the 24-h forecasting results have somewhat higher errors than the results of hour-ahead forecasting.

Table 13
Model performance: 24-h ahead forecasting (BKK building).

	Direct/recursive 24-h ahead forecasting		
Algorithm	**RMSE**	**MAPE**	**CV**
LSTM	6.24/5.30	64.90/49.61	56.11/48.60
LSTM with attention	6.35/**5.19**	74.51/47.70	60.02/46.23
BiLSTM	5.70/5.39	62.63/49.32	51.56/48.09
BiLSTM with attention	6.71/6.82	73.77/69.17	67.05/62.06

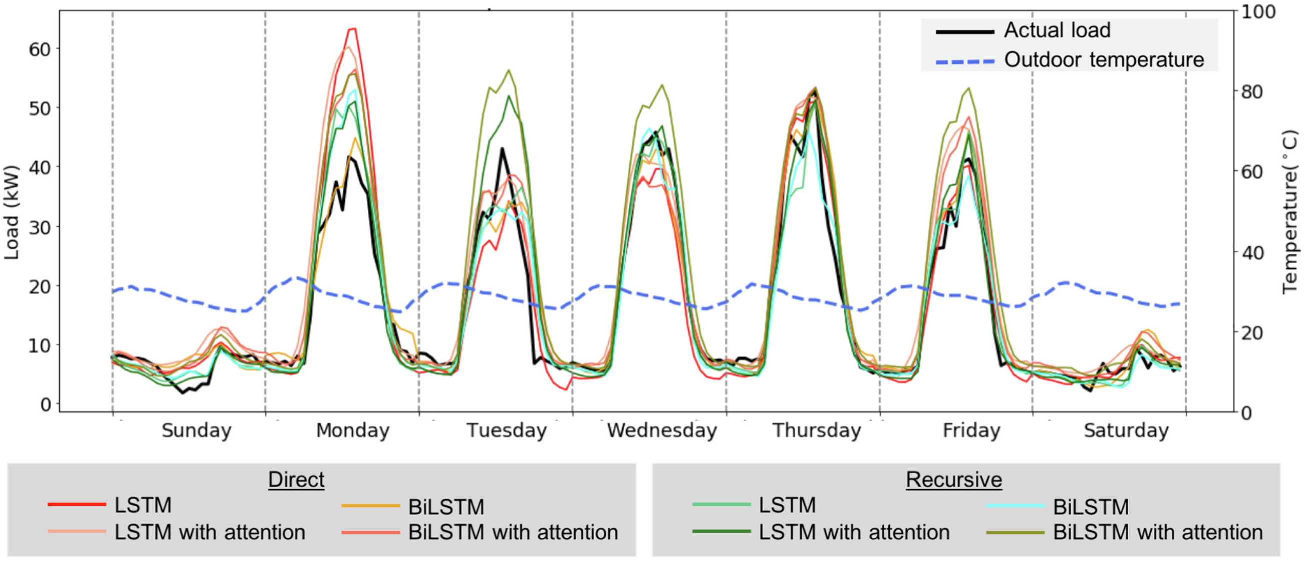


Fig. 10. Actual vs predicted load plot recursive and direct approach for the building in Bangkok: 24-h ahead forecasting.

6. Load forecasting for other buildings and comparison with the state-of-the-art

After forecasting on the building in Bangkok, the algorithms described above were tested on four other buildings located in Hyderabad (India), Virginia (USA), New York (USA), and Massachusetts (USA) for both hour-ahead and 24-h ahead forecasting. This was to evaluate the robustness of the models against different weather patterns, building types, building operations, and building locations. Error evaluation metrics, including RMSE, MAPE, and CV, for both hour-ahead and 24-h ahead are summarized in Table 14 for these buildings. Note that, for hour-ahead forecasting, all nine algorithms were run. As RNN algorithms (i.e., LSTM, LSTM with attention, BiLSTM, BiLSTM with attention) perform better in all buildings as compared to the rest of the algorithms, only these four RNN algorithms were used for 24-h ahead forecasting.

6.1. *India building*

Using the algorithms described in this paper, BiLSTM performs the best with an RMSE of 3.25, MAPE of 10.97, and CV of 13.59. The RMSE results are comparable to that of the Bangkok building because these two buildings have similar peak loads (78 kW vs. 104 kW). The results for clustering are not better than non-clustering because the building is always active during weekends, and its consumption patterns during the nighttime periods are consistent throughout the year, which is unlike the load profiles of Bangkok building. See Fig. 11(a) vs. Fig. 9. Hence, this fact explains its lower MAPE than that of the Bangkok building.

6.2. *Three U.S. Buildings*

To enable fair RMSE/MAP/CV comparison with the state-of-the-art results, this paper used the datasets from the same three U.S. buildings

Table 14
Model performance: all other buildings and comparison with the state-of-the-art results.

	Bldg 2. Hyderabad, India (Peak 78.7 kW)			**Bldg 3. Virginia, USA (Peak 74.3 kW)**			**Bldg 4. New York, USA (Peak 381.3 kW)**			**Bldg 5. Massachusetts, USA (Peak 646.2 kW)**		
	RMSE	**MAPE**	**CV**	**RMSE**	**MAPE**	**CV**	**RMSE**	**MAPE**	**CV**	**RMSE**	**MAPE**	**CV**
hour-ahead forecasting												
State-of-the-art [47]	–	–	–	–	8.28	10.92	–	5.79	7.26	–	2.31	2.84
LSTM	3.31	11.78	13.83	1.69	5.78	7.77	**5.33**	**3.97**	**5.63**	10.42	1.99	2.80
LSTM with attention	3.38	12.51	14.11	1.71	5.83	7.89	5.80	4.01	6.12	10.22	**1.87**	2.75
BiLSTM	**3.25**	**10.97**	**13.59**	1.66	5.59	7.66	5.52	3.98	5.83	10.38	1.89	2.79
BiLSTM with attention	3.44	12.40	14.36	**1.58**	**5.35**	**7.29**	5.42	**3.95**	5.73	**10.05**	**1.87**	**2.70**
CNN + LSTM	3.89	15.62	16.22	1.67	6.05	7.70	5.57	4.35	5.88	10.79	2.18	2.90
CNN + BiLSTM	3.54	13.40	14.76	1.70	6.06	7.84	5.94	4.66	6.27	10.24	1.94	2.76
Conv2D + LSTM	3.45	13.43	14.42	1.75	6.13	8.07	5.66	4.36	5.98	10.42	2.10	2.80
Conv2D + BiLSTM	3.68	12.74	15.34	1.75	5.88	8.06	5.75	4.37	6.07	10.55	2.12	2.84
Encoder-Decoder	3.43	12.86	14.30	1.73	6.13	7.98	5.47	3.99	5.77	10.84	2.08	2.92
Clustering	3.44	13.53	20.44	**1.54**	**5.39**	**7.05**	**4.94**	**3.97**	**5.40**	**9.67**	**1.79**	**2.61**
24-h ahead forecasting												
State-of-the-art [47]	–	–	–	4.25	–	–	13.45	–	–	14.35	–	–
LSTM	**5.03**	**22.37**	**23.21**	3.46	11.37	15.90	8.02	6.59	8.59	**11.27**	**2.25**	**3.09**
LSTM with attention	5.43	29.23	24.55	**3.30**	**11.64**	**15.16**	**7.21**	**5.96**	**7.87**	14.78	3.42	4.06
BiLSTM	5.55	22.78	25.66	3.40	11.86	15.66	8.46	7.14	9.37	12.60	2.67	3.45
BiLSTM with attention	5.08	23.23	23.08	3.44	13.89	17.46	15.52	10.98	15.79	12.75	2.49	3.49

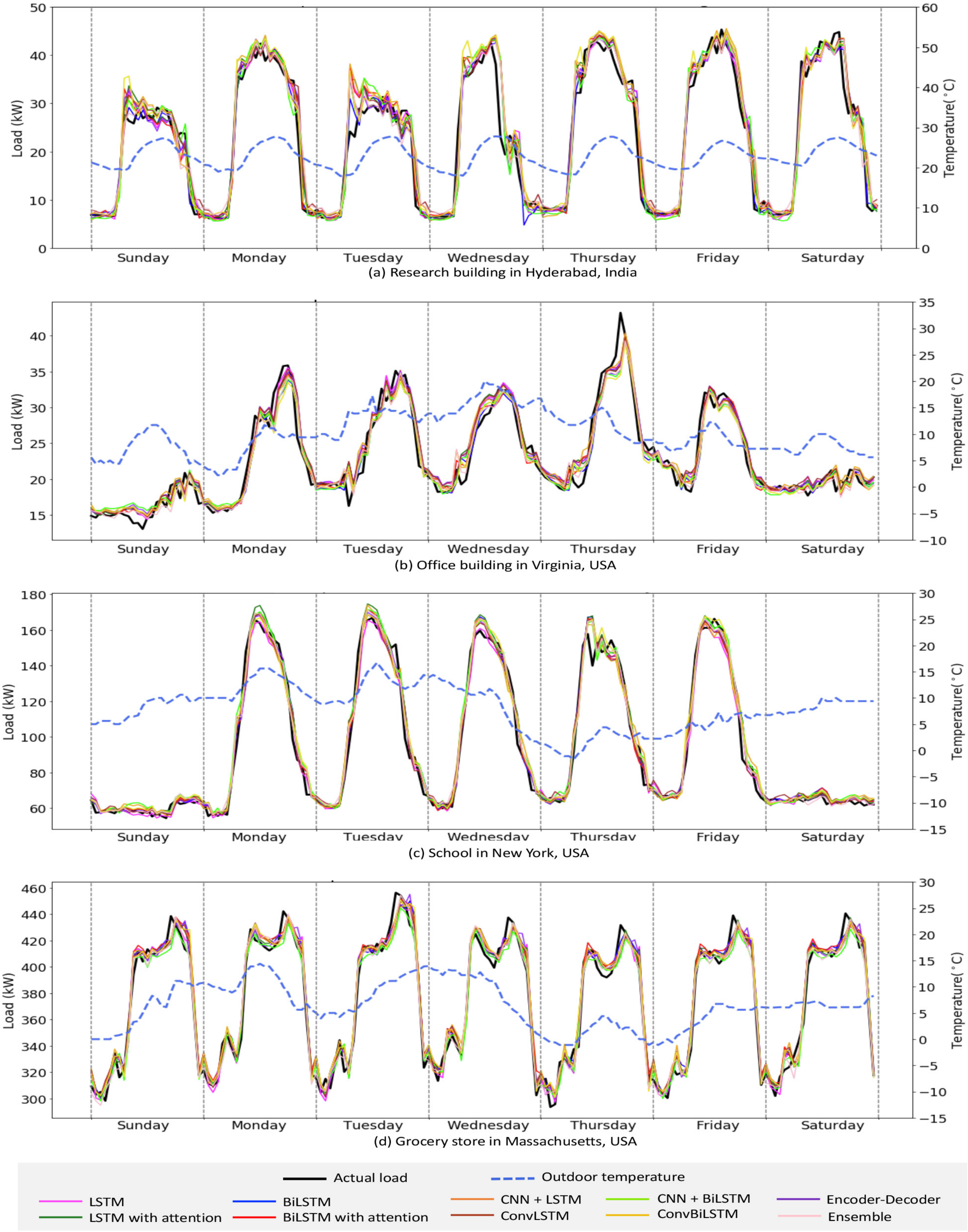


Fig. 11. Actual vs predicted loads–hour-ahead forecasting–for four buildings in: (a) Hyderabad, India; (b) Virginia, USA; (c) New York, USA and (d) Massachusetts, USA.

studied in [47].

For hour-ahead forecasting, with the forecasting methods used in this paper, the forecasting errors for all three U.S. buildings exhibit up to 35% improvement over the state-of-the-art results.

- The building in Virginia, USA: BiLSTM with attention performs the best with an RMSE of 1.58, MAPE of 5.35, and CV of 7.29. This is equivalent to 35.4% and 33.2% reduction in MAPE and CV errors respectively from the state-of-the-art results.
- The building in New York, USA: LSTM performs the best with an

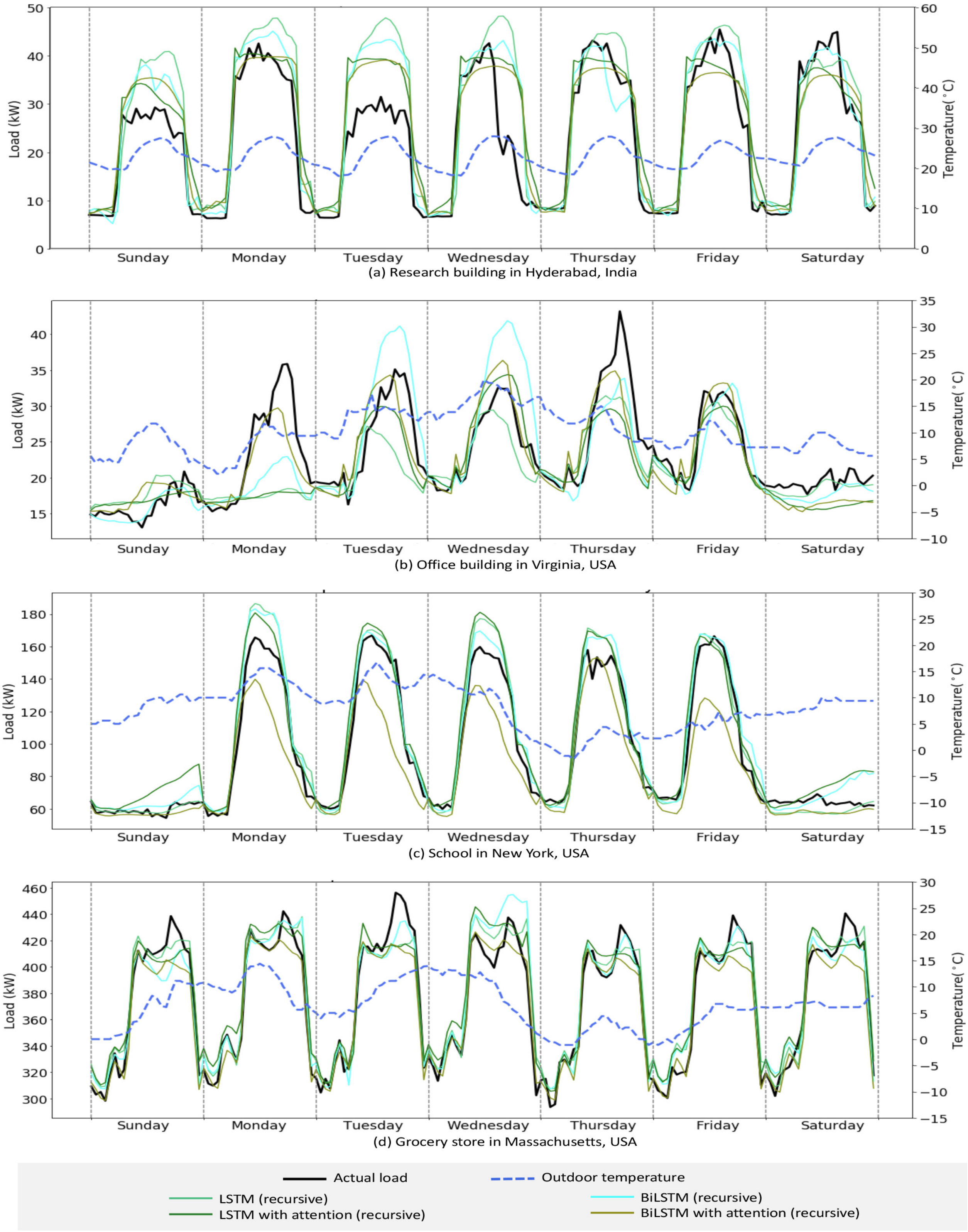


Fig. 12. Actual vs predicted loads–24-h ahead forecasting–for four buildings in: (a) Hyderabad, India; (b) Virginia, USA; (c) New York, USA and (d) Massachusetts, USA.

RMSE of 5.33, MAPE of 3.97, and CV of 5.63. This is equivalent to 31.4% and 22.5% reduction in MAPE and CV errors respectively, from the state-of-the-art results.

- The building in Massachusetts, USA: BiLSTM with attention performs the best with an RMSE of 10.05, MAPE of 1.87, and CV of 2.70. This is equivalent to 19.0% and 4.9% reduction in MAPE and CV errors respectively, from the state-of-the-art results.
- The clustering method again further improves forecasting accuracy for all U.S. buildings than the non-clustering results.

Table 15
RMSLE results for different buildings.

	Bldg 1	Bldg 2	Bldg 3	Bldg 4	Bldg 5
Hour-ahead Load Forecasting					
LSTM	0.23	0.17	0.07	0.05	0.03
LSTM with attention	0.21	0.17	0.07	0.05	0.03
BiLSTM	0.22	0.17	0.07	0.05	0.03
BiLSTM with attention	0.20	0.17	0.07	0.05	0.03
Clustering	0.19	0.17	0.065	0.05	0.03
24hour-ahead Load Forecasting					
LSTM	0.31	0.26	0.14	0.08	0.04
LSTM with attention	0.32	0.31	0.17	0.11	0.04
BiLSTM	0.30	0.23	0.15	0.08	0.04
BiLSTM with attention	0.54	0.25	0.15	0.14	0.03

Notice that while the improvement of over 20–35% is achieved when applying the presented forecasting methods on the buildings in Virginia and New York, the building in Massachusetts has slightly less improvement. The reason for this can be from the fact that the deep learning family used in this paper can capture daily and weekly patterns in building operation well. Hence, better results are observed for the buildings in Virginia and New York that exhibit such variations. On the other hand, for the building in Massachusetts, which is a grocery store that operates seven days a week, and its refrigeration system also operates 24 h a day. Hence, the current state of the art approach [47] already performs well.

For 24-h ahead forecasting, again, the forecasting errors for all three U.S. buildings exhibit up to 45% improvement using the forecasting methods presented in this paper over the state of the art.

- The building in Virginia, USA: LSTM with attention performs the best with an RMSE of 3.3, MAPE of 11.64, and CV of 15.16. This is equivalent to 22.35% reduction in RMSE error from the state-of-the-art results.
- The building in New York, USA: LSTM with attention performs the best with an RMSE of 7.21, MAPE of 5.96, and CV of 7.87. This is equivalent to 46.4% reduction in RMSE error from the state-of-the-art results.
- The building in Massachusetts, USA: LSTM performs the best with an RMSE of 11.27, MAPE of 2.25, and CV of 3.09. This is equivalent to 21.5% reduction in RMSE error from the state-of-the-art results.

The plots for actual vs predicted loads for the remaining four buildings are depicted in Figs. 11 and 12, for hour-ahead and 24-h ahead forecasting, respectively.

6.3. Comparison among Buildings using RMSLE

As mentioned in Section 2.6.3, this paper uses RMSLE to allow fair comparison of forecasting errors among buildings of different peak loads. See Table 15.

As shown, the RMLSE results reveal an interesting insight. It can be seen that the building with the highest RMSLE is the one in Bangkok, Thailand. This building has the highest difference in peak and off-peak loads. The building with the lowest RMSLE is the building in Massachusetts, USA. This building has the lowest difference in peak and off-peak load. This implies that better forecasting accuracy can be expected for buildings that do not have much variations in loads throughout a day.

7. Conclusion

This work focuses on exploring short-term electrical load forecasting methods that are robust against variations in building type, building operation, i.e., day/night, seasonal patterns, weekdays/weekends and holidays, seasonal change, as well as building locations that result in changes in the outdoor environment. In this study, input feature sets and input clustering sets have been carefully selected and models' hyperparameters have been thoroughly fine-tuned so that the resulting model performances are optimal. Research findings indicate that the ensemble technique with unsupervised k-mean clustering delivers the least load forecasting errors as variations in building loads (i.e., weekdays/weekends, day/night) are automatically clustered into groups, and the best models can be determined for load forecasting in each cluster. The result indicates that the load forecasting techniques explored in this paper provide up to 45% improvement in load forecasting performance, as compared to the state-of-the-art methods. Attention model delivers the best result in most of the cases. The deep algorithms have been tested on five different buildings and results indicate that satisfactory forecasting accuracy is achieved. Due to satisfactory forecasting results of the initial set of five buildings, it is expected that the developed method can perform well across a variety of buildings and locations.

Though better accuracy is achieved with higher resolution dataset, it comes at the cost of computing power and time. Hence, there will always be a trade off between computation power and accuracy. Future work will be to take the method presented herein as the core model of an online load forecasting platform that allows others to call for load forecasting services. Online learning will also be developed where this load forecasting service can connect to a building energy management system and the forecasting model can automatically update its parameters as real-time building loads are read. The implementation of deep neural networks for probabilistic short-term load forecasting is also a candidate for future investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] U.S. Energy Information Administration, Electricity explained: use of electricity, April 2019. <https://www.eia.gov/energyexplained/electricity/use-of-electricity.php>, last accessed on 11/20/19.

[2] U.S. Energy Information Administration, Electricity explained: Today in energy, September 2019. <https://www.eia.gov/todayinenergy/detail.php?id=41433>, last accessed on 11/20/19.

[3] Urge-Vorsatz D, Cabeza LF, Serrano S, Barreneche C, Petrichenko K. Heating and cooling energy trends and drivers in buildings. Renew Sustain Energy Rev 2015;41:85–98.

[4] Bunn D, Farmer ED. Comparative models for electrical load forecasting; January 1985.

[5] Mocanu E, Nguyen PH, Gibescu M, Kling WL. Deep learning for estimating building energy consumption. Sustain Energy, Grids Networks 2016;6:91–9.

[6] Hong T. Short term electric load forecasting., Ph.D. Dissertation; September 2010. <https://repository.lib.ncsu.edu/bitstream/handle/1840.16/6457/etd.pdf>, last accessed on 11/20/19.

[7] Friedrich L, Afshari A. Short-term forecasting of the Abu Dhabi electricity load using multiple weather variables. Energy Proc 2015;75:3014–26.

[8] Ravadanegh SA, Jahanyari N, Amini A, Taghizadeghan N. Smart distribution grid multistage expansion planning under load forecasting uncertainty. IET Gener Transm Distrib 2016;10:1136–44.

[9] Collotta M, Pau G. An innovative approach for forecasting of energy requirements to improve a smart home management system based on able. IEEE Trans Green Commun Network February 2017; 1:112–120.

[10] Sehar F, Pipattanasomporn M, Rahman S. An energy management model to study energy and peak power savings from pv and storage in demand responsive buildings. Appl Energy 2018;173:406–17.

[11] Behl M, Smarra F, Mangharam R. Dr-advisor: A data-driven demand response recommender system. Appl Energy 2016;170:30–46.

[12] Sehar F, Pipattanasomporn M, Rahman S. Demand management to mitigate impacts of plug-in electric vehicle fast charge in buildings with renewables. Energy 2017;120:651–2.

[13] Pipattanasomporn M, Kuzlu M, Rahman S. An algorithm for intelligent home energy

management and demand response analysis. Smart Grid, IEEE Trans. on 2012;3(4):2166–2173.

[14] Sehar F, Pipattanasomporn M, Rahman S. Integrated automation for optimal demand management in commercial buildings considering occupant comfort. Sustain Cities Soc 2017;28:16–29.

[15] Mashhour E, Moghaddas-Tafreshi S. Integration of distributed energy resources into low voltage grid: A market-based multiperiod optimization model. Electric Power Syst Res 2010;80(4):473–80.

[16] Lin J, Pipattanasomporn M, Rahman S. Comparative analysis of auction mechanisms and bidding strategies for p2p solar transactive energy markets. Appl Energy 2019;255.

[17] Chaouachi A, Kamel RM, Andoulsi R, Nagasaka K. Multiobjective intelligent energy management for a microgrid. IEEE Trans Industr Electron 2012;60(4):1688–99.

[18] Heimgaertner F, Hettich S, Kohlbacher O, Menth M. Scaling home automation to public buildings: A distributed multiuser setup for openhab 2. In: 2017 Global Internet of Things Summit (GIoTS); June 2017.

[19] Zhang X, Adhikari R, Pipattanasomporn M, Kuzlu M, Rahman S. Deploying iot devices to make buildings smart: Performance evaluation and deployment experience. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT); December 2016.

[20] Ahmad T, Chen H, Guo Y, Wang J. A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. Energy Build 2018;165:301–20.

[21] Foucquier A, Robert S, Suard F, Stphan L, Jay A. State of the art in building modelling and energy performances prediction: A review. Renew Sustain Energy Rev 2013;23:272–88.

[22] Zhao Hx, Magouls F. A review on the prediction of building energy consumption. Renew Sustain Energy Rev 2012;16:6:3586–3592.

[23] Amber KP, Aslam MW, Hussain SK. Electricity consumption forecasting models for administration buildings of the uk higher education sector. Energy Build 2015;123:127–36.

[24] Dudek G. Pattern-based local linear regression models for short-term load forecasting. Electric Power Syst Res 2016;130:139–47.

[25] Charytoniuk W, Chen MS, Van Olinda P. Nonparametric regression based short-term load forecasting. IEEE Trans Power Syst 1998;13(3):725–30.

[26] Song K, Baek Y, Hong D, Jang G. Short-term load forecasting for the holidays using fuzzy linear regression method. IEEE Trans Power Syst 2005;20(1):96–101.

[27] Robinson C, Hubbs J, Dilkina B, Zhang W, Guhathakurta S, Brown MA, et al. Machine learning approaches for estimating commercial building energy consumption. Appl Energy 2017;208:889–904.

[28] Kim Y, Son H, Kim S. Short term electricity load forecasting for institutional buildings. Energy Reports 2019;5:1270–80.

[29] Zhang R, Dong ZY, Xu Y, Meng K, Wong KP. Short-term load forecasting of australian national electricity market by an ensemble model of extreme learning machine. IET Gener, Transmiss Distrib 2013;7(4):391–7.

[30] Ceperic E, Ceperic V, Baric A. A strategy for short-term load forecasting by support vector regression machines. IEEE Trans Power Syst 2013;28(4):4356–64.

[31] Chen Y, Tan H. Short-term prediction of electric demand in building sector via hybrid support vector regression. Appl Energy 2017;204:1363–74.

[32] 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.

[33] 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 2017;127:1–10.

[34] 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;195:659–670.

[35] Rejc M, Pantos M. Short-term transmission-loss forecast for the slovenian transmission power system based on a fuzzy-logic decision approach. IEEE Trans Power Syst 2011;26(3):1511–21.

[36] Sudheer G, Suseelatha A. Short term load forecasting using wavelet transform combined with holt-winters and weighted nearest neighbor models. Int'l J Electr Power Energy Syst 2015;64:340–6.

[37] Moon J, Kim K, Kim Y, Hwang E. A short-term electric load forecasting scheme using 2-stage predictive analytics. In: 2018 IEEE Int'l Conf. on Big Data and Smart Computing (BigComp), 2018; 219–226.

[38] Grolinger K, L'Heureux A, Capretz MAM, Seewald L. Energy forecasting for event venues: Big data and prediction accuracy. Energy Build 2016;116:222–233.

[39] Jurado S, Nebot À, Mugica F, Avellana N. Hybrid methodologies for electricity load forecasting: entropy-based feature selection with machine learning and soft computing techniques. Energy 2015;86:276–91.

[40] Fard A, Akbari-Zadeh M. A hybrid method based on wavelet, ann and arima model for short-term load forecasting. J Exp Theoret Artif Intell 2014;26(2):167–82.

[41] Vaghefi A, Jafari MA, Bisse E, Lu Y, Brouwer J. Modeling and forecasting of cooling and electricity load demand. Appl Energy 2014;136:186–96.

[42] Massana J, Pous C, Burgas L, Melendez J, Colomer J. Short-term load forecasting in a non-residential building contrasting models and attributes. Energy Build 2015;130:322–30.

[43] Yildiz B, Bilbao JI, Sprou AB. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renew Sustain Energy Rev 2017;73:1104–22.

[44] Walter T, Price PN, Sohn MD. Uncertainty estimation improves energy measurement and verification proceduresforecasting. Appl Energy 2014;130:230–6.

[45] Gianniou P, Liu X, Heller A, Nielsen PS, Rode C. Clustering-based analysis for residential district heating data. Energy Convers Manage 2018;165:840–50.

[46] Rahman A, Srikumar V, Smith AD. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Appl Energy 2018;212:372–85.

[47] Cai M, Pipattanasomporn M, Rahman S. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. Appl Energy 2019;236:1078–1088.

[48] Marino DL, Amarasinghe K, Manic M. Building energy load forecasting using deep neural networks. In: IECON 2016 - 42nd Annual Conf. IEEE Industrial Electronics Society; October 2016.

[49] Muzaffar S, Afshari A. Short-term load forecasts using lstm networks. Energy Proc 2019;158:2922–7.

[50] Dagdougui H, Bagheri F, Le H, Dessaint L. Neural network model for short-term and very-short-term load forecasting in district buildings. Energy Build 2019;203.

[51] Fan C, Xiao F, Zhao Y. A short-term building cooling load prediction method using deep learning algorithms. Appl Energy 2017;195:222–33.

[52] Kim J, Moon J, Hwang E, Kang P. Recurrent inception convolution neural network for multi short-term load forecasting. Energy Build 2019;194:328–41.

[53] Shi H, Xu M, Li R. Deep learning for household load forecasting–a novel pooling deep RNN. IEEE Trans Smart Grid 2018;9(5):5271–80.

[54] Bedi J, Toshniwal D. Deep learning framework to forecast electricity demand. Appl Energy 2019;238:1312–26.

[55] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al. editors, Advances in Neural Information Processing Systems 30. Curran Associates, Inc., 2017. p. 5998–6008.

[56] Bourdeau M, Zhai X, Nefzaoui E, Guo X, Chatellier P. Modeling and forecasting building energy consumption: A review of data-driven techniques. Sustain Cities Soc 2019;48.

[57] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8).

[58] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors, Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012. p. 1097–1105.

[59] Ashrae - great energy predictor iii. https://www.kaggle.com/c/ashrae-energy-prediction/overview/evaluation/, last accessed on 12/23/19.

[60] Chulalongkorn university's building energy management (CUBEMS). http://www.bems.chula.ac.th/web/central/#, last accessed on 11/20/19.

[61] Bhatia A, Garg V, Mathur J. Determination of energy saving with cool roof concept using calibrated simulation: Case of a learning centre in composite indian climate. J Sol Energy Soc India; February 2012.

[62] Enernoc commerical building dataset.https://openenernoc-data.s3.amazonaws.com/anon/index.html, last accessed on 11/20/19.

[63] Weather underground. https://www.wunderground.com/, last accessed on 11/20/19.

[64] timeanddate.com. https://www.timeanddate.com/, last accessed on 11/20/19.