Energy & Buildings 216 (2020) 109921

Contents lists available at ScienceDirect

Energy & Buildings

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

Combination of short-term load forecasting models based on a stacking ensemble approach

Jihoon Moon[a], Seungwon Jung[a], Jehyeok Rew, Data curation; Visualization[a], Seungmin Rho[b,*], Eenjun Hwang, Project administration; Supervision; Writing - review & editing[a,*]

[a] School of Electrical Engineering, Korea University, Seoul 136-701, South Korea
[b] Department of Software, Sejong University, Seoul 143-747, South Korea

ARTICLE INFO

Article history:
Received 11 September 2019
Revised 28 January 2020
Accepted 5 March 2020
Available online 19 March 2020

Keywords:
Short-term load forecasting
Building energy consumption forecasting
Stacking ensemble approach
Deep neural network
Principal component regression

ABSTRACT

Building electric energy consumption forecasting is essential in establishing an energy operation strategy for building energy management systems. Because of recent developments of artificial intelligence hardware, deep neural network (DNN)-based electric energy consumption forecasting models yield excellent performances. However, constructing an optimal forecasting model using DNNs is difficult and time-consuming because several hyperparameters must be determined to obtain the best combination of neural networks. The determination of the number of hidden layers in the DNN model is challenging because it greatly affects the forecasting performance of the DNN models. In addition, the best number of hidden layers for one situation or domain is often not optimal for another domain. Hence, many efforts have been made to combine multiple DNN models with different numbers of hidden layers to achieve a better forecasting performance than that of an individual DNN model. In this study, we propose a novel scheme for the combination of short-term load forecasting models using a stacking ensemble approach (COSMOS), which enables the more accurate prediction of the building electric energy consumption. For this purpose, we first collected 15-min interval electric energy consumption data for a typical office building and split them into training, validation, and test datasets. We constructed diverse four-layer DNN-based forecasting models based on the training set and by considering the input variable configuration and training epochs. We selected optimal DNN parameters using the validation set and constructed four DNN-based forecasting models with various numbers of hidden layers. We developed a building electric energy consumption forecasting model using the test set and sliding window-based principal component regression for the calculation of the final forecasting value from the forecasting values of the four DNN models. To demonstrate the performance of our approach, we conducted several experiments using actual electric energy consumption data and verified that our model yields a better prediction performance than other forecasting methods.

1. Introduction

The energy consumption in buildings accounts for a large portion of the total global energy consumption and carbon dioxide (CO_2) emissions [1–3]. The building energy management system (BEMS) developed for the efficient management and monitoring of the building energy consumption has attracted much attention as a means of reducing the CO_2 emissions by cutting the total energy consumption in buildings [2,3]. The BEMS enhances the energy consumption efficiency while maintaining the comfort of building users by effectively analyzing the energy-saving factors based on various sensors and facilities in a building [3,4]. The BEMS also reduces the energy consumption and CO_2 emissions by actively utilizing renewable energy (RE) [3]. The BEMS is beneficial to both the suppliers of electric loads and consumers who use electric loads [5,6]. On the supply side, the BEMS is used to trade and utilize surplus electric energy from energy storage systems (ESSs) and REs in smart grids or microgrids [5]. In addition, it provides a basis for electric power corporations to decrease the operating costs of old power plants [7]. On the demand side, the BEMS is used to reduce the energy bills of consumers by controlling unnecessary electric energy consumption and coping with peak loads [3,4].

* Corresponding authors.
E-mail addresses: smrho@sejong.edu (S. Rho), ehwang04@korea.ac.kr (E. Hwang).

https://doi.org/10.1016/j.enbuild.2020.109921
0378-7788/

Abbreviations

ANN	artificial neural network
ARIMA	autoregressive integrated moving average
BEMS	building energy management system
BPNN	backpropagation neural network
CEMS	cluster or community energy management system
CHP	combined heat and power
CNN	convolutional neural network
CPU	central processing unit
DAE	deep auto-encoder
DBN	deep belief network
DNN	deep neural network
DSM	demand-side management
DT	decision tree
ELM	extreme learning machine
ELN	Elastic net
ESS	energy storage system
FIR	fuzzy inductive reasoning
GBM	gradient boosting machine
GP	genetic programming
HVAC	heating, ventilation, and air conditioning
iPSO	improved particle swarm optimization
IT	information technology
KMA	Korea Meteorological Administration
KNN	k-nearest neighbors
LSTM	long short-term memory
MAE	mean absolute error
MAPE	mean absolute percentage error
ML	machine learning
MLR	multiple linear regression
MWD	multiresolution wavelet decomposition
PCA	principal component analysis
PCR	principal component regression
PDRNN	pooling-based deep recurrent neural network
RE	renewable energy
ReLU	rectified linear units
RF	random forest
RMSE	root-mean-square error
RNN	recurrent neural network
S2S	sequence to sequence
SNN	shallow neural network
STLF	short-term load forecasting
SVR	support vector regression
VSTLF	very short-term load forecasting
XGB	extreme gradient boosting

Existing building management methods are operated by simple condition monitoring and the manual control of equipment [1,4]. However, to build a low-cost and high-efficiency system, such building management methods must be combined with software technologies [5]. For instance, artificial intelligence-based short-term load forecasting (STLF) is a representative example of software technology [8]. The STLF can be classified into two categories [9]: STLF (time horizon: one day to two weeks) and very short-term load forecasting (VSTLF; time horizon: less than one day). The VSTLF with a forecast period shorter than one hour plays a significant role in scheduling BEMS operations associated with the ESS, RE, and peak load response [7,10]. To accurately perform VSTLF, the forecasting model must effectively reflect factors affecting the fluctuations in the electric energy consumption [11–13]. In addition, the complex correlation between current and previous electric energy consumption must be considered [7]. Machine learning algorithms, such as artificial neural networks (ANNs) and support vector regression (SVR), have been applied, which consider these factors during forecasting [11,12].

Recently, it has been reported that deep neural network (DNN)-based forecasting models exhibit excellent prediction performances because they effectively reflect these factors [13]. The DNNs are used to obtain the desired knowledge (i.e., prediction values) from the large amount of training data of a forecasting model. However, the development of an optimal DNN model is time-consuming because diverse hyperparameters must be considered [14–16]. The determination of the optimal number of hidden layers of neural networks is difficult [15]. For instance, the increase in the number of hidden layers does not guarantee a performance improvement. In addition, the optimal number of hidden layers for one case or domain does often not guarantee the optimal performance in another case. Hence, there have been several efforts to combine multiple DNN models with different numbers of hidden layers to achieve a better forecasting performance than that obtained with individual DNN models [17]. Another problem related to learning-based forecasting is that the more time has passed since the last training, the less accurate is the prediction. This problem should be addressed, for instance, by considering recent electric loads [6].

In this study, we propose a novel forecasting model for the building electric energy consumption called COSMOS, which combines multiple STLF models using a stacking ensemble approach. We first split the building electric energy consumption data into training, validation, and test datasets. We constructed different input variables and training epochs using the training dataset and then identified the optimal parameters for the effective training of the forecasting model using the validation dataset. Subsequently, we set up several DNN-based forecasting models with different numbers of hidden layers using the training and validation datasets. To reflect recent electric energy consumption trends, we constructed a sliding window-based principal component regression (PCR) model using the predictions from the DNN models. During the experiments, we compared the prediction performance of our proposed model with the performances of various DNN models, machine learning (ML) methods, and our previous forecasting models based on two popular performance indicators.

The remainder of this paper is organized as follows. We briefly review previous work on ML and deep learning-based building electric energy consumption forecasting models in Section 2. We describe the data preprocessing for the construction of the input variables for the COSMOS model in Section 3. The development of the COSMOS model is explained in detail in Section 4. The experimental results are presented in Section 5 to demonstrate the superiority of our model. Finally, a conclusion and future research directions are provided in Section 6.

2. Related work

So far, many studies have been conducted on forecasting models for the building electric energy consumption based on ML techniques such as SVR, ANN, and tree-based methods [9,11]. Table 1 summarizes the information about the selected papers, and these studies are described in detail subsequently. For example, Jain et al. [18] proposed an SVR-based electric energy consumption forecasting model for a multi-family residential building in New York City. They confirmed the prediction performance of their proposed model by using the effects of the temporal (i.e., daily, hourly, 10-min interval) and spatial (i.e., the whole building, by floor, by unit) granularities. Chen and Tan [19] developed a combination of a hybrid SVR model and multiresolution wavelet decomposition (MWD) to predict the hourly electric energy consumption of a hotel or mall. Grolinger et al. [20] constructed several forecasting models for a large entertainment venue based on SVR and ANN by considering various input variable combinations. They analyzed

Table 1
Summary of several approaches for building energy consumption forecasting.

Author (Year)	Target	Granularity	Input variable	Prediction method
Palchak et al. [24]	Colorado State University	1 h	Calendar information, historical electric load, weather factor	ANN
Jain et al. [18]	Multi-family residential buildings	Daily, 1 h, 10 min	Calendar information, historical electric load, weather factor	SVR
Amber et al. [21]	Administration building of the London South Bank University	Daily	Calendar information, weather factor	MLR, GP
Jurado et al. [22]	Three buildings of the UPC	1 h	Calendar information, historical electric load	FIR, RF, ANN, ARIMA
Li et al. [26]	Great Building Energy Predictor Shootout, library building	1 h	Calendar information, historical electric load, weather factor	iPSO–ANN
Bagnasco et al. [25]	Hospital	15 min	Calendar information, historical electric load, weather factor	ANN
Grolinger et al. [20]	Event venue	Daily, 1 h, 15 min	Calendar information, historical electric load	SVR, ANN
Marino et al. [29]	Single residential customer	1 min	Calendar information, historical electric load	LSTM
Ryu et al. [27]	40 large-industry customers	1 h	Calendar information, historical electric load, weather factor	DBN, DNN
Chen and Tan [19]	Mall and hotel	1 h	Historical electric load, weather factor	SVR
Fan et al. [28]	Educational building	30 min	Calendar information, historical electric load, weather factor	MLR, ELN, RF, GBM, SVR, XGB, DNN
Ahmad et al. [23]	Hotel	1 h	Calendar information, weather factor, number of guests and rooms booked for the day	RF, ANN
Shi et al. [30]	920 smart-meter customers	30 min	Historical electric load	PDRNN
Cai et al. [31]	Academic building, primary/secondary school, grocery store	1 h	Historical electric load, weather factor	CNN, RNN
Kong et al. [32]	69 individual customers	30 min	Calendar information, historical electric load	LSTM

the prediction performances of their forecasting models by using the effect of the temporal granularity (i.e., daily, hourly, 15-min interval). Amber et al. [21] developed two daily electric energy consumption forecasting models for an administration building based on multiple linear regression (MLR) and genetic programming (GP). The results of their experiment revealed that the GP model yields slightly better forecasts than the MLR model; however, the training time of the GP model is longer than that of the MLR model. Jurado et al. [22] developed STLF models based on fuzzy inductive reasoning (FIR), random forest (RF), and ANNs. They used the forecasting models to predict the electric load of three buildings of the Technical University of Catalonia (UPC) and compared the prediction performances with that of the autoregressive integrated moving average (ARIMA). Ahmad et al. [23] proposed two heating, ventilation, and air conditioning (HVAC) energy consumption forecasting models based on ANN and RF. They compared the prediction performances of the ANN and RF using the energy consumption of a hotel in Madrid, Spain. The results of their experiment revealed that the ANN performs marginally better than the RF. Palchak et al. [24] constructed an hourly electric load forecasting model based on ANN for the development of a demand-side management (DSM) program of Colorado State University. Bagnasco et al. [25] developed an ANN-based forecasting model to forecast the day-ahead electric energy consumption of a large hospital facility. Li et al. [26] predicted the electric energy consumption of the Great Building Energy Predictor Shootout I and a library building using a hybrid improved particle swarm optimization (iPSO)–ANN method. The iPSO algorithm was applied to adjust the weights and thresholds of the ANN structure.

Recently, with the development of information technology (IT), many studies regarding the construction of accurate STLF models have been carried out using deep learning, especially DNNs. Ryu et al. [27] built two DNN-based electric energy consumption forecasting models using a pretrained deep belief network (DBN) and rectified linear units (ReLU) without pretraining. Their models yielded more accurate forecasting results than other forecasting methods such as the shallow neural network (SNN), double-seasonal Holt–Winters method, and seasonal ARIMA. Fan et al. [28] developed deep learning-based methods to achieve accurate and reliable day-ahead building cooling load forecasts. The deep learning-based techniques, such as the DNN and deep autoencoder (DAE), were analyzed in terms of the accuracy and computation efficiency. The results indicated that deep learning-based methods enhance the prediction performance mainly when a DAE is used to construct high-level features as the input variables for the forecasting models such as the MLR, extreme learning machine (ELN), RF, gradient boosting machine (GBM), SVR, extreme gradient boosting (XGB), and DNN. Several recent studies focused on improving the STLF performance by using recurrent neural network (RNN) models that can effectively process sequential data. Marino et al. [29] proposed two long short-term memory (LSTM)-based electric energy consumption forecasting models: 1) standard LSTM and 2) LSTM-based sequence to sequence (S2S) architecture. Both models were built for one-hour and one-minute time-step resolution data. The results of their experiment demonstrated that the LSTM-based S2S architecture performs well in both datasets. Shi et al. [30] proposed a household electric load forecasting model based on a pooling-based deep recurrent neural network (PDRNN), which batches a group of customer load profiles into an input pool. They used their proposed model to predict the load for 920 smart-meter customers from Ireland and compared their model with several household electric load forecasting models. The experimental results demonstrated that their proposed model outperforms the ARIMA, SVR, and RNN by 19.5%, 13.1%, and 6.5%, respectively, in terms of the root-mean-square error (RMSE). Cai et al. [31] proposed two DNN models, that is, the RNN and convolutional neu-

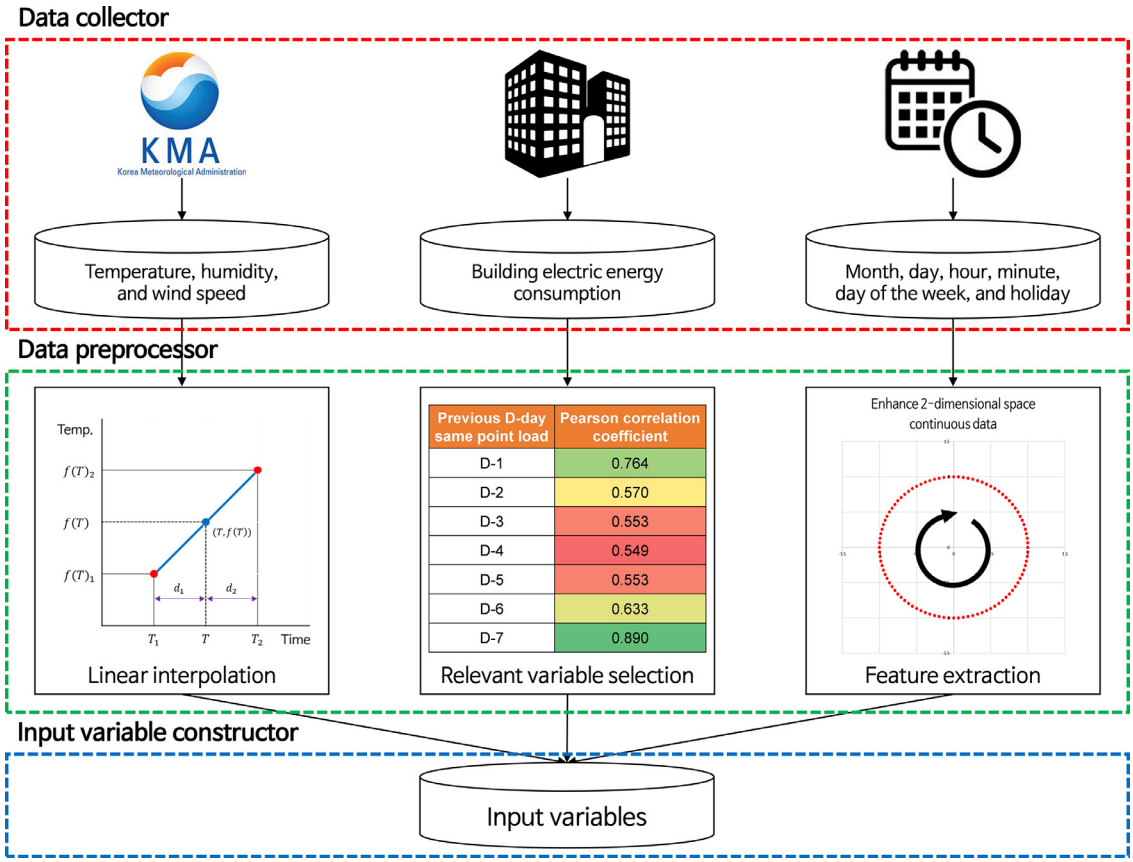


Fig. 1. Input variable construction for the DNN model.

ral network (CNN), for day-ahead multi-step electric load forecasts in commercial buildings. They compared the prediction performances with that of the seasonal ARIMA with exogenous inputs (SARIMAX) model in terms of the computational efficiency, generalizability, and prediction accuracy. The experimental results indicated that their forecasting models reduce the day-ahead multi-step forecasting error compared with the SARIMAX model. Kong et al. [32] proposed a STLF model for individual electric customers based on LSTM–RNN. The proposed model was used to predict a publicly available set of private smart-meter data. They compared the prediction performance with those of several forecasting models based on ML methods such as the backpropagation neural network (BPNN), k-nearest neighbors (KNN), and ELM. The experimental results revealed that their proposed LSTM-based model outperforms the other forecasting models with respect to the electric energy consumption forecasting of individual residential households.

The differences between the models described above and our model are as follows:

- The prediction performances of the models proposed in previous studies [18–21,23–26,29] were compared with a maximum of three ML algorithms or forecasting models. In contrast, we analyzed the prediction performance of our model and compared it with various ML algorithms and our previous forecasting models.
- Previous models proposed for building electric energy consumption forecasting targeted more than 30 min as the time resolution [19,21–24,26–28,30–32]. In contrast, our model is based on 15-min interval building electric energy consumption forecasting to obtain a more granular energy management.
- Previous proposed models based on deep learning methods [28,30–32] exhibit excellent prediction performances. However, they are difficult to apply as baseline models because they require high-performance computer specifications and incur high computational costs. In contrast, our model can quickly implement deep learning algorithms in a central processing unit (CPU) environment by using scikit-learn.

3. Data preprocessing

In this section, we describe the selection of factors affecting the building electric energy consumption and discuss the preprocessing performance to configure the input variables of the forecasting model. We first collected 15-min interval electric energy consumption data from the Korea Electric Power Corporation (KEPCO) through the i-SMART system. The building we considered in this study is a 12-story building and serves as the headquarters of a midsize company in Seoul, South Korea. In the building, heating and hot water are supplied by gas in winter only, and air-conditioning, lighting, office equipment, servers, and other facilities are powered by electric energy. The data were collected from February 1, 2015, to February 28, 2017, that is, over a period of 25 months. One of our goals was to construct a general STLF model. Therefore, we extracted time features to effectively reflect a variety of calendar information. In addition, we considered various weather factors and historical electric load data, which are fundamental elements [33]. Fig. 1 illustrates the overall data preprocessing process. The details are described in the following subsections.

3.1. Weather factor

The weather data that we used in this study are forecast data. The KMA provides weather forecasts, including Dong-Nae Forecast

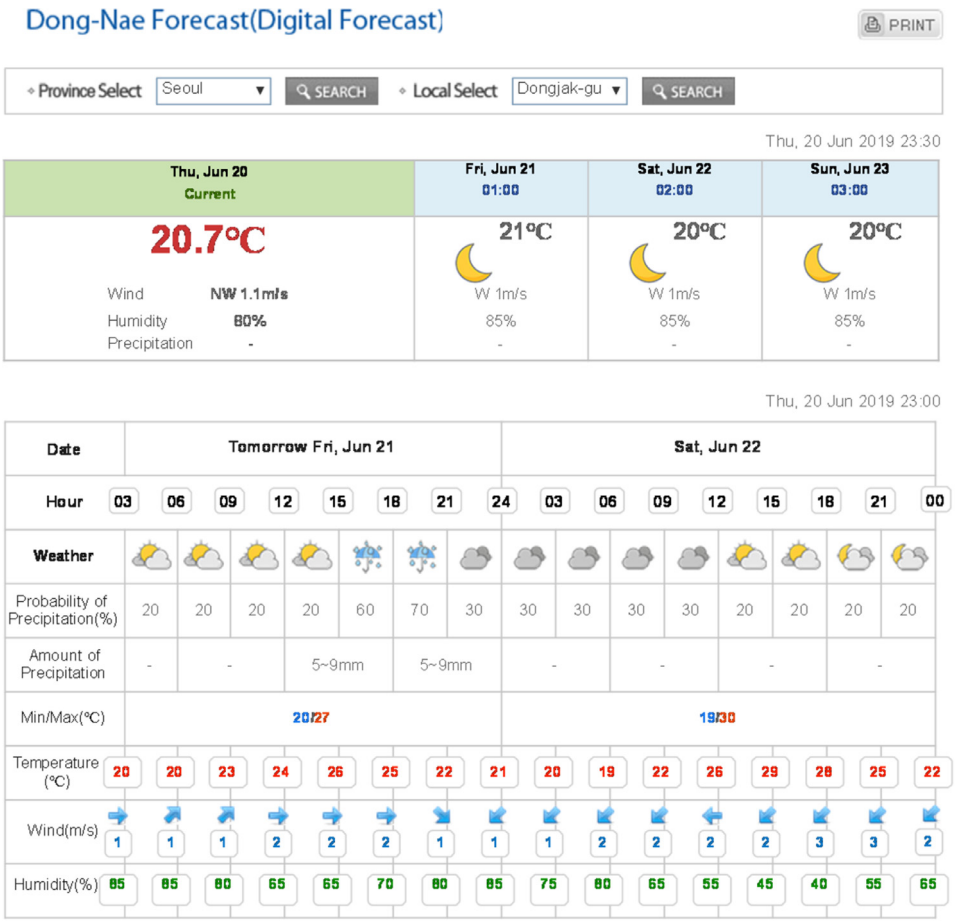


Fig. 2. Example of a Dong-Nae Forecast by the KMA [34].

Table 2
Variable description from calendar information.

No.	Input variable	Variable type
1	$Minute_x$	Continuous on [−1, 1]
2	$Minute_y$	Continuous on [−1, 1]
3	$Hour_x$	Continuous on [−1, 1]
4	$Hour_y$	Continuous on [−1, 1]
5	Day_x	Continuous on [−1, 1]
6	Day_y	Continuous on [−1, 1]
7	$Month_x$	Continuous on [−1, 1]
8	$Month_y$	Continuous on [−1, 1]
9	Monday	Binary [1: Monday, 0: other days]
10	Tuesday	Binary [1: Tuesday, 0: other days]
11	Wednesday	Binary [1: Wednesday, 0: other days]
12	Thursday	Binary [1: Thursday, 0: other days]
13	Friday	Binary [1: Friday, 0: other days]
14	Saturday	Binary [1: Saturday, 0: other days]
15	Sunday	Binary [1: Sunday, 0: other days]
16	Holiday	Binary [1: holiday, 0: weekday]

and Mid-Term Forecast, for most major areas. The Dong-Nae Forecast and the Mid-Term Forecast data include weather forecasts for three days from the present day, and day 3 to day 10, respectively. Because our forecasting model focuses on day-ahead load forecasting, we gathered weather data for the same area using Dong-Nae Forecast data provided by the KMA [34], and an example is illustrated in Fig. 2. We decided to use the forecast data for the following reasons:

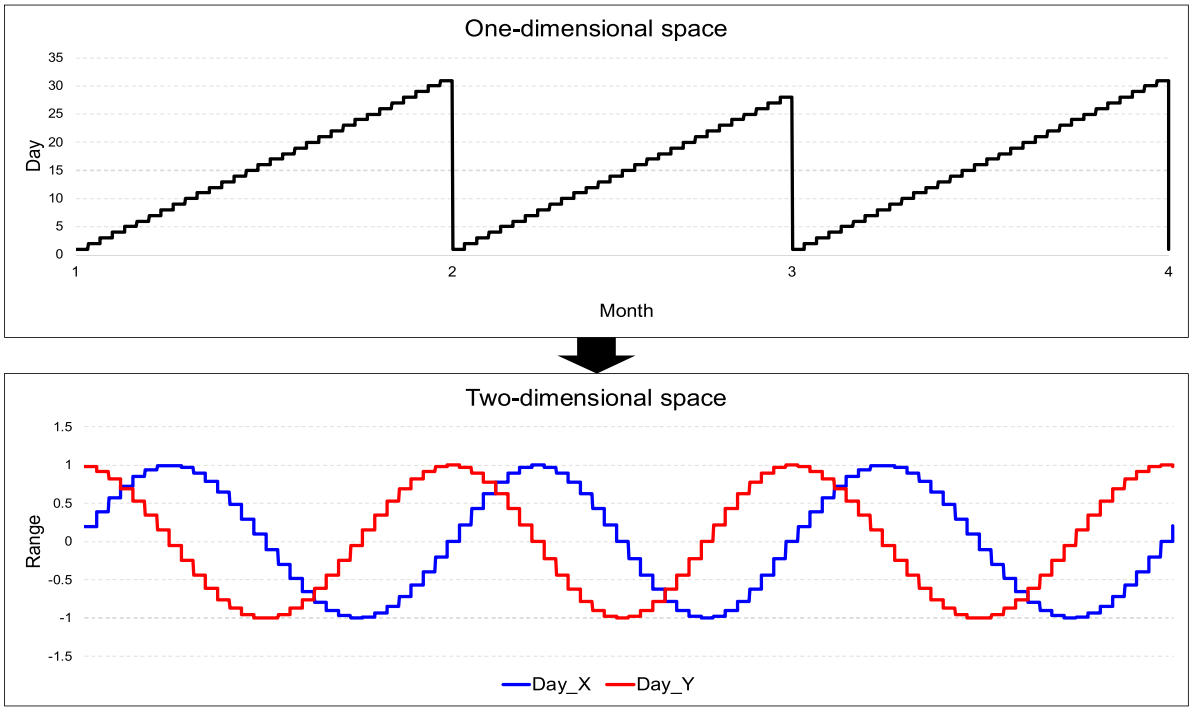
- The Dong-Nae Forecast service is very accurate and hence there is little difference between the forecast data and the measured data. Therefore, regardless of which data was used for training, the prediction performance would be almost identical. In fact, in our previous work [6,7,35], we used the measured weather data for training.
- Also, we considered additional cases to make our model more versatile. That is, if the forecast data tends to be higher (or lower) than the measured data for some reason, it would be more reasonable to use forecast data for training because the forecast data used in prediction would be higher (or lower) than the actual data.

Table 3
Statistics correlation and regression analysis.

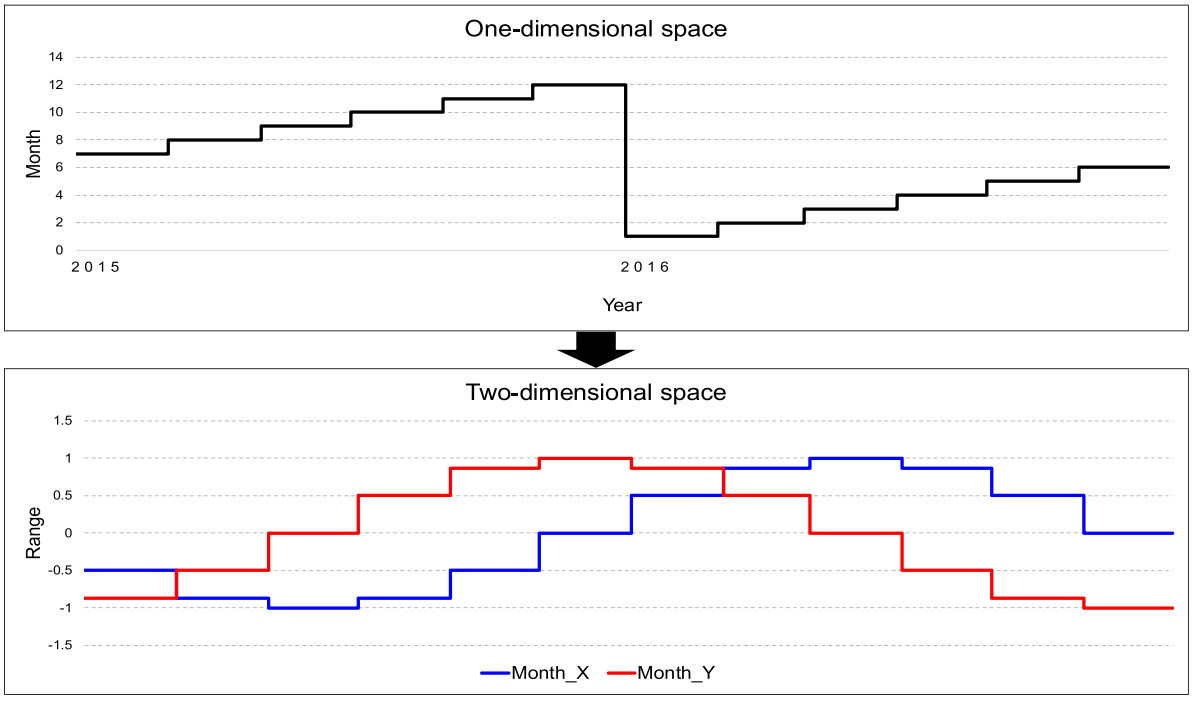
Regression statistics	One-dimensional space	Two-dimensional space
Multiple R-squared	0.0446	0.4823
Adjusted R-squared	0.0445	0.4822
Standard error	28.8	21.2

In this study, we considered three types of weather data, that is, the temperature, humidity, and wind speed, which are closely related to the electric energy consumption [7]. However, because the weather data were collected at a 3-h resolution, it is not appropriate to directly apply them to the forecasting model because of the gap with the prediction period. To obtain weather data at the same time resolution as that of the electric energy consumption, we used a linear interpolation method to estimate the weather data at a 15-min resolution:

$$f(T) = \frac{d_2}{d_1 + d_2} f(T_1) + \frac{d_1}{d_1 + d_2} f(T_2) \quad (1)$$



(a) Day



(b) Month

Fig. 3. Cases of one-dimensional space to two-dimensional space.

When the data values at two points T_1 and T_2 are $f(T_1)$ and $f(T_2)$, respectively, the data value $f(T)$ at any point $T(T_1 \leq T \leq T_2)$ between the two points T_1 and T_2 can be calculated using Eq. (1), where d_1 is the distance between T and T_1 and d_2 is the distance between T and T_2.

3.2. Calendar information

Because building electric energy consumption data are basically time series data, the periodicity of the time for the time series data must be determined before applying the DNN models. We

Table 4
Comparison of the Pearson correlation coefficients.

Previous D-day same point load	Pearson correlation coefficient
D-1	0.764
D-2	0.570
D-3	0.553
D-4	0.549
D-5	0.553
D-6	0.633
D-7	0.890

considered all input variables that could express calendar data, namely, months, days, hours, minutes, days of the week, and holidays. Overall, 16 calendar data were used to configure input variables, as shown in Table 2. The time data, such as the month, day, hour, and minute, exhibit a sequence structure. It is challenging to adequately reflect periodic information when a sequence structure is applied to DNNs [7,35]. For instance, even though 11 pm and midnight are adjacent, their sequence difference in the sequence format is 23. Additionally, even though December 31, 2015 and January 1, 2016 are adjacent, their difference is 31 in the sequence format. Consequently, we used Eqs. (2)–(9) to enhance the one-dimensional (1D) space data to continuous two-dimensional (2D) space data to reflect the periodicity [35]. The two-dimensional space data are numerical data acquired from the sin-cos transformation of one-dimensional data such as a month, day, hour, and minute. In Eqs. (6) and (7), LDM_{month} represents the last day of the month to which the day belongs, e.g., LDM_{March} is 31 and LDM_{June} is 30.

$$minute_x = \sin\left(\left(\frac{360}{60}\right) \times minute\right) \tag{2}$$

$$minute_y = \cos\left(\left(\frac{360}{60}\right) \times minute\right) \tag{3}$$

$$hour_x = \sin\left(\left(\frac{360}{24}\right) \times hour\right) \tag{4}$$

$$hour_y = \cos\left(\left(\frac{360}{24}\right) \times hour\right) \tag{5}$$

$$day_x = \sin\left(\left(\frac{360}{LDM_{month}}\right) \times day\right) \tag{6}$$

$$day_y = \cos\left(\left(\frac{360}{LDM_{month}}\right) \times day\right) \tag{7}$$

$$month_x = \sin\left(\left(\frac{360}{12}\right) \times month\right) \tag{8}$$

$$month_y = \cos\left(\left(\frac{360}{12}\right) \times month\right) \tag{9}$$

Fig. 3(a) and (b) show to enhance the sequence data on 1D space to continuous data in 2D space using Eqs. (6)–(7) and (8)–(9), respectively. Likewise, we can reflect December 2015 and January 2016 as the adjacent distance in the case of the month. In the case of the day, we also can reflect LDM_{month} and the first day of next month as the nearby distance. To verify the validity and applicability of 2D representation, we calculated several regression statistics on the building electric energy consumption in 1D space (month, day, hour, and minute) and in 2D space, as shown in Table 3. In this table, we confirmed that 2D representation can explain their correlation more effectively than 1D representation. Therefore, Eqs. (2)–(9) give a total of eight input variables to represent the date and time of the prediction time points.

The building electric energy consumption exhibits several usage patterns depending on the working hours [36]. For instance, a high electric energy consumption is observed during working hours and low electric energy consumption is observed during other hours. The days of the week and holidays are excellent indicators of the working hours. Holidays include Saturdays, Sundays, and public holidays. To reflect such usage patterns, the days of the week and holiday information were used as input variables. If the day of the week and holiday information are corresponding dates, we configured the input variables using one-hot encoding (i.e., "1" for the relevant date and "0" otherwise) to reflect the input variable configuration of the forecasting model.

3.3. Historical electric load

If the forecasting model considers only calendar and weather information for the prediction, accurate electric energy consumption forecasting may not be possible because it cannot reflect recent trends in the electric energy consumption [17]. Historical electric load data play a key role in sophisticated forecasting model construction because they can represent recent patterns and trends

Table 5
List of input variables for the proposed forecasting model.

No.	Input variable	Variable type	Data type (unit)
1	Temperature	Continuous	Weather factor (°C)
2	Humidity	Continuous	Weather factor (%)
3	Wind speed	Continuous	Weather factor (m/s)
4	$Minute_x$	Continuous on [−1, 1]	Calendar information
5	$Minute_y$	Continuous on [−1, 1]	Calendar information
6	$Hour_x$	Continuous on [−1, 1]	Calendar information
7	$Hour_y$	Continuous on [−1, 1]	Calendar information
8	Day_x	Continuous on [−1, 1]	Calendar information
9	Day_y	Continuous on [−1, 1]	Calendar information
10	$Month_x$	Continuous on [−1, 1]	Calendar information
11	$Month_y$	Continuous on [−1, 1]	Calendar information
12	Monday	Binary	Calendar information
13	Tuesday	Binary	Calendar information
14	Wednesday	Binary	Calendar information
15	Thursday	Binary	Calendar information
16	Friday	Binary	Calendar information
17	Saturday	Binary	Calendar information
18	Sunday	Binary	Calendar information
19	Holiday	Binary	Calendar information
20	Previous D-1 same point load	Continuous	Historical electric load (kW)
21	Previous D-7 same point load	Continuous	Historical electric load (kW)

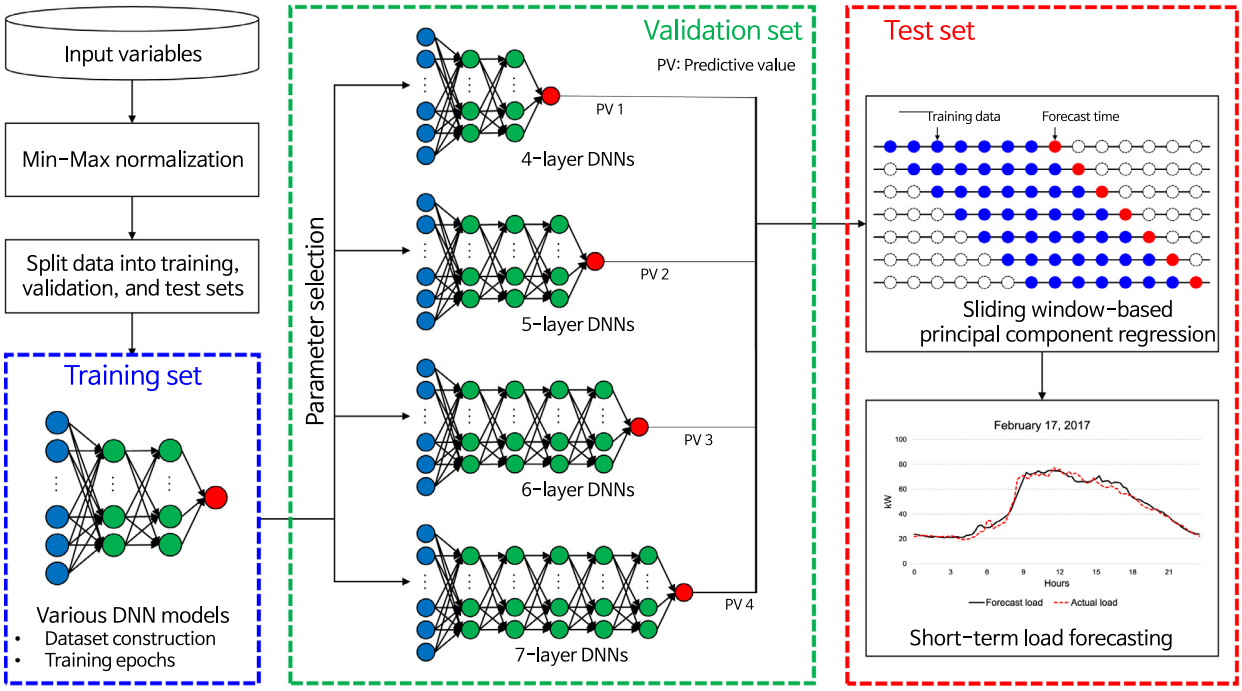


Fig. 4. Architecture of the COSMOS model.

of the electric load [9,37]. However, if too many variables are used as input for the historical electric load data, overfitting likely occurs during the model training [28]. Hence, we selected variables from the historical electric load by considering the Pearson correlation coefficient between the previous D-day same point load and current electric load and used them as input variables. Table 4 lists the Pearson correlation coefficients depending on the previous D-day. In this study, Previous D-1 and D-7 same point loads, which represent the electric load of the previous day and a week before, respectively, highly correlate with the current electric energy consumption.

We considered the electric energy consumption data of the previous day (D-1) and one-week before (D-7) at the same point. The same point load from one-week prior reflects the recent electric energy consumption pattern for the same day of the week and the same point load from the previous day indicates the most recent electric energy consumption trends. Overall, we considered a total of 21 input variables to build our proposed forecasting model, as shown in Table 5.

4. Proposed method

In this section, we describe our forecasting model, which is based on DNN and sliding window-based PCR. Fig. 4 illustrates the architecture of our model. Before training the DNN models, we performed a Min–Max normalization. Subsequently, we split the data into training, validation, and test datasets. The data from February 1, 2015, to April 30, 2016, were used as the training set; those from May 1, 2015, to September 30, 2016, were used as the validation set; and those from October 1, 2016, to February 28, 2017, were used as the test set.

4.1. Stage 1: DNN-based forecasting model

An ANN, usually called multilayer perceptron, is a computational model inspired by the structure and functional aspects of biological neural networks [6,27,38]. An ANN consists of three layers: input layer, hidden layer, and output layer. Each layer is composed of one or more nodes. The hidden layer has many parameters that affect the model performance such as the sizes of the hidden layer and node, activation function of the hidden layer, and number of epochs [6]. Therefore, the network performance depends on the network configuration. The size of the hidden layer determines how “deep” or “shallow” the network is. When an ANN has two or more hidden layers, it is called DNN [27,30]. Although an increasing number of hidden layers improves the network performance, it likely results in an overfitting problem [15,39]. However, it is challenging to determine the suitable number of hidden layers to overcome the overfitting problem because the deep learning building process can derive different values depending on the training environment or conditions and is expensive [40]. To overcome this problem, we developed a more accurate forecasting model by effectively combining several DNN models, regardless of the hidden layer sizes.

To build combined forecasting models, we constructed four DNN models by including 4-layer, 5-layer, 6-layer, and 7-layer DNNs, respectively, in the training set. We used 15 nodes because the number of hidden neurons must be 2/3 of the size of the input layer plus the size of the output layer [15,41]. In addition, we used ReLU, which is the most widely used activation function, as an activation function in the hidden layer [27]. For the other hyperparameter settings, we used the adaptive moment estimation (Adam) as optimizer algorithm [42] and the mean square error as loss function. In addition, we set the bath size and learning rate to 96 and 0.001, respectively. The number of epochs is related to the number of optimization rounds that are applied during the training. A larger number of optimization rounds reduces the error in the training data. However, the network becomes more biased to the weights of the training data (overfitting) and exhibits a low performance in terms of the generalization of the test (unseen) data [43]. Thus, we determined the optimal number of epochs by considering several cases based on the number of epochs and applied them to the DNNs.

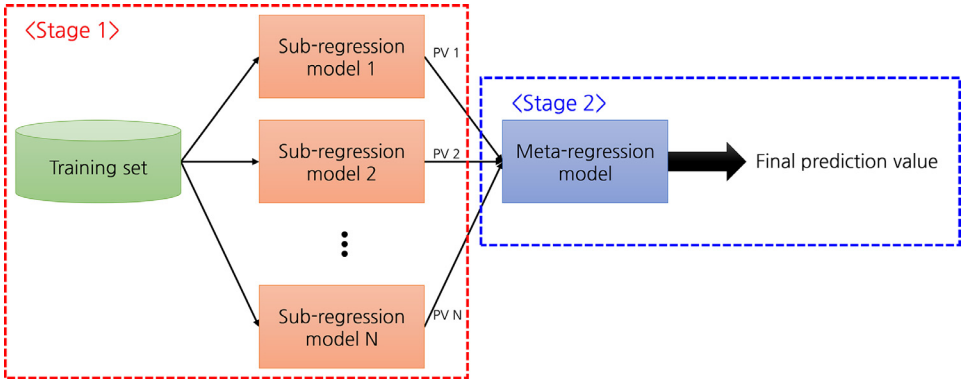


Fig. 5. Conceptual diagram of the stacking ensemble model; PV is the prediction value (output) of the sub-regression model.

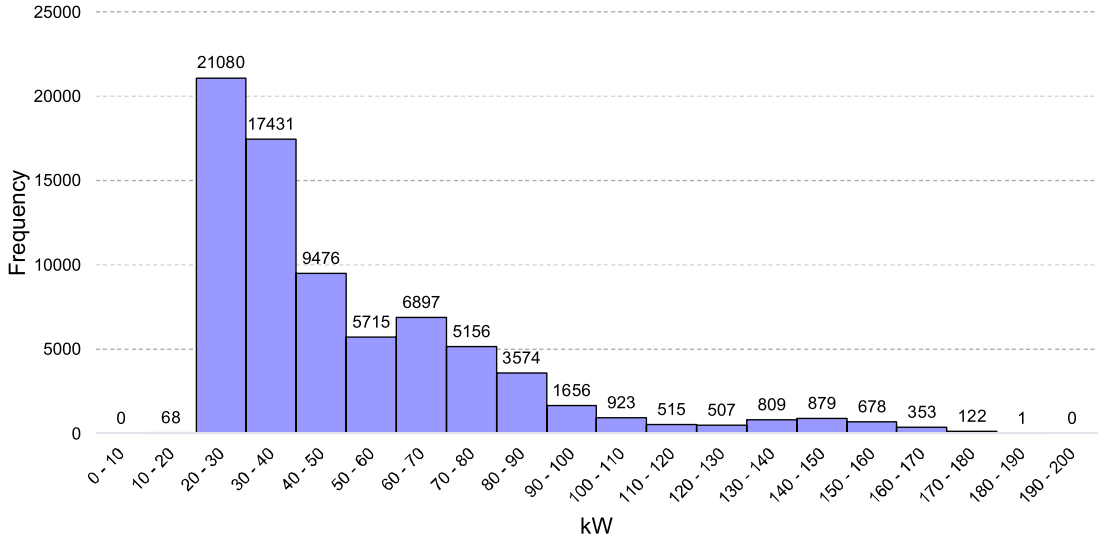


Fig. 6. Histogram of the distribution of the building electric energy consumption.

Table 6
Statistical analysis of the building electric energy consumption.

Statistics	Value
Mean	50.39
Standard error	0.11
Median	39.17
Mode	25.54
Standard deviation	29.46
Sample variance	868.17
Kurtosis	3.13
Skewness	1.75
Range	163.25
Minimum	17.04
Maximum	180.29
Sum	3671662.93
Count	72864

Table 7
Four different input variable configurations.

Case #	Input variables (count)
1	Weather factors + calendar information (19)
2	Case 1 + previous D-1 same point load (20)
3	Case 1 + previous D-7 same point load (20)
4	Case 1 + previous D-1 and D-7 same point loads (21)

4.2. Stage 2: PCR-based stacking ensemble model

The stacking ensemble approach is trained using the combined prediction values of the multiple classification or regression models via a meta-classification or meta-regression model [44,45]. In the case of the forecasting model construction, the base level or sub-regression models are trained based on a complete training set. Subsequently, a meta-regression model is trained with the outputs of the sub-regression model as input variables. The conceptual diagram of a stacking ensemble model is depicted in Fig. 5. The stacking ensemble model typically yields a better performance than any of the trained models [45–48].

For instance, Divina et al. [46] proposed a stacking ensemble learning scheme to predict short-term electric energy consumption by using a dataset regarding the electric energy consumption in Spain registered for more than nine years. They used three sub-regression models, namely, evolutionary algorithms for regression trees, RF, and ANNs. Then, they constructed a meta-regression model using GBM. They compared the prediction results obtained by the ensemble model with the prediction results obtained by the single models. Predictions obtained by their ensemble model were superior to the result of the other models. Qie et al. [47] presented a staking ensemble learning-based time series forecasting model by applying electric energy consumption datasets from Australian Energy Market Operators. They constructed several sub-regression models based on DBNs trained using a different number of epochs. Then, they built an SVR model using the prediction values of the sub-regression models to get more accurate

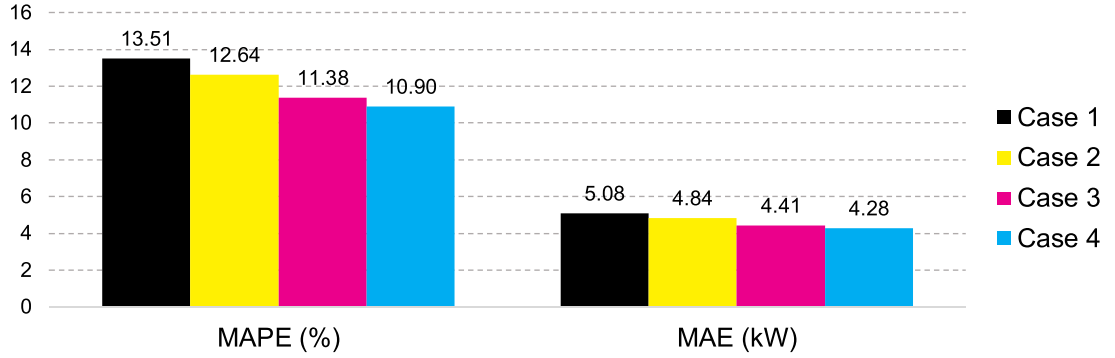


Fig. 7. Comparison of the prediction performance of four input variable configurations.

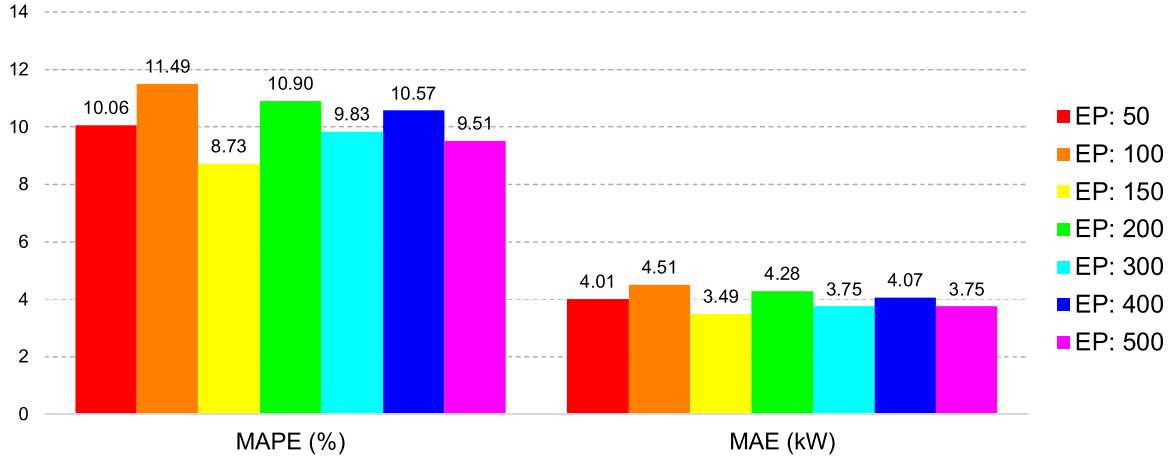


Fig. 8. Comparison of the prediction performance of different training epochs (EP: number of training epochs).

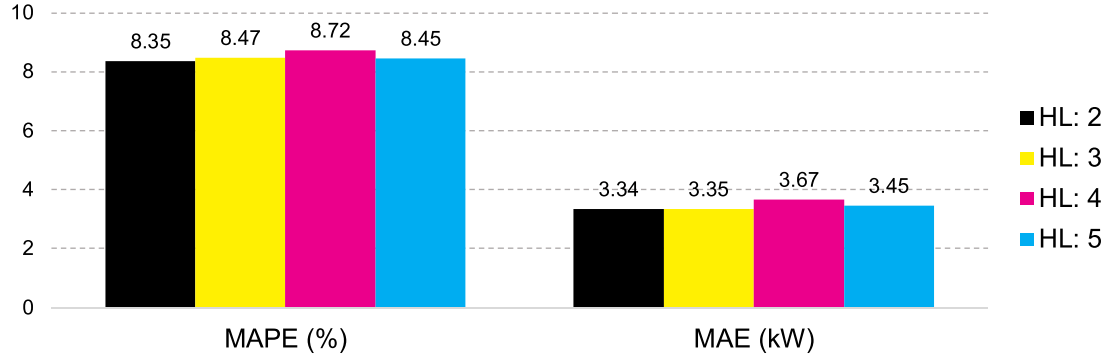


Fig. 9. Comparison of prediction performance between different number of hidden layers (HL: number of hidden layers).

forecasting results. Their prediction model demonstrated a much stronger ability on real complicated regression problems. Ribeiro et al. [48] proposed a stacking ensemble learning method to predict hourly electric load by using two real datasets from Italy and the Global Energy Forecasting Competition. They built several sub-regression models based on wavelet neural networks and then combined through simple mean, median, mode, and stacked generalization approaches. Their forecasting model compared its prediction performance with prediction performances of MLP, regression tree, mean, naïve model, and so on and showed better prediction performance than other methods.

To construct a stacking ensemble model, we combined the forecasting results of the four DNN-based models using PCR. The PCR is a regression technique based on principal component analysis (PCA) [49]. The main idea of PCR is to calculate the principal components and then use some of these components as input variables for a linear regression fitted using the typical least squares procedure. A small number of principal components is sufficient to explain the vast majority of the data variability. The PCR has several advantages over MLR; for example, it handles missing values or strongly correlated input variable columns [50]. The PCR has desirable characteristics such as dimensionality reduction, avoidance of multicollinearity between the input variables, and overfitting mitigation [46,51]. We defined X as the set of all variables needed to construct the PCR model in the test set, as defined in Eq. (10), where DNN_{4L}, DNN_{5L}, DNN_{6L}, and DNN_{7L} represent the forecast values of the 4-layer, 5-layer, 6-layer, and 7-layer DNNs, respectively. In addition, during the construction of the PCR model, we used a sliding window method to adequately reflect the recent electric energy consumption trends. Because we focused on day-ahead electric load forecasting, the end point of the sliding window is the

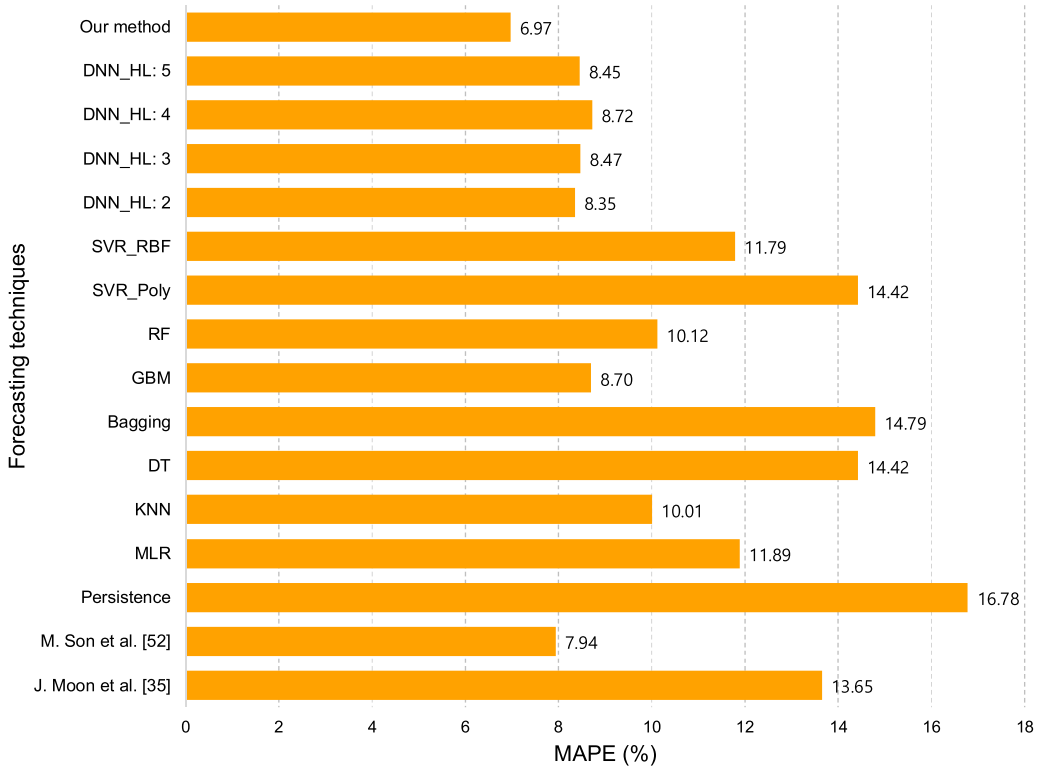


Fig. 10. Comparison of the MAPEs of the forecasting techniques.

Table 8
Comparison of the MAPEs of the PCR and MLR models.

Window size	PC 1	PC 2	PC 3	MLR	Average
1 day	7.65	7.95	8.28	8.64	8.13
2 days	7.38	7.46	7.56	7.74	7.54
3 days	7.32	7.42	7.62	7.70	7.52
4 days	7.11	7.28	7.49	7.56	7.36
5 days	7.01	7.13	7.32	7.39	7.21
6 days	6.99	7.07	7.18	7.14	7.09
7 days	6.97	7.07	7.14	7.14	7.08
8 days	6.96	7.10	7.11	7.14	7.08
9 days	6.96	7.08	7.10	7.17	7.08
10 days	6.95	7.06	7.09	7.16	7.06

Table 9
Comparison of the MAEs of the PCR and MLR models.

Window size	PC 1	PC 2	PC 3	MLR	Average
1 day	3.31	3.43	3.60	3.76	3.52
2 days	3.09	3.12	3.17	3.23	3.15
3 days	3.07	3.11	3.22	3.25	3.16
4 days	2.95	3.03	3.14	3.14	3.07
5 days	2.88	2.94	3.03	3.04	2.97
6 days	2.85	2.91	2.97	2.94	2.92
7 days	2.84	2.89	2.93	2.93	2.90
8 days	2.83	2.90	2.91	2.92	2.89
9 days	2.84	2.89	2.90	2.92	2.89
10 days	2.84	2.88	2.90	2.92	2.89

same as that of the sliding window one-day before the forecast.

$$X = \{DNN_{4L},\ DNN_{5L},\ DNN_{6L},\ DNN_{7L}\} \tag{10}$$

To construct our forecasting model, we used PCR according to the following procedure:

1 Collect X and the actual electric energy consumption data required for the PCR model.

Table 10
Hyperparameters selected for each machine learning algorithm.

ML algorithms	Selected hyperparameters
KNN	Number of neighbors: 5
DT	Maximum depth of the tree: 30
Bagging	Number of tree models in the ensemble: 25
GBM	Loss function: Huber loss
	Learning rate: 0.01
	Number of trees: 3000
	Subsample: 0.5
	Minimum number of samples for each node: 10
	Maximum depth of the tree: 5
	Maximum number of features: 21 (number of features)
RF	Number of trees: 128
	Maximum number of features: 21 (number of features)
SVR	Gamma: 0.125
	Cost: 1
	Epsilon: 0.1

2 Construct various PCR models considering several p-dimensions and sliding window sizes.
3 Compare and analyze the prediction performances of various PCR models using the test set.
4 Determine the optimal p-dimension and window size for the PCR model that achieves the best prediction performance. The resulting PCR model is the forecasting model called COSMOS.

5. Results and discussion

In this section, we investigate the prediction performance of the COSMOS model and compare it with that of other forecasting techniques and models [35,52]. We first describe the performance evaluation metrics that were used to determine the prediction performance of our model. Subsequently, we determine the optimal DNN model by considering the variable selection and optimal number of epochs and using the validation set. We then construct several

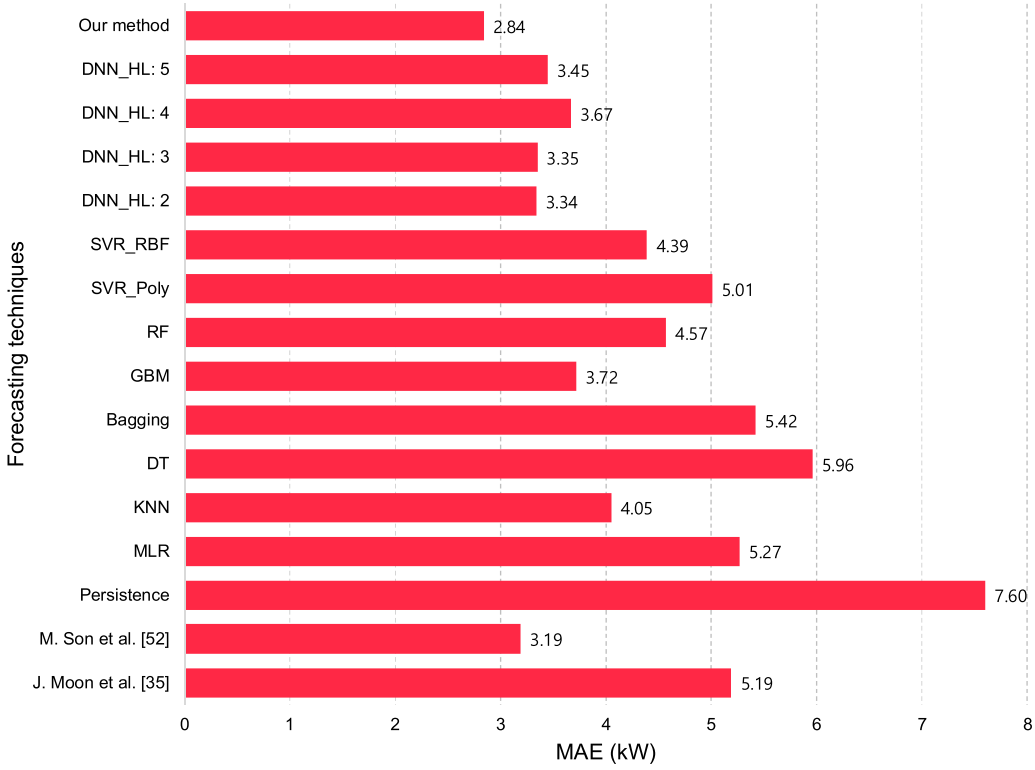


Fig. 11. Comparison of the MAEs of the forecasting techniques.

Table 11
Comparison of the MAPEs and MAEs (bolded values represent the best values).

Forecasting techniques	MAPE (%)	MAE (kW)
Shallow neural network [35]	13.65	5.19
Auto-encoder + random forest [52]	7.94	3.19
Persistence	16.78	7.60
Multiple linear regression	11.89	5.27
K-nearest neighbor	10.01	4.05
Decision tree	14.42	5.96
Bagging	14.79	5.42
Gradient boosting machine	8.70	3.72
Random forest	10.12	4.57
Support vector regression (kernel: Poly)	14.42	5.01
Support vector regression (kernel: RBF)	11.79	4.39
DNN (number of hidden layers: 2)	8.35	3.34
DNN (number of hidden layers: 3)	8.47	3.35
DNN (number of hidden layers: 4)	8.72	3.67
DNN (number of hidden layers: 5)	8.45	3.45
COSMOS	**6.97**	**2.84**

PCR-based forecasting models and determine the model that performs best. Lastly, we perform an in-depth analysis of our forecasting model and compare it with other ML techniques and models.

5.1. Performance evaluation metric

To evaluate the performance of the proposed model, we used two metrics, that is, the mean absolute percentage error (MAPE) and mean absolute error (MAE), which can be defined by using Eqs. (11) and (12), respectively. The parameter n is the number of observations and A_t and F_t are the actual and forecast values, respectively.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \times 100 \tag{11}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t| \tag{12}$$

5.2. Experimental design

To evaluate the performance of our forecasting model, we conducted several experiments. The experiments were conducted with an Intel® Core™ i7-8700k CPU with 32GB DDR4 RAM. We preprocessed the dataset in a Python environment and carried out forecasting modeling using scikit-learn [53]. Table 6 and Fig. 6 show the statistical analysis and histogram of the distribution of the building electric energy consumption, respectively.

To determine the input variable configuration, we first constructed a 4-layer DNN model with 200 training epochs by considering four different input variable configurations. We selected a 4-layer DNN because the minimum requirement of a DNN is two hidden layers. We performed our experiments by using 200 training epochs for the validation set because this is the default value for a multilayer perceptron regression in the scikit-learn environment [53].

To investigate the effect of the historical electric load on the prediction performance, we considered four different input configurations, as shown in Table 7. We then determined the best input variable configuration for constructing our model. The input variables we focused on include weather data, calendar data, D-1 (one day ago) historical load data, and D-7 (one week ago) historical load data. We used 14 nodes in the cases 1–3 and 15 nodes in case 4, and the remaining hyperparameters were the same as mentioned above.

Fig. 7 shows the MAPE and MAE results for the four input variable configurations. Because case 4 shows the best prediction performance in terms of the MAPE and MAE, we used the corresponding input variables. In addition, to determine the best number of training epochs, we constructed a 4-layer DNN model with the

Table 12
Comparison of the MAPEs by the days of the week (bolded values represent the best values).

Forecasting techniques	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Shallow neural network [35]	10.30	9.93	10.52	9.12	12.83	27.30	14.99
Auto-encoder + random forest [52]	9.18	6.37	6.47	6.55	8.40	**9.51**	8.99
Persistence	30.36	7.56	5.43	**5.82**	9.99	39.03	17.76
Multiple linear regression	18.90	7.09	6.80	7.88	11.45	20.65	24.43
K-nearest neighbor	13.16	7.64	7.10	7.28	12.63	12.69	9.47
Decision tree	16.57	13.47	13.79	13.54	16.70	18.72	18.29
Bagging	15.41	12.23	12.35	12.56	15.29	17.54	17.94
Gradient boosting machine	9.32	6.97	7.42	7.49	9.78	13.06	12.12
Random forest	7.69	6.63	6.80	6.90	9.02	28.14	21.65
Support vector regression (kernel: Poly)	8.31	8.26	7.18	8.53	8.58	26.39	32.86
Support vector regression (kernel: RBF)	8.01	8.39	8.73	8.27	8.78	19.01	20.94
DNN (number of hidden layers: 2)	8.38	6.93	6.86	7.11	8.22	11.03	9.91
DNN (number of hidden layers: 3)	7.76	7.36	7.79	6.71	8.22	11.43	10.12
DNN (number of hidden layers: 4)	9.05	5.90	6.20	6.72	8.69	12.66	11.55
DNN (number of hidden layers: 5)	8.58	6.10	6.85	6.40	8.09	14.23	8.51
COSMOS	**6.69**	**5.84**	**5.29**	5.90	**6.91**	9.83	**8.18**

Table 13
Comparison of the MAEs by the days of the week (bolded values represent the best values).

Forecasting techniques	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Shallow neural network [35]	4.23	4.36	4.49	3.91	5.24	9.71	4.30
Auto-encoder + random forest [52]	4.64	3.04	2.92	2.92	3.29	**2.93**	2.56
Persistence	18.88	4.25	2.67	2.81	3.83	14.68	5.47
Multiple linear regression	7.11	3.46	3.32	3.73	4.86	7.12	7.08
K-nearest neighbor	5.79	3.70	3.43	3.43	5.09	4.20	2.72
Decision tree	6.80	5.71	5.82	5.64	7.06	5.76	4.96
Bagging	6.25	4.99	5.03	5.10	6.37	5.31	4.90
Gradient boosting machine	4.09	3.15	3.31	3.33	4.05	4.44	3.63
Random forest	3.24	2.95	2.90	2.98	3.52	9.70	6.48
Support vector regression (kernel: Poly)	3.46	3.66	3.12	3.60	3.39	8.25	9.35
Support vector regression (kernel: RBF)	3.48	3.66	3.52	3.45	3.57	6.59	6.33
DNN (number of hidden layers: 2)	3.79	3.18	3.02	3.21	3.59	3.53	2.94
DNN (number of hidden layers: 3)	3.43	3.26	3.40	2.96	3.50	3.70	3.12
DNN (number of hidden layers: 4)	4.21	2.90	2.94	3.23	3.99	4.53	3.54
DNN (number of hidden layers: 5)	3.73	2.91	3.05	2.96	3.34	5.19	2.53
COSMOS	**3.18**	**2.79**	**2.49**	**2.75**	**2.97**	3.25	**2.44**

Table 14
Comparison of the MAPEs by the month (bolded values represent the best values).

Forecasting techniques	Oct 2016	Nov 2016	Dec 2016	Jan 2017	Feb 2017
Shallow neural network [35]	7.14	11.04	14.73	18.34	17.10
Auto-encoder + random forest [52]	6.33	7.61	6.99	10.30	8.50
Persistence	16.11	15.58	16.42	18.11	17.58
Multiple linear regression	12.51	12.67	13.04	17.43	14.27
K-nearest neighbor	9.32	7.65	7.93	13.56	11.69
Decision tree	15.41	14.09	14.48	17.35	18.31
Bagging	14.31	12.77	13.32	16.08	17.66
Gradient boosting machine	8.51	8.64	8.33	10.71	11.33
Random forest	10.33	12.91	11.92	13.19	14.34
Support vector regression (kernel: Poly)	7.96	11.75	11.50	20.27	21.20
Support vector regression (kernel: RBF)	7.71	12.02	11.46	14.06	13.94
DNN (number of hidden layers: 2)	5.87	6.41	7.78	11.73	10.15
DNN (number of hidden layers: 3)	6.10	7.85	8.75	10.67	9.17
DNN (number of hidden layers: 4)	5.64	8.08	9.39	11.03	9.45
DNN (number of hidden layers: 5)	5.68	6.65	9.03	11.50	9.26
COSMOS	**4.88**	**5.57**	**6.38**	**9.19**	**8.44**

input variables of case 4 and predicted the building electric energy consumption using the validation set. Fig. 8 shows the results. Based on the results, we set the training epochs to 150.

To construct our model, we first built four DNN-based forecasting models with two, three, four, and five hidden layers, respectively. We then predicted the building electric energy consumption of the test set with these forecasting models. The prediction performances in terms of the MAPE and MAE are shown in Fig. 9.

Fig. 9 shows that it is difficult to confirm clear differences in the prediction performances of each forecasting model compared with Figs. 7 and 8. To effectively combine these models, we first built diverse building electric energy consumption forecasting models using sliding window-based PCR for the best configuration of the COSMOS model. We set the window size based on the date, that is, between one and ten days, to select the optimal sliding window size. Because we focused on day-ahead load forecasting, we set the previous one-day same point as the end point of the win-

Table 15
Comparison of the MAEs by the month (bolded values represent the best values).

Forecasting techniques	Oct 2016	Nov 2016	Dec 2016	Jan 2017	Feb 2017
Shallow neural network [35]	2.86	4.21	5.72	6.84	6.39
Auto-encoder + random forest [52]	2.55	3.01	2.76	4.14	**3.52**
Persistence	7.25	6.72	7.65	8.28	8.13
Multiple linear regression	4.90	4.54	4.91	6.17	5.85
K-nearest neighbor	3.79	3.02	3.32	5.05	5.15
Decision tree	5.94	5.12	5.27	6.29	7.29
Bagging	5.39	4.48	4.75	5.65	6.95
Gradient boosting machine	3.44	3.25	3.31	4.02	4.65
Random forest	3.76	4.45	4.43	4.79	5.47
Support vector regression (kernel: Poly)	3.07	3.91	4.17	6.97	7.08
Support vector regression (kernel: RBF)	2.98	4.19	4.40	5.21	5.23
DNN (number of hidden layers: 2)	2.40	2.52	3.29	4.42	4.03
DNN (number of hidden layers: 3)	2.51	2.86	3.46	4.14	3.75
DNN (number of hidden layers: 4)	2.35	3.23	4.14	4.45	3.97
DNN (number of hidden layers: 5)	2.38	2.58	3.77	4.43	3.82
COSMOS	**1.91**	**2.23**	**2.69**	**3.65**	3.56

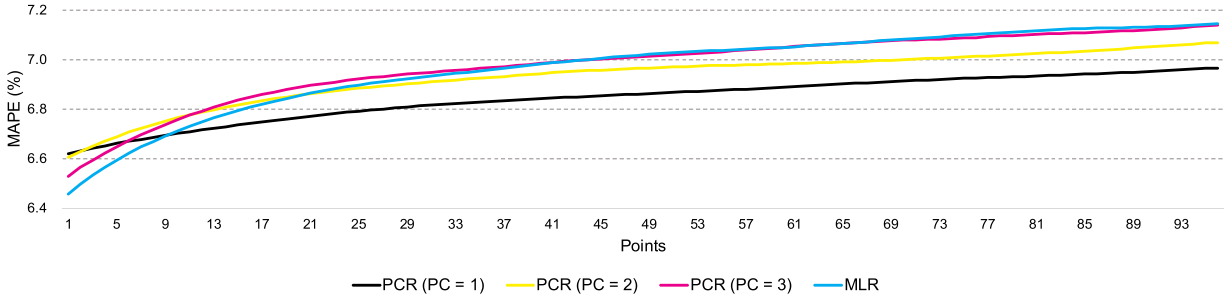


Fig. 12. MAPE results for the stacking ensemble models.

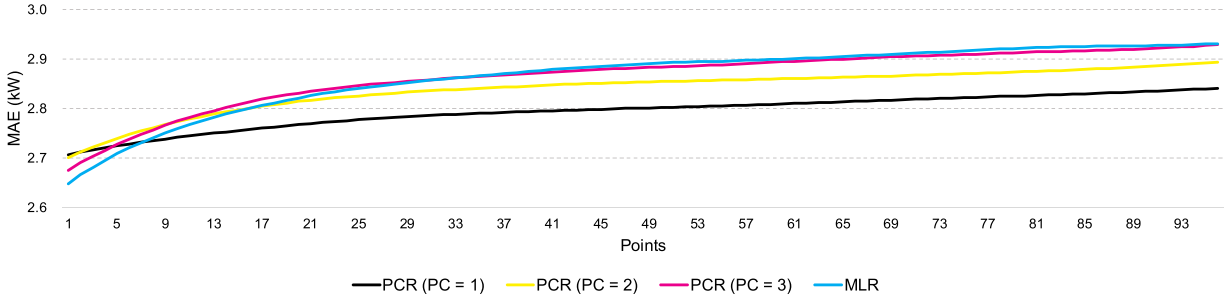


Fig. 13. MAE results for the stacking ensemble models.

dow size. For instance, if the forecasting time is 5:00 am on February 17, 2017, the one-day window size is considered to be 5:15 am on February 15, 2017, to 5:00 am on February 16, 2017. We then transformed the predictions of the four DNN-based forecasting models into one component (PC 1), two components (PC 2), and three components (PC 3) using PCA and built three different PCR-based stacking ensemble models with these components. We predicted the building electric energy consumption of the test set using these PCR-based forecasting models and one MLR model. Herein, MLR is a popular modeling technique for analyzing the relationship between a continuous dependent variable and two or more independent variables [54–57]. Its main goal is to identify a function that describes, as strictly as possible, the relationship between these variables so that the values of the dependent variables can be predicted using a range of independent variable values. MLR also allows determining the overall fit of the model and the relative contribution of each of the predictors to the total variance explained. To effectively construct a meta-regression model, MLR was used for energy consumption forecasting within the long-term [54] and short-term [55] horizons. The experimental results are presented in Tables 8 and 9.

All PCR-based forecasting models have a better prediction performance than the four DNN-based forecasting models. In addition, increasing the sliding window size leads to a better prediction accuracy. We observed that there is almost no difference in the prediction accuracy after 7 days. This indicates that it is enough to consider weekly electric energy consumption patterns for the prediction. In addition, the PCR-based model with PC 1 exhibits a better prediction performance than the other PCR- and MLR-based models. Therefore, we used PC 1 and a sliding window size of seven days for the configuration of the COSMOS model.

5.3. Comparison of the forecasting techniques

To verify the validity and applicability of the COSMOS model, we compared its prediction performance with that of various day-ahead electric load forecasting models. We built diverse build-

Table 16
Comparison of the prediction performances of multiple points-ahead load forecasting.

Points	MAPE (%)				MAE (kW)			
	PCR (PC = 1)	PCR (PC = 2)	PCR (PC = 3)	MLR	PCR (PC = 1)	PCR (PC = 2)	PCR (PC = 3)	MLR
1	6.62	6.60	6.53	6.46	2.71	2.70	2.68	2.65
2	6.63	6.63	6.56	6.50	2.71	2.71	2.69	2.67
3	6.64	6.65	6.59	6.53	2.72	2.72	2.70	2.68
4	6.65	6.67	6.62	6.56	2.72	2.73	2.72	2.70
5	6.66	6.69	6.65	6.59	2.73	2.74	2.73	2.71
6	6.67	6.71	6.67	6.62	2.73	2.75	2.74	2.72
7	6.68	6.72	6.70	6.65	2.73	2.75	2.75	2.73
8	6.69	6.74	6.72	6.67	2.74	2.76	2.76	2.74
9	6.69	6.75	6.74	6.69	2.74	2.77	2.77	2.75
10	6.70	6.76	6.76	6.71	2.74	2.77	2.77	2.76
11	6.71	6.78	6.78	6.73	2.74	2.78	2.78	2.77
12	6.72	6.79	6.79	6.75	2.75	2.78	2.79	2.78
13	6.72	6.80	6.81	6.76	2.75	2.79	2.80	2.78
14	6.73	6.81	6.82	6.78	2.75	2.79	2.80	2.79
15	6.74	6.82	6.84	6.79	2.76	2.80	2.81	2.80
16	6.74	6.83	6.85	6.81	2.76	2.80	2.81	2.80
17	6.75	6.83	6.86	6.82	2.76	2.81	2.82	2.81
18	6.75	6.84	6.87	6.83	2.76	2.81	2.82	2.81
19	6.76	6.85	6.88	6.84	2.77	2.81	2.83	2.82
20	6.77	6.86	6.89	6.85	2.77	2.81	2.83	2.82
21	6.77	6.86	6.90	6.86	2.77	2.82	2.84	2.83
22	6.78	6.87	6.90	6.87	2.77	2.82	2.84	2.83
23	6.78	6.87	6.91	6.88	2.77	2.82	2.84	2.83
24	6.79	6.88	6.92	6.89	2.78	2.82	2.84	2.84
25	6.79	6.88	6.92	6.90	2.78	2.83	2.85	2.84
26	6.80	6.89	6.93	6.90	2.78	2.83	2.85	2.84
27	6.80	6.89	6.93	6.91	2.78	2.83	2.85	2.85
28	6.80	6.90	6.94	6.92	2.78	2.83	2.85	2.85
29	6.81	6.90	6.94	6.92	2.78	2.83	2.86	2.85
30	6.81	6.91	6.95	6.93	2.79	2.83	2.86	2.86
31	6.82	6.91	6.95	6.93	2.79	2.84	2.86	2.86
32	6.82	6.91	6.95	6.94	2.79	2.84	2.86	2.86
33	6.82	6.92	6.96	6.94	2.79	2.84	2.86	2.86
34	6.83	6.92	6.96	6.95	2.79	2.84	2.86	2.86
35	6.83	6.93	6.96	6.95	2.79	2.84	2.87	2.87
36	6.83	6.93	6.97	6.96	2.79	2.84	2.87	2.87
37	6.83	6.93	6.97	6.96	2.79	2.84	2.87	2.87
38	6.84	6.94	6.98	6.97	2.79	2.84	2.87	2.87
39	6.84	6.94	6.98	6.98	2.79	2.85	2.87	2.87
40	6.84	6.94	6.98	6.98	2.80	2.85	2.87	2.88
41	6.84	6.95	6.99	6.99	2.80	2.85	2.87	2.88
42	6.85	6.95	6.99	6.99	2.80	2.85	2.88	2.88
43	6.85	6.95	7.00	7.00	2.80	2.85	2.88	2.88
44	6.85	6.96	7.00	7.00	2.80	2.85	2.88	2.88
45	6.85	6.96	7.00	7.01	2.80	2.85	2.88	2.89
46	6.86	6.96	7.01	7.01	2.80	2.85	2.88	2.89
47	6.86	6.96	7.01	7.01	2.80	2.85	2.88	2.89
48	6.86	6.96	7.01	7.02	2.80	2.85	2.88	2.89
49	6.86	6.97	7.01	7.02	2.80	2.85	2.88	2.89
50	6.87	6.97	7.02	7.03	2.80	2.85	2.88	2.89
51	6.87	6.97	7.02	7.03	2.80	2.86	2.88	2.89
52	6.87	6.97	7.02	7.03	2.80	2.86	2.89	2.89
53	6.87	6.97	7.02	7.03	2.80	2.86	2.89	2.89
54	6.87	6.98	7.03	7.04	2.81	2.86	2.89	2.90
55	6.88	6.98	7.03	7.04	2.81	2.86	2.89	2.90
56	6.88	6.98	7.04	7.04	2.81	2.86	2.89	2.90
57	6.88	6.98	7.04	7.04	2.81	2.86	2.89	2.90
58	6.88	6.98	7.04	7.04	2.81	2.86	2.89	2.90
59	6.89	6.98	7.05	7.05	2.81	2.86	2.89	2.90
60	6.89	6.98	7.05	7.05	2.81	2.86	2.89	2.90
61	6.89	6.98	7.05	7.05	2.81	2.86	2.90	2.90
62	6.89	6.99	7.06	7.06	2.81	2.86	2.90	2.90
63	6.90	6.99	7.06	7.06	2.81	2.86	2.90	2.90
64	6.90	6.99	7.06	7.06	2.81	2.86	2.90	2.90
65	6.90	6.99	7.07	7.07	2.81	2.86	2.90	2.91
66	6.90	6.99	7.07	7.07	2.81	2.86	2.90	2.91
67	6.91	6.99	7.07	7.07	2.82	2.86	2.90	2.91
68	6.91	7.00	7.07	7.08	2.82	2.87	2.90	2.91
69	6.91	7.00	7.08	7.08	2.82	2.87	2.90	2.91
70	6.91	7.00	7.08	7.08	2.82	2.87	2.91	2.91
71	6.92	7.00	7.08	7.09	2.82	2.87	2.91	2.91
72	6.92	7.00	7.08	7.09	2.82	2.87	2.91	2.91
73	6.92	7.01	7.08	7.09	2.82	2.87	2.91	2.91

(*continued on next page*)

Table 16 (*continued*)

Points	MAPE (%)				MAE (kW)			
	PCR (PC = 1)	PCR (PC = 2)	PCR (PC = 3)	MLR	PCR (PC = 1)	PCR (PC = 2)	PCR (PC = 3)	MLR
74	6.92	7.01	7.08	7.10	2.82	2.87	2.91	2.92
75	6.92	7.01	7.09	7.10	2.82	2.87	2.91	2.92
76	6.93	7.01	7.09	7.10	2.82	2.87	2.91	2.92
77	6.93	7.01	7.09	7.11	2.82	2.87	2.91	2.92
78	6.93	7.02	7.10	7.11	2.82	2.87	2.91	2.92
79	6.93	7.02	7.10	7.11	2.83	2.87	2.91	2.92
80	6.93	7.02	7.10	7.11	2.83	2.87	2.91	2.92
81	6.93	7.02	7.10	7.12	2.83	2.88	2.91	2.92
82	6.94	7.03	7.10	7.12	2.83	2.88	2.92	2.92
83	6.94	7.03	7.11	7.12	2.83	2.88	2.92	2.92
84	6.94	7.03	7.11	7.12	2.83	2.88	2.92	2.93
85	6.94	7.03	7.11	7.13	2.83	2.88	2.92	2.93
86	6.94	7.04	7.11	7.13	2.83	2.88	2.92	2.93
87	6.95	7.04	7.11	7.13	2.83	2.88	2.92	2.93
88	6.95	7.04	7.12	7.13	2.83	2.88	2.92	2.93
89	6.95	7.05	7.12	7.13	2.83	2.88	2.92	2.93
90	6.95	7.05	7.12	7.13	2.83	2.89	2.92	2.93
91	6.95	7.05	7.12	7.13	2.84	2.89	2.92	2.93
92	6.96	7.06	7.13	7.13	2.84	2.89	2.92	2.93
93	6.96	7.06	7.13	7.14	2.84	2.89	2.92	2.93
94	6.96	7.06	7.13	7.14	2.84	2.89	2.93	2.93
95	6.96	7.07	7.14	7.14	2.84	2.89	2.93	2.93
96	6.97	7.07	7.14	7.14	2.84	2.89	2.93	2.93
Avg.	6.84	6.93	6.98	6.97	2.79	2.84	2.87	2.87

ing electric energy consumption forecasting models based on the statistical approach and ML algorithms. The persistence and MLR were considered in the statistical approach. The persistence model assumes that the conditions of the predicted value (96-step ahead) are the same as those of the current value and exhibits a good accuracy due to the highly periodic nature of electric energy consumption. We used 21 input variables of the DNN models as input variables for the MLR model construction. In addition, we used several ML techniques such as KNN, decision tree (DT), bagging, GBM, RF, and SVR. The ML algorithms typically contain hyperparameters, which significantly affect the model performance. To obtain the optimal hyperparameter configuration, we determined the hyperparameters of each ML algorithm using grid search. The random states of all ML techniques were set to 42. The selected hyperparameters are listed in Table 10.

In addition, we compared the COSMOS model with the statistical approaches and ML algorithms as well as our previous forecasting models [35,52]. Previously, we proposed a shallow neural network-based STLF model to predict the electric energy consumption for university campuses [35]. This forecasting model was configured using only the weather factors and time information as input variables, similar to case 1. In another study, we proposed a 15-min interval electric load forecasting model based on an auto-encoder and RF [52]. First, we used the auto-encoder method for the traditional features, such as weather factors and time information, except for the previous electric loads. Based on the resulting features and previous electric loads, we constructed an electric load forecasting model using RF. The previous electric loads were configured at the same point in time from seven days to the day before the forecast. For the construction of a similar experimental environment, we built an SNN-based forecasting model using case 1 and used the same SNN hyperparameters as those selected for the DNN model described above. We also developed an RF-based STLF model by setting the input variables using the previous electric loads and auto-encoder-based feature extraction values of case 1. This model was implemented based on the hyperparameters of the auto-encoder and RF suggested in a previous study [52]. Figs. 10 and 11 and Table 11 present the prediction performance of diverse forecasting techniques. The results show that our proposed method exhibits the best accuracy compared with the other forecasting techniques. Unlike our previous model [35], the proposed model can reflect recent trends in electric energy consumption by using the historical electric load as input variables. Similarly, unlike our previous RF-based model [52] that considered previous electric loads (i.e., the same point in time from seven days to the day before the forecast) as input variables, the proposed model used a stacking ensemble approach to mitigate its overfitting problem.

To determine the prediction performance in detail, we analyzed the MAPE and MAE results for the days of the week and months (Tables 12–15). The COSMOS model exhibits the best prediction performance in most cases. This is because it can effectively reflect recent electric load patterns and adjust the weights of each DNN model by using the sliding window method. All models, including the COSMOS model, predict the building electric energy consumption with a better accuracy on weekdays (i.e., Mondays to Fridays) than on weekends (i.e., Saturdays and Sundays). We confirmed that it is difficult to forecast the building electric energy consumption on weekends because the electric energy consumption is close to zero. Furthermore, it was more challenging to predict the building electric energy consumptions in January than in other months because this month includes several Korean public holidays such as the Lunar New Year. Because the building electric energy consumption on Lunar New Year is low, it was difficult to accurately predict the electric energy consumption. In the future, a forecasting model should be developed that can perform more accurate predictions on weekends.

5.4. Multiple short-term load forecasting using stacking ensemble models

Multiple points-ahead electric load forecasting is essential for the preparation of real-time DSM. Although our forecasting model focuses on day-ahead (96 points ahead) building electric energy consumption forecasting, we applied the COSMOS model to predict multiple points-ahead electric loads. We predicted various points (from 1 to 95)-ahead building electric energy consumptions and compared the prediction performance with that of the other stacking ensemble models such as the PC 2- and PC 3-based PCR models and MLR.

Table 17
P-values of each normality test.

Normality tests	*P*-value
Anderson-Darling normality test	<2.2e−16
Cramer-von Mises normality test	<7.37e−10
Lilliefors (Kolmogorov-Smirnov) normality test	<2.2e−16
Pearson chi-square normality test	<2.2e−16

Figs. 12 and 13 and Table 16 show that MLR exhibits the best prediction performance up to 2 h. However, the PCR (PC = 1) as the COSMOS model exhibits the best prediction performance from 2 h to 24 h-ahead electric load forecasting. In addition, the COSMOS model predicts average values better than the other stacking ensemble models and has a higher prediction accuracy. MLR can present satisfactory prediction performance if the dependent variable has normality [56,57]. However, several normality tests exhibit very low *p*-values because of the intricate pattern of building electric energy consumption, as shown in Table 17. These results indicate that building electric energy consumption does not have normality. Hence, even though the MLR exhibited outstanding prediction performance when the prediction time was close, the performance was getting worse as the prediction time was getting farther. This is because it is difficult to reflect the probabilistic trend. Nevertheless, we can confirm that MLR is useful for VSTLF, which is required for real-time peak electric load management.

Because the strengths of our COSMOS model can be easily implemented and the model has a high prediction performance in multiple STLFs, the COSMOS model can be used in the comparison with other forecasting models in the future. If the COSMOS model is used to predict the hourly or 30-min interval electric energy consumption in the future, the input variables can be constructed by applying Eqs. (4)–(9) to the calendar information and the rest, as described above. The optimal DNN models can then be constructed for the target building by setting the sliding window sizes to 336 and 168 for the 30-min interval and hourly electric load forecasting, respectively.

6. Conclusion

Because of the development of IT, DNN-based STLF models exhibit an excellent performance. However, selecting the optimal DNN model is challenging because several hyperparameters must be considered to determine the best combination of neural networks. To solve this problem, we combined STLF models using sliding window-based PCR in this study and developed the COSMOS model. To apply the proposed model to every building, we collected and preprocessed the time factor, weather information, and historical electric load to reflect the building electric energy consumption pattern of the forecasting time. Subsequently, we built four DNN-based forecasting models and applied their forecasting values to diverse PCR models. We determined the optimal PCR model considering the PCs and sliding window size. The model with PC 1 and a seven-day window size exhibited an excellent prediction performance. We called this model COSMOS model. The COSMOS model demonstrated the best prediction performance compared with various forecasting techniques. In addition, the COSMOS model performed a stable prediction performance, even in multiple STLFs. The COSMOS model can configure DNN models even in a CPU environment and can be used a baseline model in the future due to the simple input variable and model configurations.

However, the proposed model has a few limitations. One is that it does not satisfactorily predict the building electric energy consumption on weekends. In addition, we applied the model only to a single building because of difficulties in collecting building electric energy consumption data. Hence, additional validation of the applicability of our forecasting model is needed. For this purpose, we aim to collect new building electric energy consumption data in the future and discover new variables that can reflect the target building characteristics and building electric energy consumptions on weekends. Nevertheless, we believe that our proposed stacking ensemble method is sufficient for the forecasting of the electric energy consumptions of other buildings.

Declaration of Competing Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

CRediT authorship contribution statement

Jihoon Moon: Conceptualization, Formal analysis, Methodology, Validation, Writing - original draft. **Seungwon Jung:** Software, Validation. **Jehyeok Rew:** Data curation, Visualization. **Seungmin Rho:** Data curation, Resources, Writing - review & editing. **Eenjun Hwang:** Project administration, Supervision, Writing - review & editing.

Acknowledgements

We would like to appreciate Dr. Young-Hwan Choi for providing the building electricity consumption data of Kiturami Co., Ltd. to conduct this research work. This research was supported in part by the Korea Electric Power Corporation (grant number: R18XA05) and in part by Energy Cloud R&D Program (grant number: 2019M3F2A1073184) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.enbuild.2020.109921.

Appendix A

As mentioned in Section 2, to exhibit that high computational costs are required to build deep learning models such as DAE, CNN, and LSTM in a CPU environment, we calculated training time and test time of these models by using our building electric energy consumption dataset. The input variables, periods of training and test sets, and computer environment are the same as mentioned above.

The structure of the DAE consists of five layers, and each layer consists of 21, 18, 15, 18, and 21 nodes, respectively, as shown in Fig. 14. The other hyperparameters are the same as the DNN model mentioned in Section 4.1. The CNN and LSTM structures used in the experiment can predict multiple points (from 1 to 96)-ahead electric loads. Thus, all input variables that exist in 15-min intervals during the day are considered. Based on width (W) × height (H) × channel (C), the first layer of the CNN consists of 64 filters of 48 × 21 × 1 size, and the second layer of the CNN consists of 128 filters of 5 × 1 × 64 size. If more layers are added to the CNN, the additional convolutional layer is constructed using 5 × 1 × 128 size filter. All convolutional layers use ReLU as an activation function, and the next layer of the layer has a max-pooling layer. In addition, the output layer of CNN consists of a fully-connected layer that uses ReLU as an activation function to predict the building electric energy consumption. The dimension of each LSTM layer set to 128. The output layer of LSTM consists of a fully-connected layer that uses ReLU as an activation function

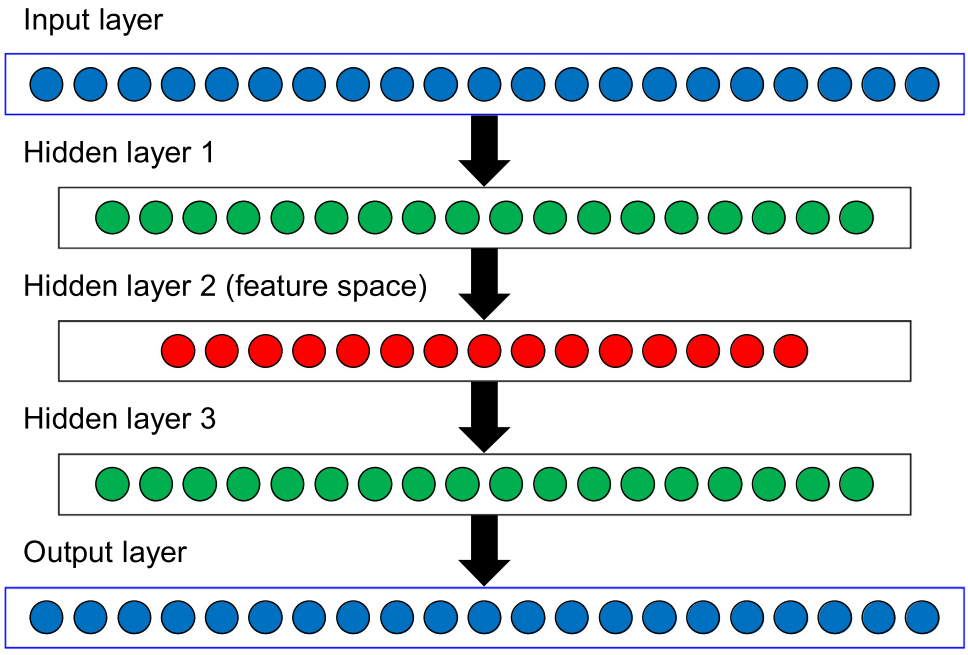


Fig. 14. Architecture of DAE network.

Table 18
Training and test times by each deep learning technique.

DL techniques	Library	Purpose	No. of hidden layers	Training time (unit: seconds)	Test time (unit: milliseconds)
DAE	TensorFlow	Feature extraction	3	63.09	-
DNN	scikit-learn	Prediction	2	35.78	6.98
			3	49.17	10.97
			4	86.76	12.94
			5	113.76	15.96
CNN	TensorFlow		2	1298.72	766.95
			3	1656.96	902.59
			4	1878.44	1035.23
			5	2049.23	1150.92
LSTM			2	9092.16	5700.76
			3	12872.48	6320.08
			4	19840.45	7568.75
			5	35743.60	11803.22

to predict the building electric energy consumption. The other hyperparameters, such as optimizer (Adam), batch size (96), learning rate (0.001), and the number of epochs (150), are the same as the DNN model mentioned in Section 4.1.

As shown in Table 18, we confirmed that other deep learning techniques were required more time than DNNs. Because considering the various combinations of hyperparameters is needed lots of time, we can see that deep learning techniques are not suitable for a CPU environment.

Appendix B

To demonstrate that the COSMOS model can be applied not only to a single commercial building but also to several building clusters containing various types of buildings, we constructed our forecasting model using electric energy consumption datasets of building clusters we used in our previous studies [35,52]. The electric energy consumption data used in the experiment were collected every 15-min interval from a private university in Seoul, South Korea, from January 1, 2015, to February 28, 2018. We presented the building cluster information and box plots of the electric energy consumption of each building cluster in Table 19 and Fig. 15, respectively.

The electric energy consumption data were divided training and test sets from January 1, 2015, to February 28, 2017 and March 1, 2017, to February 28, 2018, respectively. We constructed various STLF models using the same input variables, as described in Section 3. As shown in Tables 20 and 21, experimental re-

Table 19
Building cluster information.

Cluster #	No. of buildings	Cluster description	Types of buildings
Cluster A	16	Dormitory	Dormitory, gymnasium
Cluster B	32	Humanities and social sciences	Main hall, library, complex facility, classroom, office, etc.
Cluster C	5	Science and engineering	Laboratory, library, complex facility, classroom, office, etc.

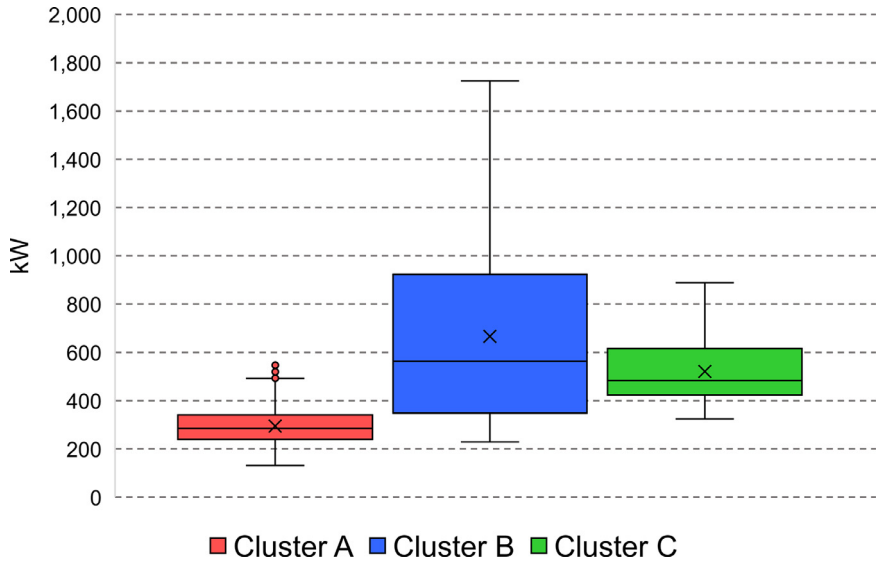


Fig. 15. Box plots by building clusters.

Table 20
Comparison of the MAPEs by the building cluster (bolded values represent the best values).

Forecasting techniques	Cluster A	Cluster B	Cluster C
Shallow neural network [35]	13.16	10.94	5.65
Auto-encoder + random forest [52]	5.78	6.90	3.59
Persistence	9.21	17.69	8.25
Multiple linear regression	6.78	14.76	5.34
K-nearest neighbor	8.48	12.44	5.49
Decision tree	8.56	17.75	6.55
Bagging	8.28	16.03	6.05
Gradient boosting machine	5.64	6.76	3.24
Random forest	6.04	6.46	3.30
Support vector regression (kernel: Poly)	5.72	8.98	3.82
Support vector regression (kernel: RBF)	5.98	7.30	3.70
DNN (number of hidden layers: 2)	5.87	6.27	3.23
DNN (number of hidden layers: 3)	5.96	6.16	3.27
DNN (number of hidden layers: 4)	6.05	6.14	3.32
DNN (number of hidden layers: 5)	6.13	6.13	3.35
COSMOS	**5.05**	**6.01**	**2.75**

Table 21
Comparison of the MAEs by the building cluster (bolded values represent the best values).

Forecasting techniques	Cluster A	Cluster B	Cluster C
Shallow neural network [35]	43.32	68.36	27.62
Auto-encoder + random forest [52]	18.24	42.96	17.79
Persistence	29.40	122.05	44.05
Multiple linear regression	21.21	78.85	26.11
K-nearest neighbor	26.66	78.39	28.02
Decision tree	27.31	99.74	32.63
Bagging	26.45	92.04	29.84
Gradient boosting machine	18.01	41.85	16.08
Random forest	19.66	41.29	16.02
Support vector regression (kernel: Poly)	17.95	51.33	18.79
Support vector regression (kernel: RBF)	19.06	44.24	18.01
DNN (number of hidden layers: 2)	18.80	38.84	15.86
DNN (number of hidden layers: 3)	19.12	38.26	16.03
DNN (number of hidden layers: 4)	19.43	38.12	16.21
DNN (number of hidden layers: 5)	19.70	38.11	16.38
COSMOS	**15.86**	**36.80**	**13.98**

sults demonstrated that our COSMOS model can also be applied to building clusters with different purposes. Accordingly, it can support efficient cluster/community energy management system (CEMS) operation.

References

[1] L. Hurtado, P. Nguyen, W. Kling, Agent-based control for building energy management in the smart grid framework, in: IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), IEEE, 2014, pp. 1–6, doi:10.1109/ISGTEurope.2014.7028937.
[2] M. Manic, D. Wijayasekara, K. Amarasinghe, J.J. Rodriguez-Andina, Building energy management systems: the age of intelligent and adaptive buildings, IEEE Ind. Electron. Mag. 10 (1) (2016) 25–39, doi:10.1109/MIE.2015.2513749.
[3] A. Ożadowicz, A new concept of active demand side management for energy efficient prosumer microgrids with smart building technologies, Energies 10 (11) (2017) 1771, doi:10.3390/en10111771.
[4] X. Jin, G. Wang, Y. Song, C. Sun, Smart building energy management based on network occupancy sensing, J. Int. Council Electr. Eng. 8 (1) (2018) 30–36, doi:10.1080/22348972.2018.1462608.
[5] R. Khan, A. Mahmood, A. Safdar, Z.A. Khan, N.A. Khan, Load forecasting, dynamic pricing and DSM in smart grid: a review, Renew. Sustain. Energy Rev. 54 (2016) 1311–1322, doi:10.1016/j.rser.2015.10.117.
[6] J. Moon, Y. Kim, M. Son, E. Hwang, Hybrid short-term load forecasting scheme using random forest and multilayer perceptron, Energies 11 (12) (2018) 3283, doi:10.3390/en11123283.
[7] J. Kim, J. Moon, E. Hwang, P. Kang, Recurrent inception convolution neural network for multi short-term load forecasting, Energy Build. 194 (2019) 328–341, doi:10.1016/j.enbuild.2019.04.034.
[8] S.F. Fux, M.J. Benz, A. Ashouri, L. Guzzella, Short-term thermal and electric load forecasting in buildings, International Conference on Clean-tech for Smart Cities & Buildings: From Nano to Urban Scale (CISBAT), 2013.
[9] T. Hong, S. Fan, Probabilistic electric load forecasting: A tutorial review, Int. J. Forecasting 32 (3) (2016) 914–938, doi:10.1016/j.ijforecast.2015.11.011.
[10] A.-Y. Yoon, H.-J. Moon, S.-I. Moon, Very short-term load forecasting based on a pattern ratio in an office building, Int. J. Smart Grid Clean Energy 5 (2) (2016) 94–99, doi:10.12720/sgce.5.2.94-99.
[11] B. Yildiz, J.I. Bilbao, A.B. Sproul, A review and analysis of regression and machine learning models on commercial building electricity load forecasting, Renew. Sustain. Energy Rev. 73 (2017) 1104–1122, doi:10.1016/j.rser.2017.02.023.
[12] L. Hernandez, A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1460–1495, doi:10.1109/SURV.2014.032014.00094.
[13] A. Almalaq, G. Edwards, A review of deep learning methods applied on load forecasting, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2017, pp. 511–516, doi:10.1109/ICMLA.2017.0-110.
[14] T. Domhan, J.T. Springenberg, F. Hutter, Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves, Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
[15] S. Karsoliya, Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture, Int. J. Eng. Trends Technol. 3 (6) (2012) 714–717.

[16] F. Yan, O. Ruwase, Y. He, T. Chilimbi, Performance modeling and scalability optimization of distributed deep learning systems, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1355–1364, doi:10.1145/2783258.2783270.
[17] E.E. LeDell, Scalable ensemble learning and computationally efficient variance estimation, Doctoral Dissertation, University of California, Berkeley, USA, 2015.
[18] R.K. Jain, K.M. Smith, P.J. Culligan, J.E. Taylor, 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 123 (2014) 168–178, doi:10.1016/j.apenergy.2014.02.057.
[19] Y. Chen, H. Tan, Short-term prediction of electric demand in building sector via hybrid support vector regression, Appl. Energy 204 (2017) 1363–1374, doi:10.1016/j.apenergy.2017.03.070.
[20] K. Grolinger, A. L'Heureux, M.A. Capretz, L. Seewald, Energy forecasting for event venues: Big data and prediction accuracy, Energy Build. 112 (2016) 222–233, doi:10.1016/j.enbuild.2015.12.010.
[21] K. Amber, M. Aslam, S. Hussain, Electricity consumption forecasting models for administration buildings of the UK higher education sector, Energy Build. 90 (2015) 127–136, doi:10.1016/j.enbuild.2015.01.008.
[22] S. Jurado, À. Nebot, F. Mugica, N. Avellana, Hybrid methodologies for electricity load forecasting: Entropy-based feature selection with machine learning and soft computing techniques, Energy 86 (2015) 276–291, doi:10.1016/j.energy.2015.04.039.
[23] M.W. Ahmad, M. Mourshed, Y. Rezgui, Trees vs neurons: comparison between random forest and ANN for high-resolution prediction of building energy consumption, Energy Build. 147 (2017) 77–89, doi:10.1016/j.enbuild.2017.04.038.
[24] D. Palchak, S. Suryanarayanan, D. Zimmerle, An artificial neural network in short-term electrical load forecasting of a university campus: a case study, J. Energy Res. Technol. 135 (3) (2013) 032001, doi:10.1115/1.4023741.
[25] A. Bagnasco, F. Fresi, M. Saviozzi, F. Silvestro, A. Vinci, Electrical consumption forecasting in hospital facilities: an application case, Energy Build. 103 (2015) 261–270, doi:10.1016/j.enbuild.2015.05.056.
[26] K. Li, C. Hu, G. Liu, W. Xue, Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis, Energy Build. 108 (2015) 106–113, doi:10.1016/j.enbuild.2015.09.002.
[27] S. Ryu, J. Noh, H. Kim, Deep neural network-based demand side short term load forecasting, Energies 10 (1) (2016) 3, doi:10.3390/en10010003.
[28] C. Fan, F. Xiao, Y. Zhao, A short-term building cooling load prediction method using deep learning algorithms, Appl. Energy 195 (2017) 222–233, doi:10.1016/j.apenergy.2017.03.064.
[29] D.L. Marino, K. Amarasinghe, M. Manic, Building energy load forecasting using deep neural networks, in: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2016, pp. 7046–7051, doi:10.1109/IECON.2016.7793413.
[30] H. Shi, M. Xu, R. Li, Deep learning for household load forecasting—a novel pooling deep RNN, IEEE Trans. Smart Grid 9 (5) (2018) 5271–5280, doi:10.1109/TSG.2017.2686012.
[31] M. Cai, M. Pipattanasomporn, S. Rahman, Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques, Appl. Energy 236 (2019) 1078–1088, doi:10.1016/j.apenergy.2018.12.042.
[32] W. Kong, Z.Y. Dong, Y. Jia, D.J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on LSTM recurrent neural network, IEEE Trans. Smart Grid 10 (1) (2017) 841–851, doi:10.1109/TSG.2017.2753802.
[33] P. Lusis, K.R. Khalilpour, L. Andrew, A. Liebman, Short-term residential load forecasting: Impact of calendar effects and forecast granularity, Appl. Energy 205 (2017) 654–669, doi:10.1016/j.apenergy.2017.07.114.
[34] KMA, Korea Meteorological Administration. Dong-Nae forecast (digital forecast), 2019. Retrieved from http://www.kma.go.kr/eng/index.jsp
[35] J. Moon, J. Park, E. Hwang, S. Jun, Forecasting power consumption for higher educational institutions based on machine learning, J. Supercomput. 74 (8) (2018) 3778–3800, doi:10.1007/s11227-017-2022-x.
[36] J. Mikulik, Energy demand patterns in an office building: A Case Study in Kraków (Southern Poland), Sustainability 10 (8) (2018) 2901, doi:10.3390/su10082901.
[37] J. Moon, K.-H. Kim, Y. Kim, E. Hwang, A short-term electric load forecasting scheme using 2-stage predictive analytics, in: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, 2018, pp. 219–226, doi:10.1109/BigComp.2018.00040.
[38] A.K. Bhateja, M. Din, ANN based distinguishing attack on RC4 stream cipher, in: Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), Springer, 2013, pp. 101–109, doi:10.1007/978-81-322-1041-2_9.
[39] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, The MIT Press, Cambridge, MA, USA, 2016 ISBN 9780262035613.
[40] T. K. Gupta and K. Raza, "Optimizing deep neural network architecture: A Tabu search based approach," *arXiv preprint* arXiv:1808.05979, 2018.
[41] J. Heaton, Introduction to Neural Networks with Java, Heaton Research, Inc., Chesterfield, MO, USA, 2008 ISBN 1-60439-008-5.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint* arXiv:1412.6980, 2014.
[43] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, in: Neural networks: Tricks of the trade: Springer, 2012, pp. 437–478.
[44] S. Whalen, G. Pandey, A comparative analysis of ensemble classifiers: case studies in genomics, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 807–816, doi:10.1109/ICDM.2013.21.
[45] Z. Ma, P. Wang, Z. Gao, R. Wang, K. Khalighi, Ensemble of machine learning algorithms using the stacked generalization approach to estimate the warfarin dose, PLoS One 13 (10) (2018) e0205872, doi:10.1371/journal.pone.0205872.
[46] F. Divina, A. Gilson, F. Goméz-Vela, M.G. Torres, J.F. Torres, Stacking ensemble learning for short-term electricity consumption forecasting, Energies 11 (4) (2018) 949, doi:10.3390/en11040949.
[47] X. Qiu, L. Zhang, Y. Ren, P.N. Suganthan, G. Amaratunga, Ensemble deep learning for regression and time series forecasting, in: 2014 IEEE Symposium on Computational Intelligence in ensemble learning (CIEL), IEEE, 2014, doi:10.1109/CIEL.2014.7015739.
[48] G.T. Ribeiro, V.C. Mariani, L. d. S. Coelho, Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting, Eng. Appl. Artif. Intell. 82 (2019) 272–281, doi:10.1016/j.engappai.2019.03.012.
[49] C.J. Merz, M.J. Pazzani, A principal components approach to combining regression estimates, Mach. Learn. 36 (1–2) (1999) 9–32, doi:10.1023/A:1007507221352.
[50] A. Folch-Fortuny, F. Arteaga, A. Ferrer, PCA model building with missing data: New proposals and a comparative study, Chemom. Intell. Lab. Syst. 146 (2015) 77–88, doi:10.1016/j.chemolab.2015.05.006.
[51] C. Marinoiu, Classic and modern in regression modelling, Econ. Insights-Trends Challenges 69 (1) (2017).
[52] M. Son, J. Moon, S. Jung, E. Hwang, A short-term load forecasting scheme based on auto-encoder and random forest, 3rd International Conference on: Applied Physics, System Science and Computers (APSAC), 2018, doi:10.1007/978-3-030-21507-1_21.
[53] F. Pedregosa, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830 Oct.
[54] M.A. Khairalla, X. Ning, N.T. AL-Jallad, M.O. El-Faroug, Short-term forecasting for energy consumption through stacking heterogeneous ensemble learning model, Energies 11 (11) (2018) 1605, doi:10.3390/en11061605.
[55] S. Park, J. Moon, E. Hwang, 2-stage electric load forecasting scheme for day-ahead CCHP scheduling, 13th IEEE International Conference on Power Electronics and Drive Systems (PEDS), IEEE, 2019.
[56] M.J. Crawley, The R Book, John Wiley and Sons, Ltd, West Sussex, England, 2007 ISBN 978-0-470-97392-9.
[57] S. Chatterjee, A.S. Hadi, Regression Analysis by Example, John Wiley and Sons, Inc., Hoboken, New Jersey, USA, 2012 ISBN 978-0-470-90584-5.