

## Article

# Machine Learning Approach to Predict Building Thermal Load Considering Feature Variable Dimensions: An Office Building Case Study

Yongbao Chen <sup>1</sup>, Yunyang Ye <sup>2,\*</sup>, Jingnan Liu <sup>1</sup>, Lixin Zhang <sup>1</sup>, Weilin Li <sup>3</sup> and Soheil Mohtaram <sup>1</sup>

<sup>1</sup> School of Energy and Power Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

<sup>2</sup> Department of Civil, Environmental and Architectural Engineering, University of Colorado Boulder, Boulder, CO 80309, USA

<sup>3</sup> School of Civil Engineering, Zhengzhou University, Zhengzhou 450001, China

\* Correspondence: yunyang.ye@colorado.edu

**Abstract:** An accurate and fast building load prediction model is critically important for guiding building energy system design, optimizing operational parameters, and balancing a power grid between energy supply and demand. A physics-based simulation tool is traditionally used to provide the building load demand; however, it is constrained by its complex model development process and requirement for engineering judgments. Machine learning algorithms (i.e., data-driven models) based on big data can bridge this gap. In this study, we used the massive energy data generated by a physics-based tool (EnergyPlus) to develop three data-driven models (i.e., LightGBM, random forest (RF), and long-short term memory (LSTM)) and compared their prediction performances. The physics-based models were developed using office prototype building models as baselines, and ranges were provided for selected key input parameters. Three different input feature dimensions (i.e., six-, nine-, and fifteen-input feature selections) were investigated, aiming to meet different demands for practical applications. We found that LightGBM significantly outperforms the RF and LSTM algorithms, not only with respect to prediction accuracy but also in regard to computation cost. The best prediction results show that the coefficient of variation of the root mean squared error (CVRMSE), squared correction coefficient ( $R^2$ ), and computation time are 5.25%, 0.9959, and 7.0 s for LightGBM, respectively, evidently better than the values for the algorithms based on RF (18.54%, 0.9482, and 44.6 s) and LSTM (22.06%, 0.9267, and 758.8 s). The findings demonstrate that a data-driven model is able to avoid the process of establishing a complicated physics-based model for predicting a building's thermal load, with similar accuracy to that of a physics-based simulation tool.

**Keywords:** load prediction; feature engineering; machine learning; LightGBM; grid-integrated buildings



**Citation:** Chen, Y.; Ye, Y.; Liu, J.; Zhang, L.; Li, W.; Mohtaram, S. Machine Learning Approach to Predict Building Thermal Load Considering Feature Variable Dimensions: An Office Building Case Study. *Buildings* **2023**, *13*, 312. <https://doi.org/10.3390/buildings13020312>

Academic Editor: Kian Jon Chua

Received: 16 December 2022

Revised: 12 January 2023

Accepted: 14 January 2023

Published: 20 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

### 1.1. Literature Study

Building thermal load prediction plays an important role in energy management and efficiency. It has wide applications, such as in determining the capacity of heating, ventilation, and air conditioning (HVAC) systems in the phase of building design [1]; providing operational optimization control in building energy systems for existing or retrofitted buildings [2]; and determining the demand response baseline for grid-integrated buildings [3–5]. In the field of building load prediction, the thermal loads for heating and cooling are the most difficult parts to accurately predict, owing to the complex, nonlinear relationships among the influencing factors (input feature variables), such as the weather conditions, building physics, and different operational behaviors. Therefore, previous studies have focused on the thermal load of buildings, especially on the loads of HVAC systems. Prediction approaches are widely categorized into three types: physics-based

models (white-box), data-driven models (black-box), and reduced-order models (gray-box) [6,7], while the data-driven models have been increasingly discussed in recent years.

White-box models, such as professional software, can predict building loads with high accuracy. Several mature software tools, including EnergyPlus, TRNSYS, and Modelica, are widely used in the engineering industry [8]. However, developing a physics-based model is time-consuming, and prior knowledge is required. Gray-box models simplify the complicated functions to a resistance and capacity model. The key shortcoming of gray-box models is that they do not consider internal heat gains and occupant behaviors [9]. A black-box model is a data-driven model based on historical data. Nowadays, with the rapid development of building-based big data and machine learning algorithms, developing a black-box model is a promising approach for load prediction. A rich set of studies have focused on machine learning algorithms, such as support vector regression (SVR) [4,10], random forest (RF) [11,12], extreme gradient boosting (XGBoost) [13–15], long-short term memory (LSTM) [16–18], artificial neural networks (ANNs) [19], and convolutional neural networks (CNN) [20]. In previous studies, the two most popular fields concerned sensitivity analyses of different dimensions of input features and hyperparameter tuning. Based on a comprehensive literature review, Table 1 lists the papers investigating different data-driven models. We found that LightGBM (similar to XGBoost but faster), RF, and LSTM have been reported as the most popular algorithms for building load predictions.

**Table 1.** Detailed descriptions and comparisons of data-driven models to predict load demands in papers.

Reference	Input Features	Data-Driven Model	Prediction Results
[13]	Day of week; Hour of day; Holiday; Outdoor dry bulb temperature; Outdoor relative humidity	XGBoost; RF; SVR; LSTM	Metrics CVRMSE XGBoost: 21.1% RF: 23.7% SVR: 25.0% LSTM: 20.2%
[21]	Total gross floor area; year of build; building height; shape form factor; vertical to horizontal ratio; length of the building; and width of the building; building morphology	LightGBM; XGBoost; RF; SVR	Metrics R <sup>2</sup> LightGBM: 0.8608 XGBoost: 0.8137 RF: 0.7959 SVR: 0.7363
[22]	Historical load; weather data; calendar rules	LightGBM; XGBoost; SVM; RF	Stacking method XGBoost and LightGBM have obtained the higher accuracy
[23]	Outdoor dry bulb temperature; Schedules	XGBoost; RF; SVR; ANN	Metrics CVRMSE XGBoost: 62% RF: 64% SVR: 64% ANN: 73%
[24]	Outdoor dry bulb temperature; Outdoor relative humidity; Wind speed; Solar radiation; Hour of day	XGBoost; RF; SVR; ANN	Metrics CVRMSE XGBoost: 4.5% RF: 4.6% SVR: 5.5% ANN: 5.1%
[25]	Relative compactness; Surface area; Wall area; Roof area; Number of floors; Orientation; Glazing area; Outdoor dry bulb temperature; Outdoor relative humidity; Solar radiation	ANN; SVM; RF; XGBoost	Metrics R <sup>2</sup> XGBoost: 0.998 RF: 0.973 SVR: 0.972 ANN: 0.968
[26]	Aspect ratio; Relative compactness; Glazing area; Roof area; Surface area; Wall area; Orientation; Number of floors; Glazing area	ANN; SVR; RF	Metrics MAE (kW) ANN: 1.15 SVR: 0.90 RF: 1.45

XGBoost was first released in 2014 and has become a powerful algorithm; most Kaggle competitions have reported it as the final winner [27,28]. XGBoost is based on a gradient boosting algorithm for assembling weak learners into a strong learner. XGBoost can be easily implemented using the Python, R, Julia, and Scala platforms [29]. Wang et al. [13] predicted long-term building thermal loads using XGBoost. Five input feature variables were considered: the day of the week, hour of the day, holiday status, temperature, and relative humidity. They found that, in shallow machine learning, XGBoost is the best algorithm. Yan et al. [23] obtained similar results. In the cooling season, they found that 11 input features could represent the main factors influencing cooling energy consumption. The prediction performances of XGBoost (CVRMSE: 62%), RF (CVRMSE: 64%), SVR (CVRMSE: 64%), and ANNs (CVRMSE: 73%) were not good relative to the results in [13]. Wang et al. [24] investigated models including XGBoost, RF, ANN, and SVR for predicting consumption from the thermal load of a residential building in Tianjin, China. The CVRMSE values of the prediction results from these models were as follows: RF (5.0%), XGBoost (5.8%), SVR (6.2%), and ANN (7.0%). Although the prediction accuracy was good, the computation cost is huge when the input feature dimensions and data size are large. In recent years, Light Gradient Boosting Machine (LightGBM) has been proposed as a novel and promising gradient boosting framework; it is similar to XGBoost. XGBoost and LightGBM are various tree-boosting methods. Shi et al. [22] concluded that LightGBM obtained a higher prediction accuracy compared to SVR in electric load forecasting. Zhang et al. [21] proposed a model of LightGBM integrated with the Shapley Additive exPlanation algorithm to predict energy usage and greenhouse gas emissions. The results show that the proposed LightGBM can achieve a higher prediction accuracy compared to XGBoost, RF, and SVR. Through the literature review, we can find that the tree-based algorithm including LightGBM and XGBoost is a promising method to obtain a better prediction result [30].

RF is a supervised learning algorithm based on decision trees. Compared with other algorithms, fewer parameters need to be tuned when using the RF model [31]. Ahmad et al. [11] compared three different algorithms for energy predictions. They selected the ambient temperature and relative humidity ratio as the input feature variable, and they found that the RF model (MAPE: 2.64%) outperforms LMSR (MAPE: 3.10%) and NARM (MAPE: 4.21%). Except the advantages of less overfitting and higher accuracy, RF presents the importance of the input features, which can be used in the model training and testing processes. A feature importance analysis chooses the main features and skips the weak features; this is critical to accelerating the computational process and ensuring the prediction accuracy. LSTM is a type of recurrent neural network (RNN) algorithm. It was first introduced by Hochreiter and Schmidhuber in 1997 [32]. Differing from a traditional neural network, LSTM passes the last step's information to the next time step (i.e., backpropagation). With these merits, LSTM comprises an inborn network for processing sequential data. It has advantages in solving complex and long time lag tasks, whereas traditional RNN algorithms are not good at this. LSTM has performed better in short-term load predictions than the linear regression, SVM, RF, and XGBoost algorithms [13,33]. The detailed theory of LightGBM, RF, and LSTM can be seen in Section 2.2.

Different algorithms have their merits for different building energy datasets, according to the previous studies, and the generalization performance of a prediction model is mainly based on the quality of data. For a specific building case, the prediction accuracy of a data-driven model is good enough now [30], while the generalization performance for exogenous buildings is still poor in the field of building energy predictions. More papers have investigated data-driven models using a specific building case, and these models are usually biased and can lead to a poor generalization [34,35]. For this purpose, the data source should cover all the possible ranges of each main variable and represent the overall energy consumption patterns of buildings, because a data-driven model cannot deduce the result for unseen data. To do that, previous studies had some attempts to acquire big data from numerous buildings. For instance, 5000 residential buildings from the Ministry of Housing Communities and Local Government (MHCLG) repository have been

used to develop a data-driven model [36]. Meteorological variables with specific ranges and a 5-min interval energy dataset from EnerNOC were used to develop a data-driven interval forecasting model for building energy predictions [37]. Despite more dimensions of information having been considered, the diversity and depth of each main variable are still worth investigating for building a well-generalized model.

Although many studies have already investigated the prediction performances of different data-driven algorithms, there are still two research gaps. The first gap is whether the data-driven model is good enough to represent the physics-based tools. It is urgent to investigate the feasibility of using a data-driven model in the field of building load forecasting. To the best knowledge of the authors, the previous studies mostly focused on the existing building with acquired historical data, which means that the model is well developed on this specific building while it usually has a poor generalization for other buildings, especially for design phase buildings. Therefore, using physics-based tools to generate a massive dataset that covers all the common energy use scenarios could be a promising way to develop a good generalization data-driven model. Another research gap concerns providing a sufficient number of key input features to determine the building energy use in data-driven models. For building owners, the difficulty in obtaining the different dimensions of input features is not equal, and some main information might be unavailable. Thus, presenting the prediction performance based on the number of main feature variables is quite useful for building owners to estimate energy consumption with a certain accuracy. Researchers have made different conclusions in the past. For instance, Yan et al. [23] concluded that 11 input features were sufficient, whereas Wang et al. [13] stated that 5 features were adequate. How many input features should be used in the model to obtain a satisfying prediction result? This is still a challenge. To address these research gaps, first, we acquired data by running massive EnergyPlus building models that represented different buildings. Note that the data can also be acquired from any physics-based tools or on-site data. Second, different dimensions of the key feature variables were selected to develop three widely used models: LightGBM, RF, and LSTM in the context of forecasting the HVAC electrical load. Notably, the HVAC electrical load was predicted as an equivalent of a thermal load, as the electricity load is more commonly used in modern grid-integrated buildings.

### *1.2. Motivations and Contributions*

In practice, it is extremely difficult to measure all of the required inputs to a physics-based building energy model. These models require thousands of input parameters derived from prior knowledge, and their simulations are typically computationally intensive. Thus, a data-driven model could be a promising approach to solve this problem. However, developing a data-driven model to represent a physics-based model well is still a big challenge. First, the value range of each input variable should fully cover the practical situations; second, the simulation scenarios and dataset sizes should be big enough; finally, the computational costs should be acceptable for the practical engineering applications. According to previous studies, weather information is widely used in data-driven models; furthermore, building physical and thermophysical variables, such as the window-to-wall ratio and total heat transfer coefficient of the envelope, have been increasingly considered. In data-driven models, the number of input features can influence the prediction accuracy and computation speed. There is a tradeoff in constructing different sizes of input features under practical conditions. Therefore, it is intriguing to conduct an overall input feature dimension investigation when building managers can obtain different dimension building information. In this context, developing a data-driven model by using the massive energy data from simulation tools to represent a physics-based model is an appealing approach to building load predictions. The main contribution of this work is developing a data-driven model by using the massive energy data from simulation tools to represent a physics-based model to building load predictions. These developed models are of high practical value despite that building energy managers have enough or limited building information,

especially for design phase buildings where the available information is lacking. The goal of this paper is to develop a suitable data-driven model to estimate the thermal load demand of buildings promptly and accurately, especially in the phase of building design. For existing building, we can calibrate and evaluate the proposed model using real data.

The remainder of this paper is organized as follows. Section 2 introduces the methodology, the theories of the selected algorithms and evaluation indexes. Section 3 presents the overall comparison results and discussion. Finally, the main conclusions and future applications are presented in Section 4.

## 2. Methodology

Data-driven models do not require establishing thermal equilibrium equations; usually, fewer inputs are required compared to physics-based simulation tools. A data-driven model uses data to deduce the hidden relationships between output (e.g., cooling load) and input feature variables (e.g., weather and building physics information) using a statistical approach. This approach is well adapted to buildings in the design phase, where detailed input parameters may be lacking. The research outline is shown in Figure 1. The methodology was designed with four main steps. The first step was EnergyPlus model development, which was the main step when using the co-simulation method of Python and EnergyPlus to obtain the required dataset. Note that the development of building energy models can be replaced by any other physics-based simulation tools. The second step processed the data and analyzed the key input features. The widely used input variables included these three types: (1) time-related information, such as the day type, occupancy, and equipment schedules; (2) weather conditions, such as the temperature, humidity, and solar radiation; and (3) building physical parameters, such as the window-to-wall ratio and R-value of the wall. The output targets are generally thermal loads or electricity consumption [38,39]. The third step selected the data-driven algorithm by reviewing previous studies and tuned the selected models. Data-driven models have gained great interest in the buildings field because of their simplicity and flexibility [40]. In this section, three promising data-driven methods are presented, including LightGBM, RF, and LSTM. Additionally, some important hyperparameters are tuned, and they can be seen in Tables A1–A3. The last step evaluated the developed models and listed the future applications. It is worth noting that simulation data was used, not real data, because we needed to change each input feature variable within a wider range to obtain enough data points (thousands of building types and millions of data points) to train a well-generalization model, which is nearly impossible to attain from real buildings.

### 2.1. Seed Model and Energy Data Source Description

#### 2.1.1. Seed Model Description

This section describes the office building models used in EnergyPlus (Version 9.0.1) and the ranges of the input features used in these cases to obtain energy data. Three categories of building energy models were developed: (1) small office building, (2) medium office building, and (3) large office building. DOE Commercial prototype building models were used as the starting point [41]. The geometry of these three building types is shown in Table 2; all of the models have a rectangular footprint. Table 2 provides key information on these three models. The small office building has 1 floor, the medium office building has 3 floors, and the large office building has 12 above-grade floors and 1 basement. Furthermore, the envelope types and HVAC system types are different. Table 2 also shows the types of exterior walls, roofs, heating and cooling systems, and HVAC system operation schedules.

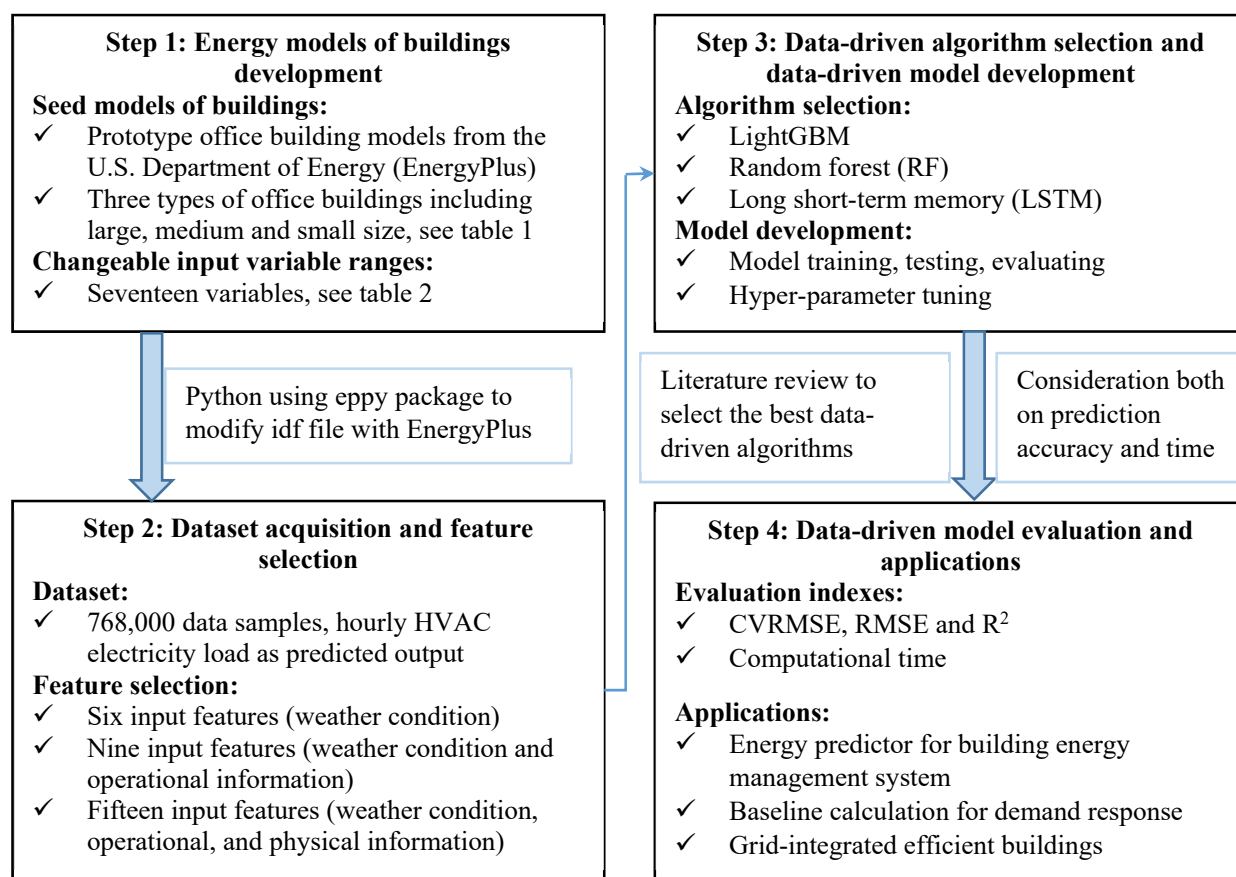
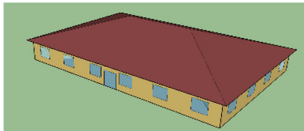
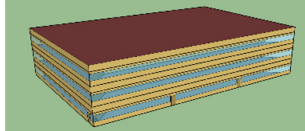
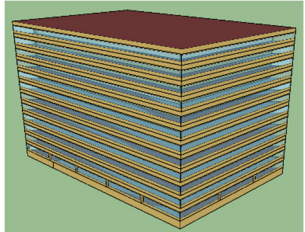


Figure 1. Research methodology and its outline.

Table 2. Seed models description in EnergyPlus [41].

Item	Small Office	Description Medium Office	Large Office
Geometry			
Total Floor Area	511 m <sup>2</sup>	4980 m <sup>2</sup>	46,321 m <sup>2</sup>
Exterior Wall Type	Wood frame walls	Steel frame walls	Mass walls
Roof Type	Attic roof with wood joint	Built-up roof	Built-up roof
Heating Type	Air source heat pump with gas furnace as backup	Gas furnace	One gas-fired boiler
Cooling Type	Air source heat pump	Packaged air conditioning	Water direct expansion cooling coil Two water-cooled centrifugal chillers
HVAC Operation Schedule	Weekdays: 6:00 am–7:00 pm	Weekdays: 6:00 am–10:00 pm Saturdays: 6:00 am–6:00 pm	Weekdays: 6:00 am–10:00 pm Saturdays: 6:00 am–6:00 pm

### 2.1.2. Energy Data Source Description

Seventeen key input variables were changed with the appropriate steps in each EnergyPlus model. Notably, for each EnergyPlus model, the first four variables (No. 1–4

in Table 3 are EnergyPlus standard TMY3 weather data) are time series data that change for each simulation time step, whereas the other variables are constant in each step once determined. Based on the default values provided by the office prototype building models, we determined the ranges of these input variables. We used  $\pm 20\%$  of the default values as the lower and upper limits for most of the input variables [42], except for the cooling and heating temperature set points. For these two input variables, we used  $\pm 1.11$  °C ( $\pm 2$  °F) of the default values as the lower and upper limits, so as to ensure that the indoor temperature was within a preferable range. Table 3 lists the input variable ranges for these three types of office buildings in detail. In order to simplify the data-driven model, we selected the most important input feature variables in Table 3, and some not important variables such as airtightness were neglected. A total of 768,000 valid data samples were generated from the above-mentioned three EnergyPlus seed models, and the dataset and full code to develop the data-driven model can be downloaded freely in [43]. When we generated the dataset, the Python package named eppy was used to co-simulate with EnergyPlus. All the variables listed in Table 3 were changed respectively in different steps, and the total simulation time to obtain all the data was about 700 h (running on a Dell Precision 7920 Tower, 20 kernel CPU).

**Table 3.** Input variable ranges for the three types of office buildings.

No.	Input Feature Variables	Unit	Range		
			Small Office	Medium Office	Large Office
1	Dry Bulb	°C	[−32.8, 37.0]	[−32.8, 37.0]	[−32.8, 37.0]
2	Relative Humidity	%	[4, 100]	[4, 100]	[4, 100]
3	Global Horizontal Radiation	Wh/m <sup>2</sup>	[0, 964]	[0, 964]	[0, 964]
4	Wind Speed	m/s	[0, 14.9]	[0, 14.9]	[0, 14.9]
5	Total Floor Area	m <sup>2</sup>	[409, 613]	[3986, 5979]	[37,056, 55,584]
6	Aspect Ratio	-	[1.2, 1.8]	[1.2, 1.8]	[1.2, 1.8]
7	Window-to-Wall Ratio	-	[0.16, 0.24]	[0.26, 0.40]	[0.32, 0.48]
8	Floor Height	m	[2.44, 3.66]	[2.19, 3.29]	[2.20, 3.29]
9	Exterior Wall Insulation R-value	(m <sup>2</sup> ·K)/W	[2.46, 3.68]	[2.25, 3.38]	[1.31, 1.97]
10	Roof Insulation R-value	(m <sup>2</sup> ·K)/W	[6.48, 9.72]	[4.25, 6.37]	[4.25, 6.37]
11	Specific Heat for Internal Thermal Mass	J/(kg·K)	[968, 1452]	[968, 1452]	[968, 1452]
12	Cooling Temperature Set Point	°C	[22.78, 25.00]	[22.89, 25.11]	[22.89, 25.11]
13	Heating Temperature Set Point	°C	[20.00, 22.22]	[19.89, 22.11]	[19.89, 22.11]
14	Fresh air volume	m <sup>3</sup> /s·m <sup>2</sup>	[0.000345, 0.000518]	[0.000345, 0.000518]	[0.000345, 0.000518]
15	People Density	m <sup>2</sup> /person	[13.27, 19.91]	[14.86, 22.29]	[14.86, 22.29]
16	Lighting Power Density	W/m <sup>2</sup>	[6.80, 10.20]	[6.80, 10.20]	[6.80, 10.20]
17	Electric Equipment Power Density	W/m <sup>2</sup>	[5.42, 8.14]	[6.46, 9.68]	[6.46, 9.68]

### 2.1.3. Input Feature Selection

Feature selection is critical for data-driven models. External weather conditions, physical parameters, and operational schedules for equipment and occupant behavior are three common input feature types [44–46]. The feature selection of each building could be different from each other. We selected the input feature variables due to three reasons: (1) the importance of input variables in physics-based thermal equilibrium equations, (2) prior knowledge in building energy consumption estimation (including the literature study and engineering experience), and (3) the difficulty to obtain in practice. Through a literature study, we found that the outdoor dry bulb temperature, outdoor relative humidity, solar radiation, day of the week, and hour of the day were the five most frequently used features in data-driven models. Except for the external climate data, the physical information of the buildings (such as the number of floors, wall area, glazing area, and window-to-wall ratio) was used to improve the prediction accuracy. Therefore, we selected 17 widely used variables that are easier to obtain and have a great impact on building energy consumption.

The complexity of the input feature size influences the computational cost and prediction accuracy. To study this impact, three input feature scenarios were investigated, aiming to meet different demands for practical applications. Table 4 shows the details of these scenarios. The key input variables represent the main weights in the machine learning models. The use of limited input information to achieve satisfactory prediction results and an acceptable time cost is a concern for building energy managers.

**Table 4.** Input feature scenarios used in our three algorithms.

Scenarios	Scenario 1: Six Input Features	Scenario 2: Nine Input Features	Scenario 3: Fifteen Input Features
Description	Weather condition	Weather condition and operational information	Weather condition, operational information, and physical parameters
Input features	Hour of the day Historical load data Dry bulb temperature (°C) Relative humidity (%) Global horizontal radiation (W·h/m <sup>2</sup> ) Wind speed (m/s)	Scenario 1 Cooling temperature Set point (°C) Heating temperature Set point (°C) Fresh air volume [m <sup>3</sup> /(s·m <sup>2</sup> )]	Scenario 2 R-value of wall [m <sup>2</sup> ·K/W] Internal mass (average specific heat of the walls) [J/(kg·K)] Window to wall ratio Floor height (m) Shape coefficient (1/m) Aspect ratio

## 2.2. Selected Data-Driven Algorithm Description

### 2.2.1. LightGBM

LightGBM is a gradient-boosting framework comprising a tree-based learning algorithm, i.e., a gradient-boosting decision tree (GBDT). The GBDT is a widely used algorithm in machine learning, owing to its efficiency and accuracy; XGBoost is a typical framework employing this algorithm. However, when the input feature dimensions and/or data size are large, as in modern buildings, the predictable data scale and computation speed remain unsatisfactory. To tackle these deficiencies, LightGBM was proposed based on two novel techniques: gradient-based one-sided sampling (GOSS) and exclusive feature bunding (EFB). These two techniques speed up the training process by up to 20 times, with almost the same accuracy as the traditional GBDT algorithm [47]. LightGBM was first released on 17 October 2016 as a part of Microsoft Corporation's "Distributed Machine Learning Toolkit" project [48]. It was designed to be distributed and efficient, with the advantages of a faster training speed, higher efficiency, lower memory usage, parallel support, and the ability to handle large-scale data. LightGBM is a promising algorithm for big data [47,49]. GBDT is a mature algorithm, and the detailed theory thereof is discussed in other references [13,28]; thus, in this study, we only introduce the theories for GOSS and EFB in detail.

GOSS is a technique for balancing data information reduction and prediction performance. GOSS reduces the computation costs by distinguishing between different gradients of instances, retaining larger gradient instances while randomly sampling smaller gradients and thereby reducing the computation memory costs. The gradient magnitude of the instance represents the training error; thus, an instance with a small gradient can be eliminated, as it is already well trained. To avoid large changes in the training data distribution from the elimination of some instances, GOSS also randomly samples small gradient instances to secure the integrity of the original data. This way, although GOSS reduces the number of instances, the generalization error is close to that calculated using the full data instances. To prove that, the variance gain  $V_j(d)$  of feature  $j$  at splitting point  $d$  is defined as shown in Equation (1).

$$V_{j|O}(d) = \frac{1}{A} \left( \frac{(\sum\{x_i \in O : x_{ij} \leq d\} g_i)^2}{n_{l|O}^j(d)} + \frac{(\sum\{x_i \in O : x_{ij} > d\} g_i)^2}{n_{r|O}^j(d)} \right) \quad (1)$$



In the above,  $A = \sum I[x_i \in O]$ ,  $n_{l|O}^j(d) = \sum I[x_i \in O : x_{ij} \leq d]$ , and  $n_{r|O}^j(d) = \sum I[x_i \in O : x_{ij} > d]$ .  $x$  is the training set with  $i$  instances,  $g_i$  is the negative gradient of the loss function, and  $O$  is the training dataset on a fixed node of the decision tree.

The training instances are initially ranked by their absolute gradients in descending order; next, the top  $a\%$  of the larger gradients are selected as subset  $A$ ; then,  $b\%$  of the remaining gradients are randomly selected as subset  $B$ . Thus, the estimated variance gain  $\tilde{V}_j(d)$  over the subset  $A \cup B$  can be defined as shown in Equation (2).

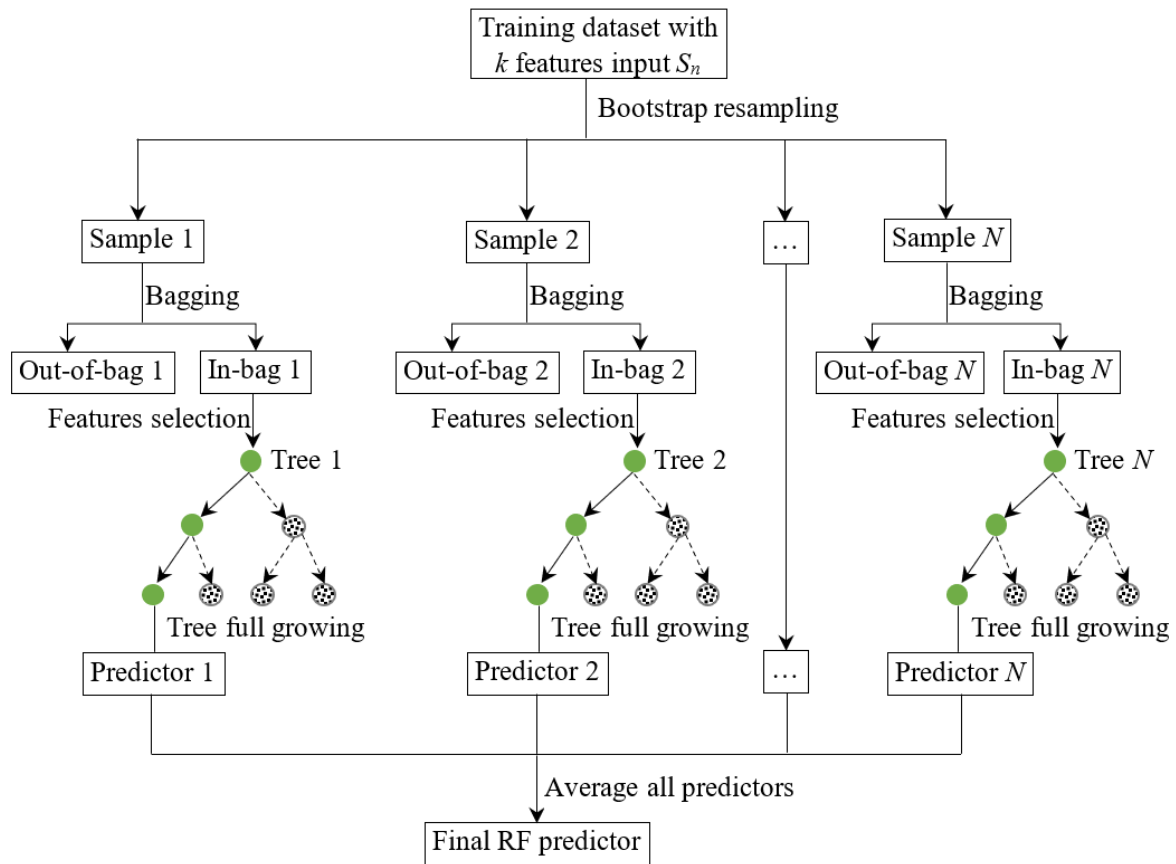
$$\tilde{V}_j(d) = \frac{1}{n} \left( \frac{\left( \sum \{x_i \in A_l\} g_i + \frac{1-a}{b} \sum \{x_i \in B_l\} g_i \right)^2}{n_l^j(d)} + \frac{\left( \sum \{x_i \in A_r\} g_i + \frac{1-a}{b} \sum \{x_i \in B_r\} g_i \right)^2}{n_r^j(d)} \right) \quad (2)$$

Here,  $A_l = \{x_i \in A : x_{ij} \leq d\}$ ,  $A_r = \{x_i \in A : x_{ij} > d\}$ ,  $B_l = \{x_i \in B : x_{ij} \leq d\}$ , and  $B_r = \{x_i \in B : x_{ij} > d\}$ . There is a theory for proving that GOSS would not lose much training accuracy as compared with the full dataset [47], i.e.,  $\varepsilon_{gen}^{GOSS}(d) = \left| \tilde{V}_j(d) - V_*(d) \right| \leq \left| \tilde{V}_j(d) - V_j(d) \right| + \left| V_j(d) - V_*(d) \right| \triangleq \varepsilon_{GOSS}(d) + \varepsilon_{gen}(d)$ .

EFB is another technique for reducing feature dimensions to improve the computational efficiency and is based on feature bundling. Usually, the bundled features are mutually exclusive, e.g., one feature is zero and the other is non-zero; therefore, these two features can be bundled together without losing information. In the case where two features are not mutually exclusive, a “conflict ratio” can be used to measure the degree of non-exclusion. When this ratio is small, the two features can be bound without excessively affecting the final accuracy. There are three steps in the EFB method. In Step 1, the features are sorted according to the total number of non-zero values; in Step 2, the conflict ratio between different features is calculated; and in Step 3, the conflict ratio is minimized by iterating through each feature and then binding the features. In this way, the time complexity is reduced from  $O(N_{data} * N_{feature})$  to  $O(N_{data} * N_{bundle})$ , where  $N_{bundle} \ll N_{feature}$ .

### 2.2.2. Random Forest (RF)

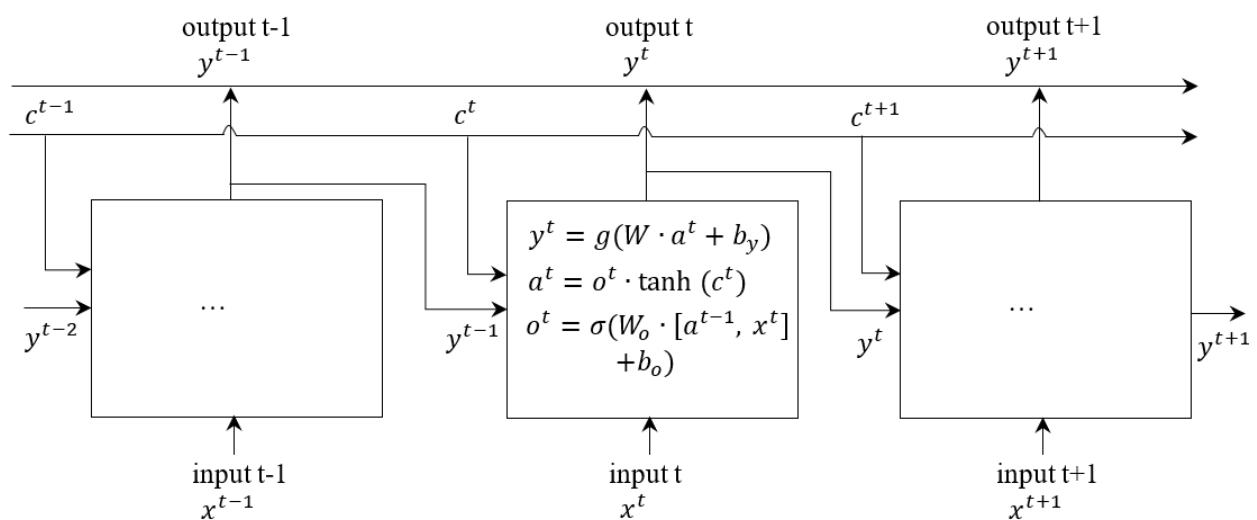
The diagrammatic prediction process of RF is shown in Figure 2. Each decision tree is randomly formed with different features and training samples, and the trees can be trained in parallel. Thus, the prediction accuracy is higher than that of a single decision tree. In the RF model, the number of trees and depth of a tree are the two key parameters that must be tuned; therefore, fewer parameters must be set than in other algorithms [12,31,50]. The RF algorithm includes four main processes: bootstrap resampling, bagging and out-of-bag error (OOBE) estimation, random feature selection, and full-depth decision tree growth [51], as shown in Figure 2. First,  $N$  samples from the training dataset  $S_n$  are randomly selected as bootstrap samples chosen with replacements, i.e., the same sample  $(X_i, Y_i)$  may appear repeatedly. Second, the bagging technique selects samples from the bootstrap samples  $N$ ; the remaining samples comprise the out-of-bag dataset. Third, there is a random selection of a predefined number  $p$  of total features  $k$ , and RF attempts to search for the best cutting among these  $p$  features. Finally, the best cutting is set by minimizing the cost function until the full-depth decision tree grows. The OOBE technique, or generalization error, is highly effective for estimating the generalization ability of the constructed model. In view of these technologies, the main advantage of the RF algorithm is its immunity to noise [51,52].



**Figure 2.** Diagrammatic process of the random forest (RF) algorithm.

### 2.2.3. Long Short-Term Memory (LSTM)

LSTM is a type of RNN algorithm. Figure 3 shows the principle of LSTM. It is an inborn network capable of accurately modeling complex multivariate sequences (such as building energy demands), although this increases its computation costs [53]. It has advantages in solving complex and long time lag tasks that traditional RNN does not. In one study [13], LSTM performed better for the load prediction than the SVM and XGBoost algorithms.



**Figure 3.** Principles of long short-term memory (LSTM).

Equations (3) and (4) define the architecture of the basic RNN algorithm. In this algorithm, only the memory of the last time step  $t-1$  can be passed to time step  $t$ . However, the longer memory of the past time steps  $t-n$  can be passed by introducing three special gates and two memory cells: input gate  $i^t$ , forget gate  $f^t$ , and output gate  $o^t$  are defined as shown in Equations (5)–(7), respectively; the candidate memory cell  $\tilde{c}^t$  and memory cell  $c^t$  are defined in Equations (8) and (9), respectively.

$$a_{RNN}^t = g\left(W_a \cdot [a^{t-1}, x^t] + b_a\right) \quad (3)$$

$$y_{RNN}^t = g'(W_a' \cdot a_{RNN}^t + b_y) \quad (4)$$

$$i^t = \sigma\left(W_i \cdot [a^{t-1}, x^t] + b_i\right) \quad (5)$$

$$f^t = \sigma\left(W_f \cdot [a^{t-1}, x^t] + b_f\right) \quad (6)$$

$$o^t = \sigma\left(W_o \cdot [a^{t-1}, x^t] + b_o\right) \quad (7)$$

$$\tilde{c}^t = \tanh\left(W_c \cdot [a^{t-1}, x^t] + b_c\right) \quad (8)$$

$$c^t = f^t \cdot \tilde{c}^{t-1} + i^t \cdot \tilde{c}^t \quad (9)$$

$$a_{LSTM}^t = o^t \cdot \tanh(c^t) \quad (10)$$

$$y_{LSTM}^t = g'(W_a' \cdot a_{LSTM}^t + b_y) \quad (11)$$

In the above,  $g(x)$  and  $g'(x)$  are two activation functions;  $a^t$  is the activation function at time step  $t$ ;  $x^t$  and  $y^t$  are the input and output at time step  $t$ , respectively;  $b$  is the bias;  $W$  is the weight factor;  $\sigma(x)$  is the sigmoid function, which is defined in Equation (13); and  $\tanh(x)$  is the hyperbolic tangent function, as defined in Equation (12).

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (12)$$

$$\sigma(x) = \frac{e^x}{e^x + 1} \quad (13)$$

### 2.3. Data-Driven Model Development Process

The development process (i.e., Step 3 in Figure 1) of our proposed data is shown in Figure 4. First, the massive dataset is divided into two sets, including training and test datasets after data pre-processing. Second, feature engineering is implemented for the model inputs, which includes the weather conditions (e.g., temperature, humidity, and solar radiation); building physical parameters (e.g., R-value of the wall, floor height, internal mass, and Shape coefficient); and operational information (e.g., temperature setting and fresh air volume). The model output is the HVAC electrical load. Then, different models can be trained, and the hyperparameters are required to tune for better results. Last is using the test dataset to test and evaluate the developed model.

### 2.4. Prediction Performance Indices

To evaluate the prediction performances of different algorithms, three indices are generally used: the CVRMSE, root mean squared error (RMSE), and squared correlation coefficient ( $R^2$ ). The CVRMSE is a scale-independent indicator that is normalized by averaging the RMSE. The CVRMSE has been used in studies [13,54] and is recommended by the American Society of Heating, Refrigerating and Air-conditioning Engineers' (ASHRAE) Guidelines 14 [55]; the RMSE is a scale-dependent indicator and thus maintains the same scale as the original

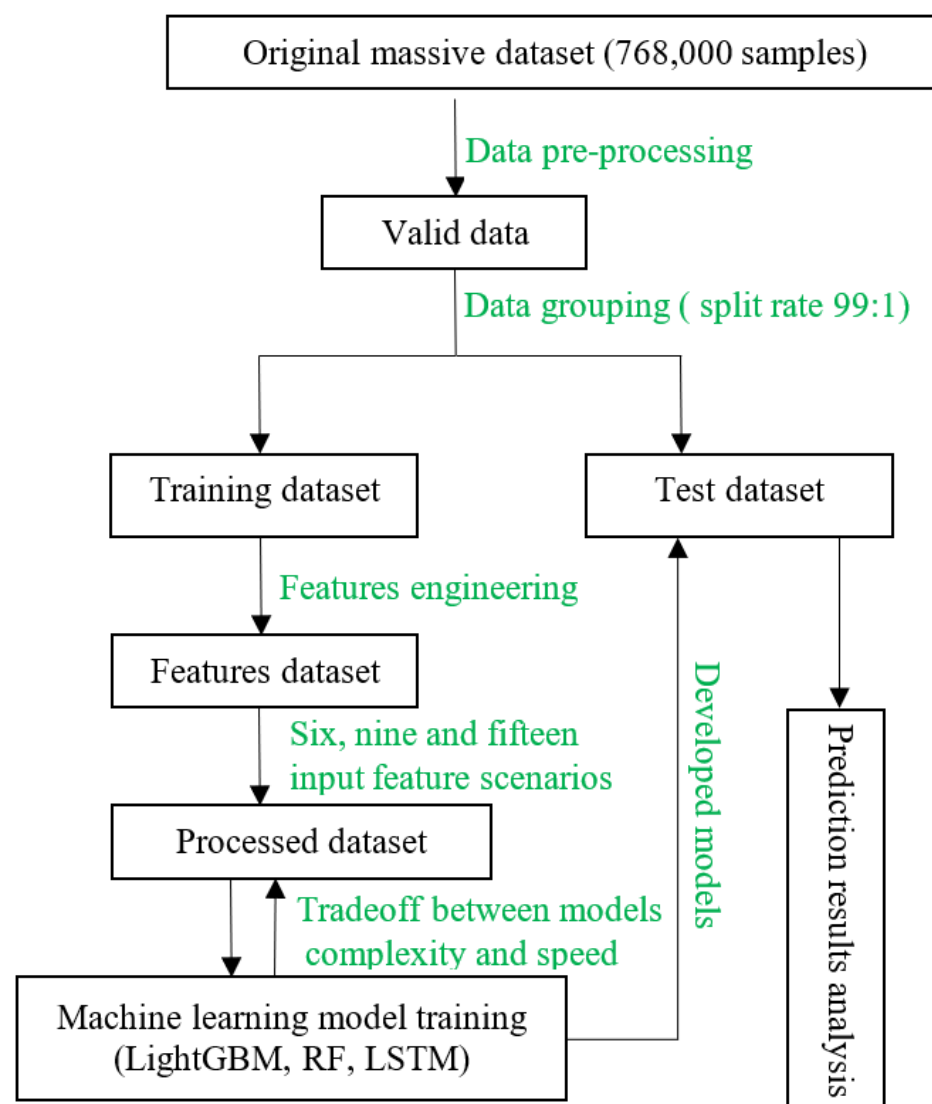
data; and  $R^2$  is the coefficient of determination, which ranges from 0 to 1 and can reflect the goodness of fit. These three indices are defined in Equations (14)–(16), respectively.

$$CVRMSE = \frac{\sqrt{\sum_{i=1}^n (\check{y}_i - y_i)^2 / n}}{\sum_{i=1}^n y_i / n} \quad (14)$$

$$RMSE = \sqrt{\sum_{i=1}^n (\check{y}_i - y_i)^2 / n} \quad (15)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \check{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (16)$$

where  $\check{y}_i$ ,  $y_i$ , and  $\bar{y}$  represent the predicted value of sample  $i$ , the actual value of sample  $i$ , and the mean value of all sample datasets, respectively;  $n$  denotes the number of samples.



**Figure 4.** Flow chart of the data-driven model development process.

### 3. Results and Discussion

The prediction performances were calculated and compared under the above-mentioned scenarios, and the prediction granularity is 1 h. For each scenario, it is worth noting that we first split the 768,000 data samples into training and test sets at a ratio of 99:1 because

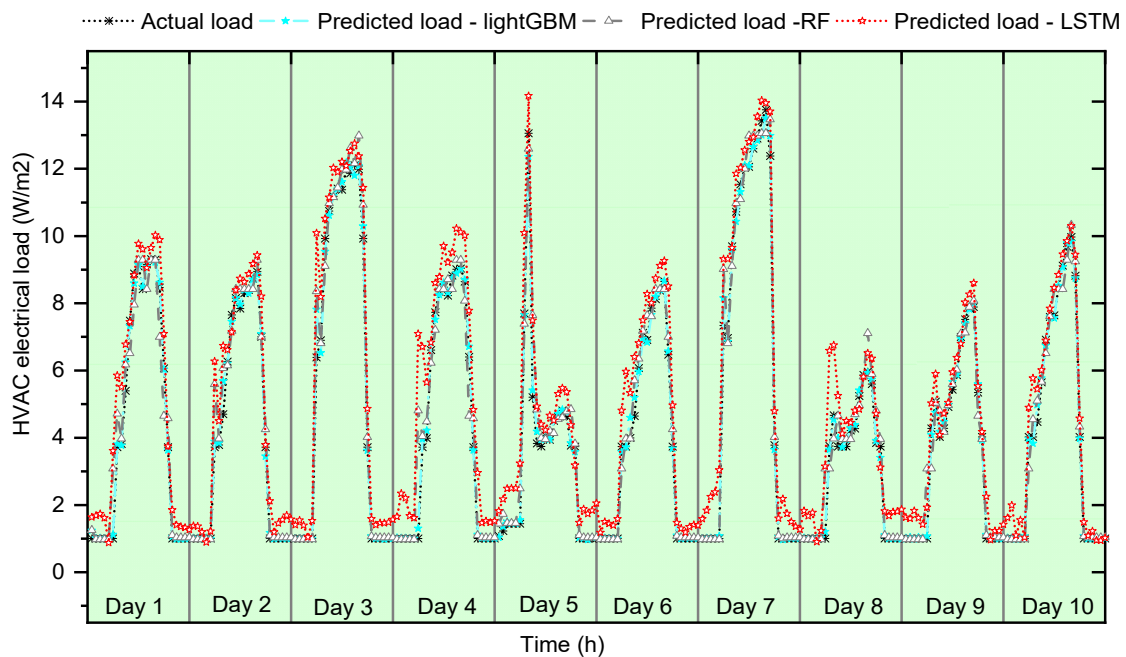
of the big size of the dataset [56], although the commonly used ratio is 80:20 or 70:30 for normal-sized datasets [36]. Second, different feature scenarios were selected as the inputs, and the hourly predicted HVAC electrical load was the output. Finally, we applied the five-fold cross-validation approach implemented in the scikit-learn Python package for the hyperparameter tuning in our models. The hyperparameter searching range and optimum results are shown in Tables A1–A3. All of the models were trained and tested on identical datasets. We compared three different machine learning algorithms (LightGBM, RF, and LSTM) under these three scenarios. To address cases with (such as existing buildings) and with no (such as buildings in the design phase) historical energy load data available, the input features were applied with and without the historical HVAC electricity load. In this paper, we obtained all the required HVAC electricity loads at the stage where we simulated all the seed models mentioned in Section 2.1. It is worth noting that we categorized the cases in the next three scenarios by distinguishing the use of historical load data or not. In all scenarios, three prediction performance indices were used to evaluate the prediction performance on the testing dataset: RMSE, CVRMSE, and  $R^2$ .

### 3.1. Scenario 1: Six Input Features

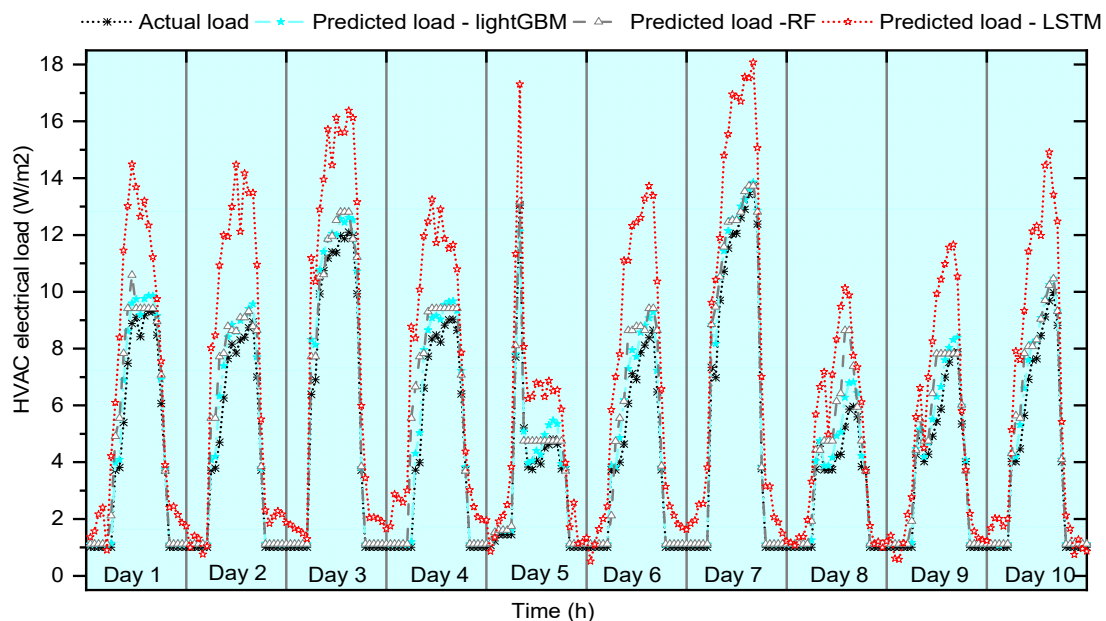
The simplified scenario used only six key input features to develop the model. As shown in Table 5, LightGBM has the highest prediction accuracy (CVRMSE: 7.14%) and fastest computational speed (5.4 s); the results from LSTM are the worst, not only in regard to accuracy (CVRMSE: 26.15%) but also to computational cost (716.5 s). If the CVRMSE is below 30% when the prediction step is hourly, the model is considerably acceptable and sufficiently close to physical reality for engineering purposes [57]. By this criterion, the LSTM result without the historical load is unacceptable. Additionally, the computation cost of the LSTM is more than a hundred times larger than that of LightGBM; this is consistent with our theoretical analysis in Section 2 and corresponds to the conclusions from the literature, i.e., that LSTM is recommended for small datasets and short-term predictions. Figure 5 presents the hourly predicted and actual HVAC electrical load profiles on the testing dataset; ten days are randomly selected from the testing data for visualization purposes. It is evident that the prediction performance of the LSTM algorithm is quite poor. All the models provide better prediction results when the historical load data is considered. The LSTM algorithm that does not use historical load data performs the worst, and the prediction deviation is large during the peak and valley load times.

**Table 5.** Comparison results of Scenario 1 with and without historical load data.

Scenario 1	Long Short-Term Memory (LSTM) (with)	LSTM (without)	LightGBM (with)	LightGBM (without)	Random Forest (RF) (with)	RF (without)
Root mean square error (RMSE)	1.07	2.26	0.29	0.71	0.76	1.02
Coefficient of variance of RMSE (CVRMSE)	26.15%	55.32%	7.14%	17.40%	18.59%	24.85%
$R^2$	0.896968	0.538813	0.992327	0.954380	0.947891	0.906964
Computation Time(s)	716.5	719.4	5.4	6.1	19.1	10.7



(a) With historical HVAC electricity load data



(b) Without historical HVAC electricity load data

**Figure 5.** Hourly prediction performances using six input features.

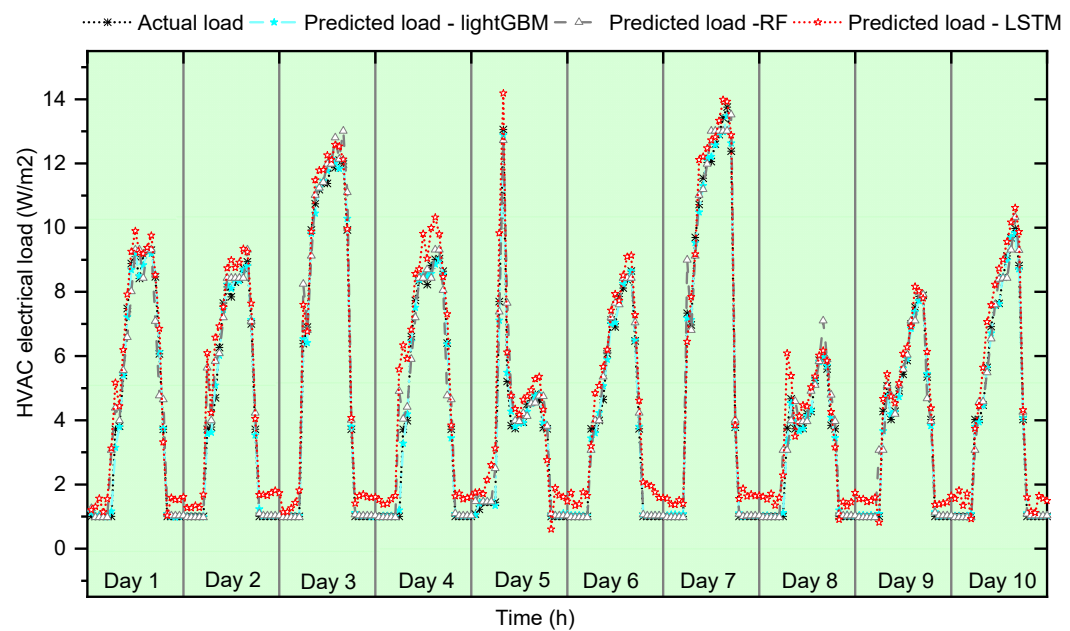
### 3.2. Scenario 2: Nine Input Features

As mentioned in Section 1, HVAC energy consumption is influenced by multiple factors, such as weather conditions, building physics, and operational parameters. Only six input features were used in Scenario 1, and the prediction results might not be convincing. Therefore, we added operational parameters (room temperature setting and fresh air volume) to boost the knowledge learning level in the training models. From comparing Tables 5 and 6, it can be seen that the prediction accuracy is slightly improved in all three models, although the computation cost also increases. By adding these operational parameters, the mean improvement percentages in the CVRMSE are approximately 10.9%, 17.6%, and 0.5% for LSTM, LightGBM, and RF, respectively. This improvement is evident in

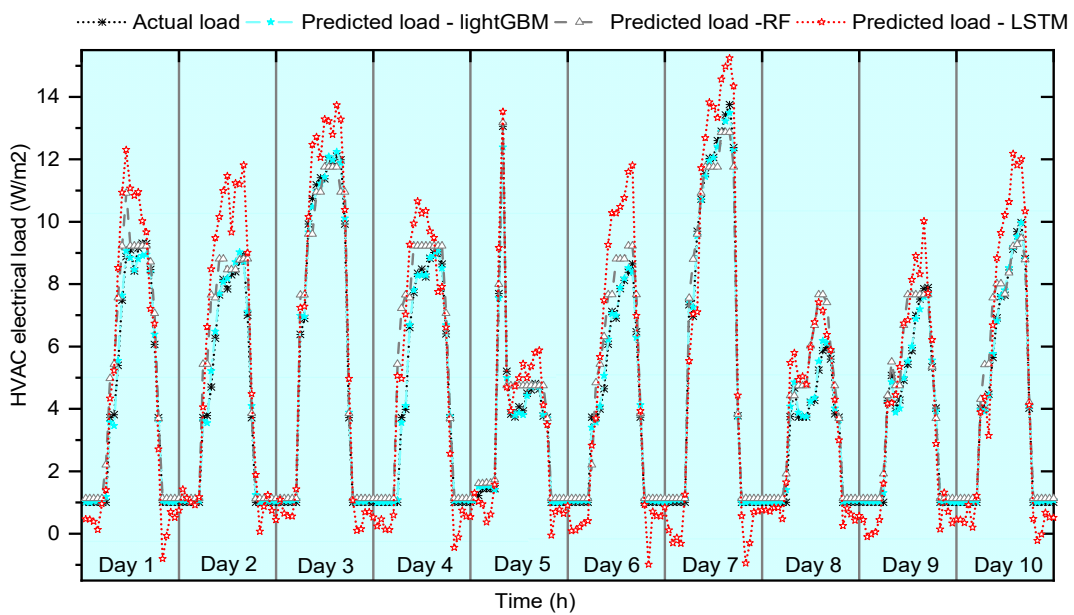
the LightGBM and LSTM models. As shown in Figure 6, the prediction deviation remains large during the peak and valley load times for the LSTM.

**Table 6.** Comparison results of Scenario 2 with and without historical load data.

Scenario 2	LSTM (with)	LSTM (without)	LightGBM (with)	LightGBM (without)	RF (with)	RF (without)
RMSE	1.04	1.83	0.25	0.57	0.76	1.01
CVRMSE	25.42%	44.79%	6.04%	13.94%	18.58%	24.61%
R <sup>2</sup>	0.902582	0.697689	0.994506	0.970706	0.947979	0.908734
Computation Time(s)	765.0	780.0	6.5	5.7	29.2	22.6



(a) With historical HVAC electricity load data



(b) Without historical HVAC electricity load data

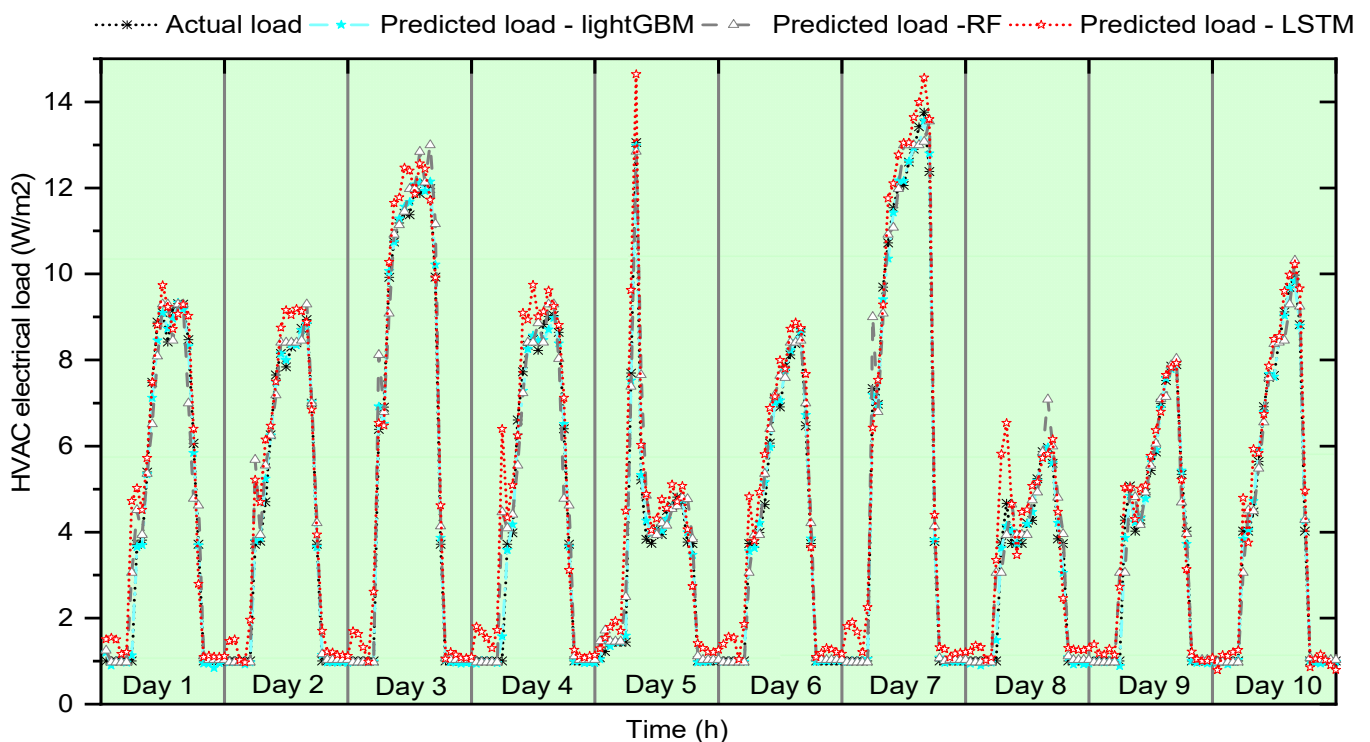
**Figure 6.** Hourly prediction performances using nine input features.

### 3.3. Scenario 3: Fifteen Input Features

In addition to the features in Scenario 2, the physical information of the building can be used to improve the prediction accuracy. We used 15 key input features to train the models in Scenario 3; three physical aspects (building shape, R-values of walls, and building thermal mass) were considered. Table 7 shows the hourly results from the different models with and without the historical load data. The best result for the CVRMSE was 5.25% in LightGBM, a promising result for the field of thermal load prediction. Furthermore, the computation time was only 7 s. The best results from the LSTM and RF approaches were 22.06% and 18.54%, respectively, i.e., close to the results from the previous study discussed in the Introduction. By adding six more building physical parameters, the mean improvement percentage of the CVRMSE was approximately 12.2%, 30.3%, and 1.6% for the LSTM, LightGBM, and RF approaches, respectively. The CVRMSE values of all three models were lower than 30%, indicating that they were all acceptable and sufficiently close to physical reality for engineering purposes. Figures 5–7 show that the prediction is gradually improved by employing additional input features.

**Table 7.** Comparison results of Scenario 3 with and without historical load data.

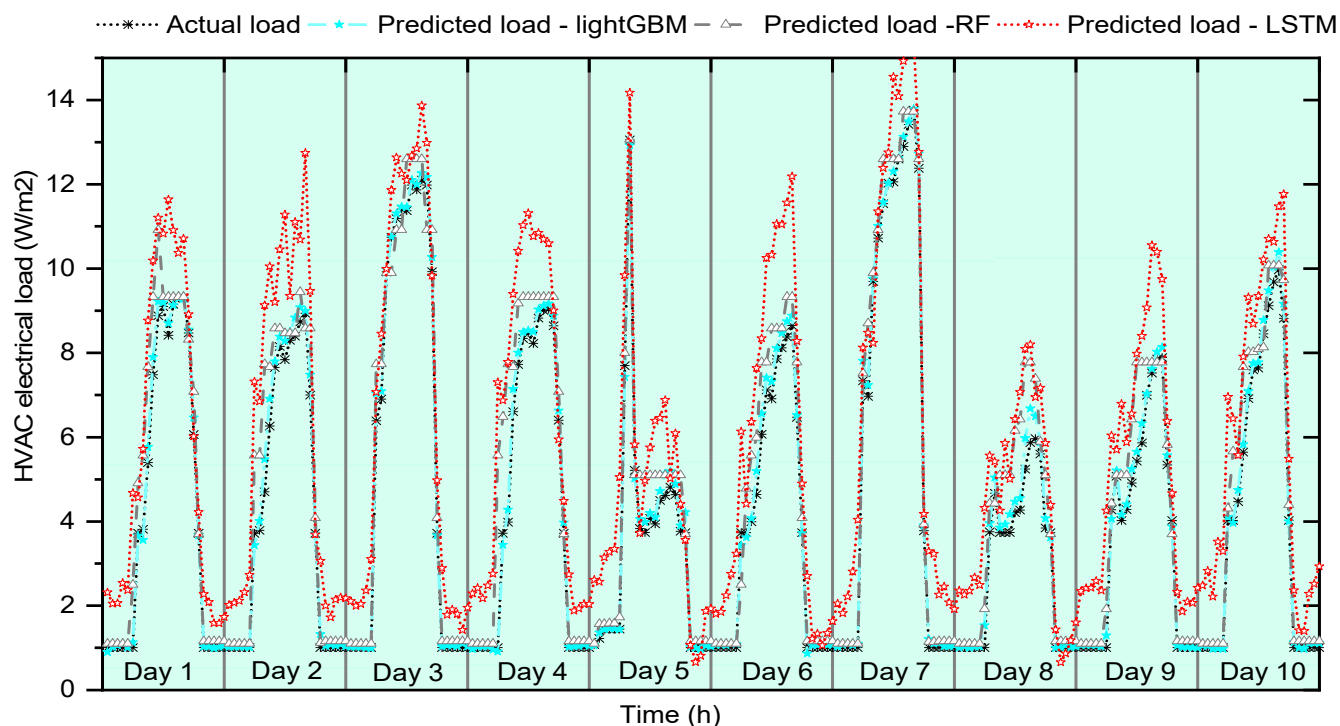
Scenario 3	LSTM (with)	LSTM (without)	LightGBM (with)	LightGBM (without)	RF (with)	RF (without)
RMSE	0.90	1.63	0.21	0.3	0.76	0.98
CVRMSE	22.06%	39.75%	5.25%	7.31%	18.54%	23.88%
R <sup>2</sup>	0.926666	0.761820	0.995854	0.991937	0.948189	0.914036
Computation Time(s)	756.8	751.4	7.0	7.4	44.6	36.5



(a) With historical HVAC electricity load data

Figure 7. Cont.





(b) Without historical HVAC electricity load data

**Figure 7.** Hourly prediction performances using fifteen input features.

### 3.4. Discussion

Generally, the prediction accuracy can be improved when more information of the building is used in the data-driven models. As shown in Figure 8, the CVRMSE of scenario 3 with the historical HVAC load data is the best in these three models, although the improvement is higher in LightGBM and LSTM. In the model of RF, different scenarios have a close prediction accuracy that means additional building information is not necessary to improve its accuracy. We can also find that the historical HVAC load data is important to improve the accuracy of the models. All in all, a CVRMSE of 7.1% can be achieved when only the weather information is used, and the highest accuracy of 5.3% reached when the weather and operational and physical information of the building structure are considered. Except for these fifteen input features, adding more information such as the occupant's behavior is worthwhile as a further study in the future.

As shown in Figure 8, LightGBM is the best algorithm in building thermal energy prediction. Generally, the more data samples trained in the model training process, the better the prediction accuracy, as more hidden knowledge between the inputs and outputs can be learned. We investigated this effect by increasing the size of the training dataset sample in LightGBM, and Figure 9 shows the results. When we used 76,800 data samples, the prediction was the worst; the prediction accuracy generally improved as the sample size increased from 76,800 to 768,000. Additional data samples improved the prediction accuracy but also increased the computation cost.

A quantified investigation of the feature importance is also interesting for researchers. Therefore, we investigated the feature importance in the LightGBM model. Figure 10 ranks the importance of these fifteen input features. In our building's case, the results showed that the day of hours, outdoor dry bulb temperature, historical load data, global horizontal radiation, and relative humidity are the five most important features for building thermal load prediction. However, other features that have lower importance values can still be used to further improve the prediction accuracy.

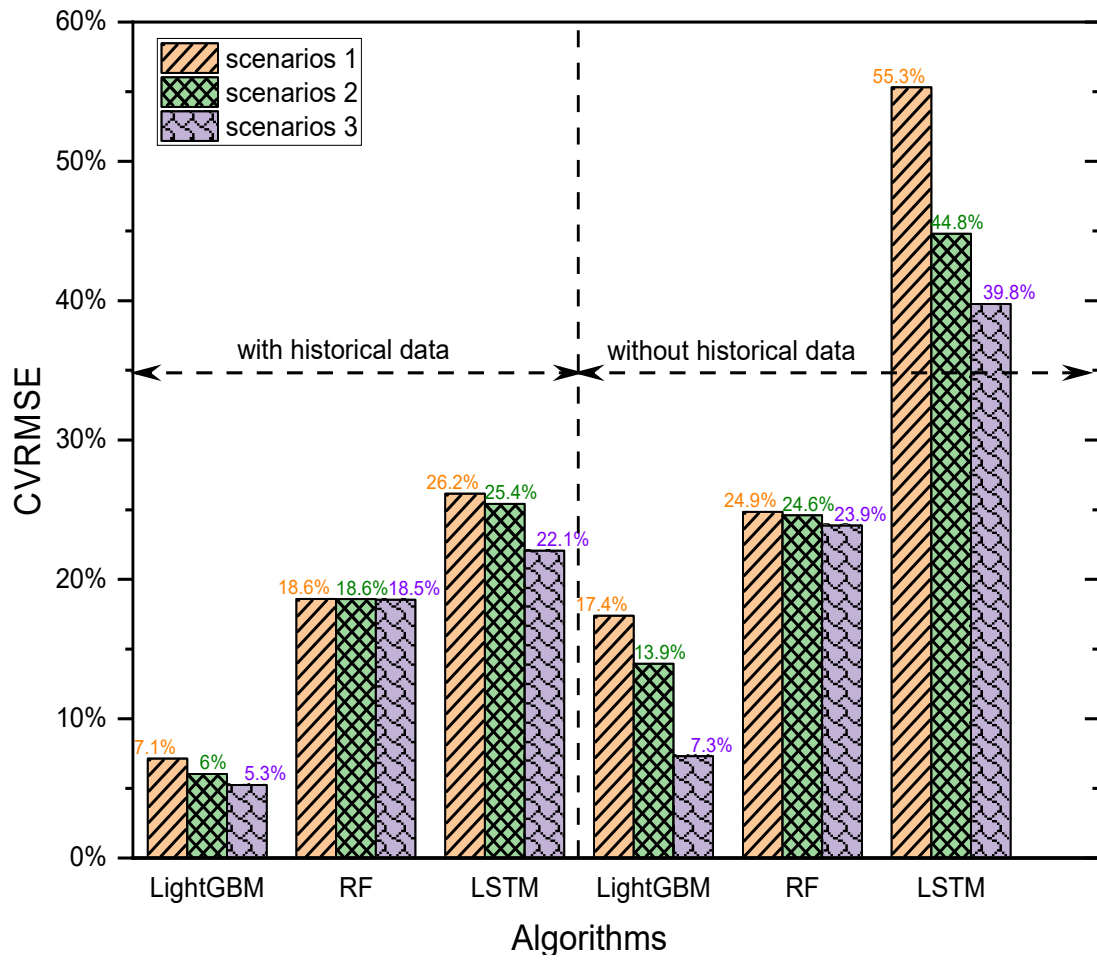


Figure 8. Prediction performances in different scenarios.

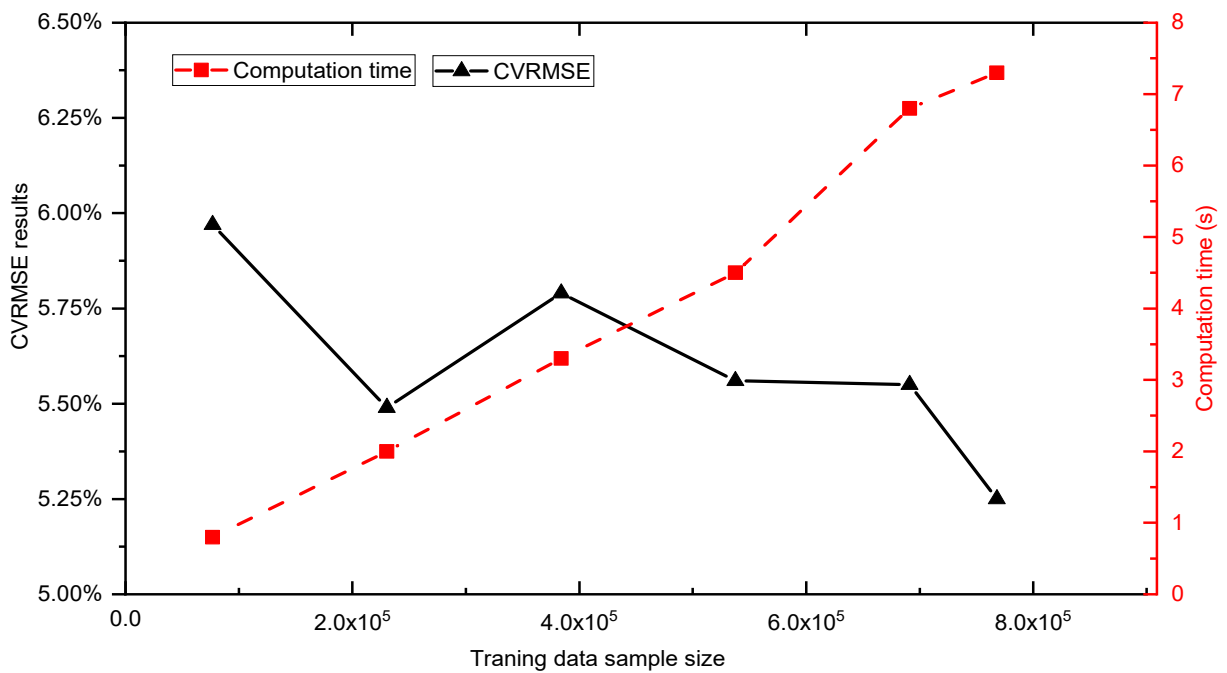
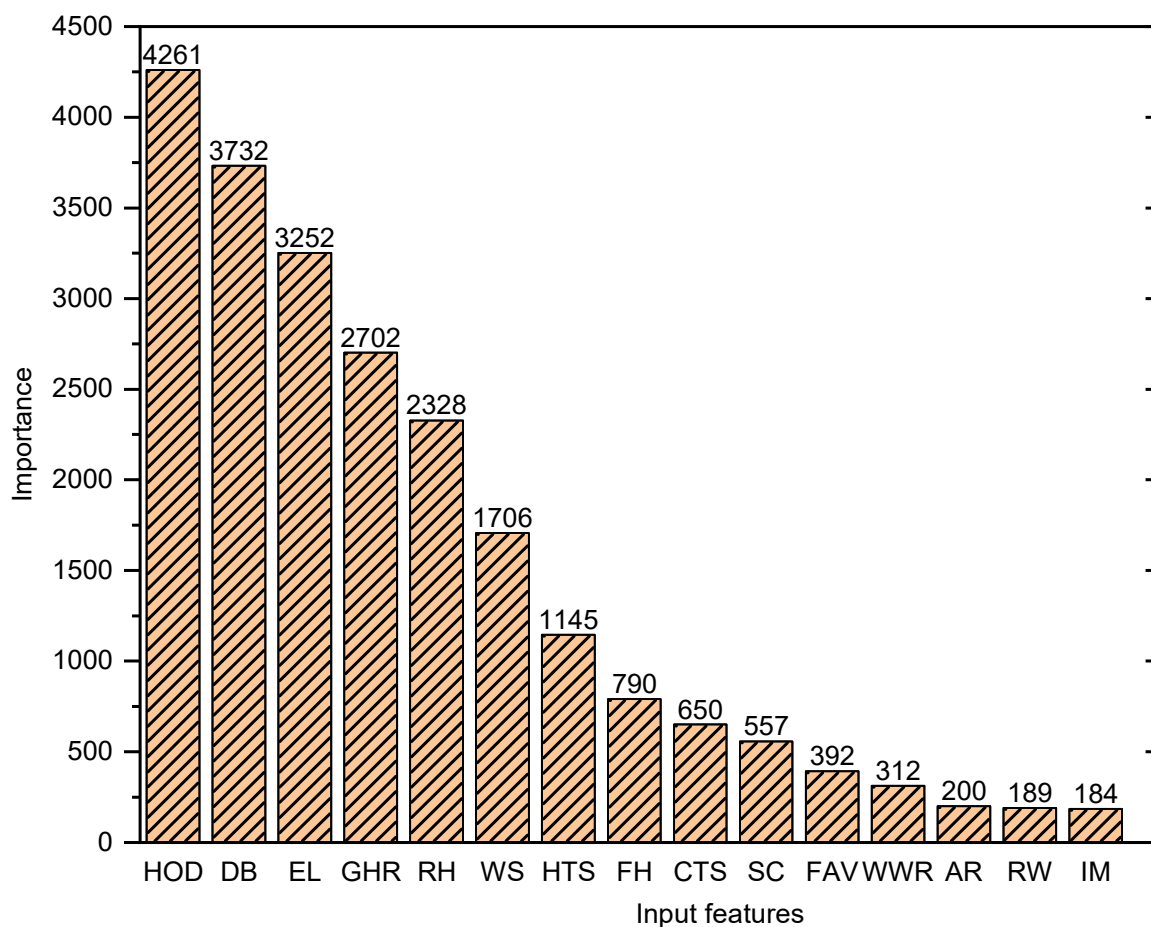
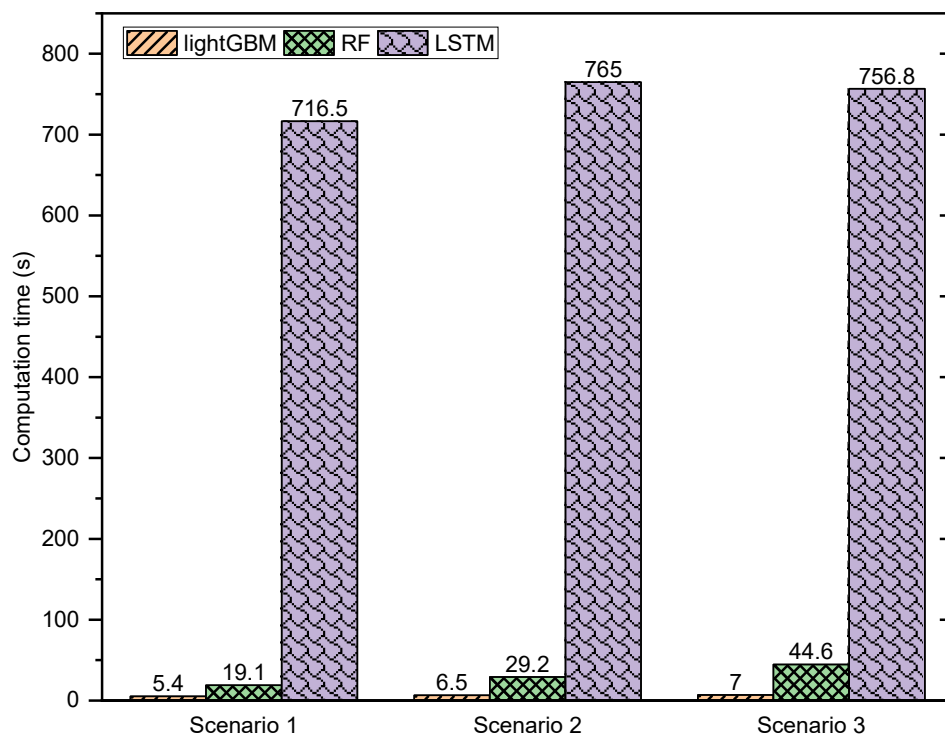


Figure 9. Prediction performance using different sizes of data samples in LightGBM.



**Figure 10.** Feature importance ranking in LightGBM. (HOD: Hour of the day, DB: Dry bulb temperature, EL: Electrical load, GHR: Global horizontal radiation, RH: Relative humidity, WS: Wind speed, HTS: Heating temperature set point, FH: Floor height, CTS: Cooling temperature set point, SC: Shape coefficient, FAV: Fresh air volume, WWR: Window-to-wall ratio, AR: Aspect ratio, RW: R-value of wall, and IM: Internal mass.)

As using more features can improve the predictive accuracy, we investigated the effect of the number of features on the accuracy and computation speed in these three algorithms. The computational expense comparison of this study was performed on Windows 10 with a 2.6 GHz processor (Intel Corporation Core i7-10700) and 16 GB RAM memory. The Spyder scientific Python-integrated development environment was used to implement the prediction tasks. As shown in Figure 11, LightGBM requires the least computation time for model training and testing. In contrast, LSTM spends much more time. Generally, the computation expense increases with the input feature size. The model development time reaches the maximum when all 15 input features are considered. The LSTM model takes much longer time than the other two models, which is not suitable for a real-time energy management control system. The computational time of LightGBM is very short (several seconds), which makes LightGBM a suitable model for a real-time control system. Note that the acquisition of the original massive dataset is time-consuming, as we spent about 700 h to obtain the dataset using a power machine (Dell Precision 7920 Tower, 20 kernel CPU). The acquisition of the dataset process is time-consuming; however, this developed model achieves higher prediction accuracy, and it can be easily generalized for various energy use scenarios and building types once it is well trained. In this way, the model developers do not need to develop a specific model for different buildings and energy use scenarios.



**Figure 11.** Computation time of the models in different scenarios.

#### 4. Conclusions

A well-developed data-driven model to represent the physics-based tools is a challenge in both academic and practical fields. Traditionally, well-designed physics-based models, i.e., white-box models, have been widely applied. However, a white-box model requires a massive amount of detailed input parameters, which can be troublesome and difficult for engineers, especially for a building in the design phase. A fast and accurate building thermal load prediction method is critically important for optimal HVAC control, energy demand-side management, smart building management, and other tasks. In this study, therefore, we ran a big amount of EnergyPlus simulations to obtain massive energy data that covers the common energy use scenarios to develop a good generalization data-driven model. Using this data source, three machine learning models were developed and compared in three different input feature scenarios. Upon completion of the investigation, the following conclusions were reached.

- (1) LightGBM is the most accurate and fastest prediction model. In the best scenario, the CVRMSE and  $R^2$  of LightGBM are 5.25% and 0.99, respectively. Compared with the results of the other two algorithms and those in the existing literature, LightGBM is the most promising and best algorithm for building thermal load prediction.
- (2) By training with the large amount of energy data generated by physics-based tools or on-site data, a data-driven model is able to represent a physics-based tool with comparable accuracy.
- (3) The dimensions of the input features influence the prediction performance. Compared with a scenario using only weather information, the CVRMSE can be further improved when physical and operational information are considered. Although better accuracy is achieved with bigger dimensions of input features, it impacts the computational speed. Therefore, there will always be a tradeoff between the prediction accuracy demand and prediction speed tolerance.

The findings and the proposed models in this study are useful for real applications, such as smart building energy management, baseline calculation of demand response programs, and grid-integrated efficient building improvements. LightGBM is strongly

recommended when dealing with large amounts of data, as it is faster and more robust. Building managers and engineers do not need to build a sophisticated physical model (such as the EnergyPlus model) to calculate the energy demand. Several basic inputs are able to reduce the tiresome tasks ordinarily required beforehand and can be used to obtain the energy demand with acceptable accuracy. In the early design phases of buildings with a lack of building information, only basic building and weather information are required to implement the prediction task. For existing buildings, not only considering the 17 variables mentioned above but more detailed building physics information, the operational schedule, and more historical load data can be used to rebuild the data-driven model and further improve the prediction performance.

**Author Contributions:** Conceptualization, Y.C. and Y.Y.; methodology, Y.C.; software, Y.Y.; writing—original draft preparation, Y.C. and Y.Y.; writing—review and editing, J.L., L.Z. and S.M.; supervision, Y.Y.; funding acquisition, Y.C. and W.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research was funded by the National Natural Science Foundation of China (No. 52208116), the China Postdoctoral Science Foundation (No. 2020M681347) and the Key R&D and Promotion Project of the Department of Science and Technology of Henan Province, China (No. 222102320113).

**Data Availability Statement:** The dataset and full code to develop the data-driven model can be downloaded freely on GitHub: <https://github.com/Bob05757/Key-inputs-setting-and-Energy-Data>.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Nomenclature

ANN	Artificial neural networks
AR	Aspect ratio
CTS	Cooling temperature set point
CV	Coefficient of variation
CVRMSE	Coefficient of variation of root mean square error
DB	Dry bulb temperature
EL	Electrical load
ELM	Extreme learning machine
FAV	Fresh air volume
FH	Floor height
GHR	Global horizontal radiation
HOD	Hour of the day
HTS	Heating temperature set point
IM	Internal mass
MAPE	Mean absolute percentage error
MARS	Multivariate adaptive regression splines
MLP	Multilayer layer perceptron
NARM	Nonlinear autoregressive model
RF	Random forest
RH	Relative humidity
RNN	Recurrent neural networks
RW	R-value of wall
SC	Shape coefficient
SVM	Support vector machine
SVR	Support vector regression
WS	Wind speed
WWR	Window to wall ratio
XGBoost	Extreme gradient boosting

## Appendix A

Hyperparameter tuning information with five-fold cross-validation in LSTM, LightGBM, and RF.

**Table A1.** Hyperparameter grid search and tuning results in LSTM.

Hyperparameter	Description	Grid Searching Range	Selected
<input type="checkbox"/> activation	Activation functions	Sigmoid; Tanh; Relu	Relu
<input type="checkbox"/> optimizer	Optimization algorithms	Adam; RMSprop; Adagrad; SGD	RMSprop
<input type="checkbox"/> loss	Loss function	Mean Square Error; Mean Absolute Error; Mean Squared Logarithmic Error	Mean Square Error
<input type="checkbox"/> units	Number of memory cells	range (20, 200, 20)	60
<input type="checkbox"/> epochs	Number of epochs	range (20, 200, 20)	100
<input type="checkbox"/> batch_size	Number of batch size	[20, 32, 60, 100, 500]	60

**Table A2.** Hyperparameter grid search and tuning results in LightGBM.

Hyper-Parameters	Description	Grid Searching Range	Selected
learning_rate	Step size shrinkage used in the update to prevent overfitting	range (0.02, 0.12, 0.02)	0.1
n_estimators	Number of estimators	range (50, 400, 50)	350
max_depth	The depth of tree model	range (3, 10, 1)	9
num_leaves	the main parameter to control the complexity of the tree model	range (5, 500, 5)	65
max_bin	the maximum number of bins stored in feature	range (5, 256, 10)	95

**Table A3.** Hyperparameter grid search and tuning results in RF.

Hyperparameters	Description	Grid Searching Range	Selected
n_estimators	The number of trees in the forest	range (10, 110, 10)	20
max_depth	The maximum depth of the tree	range (2, 20, 2)	8
min_sample_split	The minimum number of samples required to split an internal node	range (1, 11, 1)	6
max_features	The number of features to consider when looking for the best split	['auto', 'sqrt', 'log2']	'auto'

## References

- Fan, C.; Liao, Y.; Zhou, G.; Zhou, X.; Ding, Y. Improving cooling load prediction reliability for HVAC system using Monte-Carlo simulation to deal with uncertainties in input variables. *Energy Build.* **2020**, *226*, 110372. [\[CrossRef\]](#)
- Li, W.; Gong, G.; Fan, H.; Peng, P.; Chun, L.; Fang, X. A clustering-based approach for “cross-scale” load prediction on building level in HVAC systems. *Appl. Energy* **2021**, *282*, 116223. [\[CrossRef\]](#)
- Jang, Y.; Byon, E.; Jahani, E.; Cetin, K. On the long-term density prediction of peak electricity load with demand side management in buildings. *Energy Build.* **2020**, *228*, 110450. [\[CrossRef\]](#)
- Chen, Y.; Xu, P.; Chu, Y.; Li, W.; Wu, Y.; Ni, L.; Bao, Y.; Wang, K. 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. [\[CrossRef\]](#)
- Chen, Y.; Chen, Z.; Xu, P.; Li, W.; Sha, H.; Yang, Z.; Li, G.; Hu, C. Quantification of electricity flexibility in demand response: Office building case study. *Energy* **2019**, *188*, 116054. [\[CrossRef\]](#)
- Foucquier, A.; Robert, S.; Suard, F.; Stephan, L.; Jay, A. State of the art in building modelling and energy performances prediction: A review. *Renew. Sustain. Energy Rev.* **2013**, *23*, 272–288. [\[CrossRef\]](#)
- Luo, X.J.; Lukumon, O.O.; Anuoluwapo, O.A.; Olugbenga, O.A.; Hakeem, A.O.; Ashraf, A. Feature extraction and genetic algorithm enhanced adaptive deep neural network for energy consumption prediction in buildings. *Renew. Sustain. Energy Rev.* **2020**, *131*, 109980. [\[CrossRef\]](#)

8. Nageler, P.; Schweiger, G.; Pichler, M.; Brandl, D.; Mach, T.; Heimrath, R.; Schranzhofer, H.; Hochenauer, C. Validation of dynamic building energy simulation tools based on a real test-box with thermally activated building systems (TABS). *Energy Build.* **2018**, *168*, 42–55. [[CrossRef](#)]
9. Wang, Z.; Hong, T.; Piette, M.A. Data fusion in predicting internal heat gains for office buildings through a deep learning approach. *Appl. Energy* **2019**, *240*, 386–398. [[CrossRef](#)]
10. Wu, J.; Wang, Y.-G.; Tian, Y.-C.; Burrage, K.; Cao, T. Support vector regression with asymmetric loss for optimal electric load forecasting. *Energy* **2021**, *223*, 119969. [[CrossRef](#)]
11. Ahmad, T.; Chen, H. Nonlinear autoregressive and random forest approaches to forecasting electricity load for utility energy management systems. *Sustain. Cities Soc.* **2019**, *45*, 460–473. [[CrossRef](#)]
12. Lahouar, A.; Ben Hadj Slama, J. Day-ahead load forecast using random forest and expert input selection. *Energy Convers. Manag.* **2015**, *103*, 1040–1051. [[CrossRef](#)]
13. Wang, Z.; Hong, T.; Piette, M.A. Building thermal load prediction through shallow machine learning and deep learning. *Appl. Energy* **2020**, *263*, 114683. [[CrossRef](#)]
14. Cao, L.; Li, Y.; Zhang, J.; Jiang, Y.; Han, Y.; Wei, J. Electrical load prediction of healthcare buildings through single and ensemble learning. *Energy Rep.* **2020**, *6*, 2751–2767. [[CrossRef](#)]
15. Moon, J.; Park, S.; Rho, S.; Hwang, E. Robust building energy consumption forecasting using an online learning approach with R ranger. *J. Build. Eng.* **2022**, *47*, 103851. [[CrossRef](#)]
16. Wang, Y.; Gan, D.; Sun, M.; Zhang, N.; Lu, Z.; Kang, C. Probabilistic individual load forecasting using pinball loss guided LSTM. *Appl. Energy* **2019**, *235*, 10–20. [[CrossRef](#)]
17. Somu, N.; Raman, G.M.R.; Ramamritham, K. A hybrid model for building energy consumption forecasting using long short term memory networks. *Appl. Energy* **2020**, *261*, 114131. [[CrossRef](#)]
18. Xu, L.; Hu, M.; Fan, C. Probabilistic electrical load forecasting for buildings using Bayesian deep neural networks. *J. Build. Eng.* **2022**, *46*, 103853. [[CrossRef](#)]
19. Khwaja, A.S.; Anpalagan, A.; Naeem, M.; Venkatesh, B. Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting. *Electr. Power Syst. Res.* **2020**, *179*, 106080. [[CrossRef](#)]
20. Zhou, Y.; Liang, Y.; Pan, Y.; Yuan, X.; Xie, Y.; Jia, W. A Deep-Learning-Based Meta-Modeling Workflow for Thermal Load Forecasting in Buildings: Method and a Case Study. *Buildings* **2022**, *12*, 177. [[CrossRef](#)]
21. Zhang, Y.; Teoh, B.K.; Wu, M.; Chen, J.; Zhang, L. Data-driven estimation of building energy consumption and GHG emissions using explainable artificial intelligence. *Energy* **2023**, *262*, 125468. [[CrossRef](#)]
22. Shi, J.; Li, C.; Yan, X. Artificial intelligence for load forecasting: A stacking learning approach based on ensemble diversity regularization. *Energy* **2023**, *262*, 125295. [[CrossRef](#)]
23. Lu, Y.; Meng, L. A simplified prediction model for energy use of air conditioner in residential buildings based on monitoring data from the cloud platform. *Sustain. Cities Soc.* **2020**, *60*, 102194.
24. Wang, R.; Lu, S.; Li, Q. Multi-criteria comprehensive study on predictive algorithm of hourly heating energy consumption for residential buildings. *Sustain. Cities Soc.* **2019**, *49*, 101623. [[CrossRef](#)]
25. Seyedzadeh, S.; Pour Rahimian, F.; Rastogi, P.; Glesk, I. Tuning machine learning models for prediction of building energy loads. *Sustain. Cities Soc.* **2019**, *47*, 101484. [[CrossRef](#)]
26. Kumar, S.; Pal, S.K.; Singh, R.P. A novel method based on extreme learning machine to predict heating and cooling load through design and structural attributes. *Energy Build.* **2018**, *176*, 275–286. [[CrossRef](#)]
27. Kaggle Competitions. Available online: <https://www.kaggle.com/competitions> (accessed on 12 January 2023).
28. Butcher, Q. *Machine Learning with Spark-Covers XGBoost, LightGBM, Spark NLP; Distributed Deep Learning with Keras, and More*; Springer Science, Business Media New York: New York, NY, USA, 2020.
29. Get Started with XGBoost. Available online: [https://xgboost.readthedocs.io/en/latest/get\\_started.html](https://xgboost.readthedocs.io/en/latest/get_started.html) (accessed on 12 January 2023).
30. Chen, Y.; Guo, M.; Chen, Z.; Chen, Z.; Ji, Y. Physical energy and data-driven models in building energy prediction: A review. *Energy Rep.* **2022**, *16*, 2656–2671. [[CrossRef](#)]
31. Dudek, G. Short-Term Load Forecasting Using Random Forests. *Adv. Intell. Syst. Comput.* **2015**, *323*, 821–828.
32. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
33. Zang, H.; Xu, R.; Cheng, L.; Ding, T.; Liu, L.; Wei, Z.; Sun, G. Residential load forecasting based on LSTM fusing self-attention mechanism with pooling. *Energy* **2021**, *229*, 120682. [[CrossRef](#)]
34. Zhang, L.; Wen, J. Active learning strategy for high fidelity short-term data-driven building energy forecasting. *Energy Build.* **2021**, *244*, 111026. [[CrossRef](#)]
35. Hu, Y.; Cheng, X.; Wang, S.; Chen, J.; Zhao, T.; Dai, E. Times series forecasting for urban building energy consumption based on graph convolutional network. *Appl. Energy* **2022**, *307*, 118231. [[CrossRef](#)]
36. Olu-Ajayi, R.; Alaka, H.; Sulaimon, I.; Sunmola, F.; Ajayi, S. Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. *J. Build. Eng.* **2022**, *45*, 103406. [[CrossRef](#)]
37. Li, Y.; Tong, Z.; Tong, S.; Westerdahl, D. A data-driven interval forecasting model for building energy prediction using attention-based LSTM and fuzzy information granulation. *Sustain. Cities Soc.* **2022**, *76*, 103481. [[CrossRef](#)]

38. Do, H.; Cetin, K.S. Evaluation of the causes and impact of outliers on residential building energy use prediction using inverse modeling. *Build. Environ.* **2018**, *138*, 194–206. [CrossRef]
39. Guo, Y.; Wang, J.; Chen, H.; Li, G.; Liu, J.; Xu, C.; Huang, R.; Huang, Y. Machine learning-based thermal response time ahead energy demand prediction for building heating systems. *Appl. Energy* **2018**, *221*, 16–27. [CrossRef]
40. Wang, Z.; Srinivasan, R.S. A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. *Renew. Sustain. Energy Rev.* **2017**, *75*, 796–808. [CrossRef]
41. Commercial Prototype Building Models. Available online: [https://www.energycodes.gov/development/commercial/prototype\\_models](https://www.energycodes.gov/development/commercial/prototype_models) (accessed on 12 January 2023).
42. Bryan, E.; Zheng, O.; Vladimir, A.F.; Igor, M. Uncertainty and sensitivity decomposition of building energy models. *J. Build. Perform. Simu.* **2012**, *5*, 171–184.
43. Key-Inputs-Setting-and-Energy-Data. Available online: <https://github.com/Bob05757/Key-inputs-setting-and-Energy-Data> (accessed on 12 January 2023).
44. Li, Q.; Meng, Q.; Cai, J.; Yoshino, H.; Mochida, A. Applying support vector machine to predict hourly cooling load in the building. *Appl. Energy* **2009**, *86*, 2249–2256. [CrossRef]
45. Leung, M.C.; Tse, N.C.F.; Lai, L.L.; Chow, T.T. The use of occupancy space electrical power demand in building cooling load prediction. *Energy Build.* **2012**, *55*, 151–163. [CrossRef]
46. Luo, X.J.; Oyedele, L.O.; Ajayi, A.O.; Monyei, C.G.; Akinade, O.O.; Akanbi, L.A. Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands. *Adv. Eng. Inform.* **2019**, *41*, 100926. [CrossRef]
47. Guolin, K.; Qi, M.; Thomas, F.; Taifeng, W.; Wei, C.; Weidong, M.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
48. Distributed Machine Learning Toolkit-Big Data, Big Model, Flexibility, Efficiency. Available online: <http://www.dmtk.io/> (accessed on 12 January 2023).
49. Jin, R.M.; Agrawal, G. Communication and Memory Efficient Parallel Decision Tree Construction. In Proceedings of the 2003 SIAM International Conference on Data Mining, San Francisco, CA, USA, 1–3 May 2003; pp. 119–129.
50. Moon, J.; Kim, Y.; Son, M.; Hwang, E. Hybrid Short-Term Load Forecasting Scheme Using Random Forest and Multilayer Perceptron. *Energies* **2018**, *11*, 3283. [CrossRef]
51. Ahmad, M.W.; Mourshed, M.; Rezugui, Y. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy Build.* **2017**, *147*, 77–89. [CrossRef]
52. Ahmad, M.W.; Reynolds, J.; Rezugui, Y. Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *J. Clean. Prod.* **2018**, *203*, 810–821. [CrossRef]
53. Shi, Y.; Song, X.; Song, G. Productivity prediction of a multilateral-well geothermal system based on a long short-term memory and multi-layer perceptron combinational neural network. *Appl. Energy* **2021**, *282*, 116046. [CrossRef]
54. Ding, Y.; Zhang, Q.; Yuan, T.; Yang, K. Model input selection for building heating load prediction: A case study for an office building in Tianjin. *Energy Build.* **2018**, *159*, 254–270. [CrossRef]
55. American Society Of Heating VAAC. Measurement of Energy and Demand Savings. *ASHRAE Guidel* **2014**, *4*, 1–150.
56. Geron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media: Sebastopol, CA, USA, 2017.
57. Fan, C.; Xiao, F.; Zhao, Y. A short-term building cooling load prediction method using deep learning algorithms. *Appl. Energy* **2017**, *195*, 222–233. [CrossRef]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.